École Doctorale Informatique et Mathématiques

InfoMaths

Institut National des Sciences Appliquées de Lyon

Laboratoire d'Informatique pour l'Entreprise et les Systèmes de Production

C-Business et Urbanisation d'Entreprise

Thèse de Doctorat

(Spécialité Informatique)

Layth SLIMAN

(Ingénieur en Informatique)

Sous la Direction de: Frédérique BIENNIER (Professeur à l'INSA de Lyon)

Soutenu Publiquement le 27 Novembre 2008 Devant la Commission d'Examen Composé de :

Rapporteurs

Danielle BOULANGER (Professeur - Université Jean Moulin Lyon 3)
Bruno VALLESPIR (Professeur - Université de Bordeaux)

Autres Membres du Jury

Jacques MALENFANT (Président du Jury, Professeur à l'Université Pierre et Marie Curie, Paris).
Zensho NAKAO (Examinateur, Professeur à la Faculté d'ingénierie de l'Université de Ryukyus- Japon).
Youakim BADR (Co-encadrant, Maître de Conférences à l'INSA de Lyon)

ACKNOWLEDGEMENTS

First of all I would like to express my profound gratitude to my supervisor, Prof. Frédérique BIENNIER, for her invaluable support, encouragement, patience, and immense help throughout this research work.

I gratefully thank Prof. Jacques MALENFANT, the president of my dissertation committee, for attending my defence committee.

I would also like to wholeheartedly thank the members of my dissertation committee, Prof. Danielle BOULANGER and Prof. Bruno VALLESPIR, for accepting to review this dissertation.

Very special thanks go out to Prof. Zensho NAKAO for providing me the means to learn and understand during my research stay in his laboratory, as well as for attending my defence.

I appreciatively thank Dr. Youakim BADR, my research for this dissertation was made more efficient but also much more extensive through his help.

I remain indebted to Prof. Jean-Pierre CAMPAGNE, the director of LIESP laboratory, as well as all the laboratory's members, whose support has made the laboratory more than a temporary place of study or research.

This thesis gradually emerged amid my friends and colleagues, I wish I could mention each individually; they animated my three years and provided their most generous support.

I am forever grateful to my parents, Aziz and Faten, and my sister Soolafa, for their unconditional support at each turn of the road.

Layth

Abstract

Market evolution has lead most of the enterprises to focus on their core business while setting outsourcing and collaborative strategies to be able to propose the best product-service offers. This evolution increases the call for collaborative business, involving several partners sharing a common goal. This Collaborative Business environment challenges Information System re-organisation to set agile, reactive and interoperable IT supports.

To fulfil these requirements, one can reorganise the corporate information system according to the urbanisation paradigm, setting smaller rather autonomous units that can be maintained separately. Coupled to Service Oriented Architecture deployment, this approach provides more interoperable information systems. Nevertheless, traditional urbanization strategies lead to a partitioned and rather rigid information system organization (aligned on the enterprise business areas), which hinders initiating "on demand" collaborative production processes, since production process is transversal and bypasses all enterprise business areas, starting from customer and suppliers management to production management itself. To overcome these limits, we propose to adopt a new urbanization strategy that combines the transversal production logic with a service orientation to allow incremental production process building, based on goals to be reached. This lead us to propose an architecture built on the (activity, resource) couple, and thus to define business processes in a “resource driven logic” rather than "task-driven logic" in order to make the production system and its IT support more responsive and flexible.

Despite of the importance of the ICT support in Collaborative Business enactment, one must also consider “environmental” factors as key point to develop (or not) collaborative strategies. As enterprises must open their own information system and exchange “private” data and processes to achieve the collaborative process, lack of trust and security requirements can be a braking force while developing collaborative strategies. In this case, risk management and security management must go beyond technical aspects, to integrate partners’ business model. Security needs’ analysis must take into consideration business “intelligence” requirements, focusing on “business” confidentiality. This leads to identify private and shareable information and processes and impacts the implementation of collaborative activities within partners’ business processes while maintaining well-controlled levels of visibilities and coordination necessary for the achievement of the collaborative process. These needs must

then be transcribed into the implemented architecture to ensure the required security functionalities.

The architecture we propose to implement our enterprise urbanization approach is based on a service-oriented model. We extend the traditional IT service to capture semantics associated to the industrial activity so that an industrial service model is proposed. Industrial services are identified by gathering activities and resources depending on industrial constraints (production rules, product to offer…). By this way, setting a collaborative organisation consist in selecting and composing the convenient industrial services. Then security requirements are added in this model to govern access to different interfaces in a composite service. To this end, we proposed to integrate security needs and constraints into the definitions of business processes, organizational structure and technical components, providing quality of protection agreements between the involved parties in a collaborative process.

The implementation of this architecture is achieved using an industrial service bus, paying attention on industrial semantic annotations and on end-to-end security management. This bus integrates a semantic layer and a security manager component over classical ESB features to support a business services-oriented composition at the semantic level and not only at the technical level, while taking into account the overall context and security preferences of the different involved parties. The modules of our service bus are developed on the top of "PEtALS", an open source ESB.

Résumé

Les évolutions permanentes du marché ont forcé la plupart des entreprises à se focaliser sur les processus liés à leur cœur de métier. Ce recentrage les conduit alors soit à externaliser certaines parties de leurs processus, soit à former temporairement une association avec d'autres partenaires. Des tels scénarios impliquent la collaboration de plusieurs parties. Ces scénarios de collaboration imposent plusieurs contraintes sur la structure organisationnelle de l'entreprise et sur la conception et l'organisation du système d'information afin de le rendre facilement adaptable pour suivre les changements au niveau d'organisation induits par les scénarios de collaboration.

Pour que le système d'information soit facilement adaptable il est possible de le restructurer en respectant les principes d'urbanisation du système d'information et en adoptant une architecture orientée service. Toutefois, cette organisation conduit à des organisations d'entreprise assez rigides (cloisonnées selon les dimensions fonctionnelles) ne donnant pas réellement les capacités d'initier des processus collaboratifs car cette approche induit une certaine rigidité au niveau de l'organisation du processus de production qui passe les différents secteurs de l'entreprise de la gestion de clientèles et de fournisseurs au processus de production ce qui rend plus difficile la collaboration en rendant complexe la modification « à la demande » des processus. Ainsi, pour pallier les limites induites par les stratégies traditionnelles d'urbanisation dans le contexte de la production collaborative, nous proposons d'adopter une nouvelle stratégie d'urbanisation qui couple une organisation transversale du système de production de l'entreprise, et une orientation service d'une façon à permettre une construction incrémentale des processus en fonction des objectifs à atteindre, cela nous amène à proposer une architecture construite autour du couple (activité, ressources), et ainsi, à définir les processus dans une logique « pilotée par les ressources» et non plus « pilotée par les tâches» pour rendre l'organisation système de production et le SI plus réactifs et plus flexibles.

Malgré les avantages de la collaboration, ces organisations sont souvent limitées car les entreprises restent hostiles à partager leurs informations (voire des processus). Ainsi, tout scénario de collaboration doit fournir un niveau suffisant de sécurité à tous les partenaires impliqués dans un processus collaboratif. Ce niveau dépend souvent des entreprises partenaires : lorsque des entreprises potentiellement concurrentes s'allient pour un projet de collaboration, les besoins en sécurité sont accrus du fait de la perception du risque de

concurrence. L'analyse des besoins de sécurité liés à la collaboration doit prendre en compte ces aspects de «Business Security» et faire la distinction entre informations et processus privés d'une part et ceux qui sont partageables d'autre part tout en veillant à la mise en œuvre des activités au sein des entreprises partenaires et à la coordination nécessaires à la réalisation des processus collaboratifs. Ces besoins doivent ensuite être retranscrits dans l'architecture mise en œuvre pour assurer les fonctions de sécurité nécessaires.

L'architecture retenue dans notre démarche d'urbanisation d'entreprise repose sur un modèle orienté service. Nous sommes parvenus à spécifier un modèle de service industriel construit par regroupement de toutes les fonctions nécessaires autour de la fabrication du produit ce qui permet de composer des services répartis sur différents sites pour construire un processus collaboratif. Dans ce cadre, nous devons construire un modèle donnant une visibilité globale sur le niveau de sécurité à atteindre qui régit l'accès aux différentes interfaces dans un service composé de manière à assurer la sécurité pour l'ensemble des services. Pour cela, nous avons proposé d'intégrer les besoins et contraintes de sécurité dans les définitions des processus métier, les composants de la structure organisationnelle et technique de manière à proposer des accords de qualité de protection entre les parties.

La mise en œuvre de cette architecture repose sur la construction d'un middleware de services industriels capable de tenir compte et assurer la sécurité des services composés et issus d'entreprises différentes. Ce bus ajoute aux fonctionnalités d'un ESB classique, les services principaux nécessaires pour une composition orienté services métiers au niveau sémantique et non plus seulement technique et qui prend en compte le contexte globale ainsi que les préférences de la sécurité des différentes parties impliquées. Les modules de notre bus de services sont développés au dessus de l'ESB open source « PEtALS ».

CONTENTS

List of Figures

List of Tables

Chapter 1

1 General Introduction

In an ever-changing world where customer requirements, marketplace organisation and technologies fluctuate continuously, a dynamic adaptation of enterprise organisation and processes proves essential. Coupled with globalisation by which enterprises evolve in a volatile market with unforeseeable impacts, a desire for quality and delay-sensitive customers, these changes increase the need for developing collaboration as virtual companies, alliances and partnerships in a “globalized” work force.

Several problems can be identified in the case of collaboration. First, in spite of the benefits of collaboration, partners are hostile to opening their systems and want to maintain their autonomy and privacy. Any collaborative scenario must therefore provide a sufficient level of “securitisation” to all involved partners. Here, security does not only concern technical aspects such as data integrity and privacy, in fact, collaboration may bring together competitor enterprises, because of their competition, the security perception for such partners may go beyond technical aspects to cover their business model. Collaboration must take into consideration these aspects of “business” confidentiality and distinguish between private and shareable information and process regarding the implementation of activities within partners’ business processes while maintaining a good level of coordination necessary for the achievement of the collaborative process.

The problem is that the integration of partners requires sharing and collaboration between parties traditionally discouraged and even explicitly prohibited from sharing their ideas. Indeed, social, legal and organizational structures were developed to ensure the inaccessibility of information related to critical missions to external organizations.

Overcoming these challenges is more critical than technical problems, and thus requires being taken into consideration at the time an information system is developed for the technical communication with partners and suppliers.

Evidently, establishing the collaborative process starts by selecting partners. To enable such selection, enterprises must define and publish, in an objective manner, what they can offer to potential partners. In addition, based on previous discussion, they must exhaustively specify constraints, conditions and contexts under which their resources can be used, and their processes can be monitored. Without such descriptions, on-demand collaboration will be difficult, if not impossible. Potential partners can be eliminated because of the ambiguity of

their offers. As a result, processes “composition” proceeds mainly through a better description of offers and needs adequate tools which allow the identification of the compatibility offer or request, as well as for their the functional aspects (their processes input and output), as for the non-functional aspects (QoS, QoP…).

Further, these collaboration scenarios impose a reorganization of the enterprise in order to promote interoperable structures at all levels i.e. a multi perspective interoperability solution which insures interoperability at technological, organisational and business levels.

At the technical level, enterprises have to analyse their information systems in order to adapt them to dynamic collaborative strategies. That is to say the information system must be reorganized to facilitate its evolution and increase its level of technological interoperability while protecting its informational legacy. To do this, one solution is to urbanise the enterprise information system. Urbanising a corporate information system consists in reducing the IS to autonomous, small size modules. This suggests the local urbanization process. Exchange information units are set between these modules in order to de-couple them so that they can evolve or be removed separately while preserving their capacity to interact with the rest of the system. This reinforces the possibility of replacing some of these subsystems and integrating subsystems of various origins. In turn, this strengthens the capacity of IS to interact with other ISs, increasing its cooperation level. In the traditional approach, the IS urbanisation strategy is based on the enterprise’s functional or vertical architecture. This leads to specialised business support software, which augments the interoperability and the agility at the information system level but also increases rigidity at the operational level and constraints the collaboration because it does not take into account the production process logic which involves a transversal or horizontal organisation starting from the customer relationship management and supplier relationship management to the production process itself. Accordingly, a new urbanisation paradigm should be introduced; such an approach must take into consideration the particularity of the production context; an urbanization strategy must consider a “lean” organization of both: the information and the production systems to increase interoperability not only at the information system level, but also at the production system level. This should enable “on-demand” reconfigurations of production processes by integrating production constraints into the urbanisation strategy, that is to say taking into account the way business processes are related to the way the manufacturing enterprise transforms raw materials into products, such as its production process.

To promote interoperability at the organizational and business level and to allow collaborative production process construction starting from diverse specialized processes belonging to

different partners, industrial constructs should be described, published and made accessible via adapted platforms. Since production means, physical and human resources are structurally interrelated with the production process organization and execution, consistent grouping of these objects must be identified in order to be able to expose functional constructs (called in or work “industrial services”) allowing the composition of collaborative industrial processes. This involves redefining the organization through goals, identifying processes to achieve and then determining for each process the required competencies and means without -a priori- being constrained by corporate organization.

Organisational interoperability imposes adapting organisation structures and means to order to attain a dynamic organisation which is capable of providing suitable and instant responses to diverse collaboration scenarios.

To resolve this issue, we propose an urbanization strategy which emphasizes industrial organization to determine which industrial services could be compounded in a consistent manner (by following the production logic.)

Our objective is to identify industrial "services" "which will be published and offered in such a way as to allow the composition, or, in other words, to allow organizing logical groups of production resources (production islands) and managing the business flows between these islands, integrating all services required to complete a final product in response to a customer’s “on demand” order. For this, we define a production system which enables building dynamic and incremental processes using a "bottom-up " logic.

To do this, we propose to achieve a production resources re-clustering to determine “logical” groups of resources that match interoperable functional constructs while relying on IS urbanization principles in order to define our production system urbanization approach. This leads to defining an organization which takes into consideration the horizontal structure of the production system and not the vertical (functional) structure of the enterprise alone.

“Potential Industrial constructs” identification and grouping starting from production means, physical and human resources imposes decomposition and re-composition of enterprise production system in order to define a dynamic and interoperable production system, a production system which constructs composite processes in response to customers orders. This entails readapting urbanisation rules in such a way to enable orchestrating identified constructs on the basis of their interdependencies and to delineate the “amount” and means of coordination and stream controls, inter and intra production units participating in a collaborative production process.

To achieve the announced objectives, we propose an IS urbanization strategy which integrates production criterion, coupled with, Service-Oriented Architecture (SOA) in order to organise a flexible and evolving information system able to expose “composable” industrial services.

In our urbanism strategy we essentially focus on the “industrial” part of company’s activity in order to integrate the production system and the IS. As a result, we will improve the adaptability of the production to process organisation and prompt collaboration via exposing “Industrial Services” according to the SOA paradigm. We define an industrial service as a meaningful grouping of enterprise resources and activities which provide added value and can contribute in a more coarse grain production process.

In fact, SOA represents an approach to distributed computing that provides a powerful integration solution within the enterprise as well as beyond, between a company and its partners and customers by providing an abstraction layer that exposes application functionality as business-oriented services that are both location independent and discoverable on the network. Consequently, integrating this service-oriented approach while organising the business process can improve business flexibility and reactivity as the core process can be changed without affecting the way the service is included in more a complex organisation, and interoperability (as standard interfaces are published and access to parts of the corporate information system are simplified).

Another advantage of the service approach is that it ensures the reduction of costs through the bottom-up construction of composite services. In fact, the concept of "publishing and selecting services" allows a “value”-based election of available services, and a natural elimination of unnecessary activities, as a result, global “value chain” enhancement is obtained.

SOA offers new features ensuring a high level cross-enterprise integration solution allowing composition of on the fly services under collaboration contracts. Building a collaborative Service-Oriented Architecture at the production system level is a difficult task. In fact, it does not consist of wrapping software entities in the form of functional blocks with appropriate interfaces alone. Such an orientation requires a cross-layers approach to ensure the identification, modelling specifications and integration of production objects in the form of industrial services in such a way as to allow seeing the production system as a set of services some of which will be exposed to be used by actors outside the enterprise.

In addition, to enable collaborative process composition, SOA lowers barriers between businesses. The lowering of barriers forces us to rethink security, that is, security cannot be compromised to meet business goals. Since traditional security approaches assumed and took

advantage of barriers, we have to find new approaches to secure SOA applications. In other words, the traditional “islands of security” approach to application security fails in the SOA context since identity verification and access and permissions control mechanisms and policies might vary among the various back-end systems. The problem is that the service approach is limited by computing and orchestrating layers which operate at the technical level, the service composition layer acts as a layer of abstraction and hides the details of the underlying technology implementation from the users. Each service abstracts user identity and context from the underlying applications. In other words, in a service-oriented model, users can access services located on different systems at different times, and the underlying applications no longer have the user context they require to authorize specific actions. These constraints in addition to the constraints of the industrial and non-functional context and requirements make it difficult to associate the users of the overall functionality. Therefore, enterprises using the SOA paradigm need a unified identity management and security policy infrastructure that governs access to the different interfaces in a composite service in a way that provides the overall security context for systems, services and applications. This imposes the integration of security constraints into service architectures, that is to say analysing the risk related to service deployment in a pervasive environment before defining how security needs could be integrated into the processes definition and then associated to the services by incorporate them in a global framework. This involves the adaptation of organizational issues (security policies, users’ directory, identity management or service agreements), as well as their underling technical issues (security protocols, encryption and signing or authorization policies).

Seeking a conjoint organisation and information system urbanisation strategy cross-cutting industrial service composition, a lean enterprise organisation and an adaptable and secure information system suggests the alignment of enterprise organisation and processes with the information system urbanisation initiative instead of basing it only on a purely informational view.

Our conceptual solution starts from a global vision of the enterprise, describes the elements characterising its organisation, processes and information system in order to define adaptable modelling components capable of integrating the aforementioned concerns while focusing on the security and risk management issues in such an open environment. This leads us to analyse and re-think the way we identify and analyze business objects such as enterprise resources and activities in such away as to allow integrating dynamic cross-organisations process management and security criterion in their definition, taking into consideration

internal and external stakeholders, in order to allow collaborative business process to be dynamically constructed while assuring the required level of security. The solution supports the definition of conjoint business and security policies framework which assures a supple transformation from business-oriented policy into technology-oriented policy. This includes expressing the business constraints and rules as generic policies able to be integrated into the process model and then describes how system components manage and implement policies, monitoring their execution and evaluating their effectiveness.

In the light of the previous discussion, in addition to a functional description of services, our architecture takes into consideration a service's non-functional properties. More specifically, in our architecture, the published services description must include a description of contextual security and quality of protection constraints and parameters.

To this end, we extended WS-SecurityPolicy, WS-Agreement, and WSDL2 by adding semantic annotations using security ontologies to describe the contextual security parameters and security policy in service descriptions and service level agreements. The selection of service is based on algorithms for compatibility matching between requests and services security constraints along with the other QoS constraints.

To implement a collaborative industrial SOA, the information system infrastructure must support SOA principles. Enabling the infrastructure involves providing the capabilities allowing the separation between the business logic or services and the technical logic, the way these services are invoked and composed. Such capabilities are amongst the many capabilities of the Enterprise Service Bus (ESB). In addition to many functional interoperability requirements such as an asynchronous communication service or dynamic routing, ESB can provide other non-functional activities related to the quality of services, security, failure detection & recovery, reliable delivery and transaction management. Yet, mediation plays the most important role in the ESB. Mediation includes transformation and synthesis of data that can go from basic format translation in order to match a particular standard, to more sophisticated analysis using ontology or expert knowledge, adding new values to the data. Synthesis may be done over multiple data sources, having heterogeneous data types.

However, to enable ESB to support "on demand" industrial services composition, we extend its capabilities with new functionalities in order to tackle unexpected risks coming from organisational levels. Consideration of the security requirements of all involved partners requires enabling the ESB with a flexible and cross-layers security policy definition and enforcement architecture. To this end, the proposed architecture assures the awareness of complex situations through intelligent security policy composition and verification

mechanisms. The used mechanism enables a compromise of the missions, goals and preferences of involved parties and enforces context-aware security policies in order to derive context-triggered security mechanisms and actions and make context-responsive changes conforming to the security policies of involved parties and to ensure satisfactory end-to-end security enforcement.

The architecture takes in consideration the heterogeneous nature of service-based B2B paradigms and the complexity of representing a consistent and globally agreed upon security model, as each party will likely have its own perception, policy, semantics and preferences about the security and how contextual changes should impact the security. The ESB is extended with a semantic compartment enabled by security ontology to deal with heterogeneous security policies, preferences and context. Security preferences are taken into consideration as a service's non-functional properties and incorporated into Service Level Agreement (SLA, which allows building a set of contracts defining the global framework of the collaboration.

This doctoral dissertation is organized as follows: in the state of the art we focus first on enterprise modeling (chapter 2) and production management issues (chapter 3) before studying the enterprise Information System (IS) organization and related technologies (Service Oriented Architecture "SOA") (chapter 4) paying a particular attention to security issues (chapter 5) and security integration in SOA context (chapter 6). This state of the art shows the increased call for business and technological interoperability to set large scale and dynamic collaborative organizations. Furthermore, collaboration is often limited due to the lack of trust, thus such collaborative organizations require agile and contextual security support services. To overcome these problems we introduce a conjoint production and information systems urbanisation strategy which leads to an agile enterprise organization that supports an incremental bottom up production-oriented vision and takes into account both enterprise's horizontal model (i.e. company processes), and its vertical model (related to functional business areas organisation). This will be achieved by introducing an approach to design a service-oriented resource-driven enterprise model and highlight an organizational framework to support "industrial services" identification and enactment (chapter 9). In order to take into account security requirements and preferences, chapter 10 proposes to couple them within the "industrial service" description and integrate security services at the middleware layer whereas chapter 11 details the implementation of this architecture. Lastly, we conclude by summarizing our propositions and define further works and perspectives (chapter 12).

Chapter 2

2 Enterprise Modelling

2.1 Introduction

Due to rapid evolutions in marketplaces, enterprises tend to focus on their core processes and outsource parts of their processes by establishing coalitions, virtual enterprises, alliances and partnerships. Such business trends require the dynamic collaboration of many partners to fulfil common objectives and redefine enterprise organizational structures to enable dynamic adaptation of processes in order to expose their offers and services to potential partners. In this context, collaboration constraints impact the design of organizational structures, production and information systems to allow “on the fly” composition of functionalities stemming from different enterprises to exploit volatile opportunities. Enterprises must continually analyze their value chains in order to set up adapted collaborative strategies with respect to the information system and the organizational structure.

To enable global adaptability and agility at all enterprises levels, enterprise modelling must take into consideration all aspects of an enterprise, including those of its production and information systems, in order to align the information system on the organizational structures and take into account the constraint issued from the production context. Consequently, adapting to collaborative business starts by understanding the different elements of the enterprise, modelling these elements in such a way as to master the complexity of the relations maintained between them in order to enable an efficient collaborative strategy, and, as a result, enterprise modelling appears to be indispensable to reach this objective.

Furthermore, to ensure proposing coherent services to its partners, an enterprise’s production and information systems modules must be derived from the analysis of production processes to facilitate simultaneous adaptations in all enterprise levels in response to different needs during collaboration. And to allow identifying “composable” industrial services to be published and offered, or, in other words to allow organizing interoperable functional constructs and insure the ability of managing flows between these constructs, and integrate all services required to complete a final product in response to “on demand” orders.

This requires providing the tools and knowledge necessary to understand the production system, concepts that constitute it and how production system elements interact, are organized and managed to produce value and how to enhance and optimize value creation.

In this chapter, we will expose the state of the art and principles surrounding enterprise modelling, production systems organisation and management in order to identify and adopt the basic concepts and notions enabling global agility and interoperability at all enterprise levels.

In the first part, we will present a short state of the art of different enterprise modelling methodologies, focusing mostly on process-oriented approaches and approaches aiming at continuous improvement and adaptation. This will lead us to develop requirements for information system organisation and collaborative enterprise strategy.

In the second part, we will introduce production management before introducing the lean manufacturing philosophy. This part is thought to understand collaborative production requirements, identifying the concepts that must be taken into account while adapting the information system urbanisation rules to the industrial context so that collaborative organisation can be set, enabling simultaneous evolution of the production process and that of the associated information system.

2.2 Enterprise Modelling Approaches

One of the most important roles of enterprise modelling is to provide representation frameworks adapted to enterprise system analysis. Today we find several reference approaches of enterprise modelling according to their use contexts (BPM, BPE, Enterprise Architecture, Service Oriented Enterprise, Information System Interoperability, Enterprise Information Systems as "Value Chain" Carrier… etc.), most of these approaches are led by the necessity of "business process" representation in the new information system design approaches which integrate the using context, motivated by the need of the flexibility and the strategic alignment between the information system and the other components of the enterprise architecture. In other words, the main objective of the enterprise modelling is to improve the added value of the information system by bringing coherence between the information system and its use context in order to match the service provided by the SI to the enterprise's Business Process value chain.

In the context of collaboration, modelling has a very important role in analysing complex business networks involving several parties to achieve a common goal. A collaborative enterprise must be modelled in such a way that takes into account, beyond the enterprise components, and enterprise flows, external flows as well as all interested actors like clients, partners and interactions with an enterprise's environment. Thus, collaborative enterprise

modelling requires paying particular attention to the selection of a modelling strategy; such a strategy should be able to take into consideration the particularity of the collaboration context. To this end, in this part of the state of the art we will point out the various enterprise modelling approaches and tools, as well as enterprise organizational approaches in order to identify the most adapted approaches to our problem; considering the impact of collaboration on the design of the enterprise information system.

2.2.1 Functional Modelling Approaches

One can distinguish between the enterprise functionality (its activities) and the behaviour of the enterprise (the choice and the order of activities' execution i.e. which are the activities to be carried out and in which order). In other words, the activity can represent the behaviour of the resources of the enterprise, and the process represents the dynamic behaviour of the enterprise itself by taking into consideration execution context elements such as time, cost and resources availability.

An activity contributes to achieving a process, using enterprise resources in order to transform input resources state into the output resources state.

Different modelling approaches can be used to support functional modelling, for example one can refer to IDEF0 (or SADT) [NIST 93] and IDEF3 [Mayer 95] (ICAM Definition Method). IDEF0 modelling method aims to answer the following questions:

- Which functions are implemented by the system?
- Which are the objects processed by these functions?
- Which mechanism or resources are necessary for the execution of these functions?

IDEF0 is based on two primitive graphs: boxes which represent the activities and arrows which represent the relations. Arrows do not represent any notion of causality but sequencing relationships between a source activity and a destination. IDEF0 distinguishes four types of arrows:

- Input objects (can be transformed or modified by the activity) like data or raw material.
- Control inputs (procedures, rules or constraints) used to define how the activity will be executed. These inputs cannot be modified by the activity.
- Output objects (data or materials) are the physical or informational objects produced or modified by the activity.

- Mechanisms: represent the necessary means to support the execution of the activity (human resources, machines or applications).

However, IDEF0 suffers from some lacks concerning the enterprise behaviour modelling (control flow). It only makes it possible to deal with the static aspects of the enterprise, discarding the temporal and concurrent evolution and execution. The IDEF3 method was proposed to overcome these limits, IDEF3 is designed to capture and describe the operational processes of the enterprise by using a graphical notation to model the process as a sequence of steps called units of behaviour connected by joining boxes and bonds. As a result, these units form a diagram called process control flow description. This description can be associated with Object State Transition Network diagrams (OSTN) in which nodes are associated to objects states, (represented by circles, and connected via labelled arrows (one or more behaviour units associated to the actions allowing an object to change from one state to another).

IDEF3 offers a good modelling approach to represent an enterprise's operational view allowing for the development of the information system which supports enterprise activities execution. However, it reduces enterprise representation into its operational view, and lacks a global framework to represent the complexity of enterprise context, this complexity requires multi-perspective modelling methodologies which can provide a coherent representation of enterprise components from its different viewpoints along the whole enterprise life cycle. To this end, a set of reference architectures have been defined. In the following we will review the most important reference architectures.

2.2.2 Reference Architectures

These architectures aim to provide a general framework and benchmarks to users by telling them which aspects of the enterprise they must take into account, the relations which exist between these aspects and the commonly agreed terminology in the modelling domain. Among the most known architectures, we can mention CIMOSA, GIM and PERA.

2.2.2.1 CIMOSA

Open System Architecture for CIM, CIMOSA [Li 94] is a modelling architecture aiming at building an integrated production system.

The CIMOSA modelling framework is based on three fundamental modelling principles, each of them associated to a particular modelling axis (Figure 1):

1- The generic axis which consists of three levels:

- The generic level where the basic primitives of the modelling language are defined
- The partial level containing the predefined reusable partial models and structures of a given field of application.
- The specific level associated to particular enterprise models.

The generic and partial levels belong to the CIMOSA reference architecture while the specific level is associated to the particular architecture of a given enterprise.

2- The derivation axe is used to define the modelling process according to 3 main steps: requirements definition (which permits writing the specifications thanks to the language used by the end-user), design specifications (which makes it possible to specify and analyze in detail the solutions meeting the expressed needs) and the implementation description (which precisely describes the implementation of the chosen solution).

3- Generation axe defines four modelling views (functional view, information view, organizational view and resources view) in order to reduce the model complexity by focusing on some aspects while neglecting others. These views are not independent sub-models but rather mechanisms giving access to some aspects of a same integrated model by filtering some elements.

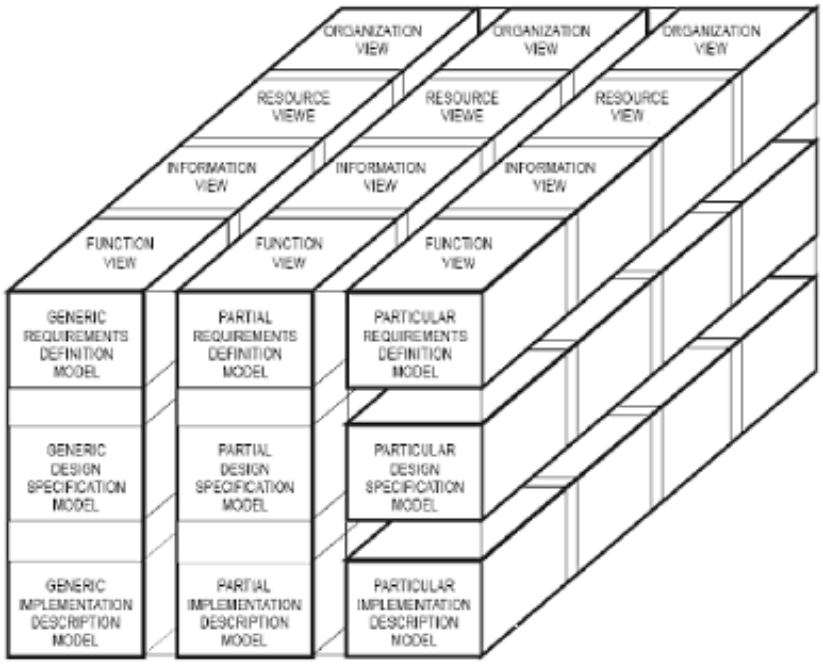

Figure 1: CIMOSA Modelling Framework, [Li 94] , page 119

CIMOSA is based on the process-operation-agent paradigm and provides the following modelling elements (constructs) according to three views (Figure 2):

- for the functional view:
    - Domain allows gathering a group of process in an independent model to deal with the complexity of the whole model.
    - Event is an instant which announces a change of state in the system and requires an action.- Process is a partially ordered sequence of steps started by an event to achieve a fixed goal. CIMOSA distinguishes two types of processes: the main processes (the processes of the highest level which are started by events) and sub-processes (which are processes defined inside the main processes).
    - Activity is an elementary step of a process, it is the position where the action takes place, and it thus requires time and resources for the execution of each of its functional operations. It has the role of processing the enterprise objects so it uses object states (defined in the object view) as inputs and outputs.

- For the informational view:
    - Enterprise objects are entities which exist in the enterprise and are used by the activity such as physical materials or data.
    - the Object viewis the state of the enterprise object at a given time. For the resources' view:
    - Resourceis an enterprise object which supports the execution of an activity. CIMOSA differentiates passive resources called components from active ones as they do not have the same characteristics.
    - Aptitudes group define the potential functionalities of a functional entity (service) or of an entity required by an activity.
- For the resource view: CIMOSA provides two constructs to model the enterprise organisational structure:
    - An organization unit describes the smallest organisation unit in an organisational structure. It is a role or a position held by a person, machine or an information processing system.
    - An organisation cell describes the organisation units gathering one or more organization units (vertical or transverse).

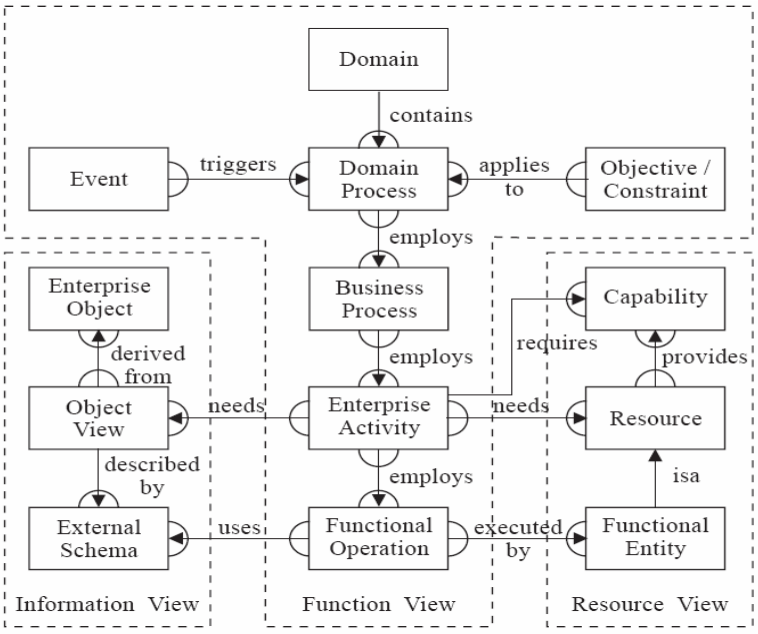

Figure 2 CIMOSA Enterprise Model Constructs [Li 94], page 134

CIMOSA carries an original answer of global integration starting from the description of requirements to the implementation phase; however the textual representation of behavioural aspects does not allow a clear visualisation of process behaviour and hinders the decision making. Additionally, it is difficult with CIMOSA to distinguish the boundaries between the different levels of modelling. Further, CIMOSA does not take into account human resource management. To overcome these drawbacks, GIM or PERA can be used.

2.2.2.2 GIM

Acronym of Grai-Idef0-Merise [Vallespir 92], it is a modelling and analysis methodology focusing on the enterprise decision system organisation. GIM considers four views (Figure 3): information (data and knowledge), decision (chain of activities and decision-making centres), physics (resources) and function (functional decomposition). The method consists in modelling the enterprise system at the conceptual level (requirements definition as they are perceived by the users) then at the structural level (definition of a technological solution validated by users) and finally at the implementation level (describe the implementation of the designed system by taking into account organisational, IT related and industrial technology related aspects. GIM does not provide a special modelling language or particular constructs, but it uses existing languages and constructs.

Although GIM provides a simple and global analysis to support the decision system, it is not adapted to most parts of industrial systems because it does not distinguish clearly between the

operational and information system and does not provide a clear specification for the underlying information system.

GIM provides an answer to the decision making question, however, it fails to provide a complete answer of human resource management and it does not take into account a model's lifecycle. To overcome these drawbacks, PERA can be used.

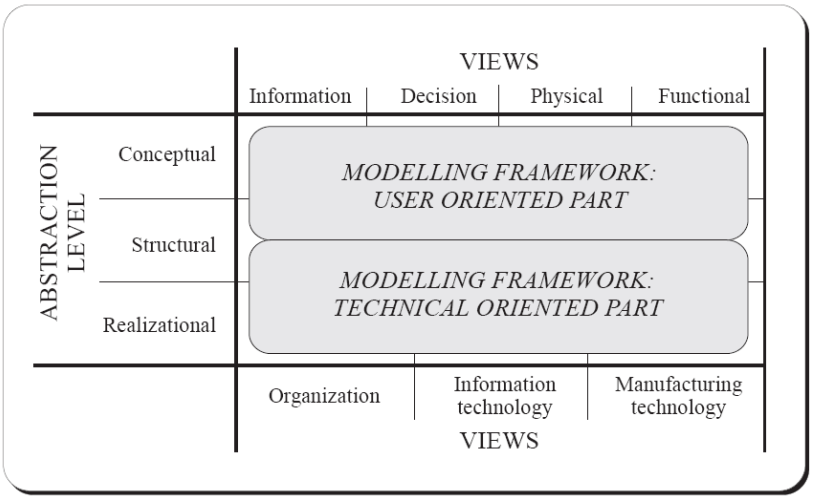

Figure 3 GIM Framework, [Li 94], page 159

2.2.2.3 PERA

PERA, acronym of Purdue Enterprise Reference Architecture [Williams 95], it is a complete engineering methodology of the industrial environment. This methodology defines all the life-cycle phases of an industrial entity. The architecture is organized in five phases (Figure 4):

- the Conceptualization phase consists of three steps: step (1) known as the identification step specifies the industrial entity study boundary and perimeter, step (2, 3) a concept step, step (2) defines the mission of the study in terms of policies concerning the product and the operational system and step (3) defines the mission of the study in terms of policies concerning the management of human resources and production management. From this two branches of methodology emerge: one for the operative part (machines and equipment) and the other for the information or order part (management of information system).

- As for the definition phase, each part this phase defines the requirements (4, 5), the tasks and the necessary entities to satisfy these requirements (6, 7) and provides flow diagrams for these entities (8, 9).
- The design phase includes two parts: the functional design part and the detailed design part. The functional design part is a specifications step allowing identification of the human tasks and the specifications relative to the human organization (11), the

architecture of the information/ management part (10) and the architecture of the operative part (12). The detailed design steps complete design for three parts: information/ management (13), human (14), and equipment (15) with a sufficient level of details for the phase of installation.

- The installation and building phase consists of (18) transforming the models resulting from the detailed design into a real implementation (machines, logistics system), (16) install the data bases and the control programs and test them, (17) prepare the personnel, carry out series of test and officialise the installation launching.
- The operational and maintenance phase corresponds to the effective use of the system; it must fulfil its mission as it defined by the administration. During the active life of the entity, certain of its parts must be reviewed and must go through reengineering or maintenance actions, they could be the information/control part (19), of the human tasks part (20) and the operative part (21).

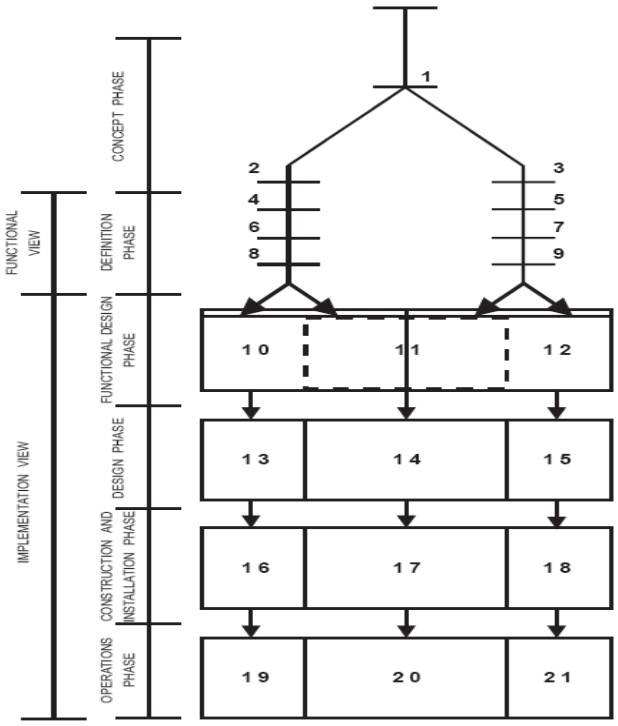

Figure 4 The Structure of Purdue Enterprise Reference Architecture [Williams 95]

PERA responds to the disadvantages observed in GIM, nevertheless, it remains partial, and does not provide a global framework to integrate all enterprise views, to this end GERAM has been developed.

2.2.2.4 GERAM

GERAM the integrated framework named GERAM, acronym of Generic Enterprise Reference Architecture and Methodology [IFIP 99] includes all the knowledge required to engineer and to integrate the enterprise. Integration aims to eliminate the organisational barriers and to improve interoperability in order to increase synergy. By this way, the enterprise will operate more efficiently and in an adaptive way. The enterprise engineering framework is designed as a collection of tools and methods which can be used to design and maintain continuously the enterprise model.

GERAM is defined by a pragmatic approach providing a framework generalized to describe the necessary components for the different steps and processes involved in enterprise engineering or enterprise integration--use cases can be for example the first definition of an enterprise framework, a re-engineering process, new models involved by the fusion, the reorganisation, the creation of a virtual enterprise or the integration of a supply chain--as well as those involved to maintain incremental changes due to continuous improvement and adaptation processes (Figure 5).

GERAM is designed to improve modelling methods integration, covering in this way several disciplines (industrial engineering, information and communication technology, automation, management science) used in the change management process. This allows using them simultaneously, instead of their stand-alone application.

An important aspect of the GERAM framework is that it unifies the different enterprise integration approaches: those based on process models as well as those based on products.

The most important component of the GERAM framework is GERA (generalized enterprise reference architecture) which includes the basic concepts to use while engineering and integrating the enterprise (for example enterprise entities, life cycle).

GERAM separates enterprise engineering methodologies (EEMs) from the modelling languages (EMLs) which are used by methodologies to describe the model, structure, contents and behaviour of the enterprise entities. These languages allow modelling the human part in enterprise operation as well as business process and their support technologies. The modelling process produces enterprise models (SME) which represent all or part of the enterprise's operations, including manufacturing or service processing, organization and management modelling, and its information and control systems.

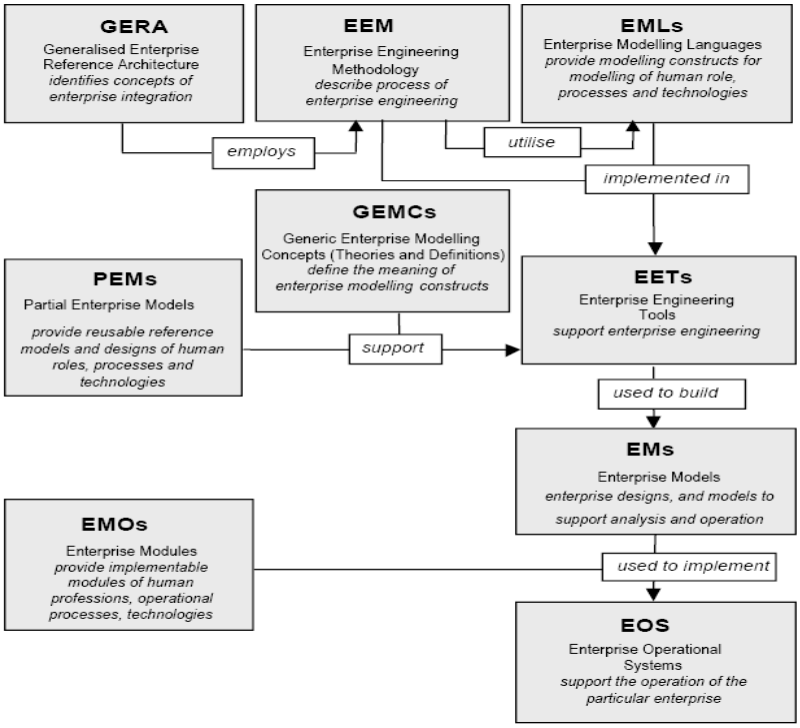

Figure 5 GERAM Framework Components, [IFIP 99], page 5

The Methodology and the languages used to model the enterprise are supported by enterprise engineering tools (EETs). The semantics of the modelling languages can be defined by ontology models, meta-models and glossaries, which are gathered in the generic enterprise modelling concepts (GEMCs). The modelling process is improved by using the partial models (PEMs) which are reusable models of the human roles, the processes and technologies. The operational enterprise models are supported by specific modules (EMOs) which provide pre-built components like human competencies profiles associated to particular professional fields, technological services infrastructure, or all other product which can be employed as component in the execution of the operational system (EOSs).

2.3 Conclusion

To adapt an enterprise system to a collaborative orientation, identification of the elements and constructs of the enterprise as well as the relations they maintain between each other must first be understood, which is evidently the role of the modelling. To this end, we have exposed different modelling frameworks and approaches. As a result, we can conclude that the multi-perspective modelling approach provides an integrated description of enterprise systems; however, the provided view is still static; and does not allow the required knowledge and tools to face the continuous change in enterprise processes and organization to adapt to dynamic collaborative production, this imposes taking into consideration production

management criteria during the modelling process, and requires new modelling approaches which take into consideration the particularity of collaborative production to enable agile, interoperable and adaptable organization of the enterprise at all levels.

Chapter3

3 Production Management

3.1 Introduction

Industrial enterprises live in a period characterized by change; they should work in a complex environment in continuous evolution, in such a context, enterprises have to restructure their organization and processes. This requires the analysis of different elements of the production system and processes in order to identify the services they can offer to enhance production process agility and adapt all enterprise aspects to dynamic collaborative strategies.

Organization, technology and human aspects structurally interact with the production process organisation and execution. Enterprises must bring consistency to these different aspects and the process framework to see global agility and to offer coherent and dynamic services to its partners. This global orientation involves redefining and considering the different factors related to production system organization and management as well as the impact of collaborative orientation on the production system organization, to this end, in the following, we will try to review the different production concepts and typologies, then we will be interested in the most important enhancement thought to improve production management efficiency and production system performance and adaptability.

3.2 Basic Concepts, Factors and Definitions

Production is the transformation operation of raw materials or components into products which have a value in the market, in accordance with the manufacturing process which consists of flows in which raw materials are transformed gradually into a final product.

The production comprises three elements in interaction [South95]: the Customer expresses a Need for a Product with certain functionality which represents the object to be realized by the Enterprise using its Processes. A need is explicit or implicit, confirmed or potential, needed or desired by a customer; a need must be formulated in requirement terms and not in solutions term. While a Product is what will be provided to a customer to satisfy his need by the functionality provided by this product [AFNOR 90]. A product corresponds to the “What” of the production system [ISO9000]. A product could be material or immaterial (service). To be designed, a complex product is initially described and broken up into components. We call

Product Breakdown Structure (PBS) this cutting of the product, otherwise called: product structure tree PST. PST is different from bill of materials [Lossent 07] which is an ordered material list, components, and subassemblies describing the product, and usually created and maintained as a particular function of the product configuration management, it defines the nature, the reference number and the number of components used, as well as the relations between components and assemblies. Another distinction is between bill of materials and product (or services) range [Lossent 07] which is all the products (and/or services) proposed by an enterprise to meet the same need.

A product is the result of production process execution; the process corresponds to the "how" or the dynamic aspect of the production system. It consists of all finalized operations. A process can be divided into sub-processes. Often a process is represented by a sequence of inter-dependent activities by relations of temporal and/or objective precedence, an activity is the description of a work to be achieved. This requires organizational specifications summarized in the question that, who does what and when? To improve production process efficiency, a set of activities is achieved (called Production control activities) [Fontanil 99], [Blondel 97]. Production control is defined as all activities achieved in order to optimize production processes by improving continuously the flows going from the suppliers to the customers by taking into account at the same time the production quality, time and, cost. The Production control is also defined as the function which allows carrying out the production operations in accordance with the quality, time; costs conditions resulted from the enterprise objectives. The objective of production control is to ensure equilibrium between the productive resources use rate, the work-in-progress and stocks level, and products delivery deadlines. Here, we should distinguish between the terms Stock and Work-in-progress. Stock [AFNOR 91] corresponds, at a given moment and a given place, to the quantity of an article not yet used either for work in progress, or for consumption purpose. This quantity is measured in a suitable storage unit. Stock is located at the interface of external enterprise flows. Work-in-progress [AFGI 91] represents the totality of articles present in the internal flow of the process, which could be on standby in front of a resource, in transfer from a resource to another, or in transformation in a resource.

An important aspect of production process is Flow [HPM]. Flows correspond to circulations of physical or informational entities through a process. In production, we can distinguish primarily two types of flow, namely material flow and information flow. Another distinction of flow relates to location in which the flow runs: external flows related only to the

provisioning and the distribution, whereas internal flows correspond to the internal production process (within the enterprise boundaries). Several types of flow management are practiced:

- Push systems [Blondel 97] when a production step is finished, it pushes the product towards the following step. In this system the product coming from the upstream triggers the following manufacturing step. Usually, this method of production implies storing the finished products before selling. For example, in the sugar industry we could not predicate the periods of beets harvest, which allow producing the sugar once collected. Thus beets should be transformed according to their availability, and thus storing sugar, without being concerned with sales.
- In Pull systems [Blondel 97]: triggering a manufacturing step cannot be done unless there is a request from the following step. With the pull system it is the downstream (the customer) who orders the upstream (the supplier).

One enhancement which could be applied to push and to pull flows is Lean Flows [Javel 04]; work in Lean flow is equivalent to work with the minimum stocks and work-in-progress. In the push system, Scheduling proves essential [Javel 04]. Scheduling is tasks execution organization, taking into account temporal constraints (needed time, sequence constraints) and constraints relating to the productive resources availability. In the production context it is the problem of insuring orders progress and control through the system. It is defined as the tasks execution planning to satisfy more than on objective ("order" and "calendar") according to productive resources availability.

An important issue related to collaborative production context is supply chain management (50), the flow management improvement by considering all the flows going from the "main supplier" to the "finale customer" as a total process. There are three flow categories circulate between the actors making the logistic chain:

- Material flow (going from the upstream towards the downstream).
- Information flow (circulating in the two directions, from the upstream towards the downstream: while following the physical flow. From the downstream to the upstream: feedback coming from consumer).
- Financial flow (going in opposite direction of the goods and which is often electronic (electronically managed information)).

Managing the logistic chain consists of smoothing circulated flows while optimizing costs as much as possible: depending on stocks, routing, handling and rupture minimization.

However, logistic chain management, Production control and scheduling functions are based in production management or computer-integrated manufacturing systems. These systems vary according to functions they integrate. MRP0 systems (Materials Requirements Planning) [South95] which are Planning systems determining the requirement of components depending on provisioning and finished products orders, can be found. By integrating the constraint of charges versus capacity of production machines (machines and equipment loads management issued from components requirements), we obtain what is called MRP1 systems [South95], while MRP2 also integrates the production costs management into requirements planning management. A software which manages all of the enterprise processes by integrating enterprise functions like human resources management, financial management or decision-making support, but also sale, distribution, stock, provisioning and electronic business is CAM software [South95](Computer-Assisted Production Management), a modular production management software which allows managing the whole of the activities related to the production of an industrial enterprise. Recently the CAM functions are commonly incorporated in enterprise resources planning (ERP) software packages which apply to all enterprise's functions. Enterprise Resource Planning or ERP [South95], the principle idea behind an ERP is to build a computer application in a modular way (inter-independent modules) by sharing a single and common data base. That creates an important difference with the pre-existent software packages because the data included is standardized and shared, which eliminates the multiple value of the same data and avoids (in theory) the ambiguity.

The other important aspect which characterizes an ERP is the systematic use of what is called a "workflow engine" (predefined automated and eventually "parameterable" system) which makes it possible to propagate data entered into the information system in all the modules of the system which might need it, according to predefined programming.

A key element of any production management system is production control panel (52), a reduced sample of statistical indicators allowing a manager to follow the evolution of results but also the dissimilarity between results and objectives, in real time (as much as possible) in focusing on those factors considered as the most significant. The data treated in this kind of systems are generally generated by a management information system.

3.3 Production Evolution and Production Typologies

The production was initially artisan and required highly qualified personnel, which led to produce limited product quantities having a great diversity. Later, demand was higher than the offer, it was therefore enough to produce then to sell. During that period, production had the

following characteristics: low diversity, mass production, little qualified personnel (thanks to work decomposition into simplified elementary tasks).

Gradually production became rather equivalent to demand; the customer had a choice of supplier. Suppliers had to produce what will be sold. It was thus necessary to make commercial forecasts, to organize the provisioning, and to control stocks.

Eventually, markets became strongly competitive, production was higher than sales. The principal characteristics of this period are: costs and quality management, short and reliable delivery time, customized and short lifecycle products.

Finally, the enterprise must evolve in a volatile market where customers are unforeseeable, unfaithful to brands, delay purchasing and yet are sensitive to quality. This period entrained the use of new methods which stress the concept of "just in time." Enterprises must produce what is already sold. The principal objectives of these methods is to compromise between minimal stocks, minimal delay and reduce production costs by limiting investments and having flexible resources.

Clearly, each enterprise has a specific context issued from many factors, like product nature, demand fluctuation or markets evolution. This requires adaptation of the production system. Many production typologies can be distinguished.

In the following paragraph, we will present a summary explaining the various criteria of production classification. We should mention that a production system can satisfy one or more of these criteria at the same time.

To reach an agile and adaptable production system the impact of market evolution, relation with customers, the nature of products, and physical and financial constraints on production system organisation and typology must be understood. This will allow us to emphasize and incorporate rules and constraints issued from these relations during the phase of production system integration in our urbanisation strategy. To this end, we focus on the next part of the state of the art in production system evolution and typology [Courtois 03] and and [Blondel 97].

3.3.1 Typologies According to Production Mode

We can classify the production according to production mode modes into four categories:

- One of a kind in which the product is complex, requires the intervention of several parties in order to be delivered at the desired moment and quality, good example is a civil engineering project.

- Continuous productions in which products are processed by means of operations perfectly synchronized on the operational time level e.g. steel and cement industry. In this kind of production systems we find dedicated production equipments with a very high level of automation.
- Mass production in which the products are carried out by the manufacture and/or assembler in huge quantity, and with very few variations. Productive Resources (workforce and machines) are specialized and adapted to specific tasks. Level of automation is generally high; a good example is food production.
- Production in small series in which it is the case of customizable products; basic products with variant. The production resources are multi-purpose, flexible, and able to change quickly from a product to another. The level of automation is generally low; a good example is clothes production.

3.3.2 Typologies According to Product Structure

We can classify the production according to product structure into three categories:

- Complex products by means of a tree with several levels "tree-structured nomenclature:" This kind corresponds to manufacturing industries such as the automobile industry, electronics, electric household equipment or furniture to name just a fez. In this class, three types of structure are distinguished:
  - Convergent "A" products are achieved starting from raw materials to assemblies. This structure is characterized by multi-levels tree arrangements each of which corresponds to subsets of the end product.
  - Convergent "T": this structure seeks to bring together the mass production and products customization. It is a modularly designed product intended to allow the achievement of customized products by combining in various ways the subassemblies of which some are standardized.
- Divergent and Parallel Structures correspond to manufacturing of little varied products, in very large series, during very long periods of time with relatively stable markets. In this class we distinguish two structures:
  - Divergent "V" in which the products are carried out starting from the transformation of very little number of raw materials: oil, milk, steel, etc.
  - Parallel in which the products are carried out starting from some raw materials slightly transformed: packing industries.

- Products Regrouping Points “X” in which the products are adapted to a specific need of customer and obtained by multiple combinations of Semi-Finished Products, this type of product corresponds with a product of the type T obtained by a combination from type A and V. We obtain a limited number of semi-finished products according to a type A structure, then, we manufacture various finished products of type V.

3.3.3 Typology According to the Circulation of Products in the Workshop

Two main classes are distinguished:

- Circulation in Job Shop in which the products circulate from machine to another according to a routing which corresponds to their manufacturing range. This type is intended to manufacture a large variety of finished product.
- Circulation in Flow Shop in which all the articles follow the same track as in dedicated transfer lines where the articles “visit” systematically each working station in the line, always in the same order.

3.3.4 Typology According to the Relation with the Customer

For the enterprise, two situations can arise, according to whether the product corresponds to customer acceptable delivery delay; in this type we can distinguish two categories:

- An immediate need indicates that the customer’s acceptable delivery delay is zero; thus, the enterprise is forced to produce on Stock, a production on stock is started by the anticipation of the demand. This type of production is employed in several contexts:
  - The range of the finished products is limited.
  - The demand of the products is relatively foreseeable.
  - The manufacturing lead time (purchase + manufacturing + assemblage + delivery) is higher than the acceptable time by the customer.
  - The seasonal variation of the product is very high to justify maintaining continuous production.
- A deferred need indicates that the customer accepts “not null” delivery delay; consequently, the enterprise can produce on demand. However, in this type of production manufacturing lead time must be lower or equal to the acceptable time by the customer. Theoretically, if the earlier condition is satisfied, no stock is necessary. If the earlier condition cannot be satisfied (the manufacturing lead time is too long) it is possible to anticipate the purchase and manufacture the components, and then carry out the assembly as soon as we receive a demand. This

implies good forecasts of sales in order to avoid an excessive stock of components. Indeed, in this type of production, the finished product can be personalized as late as possible in downstream.

3.4 Production Organization

Since the production means, physical and human resources are structurally interrelated to the production process organization and execution, and as our aim is a conjoint urbanization strategy integrating production and information systems in order to expose "industrial services," the comprehension of production organisation proves essential.

Machines and competences regrouping mode in an enterprise's workshops is a key element of production organisation. We can primarily distinguish three production organization modes [Courtois 03]:

- Functional organization regroups the machines which have the same techniques or the same functions (mechanical, electrical...etc.). Generally in this type of organisation assemblage is separated from manufacturing. Reception of the raw materials is centralized. The advantages of this organization are:
    - Job regrouping denotes people who work in the same sector are specialised in this sector. They can shift easily from a machine to another.
    - Flexibility suggests that the location of machines are independent of the manufacturing ranges, it is thus possible to manufacture a wide range of products using workshop equipment without disturbing material flow.

  Yet, this type of organisation has some disadvantages such as complex flow with many accumulation points and plenty of "work-in-progress" which includes long lead times because of flow complexity.
- The organization in production line indicates that a set of operations is carried out along the production line. We find this type of organization in continuous production. The machines are placed on line in the same order of the product range. This organisation is characterised by Simple flow management due to autoimmunization and auto-transporting and high productivity and gain in long term, but high infrastructure cost and provisioning quantities.
- Organization in cells (or small islands) denotes a set of operations carried out on a small number of stations, organized in autonomous units: An organization in cells

is constructed from small production specialized workshops each of which produce a set of similar products. It is a compromise between the line and functional establishment. This type of establishment makes it possible to decrease stocks and deadlines in the case of discontinuous processes; it has a good distribution of resources load, high flexibility, workshops autonomy, low infrastructure Cost, but needs a well trained and skilful workforce, complex management due to flows complexity and reduced global visibility.

Whatever the type and chosen mode of production organisation, the main objective is to optimize the production processes in order to improve the quality of products, reduce their cost and time to market. A key element to reach this objective is the continuous improvement of global flows coming from the suppliers to the customers. This suggests the concept of lean manufacturing [Womack 03] which is an industrial process management philosophy derived from Toyota's production.

In the following we will discuss this philosophy, examine in details the various aspects of this concept its advantages, its impact on the information system, and discuss some research carried out in related fields and how to adapt its principles to collaborative production strategy.

3.5 Lean Manufacturing

Lean Manufacturing focuses on the reduction of all losses in the production process in order to improve the total value delivered to the final customer. The main activity is to continuously re-examine different flows circulating in the enterprise with the aim of improving them and eliminating the anomalies and impediments from them. [Fontanil 99] has identified two principal types of flows in the production system; Physical flows and Informational flows. Physical flows consist of purchased and manufactured components, subassemblies sets or finished products, while informational flows consist of client orders, work orders, provisioning orders, operational cards or monitoring sheets to name a fez examples. The principal way to smooth physical flows is to better organise the workshop floor (machines and equipments positions in the workshop). Several methods were developed to meet this aim: Group technology [Suresh 98], Chainon, King, and Kiusaku algorithms [Mondon 05]. Studying these approaches in detail would go beyond the scope of our work. Other aspects can contribute also in Physical flows smoothing like avoiding machine failures, series change times reduction and workforce versatility.

Whereas smoothing informational flows is achieved by enhancing data exchange and communications means, and guaranteeing information coherence, the last issue is rather difficult to achieve due to the ever increasing quantity of information, semantic diversity and data distribution in the different information system's services and modules.
Let us start by defining the principal terms used in this field:
In the following we will discuss in more details terms, principals and methods related to lean manufacturing.

3.5.1 Terms Principals and Methods Related to Lean Manufacturing

The principal idea behind Lean thinking is the identification and elimination of useless actions in order to reach a better exploitation of enterprise resources and to increase the added value to the final customer. Added value [Fontanil 99] is the difference between the product value and the consumption of goods and services to carry out this product. According to [Womack 03] only an activity which changes the product state could add value. The identification of such activities (which add value) starts by Value Stream identification [Joiner 94], it is all essential activities (with or without added value) required to produce a product, starting from products design to products delivery including the customer services (after sale). In treating this chain, one can identify which activities add value and those which add only the cost. Activities in a value stream can be divided into two categories main activities including Upstream logistics (material handling) and downstream logistics (distribution, sale, and after sales services) and support activities, this category includes human resources management, technological infrastructure development activities. We should mention that each main activity implies its own support activities.
By identifying Value Stream, one can distinguish three types of activities:

- Those which create the value to the customer.
- Those which do not create any value but are essential and thus cannot be eliminated.
- Those which do not create a value to the customer and can be eliminated immediately.

Many kind of Loss [Monden 97] can be identified; primarily work money, space, time, and information losses. It is all extra quantity of equipment, materials, parts, and working time more than the minimum quantity necessary to fulfil the objective. Toyota, Just In Time JIT [Javel 04] concept creator classifies loss as [Monden 97]:

1- Overproduction: produce too much or too early.
2- Semi-finalized Parts stored between the operations.
3- Transported parts.

4- Useless stages of transformation.
5- Defects parts with useless labour movements.
6- Waiting: workers await machines or parts, or machines await parts.

However, loss elimination is not straight forwarded task enhancing added value require a deep reflection concerning all production system elements and it may require a fundamental changes in production system in order to reach a Process oriented Flow (5). The objective of process oriented flow is to convert the machines functional organizations in the factory into processes series (cellular manufacture), based on production families, or products. Process oriented flow is better than the traditional functional organizations since it reduces the distances, surface required and total time of production.

Another concept related to lean manufacturing is Just in Time (JIT), JIT is generally used to describe a stockless manufacturing approach where only the necessary parts are provided (or delivered by suppliers) at the right moment. JIT represents five conditions: (1) to produce at the right time, (2) with the right rhythm, (3) the right quantity, (4) with right quality, (5) in the right place. JIT philosophy mentions that flow equilibrium is much more important than speed. All operations occur according to the same rate/rhythm (cycle Duration). This is called sometimes "Takt time." A balanced production system means that all its components are conceived to work at the customer order rate/rhythm.

Nevertheless value stream smoothing according to lean thinking is not a static task achieved from time to time, it is rather a progressive and endless improvement; a continuous improvement strategy carries out (typically) incremental improvements by the participation of all enterprise actors': from workmen to managers.

Lean thinking does not concern only the internal parts of the enterprise, it involves also enterprise customers. One of the most important ideas behind lean manufacturing is to let the customer "pull" the value; starting point for Lean is that the "value" should be defined by the final customer, value must be defined in terms of desired products or services' functionality and can be defined only by the customer. A product carries the value when it satisfies the customer requirements in terms of price, delivery time and functionality.

Clearly, in lean thinking, creating value is the reason for which an enterprise exists. In the collaboration context, value has more global meaning as it should not be only created for final customers but also for all the participants involved in the process, thus the creation of the value requires that the value be only defined after the identification of all the parties involved in the process. This implies offering an adequate value for each participant involved; in other words, actors' requirements must be taken into consideration. Hence, the interdependences

and structure of the collaborative process must be well understood. A value stream becomes all -end to end- activities included in the “collaborative processes” and necessary to transform the raw materials into final product. This implies to think beyond the enterprise borders and to integrate into the process all actors contribute in the production. Generally in the case of collaboration, we can identify two important sources of loss in addition to traditional sources: Lost opportunity cost and Structural inefficiency. The cost of markets’ opportunities loss is due to badly defined collaboration strategies. The structural inefficiency loss is produced by inadequate organization structures, or bad business models construction which does not integrate all process’ owners (collaborators, suppliers, and customers). Any structural inefficiency existing in any involved enterprise at any levels could create loss and obstruct the value creation for some or all the participants. Inadequate organization structures include but are not limited to badly integrated activities into the enterprise process, useless interfaces between entities which hinder collaboration and coordination between involved parties. Lean paradigm supposes interface rationalization and co-operative environments between all involved parties to achieve goals efficiently and facilitate exchanges, starting from product design stages. This means that the internal or external interfaces between two activities must be minimized, standardised and improved in such a way that the product does not meet any resistance while moving to next process stage. That can be achieved by integrating production teams along the value chain.

Since our main concern is the integration of the production and information system to promote collaborative business, a better understanding of lean manufacturing and how to adapt its principles to collaboration scenarios proves essential. To this end, in the following, we will summarize some research tasks carried out with the aim of studying lean philosophy and its impact in the B2B context.

The research carried out in [Mahoué 01] focuses on the impact of new B2B models on the lean enterprise. This work initially examines the Internet phenomenon using four case studies: Dell Corporation, eBay, Linux and Enron. From these case studies, a set of learned lessons can be extracted which will be further employed to analyze the business tendencies in Internet world (e-World). The concluded principles are:

In this work e-World tendencies are examined alongside lean principles and the interrelations between the two sets of principles are analyzed in detail. Thanks to this analysis, a clear intersection between e-world and the lean principles is established, as a result a set of conclusions are observed:

- Business objects definition and identification in collaboration environments is difficult and objects data could be technically difficult to manage due to the multiplication and heterogeneity of information systems.
- Value steam integration is limited by many factors and especially by trust, institutional and legal constraints.
- An adaptation of the business model is difficult due to the ever-changing collaboration context and production complexity, adaptability of business model must start from the local and individual businesses and goes up to the global collaboration framework.
- There is a need to introduce a real time adaptation system to stimulate adaptive behaviour of the business model.
- There is a need for standardization.

We can conclude that attempts to develop interoperable and co-operative production information systems encounter many difficulties; these attempts are confronted with technical, organizational and legal difficulties. [Antonelli 99] examines some of these problems and suggests methods to overcome them. This research focuses on two axes, first the identification of aspects which could facilitate collaborative systems development, second, analyzing the result of the technical data exchange information systems in the logistic chain experiments. All together, these two axes have taken as a base to identify "enablers" of collaborative manufacturing information systems integration. The most significant success factor identified is that a manufacturer must have trust relations with his partners in order to establish an efficient information system which can contribute in a real collaboration. It concludes that developing an information system supporting industrial information exchange requires an enormous trust between participating companies. Technical factors, such as the choice of the software package and the common application platform seem to be less important than this factor during the development phase of an information system which binds the manufacturers with their networks of partners.

The work proposed an approach to design an information system for partners' integration according the following steps:

- Process identification identifies the process to be carried out along with its owners including all partners.
- Evaluation and synthesis clearly defines the processes (internal and external) that the proposed system should support.

- Specifications define the exact functionality that the system will provide to support these processes.
- Choose architecture easy to exploit in order to apply the functional and security specifications, while integrating all the external and internal relations of customers and suppliers
- Determine the integration strategy of all the old systems involved.
- Evaluate the alignment with the main partners business' strategies.
- Evaluate the alignment with the information technology strategy.
- Analyze the costs and the proposed system advantages to the main suppliers.

In a complementary way [Graebsch 05] explores and analyzes by a case study information and communication aspects from the "lean production" development view point. At the beginning, this work justifies focusing on the lean development. it seeks to understand which are the most frequent types of losses in information transfer, and which are the relations between the loss and the communication means, in this research work, implications and influences that information and communication impose on the lean development are discussed, studied and analyzed at the theoretical level, to conclude recommendations related to information and communication fields.

Information flow problems and limits are identified by juxtaposing information and communication concepts process with value and loss concepts.

As a result the bad quality of information and inefficiency of communication means are identified as the most important factors of loss.

Contrary to previous works which treat the impact of lean philosophy on the different aspects of the production enterprise, [Shah 03] examines the effects of three contextual factors, namely factory size, factory age and unionization level on the possibility of applying the lean production systems. As a result, the work conclude that that old factories are more resistant to the changes required to apply lean practices; the results show that the old factories are less ready to apply only some practices of lean. However, the age of the factory is not an important factor for many other lean practices. In the same way, the size of the factory does not constitute a criterion which can justify or not lean practices adoption: no difference was found in the cross-functional quality and labour management programs execution possibility for this aspect.

The impacts of enterprise size on the lean strategy adoption were also discussed in [Seitz 03] but this time from another viewpoint: the case of small enterprise. This work presents a new

framework of lean enterprise for the small companies. The lean paradigm is explored and the specific requirements for the small companies are considered. A specific theory called "natural leanness" is explored to explain the lean behaviour of small enterprises in the absence of a formal lean architecture.

Authors affirm that the small companies have a "naturally" lean behaviour. This behaviour is \implicit rather than explicit. This "natural leanness" does not contain an explicit architecture or a system infrastructure which could support the long-term growth, thus the same elements which provide its lean behaviour limit the future growth.

This thesis identifies and describes the current lean tools which can be used without massive modification companies' structures. Two new tools are presented to facilitate companies' lean transformation: the first is a "Dependency Structure Matrix" technique (DSM) to deduce the stakeholders' values. The second is the use of "Throughput Accounting system" to measure the progress of a lean transformation in respect to organization goals. The resulting collection of tools allows small enterprises to gain the strong points of lean without adding useless costs, and to create an operational framework which could prevent the auto-limitive behaviour regarding growth possibility.

3.6 Conclusion

Collaboration involves an agile organization of production system and processes in order to enable the composition of coherent, end to end production processes starting from different partner processes, an enterprise's production system is not, however, the only aspect affected by its collaborative orientation. To it, several constraints on the enterprise information system (IS) and organization must be added.

Collaborative production processes are transversal due to their nature i.e. it involves a transversal or horizontal organisation, process management must be reorganised in such a way to allow collaborative process construction starting from divers processes belonging to different partners, in other word to increase the level of enterprise organization interoperability.

At the implementation level as the enterprise's IS supports and/or carries out its processes and is used to manage company activities and since collaborative production needs an intensive use of information and communication technologies. It is therefore necessary to align the organization of the information system on the enterprise organization and to guarantee information system agility in order to be able to respond quickly to changes resulting from the evolution of the enterprise's business process organisation due to collaboration.

To by-pass these constraints, the information system must be reorganized to facilitate its evolution and increase enterprise interoperability both at organisational and technological levels. Organisational interoperability imposes adapting organisation structures and means to order to attain a dynamic organisation which is capable to provide suitable and instant responses to diverse collaboration scenarios.

Chapter 4

4 Operational and Information Systems Organization

4.1 Introduction

In the previous chapter we discussed the main enterprise modelling frameworks and production organisation methods, it is essential to take into account and well situate these approaches with regard to our areas of interests, namely an agile and conjoint organization of both enterprise information and production systems as well as the impact which could have the collaborative orientation and continuous adaptation on the global organization of the enterprise operational and support systems in order to approve the most convenient approaches which will impact our technical choices and serve as base of our further proposition to integrate production and information system.

To promote a dynamic collaboration, two issues should be taken into account: first of all, at the organizational level, we must reorganize the enterprise in order to consider interoperability and agility constraints, as well as the particularity of collaborative production, seeking a dynamic collaborative production involves agile and adaptable organizations which are able to reduce to the minimum extent the impact of the contentious change in the environment issued from the dynamic collaboration, for instance, by confine these impacts to local systems.

Second, since collaborative businesses involve an intensive use of communication and information technologies, the choice of the software solutions and the common application platform which binds the manufacturers with their networks of partners proves crucial. Such solutions should ensure flexibility, reactivity and interoperability at both production and information levels i.e. the ability to compound dynamically business functions issued from divers and heterogeneous systems. Nevertheless, traditional approaches of information systems organization generate fragmented, complex and rather rigid systems, this hinders the dynamic adaptation which is an essential feature for adopting collaborative production, consequently new approach of information system organization is needed, an approach insuring the adaptability and the integration of production constraints into the information system. Information system urbanization strategy coupled with service-oriented technologies can be exploited to organise a flexible and evolving information system.

Integrating service-oriented approach while organising business processes can improve its flexibility, reactivity and interoperability, in the other side, information system urbanization

leads to rather de-coupled business support software which can evolve or be removed separately while preserving their capacity to interact with the rest of the system increasing the level of information system adaptability and agility.

In the following section, we will start by introducing the main organizational approaches which could be applied on the production and information systems, discussing their advantages and drawbacks in the context of collaboration, then we will introduce the notion of Information System Urbanization, thought to bring to light an agile and evolvable information system, We will outline then the different definitions related to SOA in order to couple these approaches to be applied simultaneously to production system and its supporting information system. This will enable us in the contribution to describe our own process of migrating towards an urbanised and service oriented production enterprise.

4.2 Enterprise Organizational Approaches

Structuring the organization of an enterprise into "Business Areas" can result from the application of various criteria such as actors, functions or specialties, processes, product lines, segments of customers or markets, geographic areas, or a combination of some or all of these criterions.

Hierarchical and essentially functional, these organizations are considered the dominating approach in organizing and structuring companies. This leads to consider hierarchies of functional units which are responsible of carrying out different activities and thus structuring these activities within organizational functional units rather than considering cross-organization interrelated activities. This hierarchical structure produces specialised groups of actors which focus on their own activities. This functional specialization has led to many problems [Simon 95]. In particular, information subsystems implemented in different units were likely to be incompatible with each other resulting in product delays, increase in complex workflow, multiple transactions, considerable costs and quality of service problems [Vanhave. 99].

A real challenge stems from managing horizontal processes that cut across functional boundaries and fall under the organizational purview of multiple operating managers. Since an enterprise embraces process management, it becomes difficult and undesirable to maintain the crystal-clear lines of authority and responsibility typical of functional command and control. This conventional command-and-control structure simply does not facilitate required cross-functional flexibility to satisfy process goals. Indeed, collaboration between parties

requires the organization of a common process. This leads to privilege a process-based approach.

Organizational architectures that support cross-enterprise horizontal processes are complex. They entail a further redesign of traditional and well-established organizational structures to enable effective integration of internal capabilities between various business areas. In the case of a true cross-enterprise process, a company must modify processes to incorporate and fuse the capabilities and competencies of independent business areas. This approach promotes goals, activities, and resources needed to achieve goals, rather than organizational hierarchy. In this context, the first mission includes the identification of activities, their integration into the process, evaluation of their interdependence and ability to meet results according to stated objectives.

One structure often used to facilitate cross-functional integration is commonly referred to as a matrix organization [Gunn 07]. Under this approach, two managers share responsibility for an enterprise. One senior business manager focuses on financial and operational aspects, retaining responsibility for specific business units that are often organized around product categories, geographic proximity or class of business. The other manager devoted to resources, he is responsible for the deployment of human and physical assets to meet the requirements of processes enactment. Individual business managers are assigned skilled personnel from resource pools on the basis of projected requirements. While skilled personnel are directly responsible to the resource manager, they are supervised by the business manager. The business manager has direct authority for work design, temporary reassignment of functional staff and project control. Upon the completion of a business assignment, skilled personnel are re-assigned by the resource manager. The matrix model, however, is more difficult to implement in manufacturing. Companies in such sectors have complex material flows stemming from several factors including the company's global strategy, nature of products, client need urgency, product diversity and customization, the product implication level in the various production sectors and, finally, local constraints related to each production sector such as the local charge or machine capacity.

Additionally, production processes are highly complex and involve well-trained human resource competencies, as well as specialized and sophisticated devices and equipments for each product category. Such business sectors require an organizational approach similar to the matrix, but one which takes into account these constraints. This approach aids in the management of various production processes, their resulted flow and involved objects and accurate sharing of human and material resources. Such an approach should enable each

company's resource object to be exposed from multiple points of view with a multi-faceted object representation. This enables the management of different production aspects while conserving global coherence and production system integration along with company organization.

Corporate organizational approaches also distinguish between bottoms-up and top-down Organization. In the top-down approach, the overall process model, which serves as departure point, is broken down into its individual elements. This implies that, according to specific business requirements, most or all of the components are designed. theses components are executed in a predefined order and sequences when certain distinct conditions occur. The top-down approach is not convenient for collaboration scenarios in which processes are often dynamically defined (i.e. on-demand). It may also lead to a loss of business opportunities. Additionally, since top-down approach suggests predefined process, it leads to specialised and fragmented supporting information systems, and restrict organization adaptability, as we need to align the organization on the exciting information system modules each time there is a need to adapt the organization to changes in the environment.

Inversely, a bottom-up approach is used if specific functions have been defined and the goal is to combine them into a single process. The main focus when using bottom-up approach is the definition of process phases themselves using an available set of fine-grained components (activities), it is easy then to add to lower, foundational layers new activities and standardize the interface in upper layers. Bottom-up approach advantages can be grasped if required features are known in the light of collaboration [Reithofe 97]. However, bottom-up strategies may result in a tangle of elements and subsystems, developed in isolation and subject to local optimization as opposed to meeting a global purpose [Gadomski 98].

Another possibility is to navigate between the two approaches. Solely using a top-down approach may lead to a loss of opportunities. In contrast, a bottom-up vision can engender a situation in which we make superfluous items. Thus, a top-down approach of well-defined scenarios and best practices can first be used down to basic functional elements after which a transition can be made using the obtained elements to build new scenarios and new undefined processes.

To by-pass these constraints, we must rationalize and reduce information system complexity by simplifying and homogenising its functionalities into elementary activities in order to facilitate its evolution and increase its level of its modules "composability." To achieve this goal, an IS urbanization strategy coupled with service-oriented technologies can be used in order to organise a flexible and evolving information system.

4.3 Enterprise Information System Urbanization

4.3.1 Introduction

Collaborative scenarios involve the collaboration of many parties to fulfil common objectives. This collaboration entails many constraints on the company organizational structure and on the way by which information systems are designed and organized. As collaboration bring together several parties in a composite process formed from various processes carried out by different partners in order to achieve a common goal. This imposes several constraints not only on the enterprise organizational structure but also on the design and organization of information system (IS) which supports the creation and implementation of these processes. However, enterprise information system is traditionally designed to operate in a relatively closed environment where interactions with the environment is strictly reduced to some predefined forms of information exchanges while, in the collaboration context, enterprises information should be designed to be ready to "inter-operate" in a collaborative process. Furthermore, traditional information system organization strategies are designed to withstand in relatively stable environments where changes in the enterprise's context are limited in one hand and without taking into consideration the collaboration possibility in the other hand. Thus, enterprises must adopt new structural strategies for their information systems that can take into account changes resulted from such scenarios. The problem is to make the enterprise and its information system as responsive as possible with regard to changes in enterprise markets and strategy, while protecting information assets, local autonomy and enterprise operational performance. To reach the required level of information system agility, that is to say the ability to adapt to structural changes, it is possible to restructure the information system using principle of urbanisation.

4.3.2 Basics Concepts of Enterprise Information System Urbanization

Enterprise information system Urbanization aims at simplifying it, optimizing its added value and making it more reactive and flexible to fit the enterprise strategic evolutions paying attention to technological opportunities. It involves the information system re-organisation as well as the organisation of a progressive and continuous transformation of the information system. Urbanising the information system aims at designing a modular applicative architecture (supported by an adapted technical structure), consistent with the business architecture and aligned with the organization's strategy [URBA-SI 02].

Traditionally the enterprise information system urbanization initiative consists in studying the different business areas (production, administration, sales, etc), in order to carry out a cartography, then studying in the same manner its information system [Longépé 04]. It relies on cutting out the information system in subsets (zone, district, and neighbourhood or block). A zone gathers districts which gather small neighbourhood. The basic element are homogeneous replaceable and developable separately information system components (taking into account the nature of information handled in each of them).

The neighbourhood or block corresponds to the lower level of decomposition of the information system; the block is an entity of the system associated to a functional purpose and includes data processing and data access functions in order to achieve this objective. A small block emits standardized results which can be used by other small blocks [Longépé 04]. Although the principal objective of the urbanization is to have an evolutionary and adaptive system to face continuous environmental changes, the information system issued by a traditional urbanisation strategy fails to respond to on-demand production and dynamic collaboration between partners. In fact, a company must adapt to market changes by production structures and the organization of its processes and not the flow of its information alone. Production enterprises have complex material flow issues due to several factors including an enterprise's global strategy, the nature of products, urgency in client demand, product diversity and customization, product implication level in the various production sectors and, finally, local constraints related to each production sector such as local charge and machine capacity. Additionally, production processes are very complex and involve well-trained human resource competencies and the use of specialized and sophisticated devices and equipment for each product category. The production business sector requires an organizational approach which considers these constraints. Such an approach allows the various production processes and their resulted flow and involved objects to be managed and ensures the accurate sharing of human and material resources. Thus the challenge remains in allowing the information system and the organizational structure to simultaneously evolve taking into account these production constraints. In other words, what is needed is a conjoint production and information system urbanisation strategy which suggests the alignment of production system organisation and processes with the information system urbanisation initiative instead of basing it on a pure informational view only.

To get an urbanised information system we must comply with the city planning rules which consist in obligations, prohibitions, and limitations in order to constrain the component building process. These rules according to [Sassoon 98] are as the following:

1- Membership: in the hierarchical decomposition (zone, district, block), a block belongs to one and only one district which belongs to one and only one zone.
2- Autonomy suggests that there is no dependence between the blocks, i.e., a block can process one event from beginning to end, without call for processes carried out in other block.
3- Asynchronous: a block can process an event and send the result without waiting
   answer of the recipient.
4- Anchoring point: information of each block is sent only by one point.
5- Property of the data: each block is responsible for its own data, in other words updating these data is carried out only by this block.
6- Data Standardization: the results produced by each block are bound with recognized norms and standards.
7- Crossing point: the blocks do not communicate directly; they communicate by an entity which is responsible of data flow management.

In the following we will explore will analyze its different approaches in order to establish the lack and then to propose a set of rules and methods allowing the alignment of information and production system urbanisation initiative.

4.3.3 Urbanization Approaches

4.3.3.1 Approaches Based on Enterprise's Functional Architecture

Several works have been carried out in order to solve the different problems related to this topic. According to [Le Roux 04] the objectives of urbanisation are:

1- To Allow establishing the link between: mission of the organization, exchanges, internal and external actors, business event, business process, information, applications and infrastructure.
2- To provide a complete description of IS on functional, organisational and technical levels.
3- Being a decision-making support tool to understand change impacts during the evolutions (strategic, business, functional, applicative, and technical).

This approach supposes that an organization is structured around an objective (sale, production…) to which the totality of organisation's resources are dedicated. He defines thus an internal (the organization agents) and an external (partners, customers). The information is

treated to satisfy this objective by implementing various supports. In this approach the author proposes to see the enterprise according to four view points:

- External view: it is the Partners and customers view, it highlights and formalizes the organization's missions. It establishes the link between an external and an organization ensuring a mission by delivering (selling) services. In this view a business event is an element which triggers a process of an organization (in this view we must have an exhaustive knowledge about the business events that an organization must treat; it represents "Why" and a part of "How" (the external aspects of "how": partners, suppliers).
- Internal view (organisational, functional): It describes the organization agents who collaborate to achieve the objectives, this view highlights the way in which the organization deals with its missions. An organization is structured of sub-structures defined on the basis of competence and which groups actors having the capacities to carry out acts of management with certain nature and with certain range. In this view we define the functional units and their competences, the actors of the organization and their scope of responsibility, decision-making power and information access rights. To satisfy a customer the enterprise produces an added value "objective" starting from elements obtained from suppliers and thanks to the assistance of partners. Therefore it carries out successive co-ordinates and imbricate activities, which transform an input "product" into an output product starting from an external request "a business event". This view is based on business process analyses which leads to a description of the organization missions "who does what, when and how?"
- Informational view: this view capture the information treated or manipulated by the organization to satisfy its objectives. This view represents the "what": which information to be treated and how to treat them. Generally, the information treated in the context of an information system can be considered according to the level of formalization:
    - Structured Information: it includes the business objects stored in the data bases, structured information models or business objects model allowing creating views corresponding to the different types of exchange.
    - Unstructured Information (the whole of documentary reference frame). It represents a part of the enterprise knowledge and "know how" which cannot be structured in a data base.

- Applicative view: it describes the data-processing tools available to users, and needed to support the business operational execution. It covers two visions:
    - Vision of applications and inter-applicative flows.
    - Technical architecture: technical choices and physical locations.

These four points of view are introduced in this work by a model of four facades linked to each other allowing to build a complete view of IS, elements of each facade are modelled using UML:

1. Event-driven and partnership model: this facade describes why the organization exists by describing the external view of the enterprise, which is a translation of its missions, it includes:
    - The external actors (partners, customers).
    - The business events (which trigger the internal processes).
    - Mission.
    - Exchange Protocols.
2. Business process model it is the functional model of the organization. This model must take into account:
    - The external business events and actors to establish the link with the event-driven and partnership model.
    - The functional units and of the internal actors to present a dynamic and functional view of the organization.
    - The sequence and the management actions specifications in order to present a view of the functional model.
    - It must materialize the information exchange between 'process, activity, operation,' to present an organizational flow model.
    - This model's elements are: process, procedure, activity and operation, internal actor, information and message.
3. Business objects and exchange formats model: It includes two parts:
    - A model of the business objects (central reference frame of an organization).
    - A model of exchange formats (partial view of business objects).
4. Applications and flows cartography, this model offers a vision of the data processing tools which serve the business. The elements of this model are: the ground occupation plan, application, flow, data reference frame:
    1- The applications described in a detailed and homogeneous way.

2- Reference frames of data in which they could access.
3- Inter-applicative Flows, nature, amount, frequency, criticality.

In a complementary way, [Longépé 04] presents an information system urbanization methodology where setting out cartographies is the main step to follow to get the information system urbanism scheme. These maps are used to describe existing and targeted information systems. We distinguish four main types of cartography: business, functional, applicative and technical. The methodology is based on a reference framework integrating four views from the information system:

1- Business view: it describes all the business activities of the enterprise that the information system must support.
2- Functional view: it describes the information system functions which support the business process.
3- Applicative view: it describes all the applicative elements of a data processing system.
4- Technical view: it describes all the hardware, software and technologies used to support the functionality of the information system.

The methodology starts by placing the IS urbanisation scheme within the general frame of enterprise's strategy to allow defining a target information system which is aligned with business process, itself aligned with the enterprise strategy.

The strategy is modelled by a hierarchy of goals represented by ISHIKAWA diagrams, breaking up them into sub-objectives until reaching the lower level in which all the objectives are achieved by at least one process. This diagram is constrained by the following rules (urbanisation rules for the diagram of objectives):

- The same objective appears once and only once in the diagram.
- If an objective is divided in sub-objectives the list of those must be exhaustive.
- An objective of the lower level must be associated to one or more of performance indicators.

In this phase it is necessary to carry out the enterprise diagram in order to isolate all the functional entities (sale and commercial, administration and finance, production and logistics…) to detail their activities and the information flow exchanged between them, and to identify the customer, suppliers, partners… etc.

It is also necessary in this phase to establish the IS chart according to the following principles:

1- Autonomy of business: supposes that the IS serving each business area is independent of the others (as much as possible).

2- Group consistency.
3- Standardisation (to a reduced the amount of data (business objects)).
4- IS consolidation, i.e. IS which are essential to and imposed on all the others ISs.
5- Synergy: the possibility of defining a common IS between two or more business areas.

A second stage consists in identifying the business processes. This phase is not a tool to determine the vision or the mission nor the enterprise strategy: these elements are supposed to be already defined and are used to deduce the processes to be analyzed. This stage starts setting out processes cartography which makes it possible to give a global vision about all the processes, to categorize (to classify) them and to explore their interrelations.

A process is defined as a network of activities which treat a triggering event. It must be defined independently of any organization, and of all existing systems in the enterprise with (in the opposite of the procedure which is a process augmented by an organizational dimension, i.e. a business process described in a given organization structure.

After identifying the processes, we set up a matrix of which the columns are the strategic objectives (the diagram of the objectives) and the lines are the processes identified in the cartography of process. By crisscross between the lines and the columns we indicate the contributions of the processes to achieve the goal concerned.

This stage is constrained by the following rules (process model urbanisation rules):

1- An activity of a process belongs to one and only one IS, i.e. it cannot call more than one IS.
2- Any transformation of the properties of an object results from an activity, and an activity concerns only one object which could be composed of many objects.
3- An elementary activity cannot be stopped after triggering it.
4- All the activities can have normal end and could be subject to temporal events.

The third stage is charting the functional architecture (representing the cartography) which answers the question “what” without taking into account the actors of the organization. It is the base on which the SI will be constructed. Functional Architecture consists of: Functional zones, Districts and Small blocks.

The urbanisation rules for the functional architecture are the following:

1- Membership: in the hierarchical decomposition (zone, district, block) a block belongs only to one district which belongs only and only to one zone.

2- Asynchronism: a block can treat an event and send the result without waiting the answer of the recipient.
3- Information of each block could be sent by one point.
4- Data ownership: each block is responsible for its own data, i.e. updating these data is allowed only by this block.
5- Passage point: the blocks do not communicate directly; they communicate through an entity which is responsible of the Data flow Management.

The fourth and the last stage starts by charting the applicative cartography in order to show the different components of the current IS, and to show the flows inside these components applicative. It is then necessary to constitute the target applicative architecture which based on the target functional architecture by introducing new concepts related to applicative architecture:

1- Flow manager: allows applications to communicate without any knowledge about location, hardware or protocols used.
2- The message: represent flow travel inside the enterprise or exchanged with the environment.

   3- Front office: the whole of customer oriented services, which could be activated by the customer.
3- Back office: all of product oriented services.
4- Middle office: services which makes the correspondence between customer and product (between front and back offices) and could not be activated directly by the customer.

4.3.3.2 Process-Oriented Urbanization Methodology

In this context we can mention [Chapron 04] whose aim is at defining an information system "evolution management methodology" taking into account the current system state, the technological level and the organisational environment. This work aims to provide formal methodology and tools to plan and control the evolution of the information system. It is also a pragmatic step which could be implemented in the enterprise to manage the organisational changes.

The author introduces an evolution management methodology based on a global vision of the enterprise, based on its business processes. It consists in a conceptual framework allowing managing the evolutions of the information system (called in this work "information system organisational urbanism").

This methodology structures the change management process according to three stages. It also includes a dedicated representation of the information system, based on a specific meta-model adapted to the evolution management problem. This methodology, based on a business process oriented approach (including the formalization of the organisational business processes and their contexts) proposes a formal model allowing to represent and to monitor the change processes in the information system. Therefore, the author introduces a meta-model to set clearly interfaces between the processes and their environment. This modelling approach puts the stress on organisational flexibility, introducing the decoupling and decomposition concept of flexibility in sub-elements. In this order the author has introduced two conceptual decoupling of the meta- model:

1- Between the processes and the organisational environment. This is based on different generic views:
    a. The operational view includes the following concepts:
        i. The role concept: it is the interface between the process activities and the actors.
        2- The functionality concept: it is the service required to achieve a business activity.
        ii. The capacity concept: it is an interface between the applications and the hardware (the necessary hardware resources needed to support the right applications' functionality).
        iii. The characteristic concept: it is the interface between the physical process and the physical resources.
    b. The "Organisational Environment" view gathers all the objects involved in the process environment, i.e. the objects which can impact the process or be impacted by its behaviour. In this view we find several elements:
        I. The organisational entities: include the organisational units and the actors of the enterprise who could be involved in the process.
        II. The architecture entities: contain the data bases and the applications of the enterprise.
        III. The infrastructure entities: contain the whole components on which the applications of the enterprise are based (servers…).
2- Between businesses oriented modelling and IT oriented modelling: It is based on the definition of two views:

a. Business process view: describes all the processes of the enterprise according to different granularity levels. In this view, processes are described thanks to models associated to three different granularity levels:
    I. Frame process: this model describes the process used in given field of the enterprise and its interrelationship.
    II. Generic processes: this model describes the generic steps necessary to carry out each process.
    III. Instantiated Processes: this model describes the different processes implementation in the organisation.
b. IT view: this view includes various types of process: physical, informational, data processing.

In this approach, the business object concept is defined as a set of information handled by informational processes. The meanings associated to these different business objects are shared by all the actors whatever the process.

In a second part of this paper, the author proposes an approach to obtain a refined management of different processes in the enterprise in order to understand the relationships which link them. To do that one must identify the zones in which processes are strongly dependent (interdependent) in order to make them evolve jointly and to facilitate the change management process.

This approach relies on business process grouping (clustering) methods to delaminate the boundaries of the coherent zones (in which the processes are strongly dependent) with a strong decoupling between the zones identified. The author claims that the organizations are made up of multiple interdependent elements, and thus, any change on an element has an impact on the other elements which are linked (be dependent on) to the first one. Thus to obtain a good business processes management, it is necessary to identify the processes crossing the different organisational sectors, to rationalize flows and to manage their interdependencies because the dependence represents an element of stability of the enterprise translated by the capacity to maintain the coherence between enterprise processes after the changes or evolutions. The interdependence thus represents a constraint for evolution management. Thus it is necessary to control the interdependences with respect to the evolution management. The second step thus consists in identifying the objects used in a process and analyzing the correspondences in other process to quantify the dependences between these two processes.

To reach this goal the author chooses to identify processes dependencies thanks to 3 main components: actors, applications and information.

1- Actor based dependences can be expressed according to 3 types:
    - Actor sharing: two processes share exactly the same human resource.
    - Structural dependence: two processes require two different but related actors.
    - Role based dependence: human resources are identical or not but they have to play a same role.
2- Resources based dependences are split into 2 sub-types:
    - Application sharing: two processes use the same applications.
    - Dependence related to the data flow: processes use two different applications who exchange a high data flow level.
3- Information based dependences:
    - Business object sharing: two processes handle one or more identical business objects.
    - Event-driven dependence: a business object plays the role of a trigger event between two processes.

The author proposes to measure all these dependences types and to incorporate them into an elementary dependences matrix. These matrixes can then be processed to produce a dependence graph (interrelationships). Once this graph is defined, one can apply a hierarchical classification method in order to define groups of (strongly dependent) processes to provide a process clusters map. These maps can then be used by managers as diagnosis tools for the information system. Therefore the author introduced four diagnoses related to the various goals:

1. The first diagnosis deals both with the quality of the cartographies and with structural information that they could provides include about the evolution of a system in order to clarify the evolution of system specifications (characteristics) at the time of a transition from one state to another.
2. The second diagnosis focuses on the calculation of the required effort to move the system from one state to another. It also concerns the alignment between information processing system and the business needs by introducing a synthetic vision of actual existing constraints in the organization, constraints which must be taken into account in the applications.

3. The third diagnosis includes the comparison of various factors (actors, resources, information) to identify variations while setting clusters to potentially highlight dysfunction or improvements source cases.
4. The fourth diagnosis concerns the clusters' internal analysis in order to help in the development of the target system as well as an evolution scenario i.e. the constitution of consistent management units with respect to the evolution while taking in consideration two indicators: the processes maturity and the processes contribution to the enterprise strategy.

The last stage of the methodology is dedicated to the real application of the previously defined steps in a real enterprise to meet the needs concerning the problems of evolution management; this stage includes three steps:

- Modelling the enterprise's information system using the meta-model.
- Implementing the process clustering methodology, to provide information system cartography.
- Carrying out the information system diagnosis according to the various indicators.

4.3.4 Discussion

In both [Longépé 04] and [Le Roux 04], the IS urbanism strategy is based on the vertical enterprise's functional architecture without taking into account the production process logic which involves a transversal or horizontal organisation. This leads to the use of specialised business support software but also increases rigidity at the operational level and aggravates setting collaborative strategies as collaboration processes are dynamic by their nature and fully impact the enterprise. Additionally, although [Le Roux 04] presents a clear and simple approach to model and describe the enterprise from different view points, for the urbanization problematic, a method to exploit such a model has yet to be clarified to achieve a progressive transformation towards the target system. While [Longépé 04] presents an integrated description of the different views and how to deduce them from each other and manage the transformation between the current and target system, the external view which could impact the system design is discarded.

Despite the fact that the approach highlighted in [Chapron 04] uses a process-oriented methodology to avoid many limitations seen in traditional approaches, the context of collaborative production is not taken into consideration. This lack results in some shortcomings:

- The approach does not take the external view into consideration despite the fact that it could impact the dependency between processes.
- The approach considers that resource sharing always creates dependency between processes which is not the case, especially for shareable resources.
- The approach is based on four elements to analyze dependence, namely: applications, actors, events and information. It has not taken into consideration other important elements in the context of production such as storage points, raw materials and material resources including tools and machines.
- This approach treats the coordination between processes as the only dependence criteria between processes, while in the context of collaboration, coordination represents in of itself only one factor of dependency to which, for example, can be added communication and cooperation.

4.3.5 Conclusion

In brief, urbanization consists of studying existing systems then specifying the future systems in order to get an adaptable information system and facilitate reusing or extending existing functionality to meet the future business needs, as we have seen via the previous review, urbanizing a corporate information system enhances the agility at the information system level but it increases rigidity at the operational level because it does not take into account the transversal nature of production process, leading to a fragmented information system, which farther hinder the collaboration.

What we need is a new urbanization paradigm which takes into consideration the particularity of the collaborative production context; the urbanization strategy must be redefined to fit collaborative production needs. This leads us to re-think the way we identify and analysis the production and the information systems in order to “expose” business objects integrating industrial management criteria in their definition. As a result, enterprises can offer their resources or results of their business processes via the description of these resources, while specifying constraints, conditions and means necessary to manage the access of these resources. In this context an important issue should be addressed while developing an information system supporting collaborative production; the choice of common application platform which binds the manufacturers with their networks of partners. Such solutions should ensure flexibility, reactivity and interoperability at both production and information levels i.e. the ability to compound dynamically business functions issued from diverse and heterogeneous systems.

Our aim is to define a framework to collaborative production information system, especially for dynamic collaboration scenarios, in such scenarios many functional modules (developed without any central control), are composed to meet the collaboration objective; starting small and building towards business goals. In such a context discovering available functional modules is one of the more challenging parts of designing the information and production systems, we need an approach where everyone would look in a “directory” to find the functionality they need and adopt the best quality or least expensive. This argument goes, for adopting a bottom up organization. Using this approach we analyse all available functions that can be exposed to the environment and for each we determine whether or not the exposed functionality is independent of other functionality. Any dependency means that there's additional functionality needed to be able to freely expose this function, this will leads to construct incrementally the production and information system. All these requirements are considered as natural features of Service Oriented Architecture SOA, in fact, SOA represents an approach to distributed computing that provides a powerful business integration solution within the enterprise as well as beyond the enterprise; between a company and its partners and customers, by providing an abstraction layer that exposes application functionality as business-oriented services that are both location independent and discoverable on the network. Consequently, integrating this service oriented approach while organising business process can improve business flexibility and reactivity; as the core process can be changed without affecting the way the service is included in more complex organisation, interoperability; as standard interfaces are published, accesses to parts of the corporate information system are simplified.

In the following section, we will present the main SOA standards, and namely those of Web service technology paying a particular attention to the security issue as an important concern in the collaboration scenarios.

4.4 Service Oriented Architecture

Service-Oriented Architecture (SOA) is an approach used to design and implement information systems with the help of basic components, called services, implementing specific functions.

Services are encapsulated applications presented as independent components offered by various companies and exposed by means of a service broker. These services could be composed to create new services.

The service is organised into two parts: first, the published interface description defines exactly what services are able to do, the information they exchange and the way they can be called. Second, the “internal part” includes the exact process that will be used to achieve the supported function. Services can be used either to implement “standard” access to the corporate information system or support particular functions used in business processes.

Such service-oriented architecture increases reusing abilities as basic services can be included in different “service chains” to complete various tasks and achieve flexibility as only the published interface is fixed, the internal part can be changed easily to fit organisational or technological changes. Moreover these “services” which can be distributed on different sites and companies can be used to reconfigure a new collaborative business process “on demand” with help from the selection and composition process.

To ensure « technological » interoperability, a service-oriented architecture is based on the following principles [Nickulland 05]:

- Interface support based on market standards (such as SOAP, WSDL and UDDI among others) and Discovery, a dynamic producer for the consumer via an intermediate party.
- In regards to separation between the interface and its implementation, the interface should contain everything that must be known about the service, its mission, interface inputs and outputs and the protocols used to access it.
- Finally, the element of technological neutrality decouples the interface’s functional input and output description and those of technical access protocols.

In such architecture, the integration of services creates a set of challenges on both conceptual and technical levels (Figure 6). On one hand, service granularity in the information system is technical and thus too weak to describe a business function. Inversely, the business process has a very high description ability, which must be broken down into segments [Brende 05]. In other words, the organization of arrangements, call sequences and access rights must be ensured for these applicative modules in order to compose functional entities capable of exposing meaningful business functions and finding the qualified entity which can ensure the dynamic composition, execution and coordination of services from multiple and heterogeneous sources.

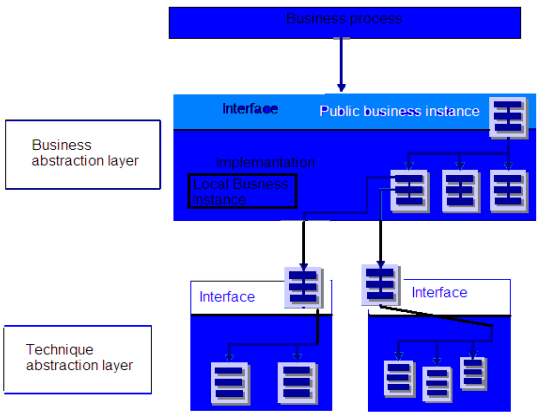

Figure 6 The Dual Layers of Abstraction [Brende 05]

Building a collaborative process via SOA paradigm requires overcoming several obstacles:

- While individual services are rather simple, in the collaboration context, services are usually results of combinations of services stemming from several systems; there is a need for modelling the interactions and dependencies between services stemming from multiple sources.
- The implementation and coordination of complex processes require streaming real-time information and events from the process currently in execution towards multiple destinations. We must therefore ensure effective communication in asynchronous and synchronous scenarios, secure routing, security and management, message sequence transmission.
- For security reasons, business process should be represented in an abstracted manner in such a way as to allow the representation of the business logic and flows, while limiting the visibility of services execution details form the consumer. This requires providing services interfaces with a sufficient level of abstraction to allow exposing visible "business" Component as defined by the contract.

The solutions to these problems is based on the definition of third party (i.e. orchestrator) who has the role of achieving the dynamic composition of functional processes as well as managing business components coordination and execution (in response to business events or a pre-programmed way) taking into account inter-dependencies between process components. “Services Orchestrator” (Figure 7) guides the construction and the execution and the coordination of the complex collaborative services and manages the state of composing services as well as their intermediate results. It organizes and sequences the calls of services for technical or business proposes, in response to events. In other words it has the role of integrating a set of services distributed at several physical points and accessible separately to accomplish a task or Implement a business process. "The role of SOA orchestrator will be to

manage these chains of inter-connected services in the best way. At various levels, the process can be modelled and implemented directly from the orchestrator, which will have the role of invoking the relevant Applications " [Brende 05].

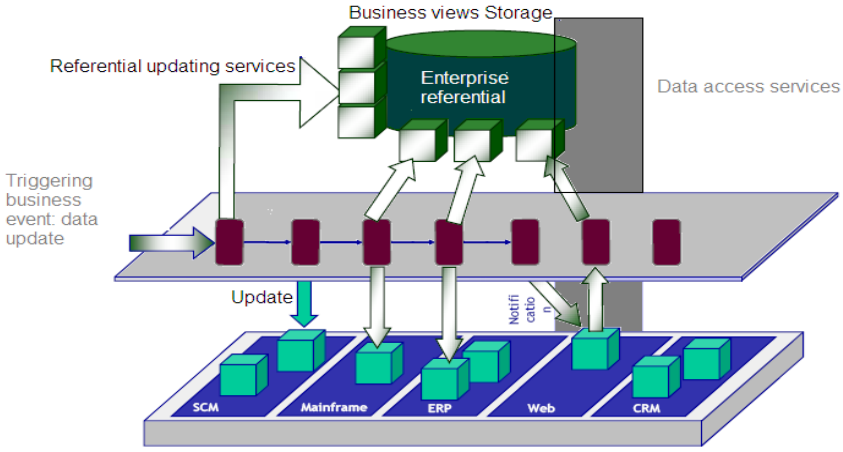

Figure 7 SOA Orchestrator [Brende 05]

At the business layer, organising inter-enterprise collaborative service chains involves adapting the composition process to support dynamic collaborative process enactment according to changing context and needs, as well as respecting and consequently integrating the policies imposed by the different "service offering" enterprises.

Thus, a significant attention should be paid to extract the business rules defined by the different organizations in order to define the sequence in which tasks are done in conjunction with the responsibilities, conditions, and any other aspect in the process steps, as well as describing how to orchestrate the business flow in order to enable automation of tasks, as well as identifying monitoring checkpoints and match them to the performance indicators to in order to improve the process execution while being aware of the context. This is usually done transforming business logic into well-defined services invocations using modelling techniques to match business behaviour and functionality into software services within the information systems without involving huge programming efforts, this is achieved by Business Process Management Engine, This usually includes [L'encyclopédie e-Business]:

1. A modelling tool used to formalize the description of the functions performed in the in the enterprise process, in computer applications.
2. A set of development tools to formalize the logic that governs the enterprise process into a set of operation rules.

3. an execution engine that supervises processes workflow in conjunction with a rules engine that assess the state of all the objects involved in the enactment of the process and validate conditions which allow to begin, continue or stop the execution.
4. A repository which keeps the definitions of the processes, the rules that should trigger their execution, integrity and security constraints, and the measures relating to the enterprise business domain.
5. Administration tools that will adjust the settings of the entire system and obtain performance indicators and statistics based on data collected during the execution process.

This Business Process Management can be seen, using SOA terms, as the "Services Orchestrator".

In order to implement an SOA, both applications and infrastructure must support SOA principles. Enabling an application for SOA involves the creation of service interfaces to existing or new functions. Enabling the infrastructure involves providing the capabilities allowing routing and delivering service requests to the correct service provider and supporting the replacement of service implementation by another without affecting the clients of that service. This requires that the infrastructure allows client code to invoke services independently of the service location and the communication protocol. Such service routing and substitution are amongst the many capabilities of the ESB (Figure 8). ESB as a set of infrastructure capabilities implemented by middleware technology usually based on recognized standards which provide foundational services for more complex architectures via an event-driven and standards-based messaging engine (the bus). The ESB supports service, message, and event-based interactions in a heterogeneous environment, with appropriate service levels and manageability.

The most important role of ESB is the mediation; mediation includes tasks such as transformation and synthesis of data that can go from basic format translation in order to match a particular standard, to more sophisticated analysis using ontology or expert knowledge, adding new values to the data. Synthesis may be done over multiple data sources, having heterogeneous data type. For example, one may have to apply a currency conversion between services from different countries or may want to extract the global evolution of the activity of a plan over a week period from a database containing daily information. The aim of mediation is thus to abstract data and extract the domain knowledge in order to ease the decision making [Herault 05].

It also provides response to many functional interoperability requirements like an asynchronous communication service with dynamic routing and dispatch of requests potentially to multiple receivers in order to perform load balancing or to respond to failure of a data source for instance], as well as publish-and-subscribe semantics that support an event-driven model in which any number of service consumers can respond to a business event. These rich messaging semantics enhance service re-use by make it easy to dynamically add new service producers and consumers to the scenario at runtime, without requiring a recoding of a process or service [Chappell 04].

In addition it can provide other non-functional activities related to quality of services management such as incomplete data management, quality measurement, tracing, security, caching, or failure detection and recovery, reliable delivery, transaction management etc [Herault 05].

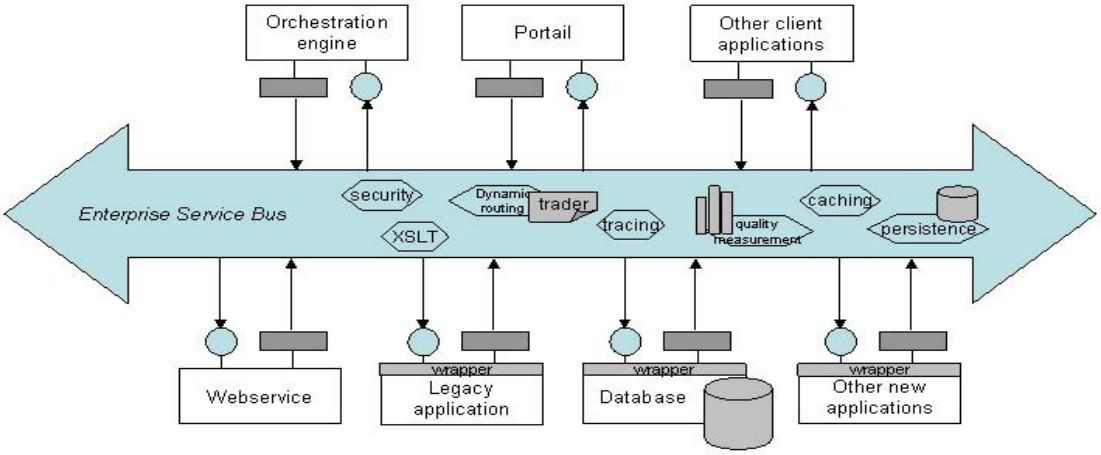

Figure 8 Enterprise Services Bus [Herault 05]

In the following we will explore the main features and technologies thought to support Service Oriented. We will start by a introducing the basic « typical » Service standards and then we will be more interested in the security issues as they represent an important obstacle which hinders collaboration.

A key element for constructing the enterprise information system in service oriented manner is to understand the complexity the enterprise itself. However, because of the multifaceted nature of an enterprise, the modelling needs to take range of different perspectives at different levels of abstraction to be able to properly understand and describe the enterprise in terms of its architecture; this leads us to notion of Enterprise Architecture.

4.5 Enterprise Architecture Reference Frameworks

Enterprise Architecture is a representation of any collection of collaborative organisations and sub-organisations, that need to interact at all levels to achieve common set of goals [Forder 07].

An Enterprise Architecture can present a clear vision of an enterprise in all of its dimensions and complexity both in terms of its existing state and its desired or future state in order to support all aspects of the enterprise including strategic, organisational structures, business processes and information needs [Forder 07].The guidelines and rules to define describe and develop an architecture, which can be the current or future architecture, are defined by an Architecture Framework. The framework provides a set of standard building blocks that help ensure consistency both within a single architecture and across several architectures. Typically, an Architecture Framework is thought for describing an Enterprise Architecture in terms of a number of viewpoints, usually: Strategic, Business processes, information needs and constraints, and the underlying technology and its constraints to support the business needs. Each of the viewpoints also need to be described in terms of views that show behaviour and structure at varying levels of abstraction. The complexity of the enterprise is captured by capturing the entities and concepts within the viewpoints and underlying views ass well as by describing the complex dependencies that exist between them.

There are several architecture frameworks, they are not necessarily in competition with each other; in fact they are often complementary. In this context we can mention The Open Group Architecture Framework (TOGAF) is a framework for Enterprise Architecture which provides an approach to the design, planning, implementation, and governance of enterprise information architecture. As indicated in (Figure 9) the architecture is typically focused at four levels or domains: Business; Application; Data; Technology.

The 4 domains are modelled by following a methodology comprising 9 discrete but iterative phases.

DoDAF is yet another framework, it is especially suited to large systems with complex integration and interoperability challenges, and is apparently unique in its use of "operational views" detailing the external customer's operating domain in which the developing system will operate. These products are organized under four views: All View, Operational View, Systems View, and the Technical Standards View; OV, SV, TV, AV respectively (Figure 10).

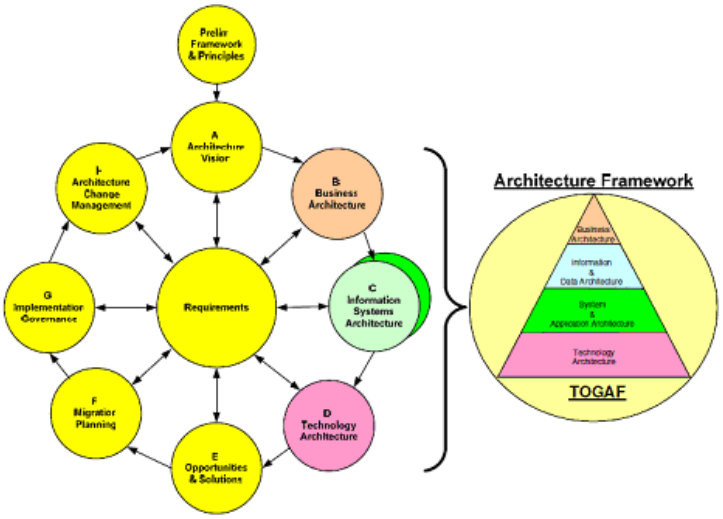

Figure 9 The Open Group Architecture Framework (TOGAF) [Forder 07]

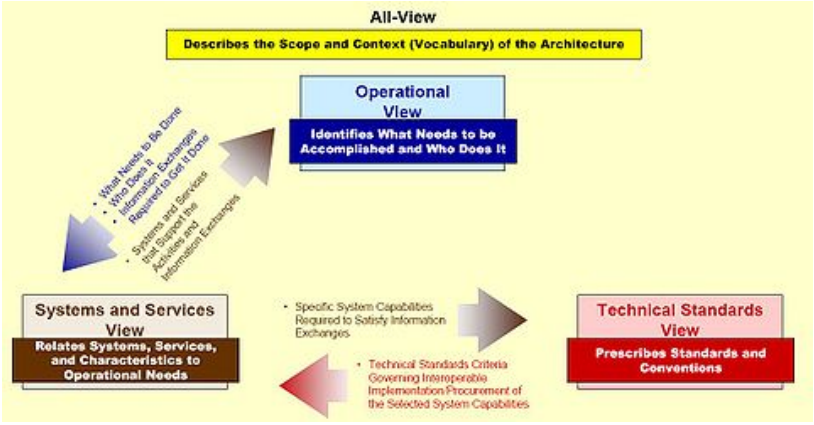

Figure 10 DoDAF Views and Their Relations [USDoD 07]

Many frameworks are based on DoDAF among them we can mention MODAF. MODAF adds a number of new views needed to support processes and structures and seeks to better integrate the concepts from the Strategic and Acquisition views with the core views inherited from DoDAF (OV, SV, TV, AV) [Forder 07].

The NATO Architecture Framework NAF is also based upon DoDAF and informed by the additional MODAF views [NATO 07]. NAF also introduces a number of new "Service Views" to support Service Orientated Architecture.

Another interesting framework is Federal Enterprise Architecture (FEA) which is the Enterprise Architecture of a Federal government. The FEA is built using a collection of reference models that develop a common taxonomy and ontology for describing IT resources, these references models are: Performance Reference Model (PRM), Business Reference Model (BRM), Service Component Reference Model (SRM), Technical Reference Model, and the Data Reference Model (DRM).

The PRM is a standardized framework to measure the performance of major IT investments and their contribution to program performance.

BRM is a function-driven framework for describing the business operations of the Federal Government independent of the agencies that perform them, providing an organized, hierarchical construct for describing business operations of the Federal government using a functionally driven approach.

SRM is a business and performance-driven, functional framework that classifies Service Components with respect to how they support business and/or performance objectives.

DRM describes the data and information that support government program and business line operations. This model enables agencies to describe the types of interaction and exchanges that occur between the Federal Government and citizens.

Finally the TRM is a component-driven, technical framework categorizing the standards and technologies to support and enable the delivery of Service Components and capabilities [OMB 07].

As the most part of complex system analyses approaches, Enterprise Architecture Frameworks use Pattern Oriented Modelling.

4.6 Pattern Oriented Modelling

The Pattern Modelling is based upon the idea that system development process is a sequence of analysis observations and design decisions that identify problems that must be solved, and then apply solutions that addresses each of these problems by addressing subordinate problems that that particular problem may have dependencies upon in a bottom-up way.

Pattern is a structured description of proven solution (activities) to achieve some goals or to resolve a frequently appearing problem. A pattern is simply a well-defined, proven technique or approach that is applied to solve a problem that occurs in a specific context. It is a description or template for how to solve a problem that can be used in many different situations. The idea is to write down best practices and lessons learned from a given problem domain. Pattern includes a description of the goal to be achieved (problem to be resolved) and the description of the core of the solution or the activities to be done in order to achieve the desired goal.

Pattern Model can be determined via a Problem Model; a Problem Model defines the set of problems that the system must address. The Problem Model defines the hierarchical relationships that exist between these problems. Each problem exists at a given level of abstraction within the problem space. Problems may have dependencies on other problems.

The solution for some problems must be defined before the solution for other, dependent problems can determined to determine problems having dependencies on other problems. A problem can be represent by a problem frame [Jackson 00], Problem Frame is a frame or schema defining in an intuitive manner a class of problems identified by its context, needs and specifications [Choppy 04]. Usually a pattern consists of at least four parties:

- Initial environment or rules: describes the preconditions under which the pattern should be invoked, it represent the “when” and “where” and sometimes the “on what”.
- Goal or Problem: describe the reached objective by the pattern application “the why”.
- Operation or solution: the abstraction which outline the proven solution i.e. operations to be followed in order to reach the thought goal it is the “how” or code (it could be a generative model).
- Consequences: A description of the results, side effects, and trade offs caused by using the pattern.

More advanced pattern description can be found in the literatures depending on its application domain, for instance a commonly used documentation format is the one used by Design Patterns. A design pattern is a general reusable solution to a commonly occurring problem in software design. A design pattern is not a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. It contains the following sections [Gamma 95]:

- Pattern Name and Classification: A descriptive and unique name that helps in identifying and referring to the pattern.
- Intent: A description of the goal behind the pattern and the reason for using it.
- Also Known As: Other names for the pattern.
- Motivation (Forces): A scenario consisting of a problem and a context in which this pattern can be used.
- Applicability: Situations in which this pattern is usable; the context for the pattern.
- Structure: A graphical representation of the pattern.
- Participants: A listing of the classes and objects used in the pattern and their roles in the design.
- Collaboration: A description of how classes and objects used in the pattern interact with each other.
- Consequences: A description of the results, side effects, and trade offs caused by using the pattern.

- Implementation: A description of an implementation of the pattern; the solution part of the pattern.
- Sample Code: An illustration of how the pattern can be used in a programming language
- Known Uses: Examples of real usages of the pattern.
- Related Patterns: Other patterns that have some relationship with the pattern; discussion of the differences between the pattern and similar patterns.

Recently a new software and information system engineering domain basing in « pattern » idea has appeared, it is the Component Oriented Architecture [Szyperski 97]. The basic idea behind this approach is to construct the system (or a software solution) using a pre-fabricated, encapsulated and rather independent constructs called components. This components must satisfy the following conditions [Choppy 04]:

- All services obtained from a component via well defined interfaces.
- The specifications of a component must contain all required information for its use.
- A component adheres a “component model” which is a set of syntactical and communication standards and conventions. The “component model” insures the interoperability between all the components adhered to it.

After this short theoretical background, in the following we will explore the main features and technologies thought to support Service Orientation. We will start by a introducing the basic « typical » Service standards and then we will be more interested in the security issues as they represent an important obstacle which hinders collaboration.

4.7 Basic Service Standards

At the implementation level, service oriented architecture appears as a convenient way to support “technological interoperability” since it has been developed basing on “de-facto” standards allowing an open and distributed implementation. Currently, the most popular implementation of this idea is the web services (Figure 11).

A web service can be defined as any software that can be described using the Web Services Description Language (WSDL). WSDL specifies these details about a service:

- What operations the service provides.
- How to invoke those operations, specifically: the format and protocols we can use to send the data for those operations.
- The location (address) of the service.

There is a companion standard called UDDI (Universal Description, Discovery, and Integration) that provides for discovery of services and retrieval of their WSDL descriptions. Together, these standards let applications discover and use web services. Several frameworks use the details from WSDL at runtime, providing loosely coupled integration between service consumers and service providers.

The most popular data format and protocol in use for web services is SOAP.

Here, we are going to focus on SOAP-based web services exclusively, as they are the most popular ones. We will start by introduce the typical standards used in the implementation of the service, and then we will be particularly interested in discussing the security tends in the context of SOA.

4.7.1 UDDI Universal Description, Discovery and Integration

UDDI [Clement 04] stands for Universal Description, Discovery and Integration, UDDI is a directory for storing information about web services; a directory of web service interfaces described by WSDL, it facilitates the discovering of available services for the clients by exposing their interfaces and describing the functionality they can provide, the principal protocol used to communicate with UDDI is SOAP. The process starts by sending a client's request using one of the protocols supported by the client's application then the UDDI answers by sending the WSDL description of the suitable service encapsulated in a SOAP message. For instance, if flight reservation industry published an UDDI standard for, airlines could register their services into an UDDI directory. Travel agencies could then search the UDDI directory to find the airline's reservation interface. When the interface is found, the travel agency can communicate with the service immediately because it uses a well-defined reservation interface.

4.7.2 WSDL for Web Services Description Language

WSDL [Christensen 01] stands for Web Services Description Language. It is used to describe and to locate Web services. It is written in XML. WSDL contains set of definitions to describe a service. A WSDL document describes a web service using these major elements:

<portType> The operations performed by the service. It is the most important WSDL element. It describes a service, the operations that can be performed, and what messages are involved. This element can be compared to a function library (or a module, or a class) in a traditional programming language.

<message> The messages used by the service. It defines the data elements of an operation. Each message can consist of one or more <part>. The parts can be compared to the parameters of a function call in a traditional programming language.

<types> The data types used by the service. It defines the data types that are used by the web service. For maximum platform neutrality, WSDL uses XML Schema syntax to define data types.

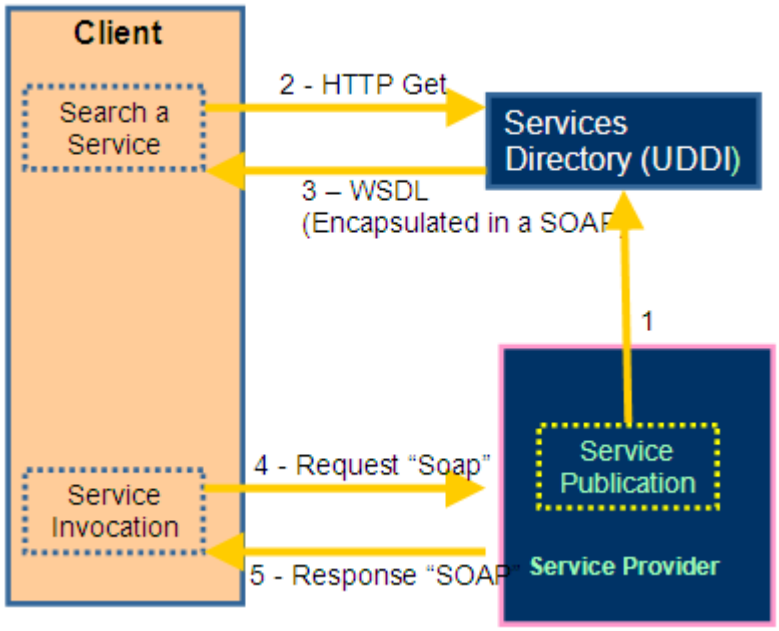

Figure 11 Standards Used in the Implementation of Services

```
<binding> The communication protocols used by the service.
<message name="getVoyageRequest">
<part name="Destination" type="xs:string"/>
<part name="Date" type="xs:string"/>
</message>
----
<message name="getVoyageResponse">
<part name=" Price" type="xs:string"/>
</message>
------
<portType name="VoyageDirectory">
<operation name="getVoyage">
<input message="getVoyageRequest"/>
<output message="getVoyageResponse"/>
</operation>
</portType>
```

Figure 12 Example of WSDL

(Figure 12) gives an example, in this example <portType> element defines " VoyageDirectory " as the name of a port, and " getVoyage " as the name of an operation. The "getVoyage" operation has an input message called "getVoyageRequest " and an output message called " getVoyageResponse ". Compared to traditional programming, "VoyageDirectory" is a

function library, "getVoyage" is a function with "getVoyageRequest" as the input parameter and "getVoyageReponse" as the return parameter.

4.7.3 SOAP Simple Object Access Protocol

SOAP [Haas 07] stands for Simple Object Access Protocol. SOAP is an XML based protocol which allows accessing an object across the network, this protocol is transport neutral; SOAP can be used across HTTP, FTP, and SMTP …etc. SOAP includes a whole stack of "composable" Web service specifications (WS*): WS-Security for inserting security tokens into SOAP headers, WS-ReliableMessaging, WS-Transactions etc.

SOAP is used for both Messaging and remote procedure call (RPC). It has a simple structure; in the following we expose the main elements of SOAP message:

- **SOAP Header Element:**

  The optional SOAP Header element contains application specific information about the SOAP message (like security). Header element is the first child element of the Envelope element. The SOAP header can contain two important attributes: The "mustUnderstand" and "actor" attributes:

  - It can be used to indicate whether the recipient must or not process a header entry, that is, if mustUnderstand="1" in a child element of the header element, it indicates that the receiver processing the header must recognize the element; otherwise an error message is generated when processing the header.
  - Actor Attribute: usually a SOAP message passes different points along the message path from a sender to a receiver. Not all elements of the SOAP message may be intended for the ultimate endpoint, instead, some elements may be intended for one or more of the intermediate nodes. The SOAP actor attribute is used to address the Header element to a particular node across the message path.

- **SOAP Envelope & Body Elements:**

  The SOAP Envelope declares the XML document as a SOAP message. It is the root element of a SOAP message.

  SOAP Body element is mandatory, since it contains the actual SOAP message intended for the ultimate endpoint. (Figure 13) shows an example of a SOAP message:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-
envelope
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Header>
<user:credentials soap:mustUnderstand="1">
<username>Layth</username>
<password>Okinawa</password>
</user:credentials>
<m:visitor xmlns:m=http://www.u-ryukyus.ac.jp/transaction/
<soap:actor="http://www.u-ryukyus.ac.jp/appml/"> 234 </m:visitor>
</soap:Header>
<soap:Body>
<user:context>
<position bulding="eee2" floor="3"/>
<role>visitorResercher</role>
 <time>10.00</time>
</user:context>
</soap:Body>
</soap:Envelope>
```

Figure 13 Example of a SOAP Message

4.8 Conclusion

Information System agility raises challenging issues in relation to the interoperability, adaptability and openness of the production and business processes. The Service-Oriented Architecture aims to compose and implement information systems based on fine-grained components, called services, to provide specific business functionalities. Taking into account the service-oriented approach in business processes improves flexibility, reactivity and the interoperability of information systems. However, collaborative business requires an enormous trust between participating companies, the choice of technical factors which ensure security specially if we adopt SOA paradigm, should be issued from the business risk analyses of the different involved parties from one side, and should take into consideration the dynamic nature of the collaboration, i.e. "on-the-fly" composition of functionalities issued from heterogeneous systems and possibly heterogeneous or even conflicting security interests and measures.

In the following we will be interested in SOA security issues as they represent an important obstacle which hinders collaboration.

Chapter 5

5 Security Issues

5.1 Introduction

SOA promotes collaborative process via service composition, however, this approach imposes rethinking security. In fact, Traditional information systems are designed to operate in a rather closed environment where the interaction with the environment, and thus security issues, are strictly thought with regard to information exchange. In contrast, in the context of collaboration, the information systems of involved enterprises must "interoperate" in a collaborative process.

In this context, we must note that a service is not an application that moves, but rather a part of a program remotely accessible. In fact, processing still run on a defined endpoint while the service wrapper (interface) facilitates the access to service's functionality. Hence, we are still limited by the existing computing and orchestrating layers which operate at the technical level, the service composition layer acts as a layer of abstraction and hides the details of the underlying technology implementation from the users, each service abstracts the user identity and context from the underlying applications. In the other side, users can access services located on different systems at different times, and the underlying applications no longer have the context they require to authorize specific actions.

Additionally, the identity verification as well as access and permissions control mechanisms and policies might vary among the various back-end systems, consequently users might have different identities, privileges, and policies for each system.

Therefore, service composition process must assumes global mastering of security issues i.e. unified identity management and security policy infrastructure that governs the access to the different interfaces in a composite service in a way that provides the overall security context for the systems, services and applications. This imposes the integration of the security constraints into the service architectures,

To enable the establishment of "on-demand" collaboration via at the business level, enterprises must define and publish what they can offer while specifying constraints, conditions and contexts, under which their services can be invoked, and their processes can be monitored. This can be achieved by analysing the risk related to service deployment in a pervasive environment before defining how security needs could be integrated into the

processes definition and then associated to the services by incorporate them in a global framework. This involves the adaptation of organizational issues (security policies, users' directory, identity management, service agreements ...), as well as their underling technical issues (security protocols, encryption and signing, authorization policies ...). We will hereby emphasize the security aspects (e.g. encryption and signing, authorization ... etc) which is a main concern when different companies, using heterogeneous security systems need to interact [Herault 05].

5.2 Enterprise Security Strategy Definition and Risk Analyses

Security concerns aim to protect various resources of the enterprise from undesired access which directly or indirectly impacts enterprise activities. Integrating security in the main constructs of the enterprise's organization and processes is achieved by determining security requirements governing their behaviour early at the stage of the system design.

Security does not only concern internal actors of the enterprise but also their customers. Customers have preferences and requirements concerning security especially in the e-service environment where services are executed on the provider side. In addition, costumer and enterprise contexts should be considered during service initiation and execution phases. Contextual information impacts applied security policy and measures. As a result, contextual information and customer preferences should be identified and propagated through the service chain.

In the following section we discuss risk analysis and security evaluation methodologies in, and then we will survey current security standards, especially in the context of B2B. We illustrate the integration in the enterprise organisation and process level, the acquisition of customer security preferences and the propagation of security-related context issues in e-services.

5.2.1 Risk Analysis

Improving security in enterprise information system relies on analysing threats, risks and vulnerabilities to specify appropriate countermeasures. The analysis identifies internal and external system stakeholders, their activities and required resources to figure out potential events that could negatively impact information system reliability. The analysis outlines the probability of risks and key indicators that are used to calculate risk occurrence. The analysis

supports the definition of security policies and how system components manage and implement policies, monitoring their execution and evaluating their effectiveness.

Many methodologies of risk assessments are proposed based on static values such as purchase cost or maintenance expense or process-oriented methodologies. For this purpose, the work in [Alberts 01] provides a cross-analysis of the organisational structure and related technical solutions to identify threats, analyse risks and define countermeasures. Done both at technical and organisational levels, this method provides a hierarchical description to guide vulnerability. The method consists in the identification of asset, access mode, actor involved, motivations, effect linked to the nature of action and impact linked to its costs. With the help of this systematic analysis, risks can be identified and then associated to technical actions reducing their consequences. However, this method does not provide a global point of view on the infrastructure. In [Eom 07] authors proposed business process-oriented risk assessment methodology to assess information systems. In this work, information system components are classified with respect to their utilisation context and business contribution. Authors then applied weight to analyze asset according to business process oriented classification factors.

However, their analysing method is limited by the organisation boundary in that they did not take into consideration the external actors and environment related to the collaboration possibility. Additionally, they do not propose a methodology to generate an appropriate security policy from risk assessment. The work in [Kearney 07] considers security as part of risk management by which uncertain events affect the system in terms of net productive values and through system properties such as confidentiality, integrity, availability, privacy and accountability. The work describes three sub-models; the value model, threat model and security model, which are defined in a meta-model based on the agent concept. The value model is built in order to recognise the system and its environment. The security model defines the infrastructure to protect the system whereas the threat model defines threat scenarios with associated risk distributions. These models are considered in a risk analysis phase to verify whether the system needs further tuning.

Roughly speaking, the analysis recognizes events that impact information system functionalities. Still, there is a need for systematic frameworks which provide end-to-end to analyze security requirements and define security strategy.

5.2.2 Security Strategy Identification Methodologies

In the emergence of B2B and C2B Internet-based paradigms, threats related to huge quantity of exchanged information via internet makes enterprises aware of securing infrastructure as well as integrating trust and security requirements in the design of business processes and information systems. However, building a security strategy is often reduced to the implementation of a "secure infrastructure".

Methods and standards have been defined at the international level of certification. For example, DoD Rainbow [USDoD 85] provides a framework to implement security policies specification. The security policy is defined with labelled information parts such as resources along with information access rights and actors' access rights descriptions. Common criteria [CCO 04] consists of both criteria definition and evaluation methods. It defines a security concept model mixing threats and responsibilities to protect system components. These standards provide technical solutions to threats and vulnerabilities. Recently, security requirements have take into account the design of business processes and physical implementations [Biennier 05].

ISO 17799 [ISO 00], for example, proposes security standards with respect to organisational and technical elements. This approach introduces a method to define, implement and manage a consistent security policy which deals with communication and physical security requirements such as building security or access control and provides information confidentiality, availability and integrity. It also proposes a methodology to integrate both organisational and technical point of views in a common security policy. Unfortunately, the ISO 17799 standard does not provide a framework to integrate security requirements in Business Process models and does not define a systematic risk analysis methodology [Biennier 05].

All the aforementioned security standards rely on centralised models, however, B2B and B2C paradigms extensively use the web service technologies in decentralised connective models, Moreover, these methods are designed to define and implement a convenient security strategy for an enterprise, focusing on the information system infrastructure (mostly the network organization, integration of "secured" software components and SSO or RBAC systems). These different elements fit well "static" and intra-enterprise context but they do not fit the inter-enterprise collaborative context as BP are built dynamically depending on the context.

Information security aims to protect firm resources from undesired access by users and applications. Integrating security in the firm structural organisation and processes is achieved by identifying security requirements and controlling access to firm patrimonial assets. Controlling access makes it possible to reduce risks that can dramatically threaten firm resources.

5.3 Integrating Security in the Enterprise Organisation and Processes

The definition of a security strategy leads to deploy security policies to control access based on roles which can be integrated into enterprise workflow specifications and user authentication system [Li 02] Role-Based Access Control (RBAC) [Ferraiolo 95] is a security model which relies on the enterprise organization view to provide authorization to access resources. The RBAC model creates an indirect relationship between rights and actors through the roles played by the actors. Thus, security policies are defined to govern roles, not individual users and facilitate the management of security policies. This model consists of six components: users or agents, roles, sessions, transactions, objects and permissions. To this, the notion of constraints is added on the top of the relations in order to establish a high level of organizational policies. RBAC affects users of the roles they can play, these roles are associated with permissions which provide access to resources [Chen 07]. However, the RBAC model does not take in consideration the notion of using or execution context.

The work in [Weber 05] extends the RBAC for adaptive process management systems. The approach defines two types of access rights: user-dependent access rights which restrict process changes and, process-dependent access rights which allow change commands within a particular context. The DRBAC [Zhang 06] proposes dynamic roles to access resources by defining policy context sets and trigger changes in role state via role state machines. These state machines transform roles according to context changes. States are associated to the role hierarchy and trigger conditions describing the context. DRBAC does not consider the collaborative business context where dynamic ad-hoc groups of users of different policies and contexts conjointly work to carry out a common objective.. The Generalized Role-Based Access Control (GRBAC) model [Covington 01] applies roles to system entities i.e. subject roles, environment roles, and object roles. Contextual information is checked to grant access. Due to the multitude of system entities,, management of roles becomes difficult to maintain. The work in [Ray 06] extends the RBAC model with the physical location of different components to decide whether a role has an access right to resources.

[Yialelis 96] proposes role-based access control security architecture to support distributed and multi-organisational enterprise. In this work the notion of domain is used to specify various security policies according to objects groups. The organisational structure is represented by the Roles and inter-role relationships. This architecture provides ability to refine the abstract organisational policies to enactable rules. Though this work presents well-designed and integrated policy-based security architecture, it suffers from some drawbacks: the notion of context has been neglected which, in this case, impacts the totality of the architecture. Additionally, the work treats the problem from a purely security-oriented viewpoint without taking into account the business dimension such as process or workflow and their related problem of coordination, dynamic configuration.

In order to take into account the workflow particularity, the RBAC model is extended to the DW-RBAC [Wainer 07] by adding the "doer" relationship. This relationship means that the user can complete the task within specified cases. It also adds dynamics to monitor the integrity of workflow. The delegation is defined in two ways. In the first option, a user can delegate to another user the right to perform a task. In the second method, it can also give the right to delegate. This model provides two types of delegations: specific type delegations applied only to particular cases and the generic type delegations which is a context independent delegation of a task.

5.4 General Security Topics

The security mechanism and measures have many objectives each of which is applied to assure certain aspects of security, the process of ‘‘securing” an information system has two functions, first, the access control which incorporate the authentication i.e. verifying user identity and the authorization i.e. verifying whether or not to permit action on a resource. Authorization is achieved according to constraints issued from confidentiality; protecting sensitive data and privacy.

In the other side, at the information exchange level, security concerns the data integrity and non-repudiation; detecting data tampering and making sure neither the sender nor the receiver can deny the message they sent nor received, this is usually achieved by many techniques like the digital signature and cryptography. However, both functionalities are governed by a security policy issued from business and organisation strategy.

5.4.1 Authentication

Authentication is the process of establishing that one is who they claim to be; that you are authentic [Abele 09]. Identity is the set of behavioural or personal characteristics by which an individual is recognizable from other members of a group [Farlex 09].

Most applications require a user to establish his or her identity before requests are served.

This is either because security restrictions require that applications be provided only to authorized users and most often it is necessary to determine user identity to figure out if a user is authorized or not. Our application logic requires the knowledge of who the user is in order to customize its behaviour according to user's identity. A user claims an identity by sending an identifier such as a username, or digital certificate.

5.4.1.1 Password Based Authentication

As identifiers may be public knowledge or may be easily guessable. An identifier by itself, in general, cannot establish user identity. In most situations, an identity claim consists of an identifier as well as proof of identity which is some secret that only the identity owner and the application know or verify.

Providing a preregistered password as proof of identity is one of the most common authentication ways.

When using a password to authenticate a user, an application needs to verify the password presented by the user. For this purpose, the application can consult a password store. There are several locations where applications stored passwords: In Unix, passwords were traditionally stored in the file system. Most modern applications include support for accessing the passwords from directory using a protocol named Lightweight Directory Access Protocol (LDAP) [Kanneganti 07].

5.4.1.2 Kerberos

Kerberos [MIT 2009] depends on encryption (PKI) technology, under the Kerberos scheme, every user registers a long term key in the Key Distribution Center (KDC), the KDC holds the long-term keys for all the Servers and users. The only two parties that know a user's long term key are the user and the KDC itself. When a user wants to authenticate and communicate with a server, she simply requests what is known as a session key.

The KDC dynamically generates a session key and sends two copies of it—one to the user (the client) and one to the server. Each copy of the session key is encrypted using a well-known encryption function that is parameterized by the long-term key of the intended recipient. Each party can use its long-term key to decrypt the session key copy provided by

KDC. Subsequently, the client and the server use the session key to encrypt all communication between them. Kerberos addresses authentication problem by introducing the concept of a ticket [Kanneganti 07].

A ticket consists of the client's identity information and the session key granted to the client by the KDC for a particular service, and is encrypted using the service's long-term key. As the service's long-term key is not known to anyone except the KDC and the service itself, a ticket cannot be forged. The ticket can be used for authenticating the client as well as securing the communications between the client and the target service.

5.4.1.3 Digital Certificate Based Authentication

In this paradigm a central store trusted by the application is called a Certificate Authority or CA. The information signed by a central store a public key, the name of the entity that owns the key, and other details such as expiry date, is called a "digital" certificate. In other words, a certificate not only provides the public key of the certified party, but also provides the certified party's identification information [Sun 08].

Public key certificate, when combined with digital signatures, provide an alternative to the authentication mechanisms we have discussed in the previous two sections, namely password-based schemes and Kerberos. Using public key, each party can prove its identity to the other by signing the message with its private key and attaching its certificate to the message. The other party can validate the sender's public key and identity information by checking the CA's signature in the certificate. The request must be verified to see that it came from the party identified in the certificate by checking the signature provided.

One interesting issue related to this concept is certificate chain which is an ordered sequence of security certificates, in which the first security certificate contains security-relevant information, and each subsequent security certificate contains security information which can be used in the verification of previous security certificates. Certificate authority (CA) hierarchies are reflected in certificate chains. A certificate chain traces a path of certificates from a branch in the hierarchy to the root of the hierarchy [Sun 08].

The purpose of certificate chain is to establish a chain of trust from a peer certificate to a trusted CA certificate. The CA vouches for the identity in the peer certificate by signing it. If the CA is one that you trust (indicated by the presence of a copy of the CA certificate in your root certificate directory), this implies you can trust the signed peer certificate as well.

However, in the service world there is still problems; if a service is invoked in different ways, how can authentication take place? For instance, if the service is invoked within the same

enterprise, we can use the LDAP directory. But, if it is invoked from outside the enterprise, that repository is of no use. Moreover, when a service uses another service, how does it provide the credentials? Can we trust the authentication done by one service and reuse it? How do we make sure we communicate the results of authentication between services? All these questions are answered by the different Federated Identity Management approaches [Kanneganti 07].

5.4.1.4 Federated Identity Management

Securing corporate information system requires defining a convenient access control policy by imposing constraints on resource access according to business strategies depending on requestors' identity (i.e. identity claims, roles, preferences, context…etc.). Such a context involves dealing with authentication and authorization policies of various identity owners who may join and leave dynamically. Assuring authentication and authorization in requires cross-boundaries access control to deal with various services directories and divers identity providers and manage identity requestors, such as users or services related information, which are usually stored in a particular registers as part of corporate directory.

Identity federation solutions allow creating such architecture by adopting a cross-organisation authentication and authorisation mechanisms. It describes enabling technologies and standards of identity information usability across autonomous security domains and involves many issues such as trust management, cross-organization single sign-on, cross-organization user account provisioning (eliminating need for multiple account registration through automatic systems) and cross- organization user attributes exchange.

All together, these issues enable single set of user security credentials to be used and allow access to multitude federated resources. The federated identity paradigm offers unique, simple and transparent authentication mechanism across multiple applications while keeping the final authorization decision to the end application or service.

The Federated Identity Management is initiated by OASIS and the Liberty Alliance. The Federated Identity Management enables authenticated identity across multiple organisations. It avoids centralized storage of user information, while allowing users to link multiple local identities related to different accounts. A key element of Federated identity is Circle of Trust that identifies that a user belongs to a community to access certain services.

It broadly covers three specifications:

1. Identity Federation Framework (ID-FF) [Wason 05]: allows features for SSO, account linkages, anonymity, affiliations and options for meta-data exchange.

2. Identity Web Services Framework (ID-WSF) [Madsen 08]: allows features for Permission Based Attribute Sharing, Identity Service Discovery, Interaction Service Security Profiles and Identity Services Templates.
3. Identity Services Interfaces Specifications (ID-SIS) [Cahill 08]: allows for interoperable services to be built over ID-WSF. Interoperability is offered through use of context dependent agreed schemas.

Many projects and initiatives have created architectures and open-source technologies and/or specifications for federated identity-based authentication and authorization like Shibboleth [Scavo 05], OpenAthens [1], OpenID[2], among these works we can distinguish Shibboleth which has the advantage on other architectures by introducing the notion of Where Are You From (WAYF) service. This component allows providing seamless access control for collaborating organisations by supporting a flexible cross-organisation authentication mechanism, while allowing enforcing local authorisation policies to control what resources authenticated users are allowed access to.

5.4.1.5 Shibboleth

Shibboleth is an Internet2 middleware which supports the establishment of federated authentication and convey security attributes across independent organisations [Scavo 05]. User authentication at a local Identity Provider (IdP) can support Single Sign-On across a federated organisations where security attributes and assertions are released and used by service providers (SP).

The key element of Shibboleth is trust; through Shibboleth, each service provider has pre-established trust relationships with a set of Identity providers (IdPs). Thus, end users having single usernames and passwords from their home organisation are provided with seamless access to various service providers resources. The Shibboleth architecture consists of Identity Providers, Service Providers and Where Are You From (WAYF) services. When a user attempts to access a service he/she is redirected to a WAYF server that asks the user to pick his home (IdP) from a list of known and trusted service providers. The user then is redirected to his/her authentication server. The home site redirects the user back to the SP while providing a digitally signed SAML [Madsen 05] (Security Assertion Markup Language) authentication assertion message i.e. token, asserting that the user has been successfully authenticated and providing the SP with a temporary pseudonym for the user (the handle), the

[1] - Eduserv, OpenAthens framework: http://www.athensams.net
[2] - OpenID Foundation, OpenID: http://openid.net/

location of the attribute authority at the IdP site and the resource URL that the user was previously trying to access. The digital signature on the SAML authentication assertion is verified, the resource site then returns the handle to the IdP's attribute authority in a SAML attribute query message, depending on which the IdP's attribute authority returns a signed SAML attribute assertion message (called Attribute Certificate AC). Depending on the attributes assertions, SPs within a federation are still autonomous and are able to decide whether the provided attributes are sufficient for access resources and which attributes they are able to release to an SP according to the degree of trust it keep with them.

As we observe the trust notion is fundamental to deal with identity federation. To this end, the SOA Reference architecture proposed by the OASIS [McCabe 08] proposes a trust model between services stakeholders and service customers. In the next following paragraph we present an abstract description of OASIS trust model.

5.4.2 Authorisation

Once a user is authenticated, an application needs to determine whether the identified user is authorized to access the functionality he is requesting. The decision to grant access may depend on multiple criteria, such as the action that is being requested, the resource on which the action is being requested, and the groups to which the authenticated user belongs or the roles that the user plays.

5.4.2.1 Role-Based Access Control and Attribute Based Access Control

Most applications come with their own access control model; that is, the logic for deciding whether or not to grant access for a particular action on a resource is hard-coded into the application [Kanneganti 07]. Some of the information used to decide whether to grant access is often pulled from a directory server or a configuration repository. The two most common access control models in service world are Role-Based Access Control (RBAC) and attribute based access control ABAC. In RBAC permissions for each action on a resource are granted to one or more role. Information on what roles is granted to which users can be maintained in a directory. RBAC is discussed in details in another part of this document. ARBC access control works differently. Administrators associate a list of rules with each resource. Each resource may be associated with a set of rules describing authorised user's attributes such as age or sex. RBAC and ABAC can be combined to provide a powerful authorisation model. These two approaches have been discussed in section 5.3.

5.4.2.2 Key Oriented Access Control

The idea is to rely in the local authorities and trust resulted from business relation in such a way to avoid the fixed hierarchical authority structure.

In such scenario the local entities controlling the access to the services act as an authority for those who depend on them for the services using the delegation mechanism. The delegation is achieved by signing delegation certificates. Delegation certificates are used by an entity to delegate some of its rights to other entities which in their turn can delegate these rights and so on. Finally, the certificates can form a complicated network that reflects the underling business relations between the owners of the private key.

By taking into consideration the delegation the system avoids the dependence on global trusted name/key services. If any names are used they are not globally distinguished names but relative to local business communities [Aura 99].

In this way we can ensure the distributing ability, scalability, and the locality of policy decision while preserving the global coherence of policy enforcement.

A delegation certificate is a signed message with which an entity (issuer) represented by its key grants access rights to another entity (subject) represented also by its key.

The authorisations granted according to the rights mentioned in the certificate are application dependent i.e. each application maintains its own rules for mapping the mentioned rights to its own authorisation domain.

When a delegated subject wants to access a service it signs a request attached to the delegation certificate.

The subject can re-delegate the rights it received if the re-delegation is not explicitly forbidden in the certificate. This can form a chain of delegation certificate in which the original issuer is the service provider and the final subject is the requestor.

The rights passed through a chain are the intersection between the rights possessed by the first issuer and all the rights mentioned in all the certificates of the chain. It is possible to put condition on the certificates in order to limit the re-delegation to a finite number [Aura 99].

For more complex verification such as security policy enforcement, we can use the conditional certificate. A conditional certificate explicitly states that the rights are granted to the subject only if all the condition in the list are satisfied, in order to use the certificate, the subject must provide a proof that the conditions are fulfilled by attaching appropriate certificates chains, one for each condition.

5.4.3 Data Confidentiality and Integrity

Data exchanged over a network needs to be protected from curious eyes. Encryption is the standard technique used to protect confidentiality of exchanged data.

Traditionally, applications establish a secure channel for data exchange using Secure Sockets Layer (SSL)/Transport Layer Security (TLS) [Dierks 08] by encrypting exchanged data. SSL is a set of rules regulating encrypted communication between two sides, i.e. client and server.

SSL is used to form a secure connection between clients and servers so that they can communicate insuring confidentiality using messages encryption, integrity using signatures and authentication using digital certificates [Kanneganti 07].

This type of secure connection ensures that all data exchanged between sender and receiver is encrypted using public-key certificates as a means of authenticating principles. As part of the initial SSL handshake process, the server presents its certificate to the client to authenticate the server's identity. The authentication process uses public-key encryption and digital signatures authenticate the server, i.e. the server's certificate is valid. Once the server has been authenticated, the client and server use symmetric-key encryption, to encrypt all the information they exchange for the remainder of the SSL session and message digests to detect any tampering that may have occurred [Kanneganti 07].

5.5 Conclusion

We have discussed general security issues applied to resolve and manage security for traditional enterprise systems, however, in SOA there is additional complexity as information cross enterprise boundaries. Traditional data confidentiality strategies become less useful. For instance SSL/TLS can not protect the confidentiality of a message when it is passing an intermediary service throw the message path in a chain of services. Clearly, we need to enhance confidentiality techniques to overcome these new issues.

As with data confidentiality, the traditional strategy to protect data integrity and non-repudiation is not enough when higher-level services bring together lower-level services from different parties. A secure channel provided by SSL/TLS cannot prevent an intermediate service from providing different information for the end service than what the customer provided. We need new and better ways of ensuring data integrity and non-repudiation [Kanneganti 07].

In collaborative SOA context, we should consider security concerns from internal and external enterprise stakeholder view points; each stakeholder has its own perception of these concerns, this imposes being aware of all involve parties perceptions of these concerns

especially in the SOA context where services are executed on the provider side offering functionality to the customer.

In SOA a composite service puts together capabilities of various services. An action in the composite service consists of multiple actions and multiple services calling, the composite service should check the access control rules of all “composing” services before initiating an action. This requires that all access control rules of each composing service are available to the orchestrator. Unfortunately, usually, authorization rules are hard-coded into applications in an opaque way, this reduces dramatically the reusability of the services when services with different access control models are brought together to achieve a composite service. Any authorization strategy for SOA will have to address these issues.

Chapter 6

6 SOA Security

6.1 Introduction

The service-oriented architecture supports the dynamic building of composite services provided by different enterprises to deliver variety of innovative “on demand” services. However, collaboration via SOA raises many challenges especially in terms security; in cross-organisation context different information systems having different Quality of Protection (QoP) preferences and security policies interoperate to carry out collaborative services. In such a context, insuring end-to-end security involves dealing with various security policies and strategies, authentication and authorization systems as well as an ever increasing number of partners who may join and leave dynamically. This requires flexible security solution to deal with heterogeneous services infrastructures issued from dynamic "network" of collaborative partners each of them operates in different context and uses different policy to enforce security upon its own resources.

It is clear that traditional approaches to various aspects of application security will no longer suffice. New standards tools and security infrastructure are needed to allow overcoming long-standing barriers between applications belonging to various partners and dealing with new issues such as context propagation, heterogeneous policies matching and enforcement, cross-boundary identity and access control management. Fortunately, SOA allows a few new approaches, thanks to the standards it supports, that fit the changed requirements of application security discussed previously. In this section, we describe a new security approaches, namely message-level security and Federated Identity Management. We will discuss also some important issues related to SOA security like customer preferences acquisition and conformity, and security policy concerns.

6.2 OASIS Reference Architecture Trust Model

The OASIS reference architecture proposes a trust model between services stakeholders and service customers in order to define who can perform an action or raise an event. The Trust Domain is defined to model abstract concepts over policy-based trusted social groups. It is defined as “An abstract space of actions which all share a common trust requirement; i.e., all participants that perform any of the actions must be in the same trust relationship”. This leads

to define several trust level associated to different “social groups” mixing stakeholders and participants and trust relationships (Figure 14). Trust relationships between participants can be used to derive decentralised authentication and authorisation so that trust chain can be established for a global service chain.

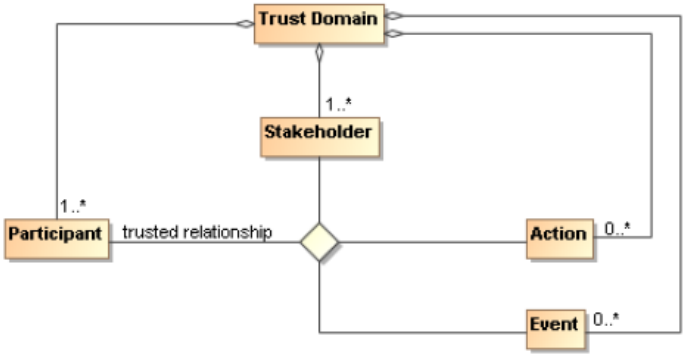

Figure 14 Authorisation Model [McCabe 08] p. 90

Such a credential-based security mechanism (Figure 15) expresses security choices that are then used to define security mechanism to enforce security. To be able to communicate these choices, they can be described in policies. Policies are defined as a set of rules that can be tested by decision points whereas enforcement points are used as relays in the distributed system (Figure 16).

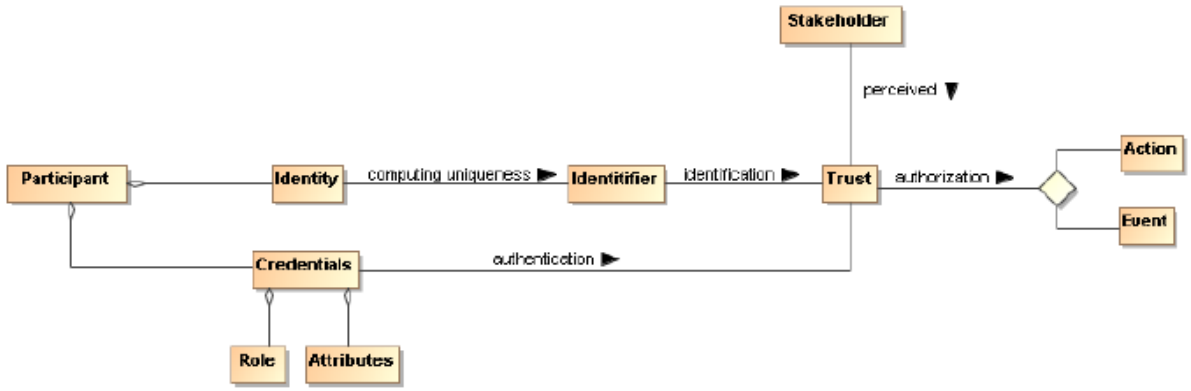

Figure 15 Decentralised Authorisation Model [McCabe 08] p. 92

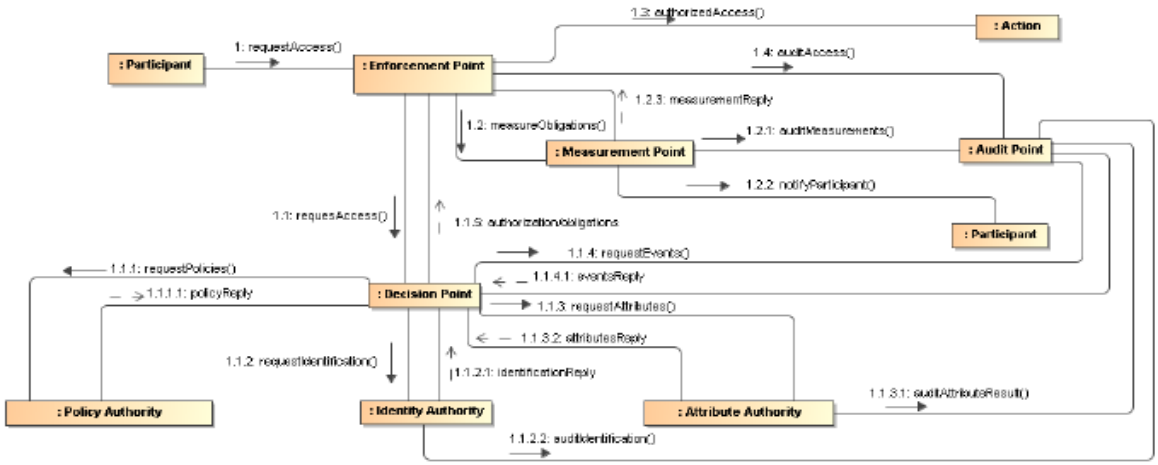

Figure 16 Organisation of a Security policy Model [McCabe 08] p. 92

In addition to services security requirements, requestors should be able to specify security requirements and preferences. Security requirements could be classified into two categories: Service security quality such as security level and the mechanisms used to meet it, and privacy such as personal data management. The management of these two aspects can be organised in four sub-domains:

1. User-side security management: user security and privacy preferences.
2. Service side security: service security and personal data management policy.
3. Interaction management: policy matching and negotiation management.
4. Propagation management: trust and context establishment and management.

As follows we will discuss technologies and standards related to user preferences and service policy representation and matching.

6.3 Customer Preferences Acquisition and Conformity

In the context of collaboration via SOA, a huge quantity of personal data is exchanged in order to achieve various types of transactions using web services. Users usually impose some restrictions concerning their personal data use and defuse. This forces the different service providers to offer tools and means allowing the expression of customer requirements and preferences. In the following we expose some languages used in this domain.

P3P [W3C 03] consists of specifications allowing a service to describe its personal data management policy such as how it processes a user's personal data, allowing to a user to compare and match his or her preferences and requirements with regards to service policy. This language is based on XML. The service provider expresses its policy in a P3P file using XML tags. The most important element of P3P file is as follows [Cranor 03]:

1. Entity: makes it possible to listen to the information about enterprise activity and organisation as well as the person responsible for the site.
2. Direct access: mentions if some personal data will be stored in the database.
3. Dispute: describes how security policy is adapted in the case of conflict with user preferences.
4. Data: lists the collected data categories
5. Purpose: describes how these data will be used.
6. Recipient: how and under what conditions these data will be shared.
7. Retention: the time during which these data will be stored.
8. Consequence: legible description allowing a correct use of the service.

Each of these elements is expressed as an XML tag in the P3P file. User preferences and requirements could be expressed using sub-tags and attributes. The real implementation of such specifications requires an automatic processing of user preferences and service policy (without the explicit intervention of human actors); to date no standard mechanism is specified for this purpose. In this order [W3C 02] propose APPEL (a P3P Preference Exchange Language), It is a W3C specification allowing a user to specify how his or her personal data should be processed or handled. These specifications are formulated into clauses or rules allowing formulating the behaviour to be adopted concerning each aspect of the personal data. The APPEL specifications also define research and matching algorithms allowing automatic matching of user security preferences to services security policy. However, the APPEL matching mechanism is not very efficient given that it only permits specifying the valid element in a P3P policy; it does not make it possible to specify non-valid elements. In order to overcome this problem, [Agrawal 05] proposes another language called Xpref. This language allows the description of user preferences according to two rule categories:

- Acceptable rules: make it possible to specify acceptable rules in P3P policy.
- Non- acceptable rules: permit the specification of non-acceptable rules in P3P policy.

Automatic processing of user preferences and service policies requires the introduction of service semantics in order to render the actions related to this matching more intelligent. In this domain we can mention SWAPPEL [W3C 04] (Semantic Web APPEL). The principal specifications of this language are thought to integrate a semantic rules management described using some adapted languages like SWRL (Semantic Web Rule Language). It consists of a head based on the syntax of APPEL and a rule body representing the behaviour, description, prompt and prompt message as child tags. It uses a standard query language to match elements of the P3P policy. We should mention here that SWAPPEL, P3P and Xperf integrate mechanisms concerning personal data management negotiation.

User preferences as a whole can be acquired both manually and automatically. In the latter case, we can easily note the dynamic nature of these references depending on the context and thus depending on the information characterizing the environment and the situation of the user [Razmerita 05]. Services should be aware not only of explicit user preference and requirements but also of requirements stemming from this context which could impact customer preferences and the chosen set of service policy; this context should be able to be characterized and transferred along the service chain.

Many studies have been conducted to determine elements characterizing and propagating customer context in order to better understand the user's environment. This could be translated by the usual questions; who: identity, what: activity, where: location, when: time, and why: objective. Among these studies, [Abowd 99] introduces the concept of Activity, reflecting a final objective aimed by the user. The work in [Rafiy 07] models the user context by gathering the elements necessary to understand of the user's environment. In this model, in addition to the information related to user's activity (transaction) user's identity and time, we find information characterizing user Beliefs. For example, the user that uses a shopping site does not usually wish to communicate his or her personal information due to beliefs that he or she will be subject to telemarketing. The implementation of such a context has been achieved by implementing the context definition in the form of an XML schema by extending the concept of PeCAN (Personal Context Agent Networking) to put together timely and effective matching between user context and preference on one hand and the service policy on the other hand.

To assure that the mechanisms and technologies described above are applicable along the service chain, a mechanism of user attributes exchange and transmission is necessary in order to avoid multi point users to service connection establishment. The main idea here is to create a token once the user is authenticated for the first time, then this token can be transmitted to other services. This token could be considered as user context description.

Inspired by SSO Single Sign On of the distributed web application, [Gross 03] proposes using SAML technology to propagate user context. SAML is a standard XML based language and protocol thought to exchange security assertions. SAML assertions help in security context attributes transmission, including user attributes. SAML assertions are created to convey the security context to other entities that depend on this context for their security or business logic. The entity that manages security communicates the security context to all nodes in the message path. SAML fulfils these requirements as it provides a language for expressing the service security context. SAML is an XML-based standard for exchanging authentication and authorization data between security domains, that is, between an identity provider (a producer of assertions) and a service provider (a consumer of assertions). SAML makes it possible to exchange security information between services using three kinds of assertions:

- Authentication assertion: asserts authentication results, after a service authenticates a user.

- Attribute assertion: asserts user attributes, for instance validating the username and password or asserting the identity of the user. It can assert more information, such as the groups the user belongs to, the user's preferences, and the user's location..
- Authorization Decision assertion: asserts the authorization decisions by conveying the kind of access granted to the user for given resources.

In addition to these assertions, context exchanged by SAML can include security metadata making security decisions possible. SAML can be integrated into existing frameworks like SOAP/HTTP. Although SAML provides solutions for many problems, it suffers from some drawbacks. For example, it cannot manage problems like cryptography and global sign out for a chain of service, expressing and exchanging trust issues, and security meta-data exchange, these questions are better treated in the Roadmap [IBM 02] which proposes specifications and a security model for services by integrating security standards and mechanisms including a message security model, service endpoint policy, trust model and privacy model. The proposed specifications are resumed as follows:

- WS-Policy: XML based specification which describes a model and its corresponding syntax allowing a service to express its security policies capabilities and constraints.
- WS-Trust: describes a framework for trust models that enables Web services to securely interoperate.
- WS-Privacy: provides a framework to bind organizational privacy assertions and requesters privacy preferences enabling an automated processing of these preference and privacy assertions.
- WS-Secure Conversation: describes how to handle and authenticate messages and security context exchanges.
- WS-Federation: describes how to broker trust relationships in a heterogeneous federated environment.
- WS-Authorization: describes how to manage authorization data and authorization policies.
- WS-Security: a protocol describes how to append signature encryption headers and security tokens to SOAP messages.

Here we will emphasize WS-Security as it is the most popular way to express message level security-related information such as encryption and signature to insure data integrity and confidentiality, since message is considered as standard mean to exchange information in service context.

6.4 Message-level Security

With Message-level security approach, different elements of a message can be protected differently to make them usable only by intended user or service in the message path. Protecting different parts of a message differently is commonly referred to as message-level security [Kanneganti 07]. A standard called WS-Security [Lawrence 06] allows message-level security to be implemented with SOAP. In the following we will describe this standard.

WS-Security defines a Security header in which security claims are inserted to ensure message-level security. A security claim is any security statement, such as "My name is X.", "X is authorized to access this resource" This message is signed by X," or "This message is encrypted using X's public key."

SOAP header entry is used to carry security-related information using security tokens. A security token is a collection of one or more security claims. For example, a UsernameToken includes the name of the message's sender. WS-Security defines many kinds of security tokens to conduct different security claims. WS-Security token profile defined how the different kind of tokens should look like and in which way should be used. In addition to standard security tokens that can express the most popular security claims, WS-Security allows defining customized security tokens using custom elements and attributes within Security header entries; however, since these are not standard, the semantics of such tokens should be agreed between sender and receiver.

WS-Security left the way a service can verify or process security claims to the implementation logic, that is, it does not impose a standard way by which a service process the claims, this aspect is usually expressed as part of security policy.

6.5 Service Security Policy Concerns

Each service or resource must be able to express its capabilities and requirements concerning security by transforming them into a security rules such as security policy.

The idea behind policy-driven security is that security requirements and mechanisms must not be hard-coded into applications. Instead, enterprise security should be declared separately as a "security policy."

A security policy declaration separates security logic from business logic, leaving the former to security specialists. In this way, it becomes easier to ensure consistency of security enforcement across multiple services. In the other hand, policies of involved service providers

can be easily compared to bring coherence to thier security implementations, as a result, interoperability is enhanced.

WS-SecurityPolicy [Lawrence 06] provides security-related assertions that can be used to express the requirements, capabilities, and constraints of services security implementations.

The assertions defined by WS-SecurityPolicy apply to SOAP Message Security (Figure 18), WS-Trust and WS-SecureConversation, and can be classified into four groups based on the policy subjects they can be associated with [Kanneganti 07] (Figure 17):

1. Endpoints level Policy assertions: security policies that can be set at the endpoint level.
2. Operations level Policy assertions: security policies that can be set per operation.
3. Messages level Policy assertions: security policies defined at the message level.
4. Nested Policy assertions: these assertions cannot be directly associated with a subject can only be nested into other assertions.

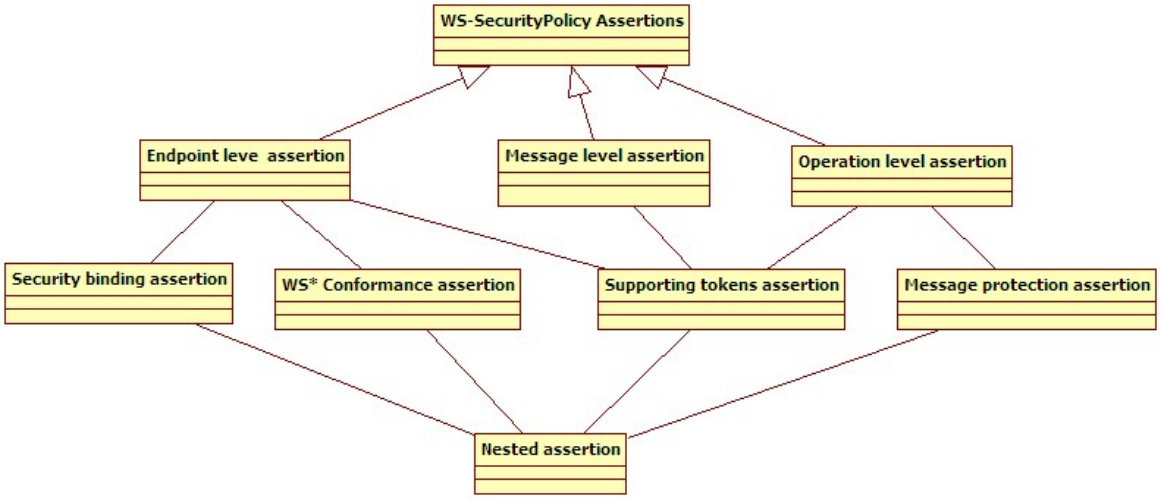

Figure 17 Typology of WS-SecuritPolicy Assertions

```
<wsp:Policy wsu:Id="X509EndpointPolicy" >
   <sp:AsymmetricBinding>
      <wsp:Policy>
         <sp:IncludeTimestamp />
         <sp:OnlySignEntireHeadersAndBody />
      </wsp:Policy>
   </sp:AsymmetricBinding>
</wsp:Policy>
```

Figure 18 Example of WS-Security Policy

Although WS-SecurityPolicy supports partly interoperability, it falls short in providing a full security implementation of services. For instance, as we can note from above discussion, it

does not support assertions whose subject is a service. Many other details are needed by a security policy statement but that are not covered by WS-SecurityPolicy. For instance, we can neither specify the directory to be used for authentication, nor specify the Certificates Authority to be used for certificates, we cannot specify whether the token to be used must contain an authorization assertion or not. Thus, we should rely on other extensions to fill this gap. There are other standards that focus on codification of domain specific policies.

One of the languages thought to support the expression of such policy is the XACML (Extensible Access Control Markup Language) [Anderson 05]. XACML is an OASIS standard based on XML. It permits the description of access control specifications. The XACML element is composed of a set of security policies (<policy set>) which are formulated as predicates or functions inputting requestor characteristics such as role or environment (date, time..) and outputting a positive or negative answer (authorized or not). The first drawback of this language is its processing complexity. Additionally, business holders lack the required knowledge to treat such a language.

To overcome these drawbacks, many works have been carried out. Among these, we can mention SECTET-PL [Alam 06] which is similar to Object Constraint Language of UML (OCL). With SECTET-PL we can transform UML specifications into XAMCL specifications.

In addition to the dynamic composition of programs or actions, web services compose or compute requested information from various data sources and provide a dynamic generation of new of data which have to be dynamically classified and secured according to their mutual source policies. In other words, in analogy to the dynamic composition of web services, there is a need for a dynamic and consistent composition of the related security policies of all participants. To satisfy these needs, [Han 06] adopts techniques from language-based information flow control through modelling confidentiality of data as restrictions on the flow of information between domains of a global system to control the privacy of dynamically-created data according to the policies of involved web services. Each piece of data is augmented by a type specifying its classification with respect to various user-defined security categories. Customers formulate their security requirements by attaching types to customer data and individual web services or classes of web services. As a result, web services can only be used for service composition if they provide the necessary clearance for the information they would receive when executing the service. Additionally, once new data is synthesized, it is automatically classified according to the classifications of the data that have been used to compute it. In this approach, a mechanism to synthesize a common security policy for all participating web services with respect to customer policy is described. A service is

considered secure if the low-level privacy output of the service does not depend on the high-level privacy input. Each customer has its own perception about the security categorization of individual or families of web services and no central authority is required.

The security policy can be seen as a set of assertions concerning used resources and/or activity enactment (which users may use the resources, when and where resources could be used etc…). At the service level, the security policy can be seen as a set of metadata on a service. These metadata must facilitate the discovery and selection of services, in this case, security policy and its related enforcement context must be described using common semantically-agreed well-structured vocabularies such as ontology. The ontology allows us to define a vocabulary ensuring good and common expression of concepts in a particular domain. The US Navy Research Laboratory has developed an ontology for the domain of computer security based on a former ontology. In this domain of DAML Security ontology, NRL-SO is divided into several sub-ontologies covering the main areas of the Security.

The following elements are included in the synthesis presented in [USDoD 04]:

- Service Security Ontology: introduces new and needed security concepts.
- Agent Security sub-ontology defines resource query functionality.
- Main Security Ontology Information: is the principal ontology of NRL-SO it defines the main security concepts by three sub-divisions: protocols, mechanisms, and policies.
- Information Object Ontology: sub-ontology that defines information processing concepts.
- Security Algorithms Ontology: sub-ontology that defines all the concepts related to security algorithms such as key and signature.
- Credentials Ontology: sub-ontology defining all the concepts related to certificates.
- Security Assurance Ontology: sub-ontology defining all concepts related to the assurance of processed or manipulated elements.

With the NRL ontology, the user and the service can describe their security abilities and needs using objective security terms and properties. With these descriptions translated into XML, it will be possible for a service to confirm user needs and preferences.

The question that arises is what happens if no appropriate service is found? Could a compromise between user preferences and service policy be concluded? To answer this question a negotiation phase must be established.

6.6 Negotiation and Matching Customer Preferences and Service Policy

In the case of conflict between customer preferences and service policies, a negotiation phase should be established in order to assure a compromise between them. A framework to manage this kind of negotiation is needed. In this light, [Trabelsi 07] proposes extending the service description and user query by including them in the service security policy and user preferences respectively. In this case, the service broker is charged of the task of matching client preferences and service policy, and establishing a negotiation phase in the case of a matching failure. Thus an extended XACML ontology is used for a common understanding of the diffident semantics describing the client request and service content. However, in this work, a negotiation framework has not been proposed. [Han 06] proposes a framework that achieves such negotiation. This framework distinguishes between security objectives and a description of security attributes. The former describes high level security objectives like confidentiality or reliability, while the later describes the elements with which we can reach these objectives. Only security objectives are published. The first phase consists of automatic negotiation in order to match user preferences and service security policy objectives. At the end of this phase, a security contract is generated to each service involved including the chosen security attributes to be imposed on each service. In this approach, two drawbacks can be observed: first, user preferences are formulated as objectives. A user cannot impose preferences using technical terms and attributes. Additionally, no personal data management policies can be negotiated which is an important question treated in many research works. [Preibusch 06] proposes a negotiation framework which offers an automatic negotiation between P3P service side security policy and APPEL, Xpref client side security preferences.

6.7 Conclusion

We have disused in this part of the state of the art different issues related to SOA security trust, requestor and provider security requirements and preferences negotiation and matching, security context acquisition and propagation as well as the different tools and solutions used to express and treat these issues (Figure 19), however, in the context of collaborative business SOA there are much more concerns to be taken into account, in fact, collaborative SOA involves distributed services stemming from various firms; security policies should adapt service behaviours according to the ever- changing context of different involved partners, their locations, third party applications and resources. Furthermore, customers and service providers may exchange data with respect to different perceptions of using and securing these

data. A framework for service security policies must take into account service composition and incorporate security requirements including their preferences and context. For this, a global Quality of Protection (QoP) agreement should be established by stakeholders and must be guaranteed by service deployment. Despite numerous academic and industrial activities related to security, user preferences and context-aware environments, there is not currently a global approach to provide end-to-end systematic analyse, design and integration of adaptable, cross-layer, distributed and service oriented security to process and organisational structures down to disparate and heterogeneous applications and computing platforms.

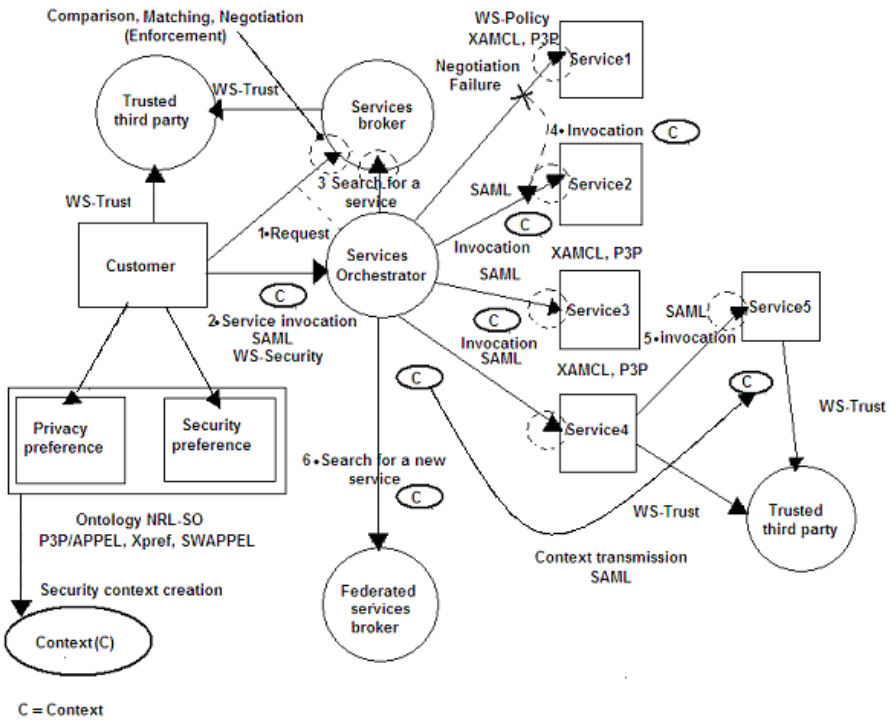

Figure 19 Integration of the Different Elements Presented in the State of the Art

Chapter 7

7 Conclusion of the State of the Art

In the state of the art we have introduced different issues related to our subject ranging from modelling approaches, and production organization issues, to SOA and information system urbanization approaches, we concluded from the previous that traditional modelling and urbanization designing methods make use of a rigid system view which assumes that the business process acts in a pre-programmed and supervised way, this limits the flexibility and autonomy of process enactment in the collaborative contexts since cross-enterprise business process involves a number of components designed and configured independently and thus so far to interact coherently. In the other hand, we concluded that traditional security solutions fail in the context of collaboration, as in collaboration scenarios, involved enterprises are forced to open their systems, not only on data exchange level, but also to achieve more complex interactions required to coordinate and orchestrate the achievement of collaborative process.

A compromise should be made to bring together interoperability and partners' autonomy along with security requirements of each. Our trend is to propose an approach allowing building a flexible collaborative process from useable components in a decentralized way allowing enterprise to exploit market opportunities and to thrive under conditions of unpredictable changes. This proposal should consider the collaboration at all enterprise life cycle phases and should support a process which integrates production and information systems while meeting the security constraints and preferences of the parties involved.

Chapter 8

8 Towards a Collaborative Service-Oriented Organization

8.1 Introduction

Collaborative production processes are transversal due to their nature, for example, they involve a transversal or horizontal organization starting from customer relationship management and supplier relationship management to the production process itself. An enterprise must thus reorganize process management to balance power relationships between such transversal process owners and "vertical" business unit managers in each enterprise. In other words, it has to simultaneously improve the "vertical performance" level and its amount of "horizontal enactment."

Collaborative Production processes are constructed from diverse specialized processes issued from different partners. To allow the discovery of potential partners and the establishment of such "composite processes," "composing" industrial constructs should be described, published and rendered accessible via adapted platforms. To do this, two requirements must be satisfied:

Since production means, physical and human resources are structurally interrelated to the production process organization and execution, we should identify consistent grouping of these objects in order to be able to expose functional constructs (called in or work "industrial services") allowing composing collaborative industrial processes. This involves redefining the organization through goals, identifying processes to achieve and then determining for each process the required competencies and means without being -a priori- constrained by corporate organization. Publishing these functional constructs involves integrating into their descriptions -in addition to functional aspects (like product or service description, coordination management…etc), non-functional terms related to Quality of Services offered by each party, policies and standards to be respected and privacy and confidentiality constraints. Privacy and confidentiality constraints have a special importance in the collaboration context as they can determine which part of the process can be exposed as white, gray, or black boxes, that is to say process enactment visibility.

Clearly, fulfilling collaborative process objectives requires the participation of many independent parties, each of them has their own processes, data, production and information systems, this requires each party to adapt itself to a collaboration scenario.

First step to enable collaboration, partners should move from dedicated organisation and infrastructure to interoperable ones.

Generally, the term interoperability covers three aspects i.e. technological, semantic and organisational interoperability. Technological and semantic operability allow heterogeneous systems issued from different business domains to exchange messages and interact in a seamless way and without particular efforts, while marinating the autonomy and privacy of local systems, whereas organisational interoperability goes within enterprise organisation boundary, and impose adapting organisation structures and means to order to attain a dynamic organisation which is capable to provide suitable and instant responses to diverse collaboration scenarios.

8.2 Infrastructure Choice

Service-Oriented Architecture offers the aforementioned abilities i.e. publishing, discovering and composing functionalities within and across the enterprise's boundaries to offer new and more coarse-grained business processes, as well as a framework and a set of enabling technologies to support the organisation and management of the collaboration including its functional and non-functional aspects. Nevertheless, from the previous discussion, we can conclude that to support efficient collaboration and establish a common ground for collaboration between enterprises, we must pay a particular attention to two main issues:

1. Interoperability between different partners' production and information systems involves providing the capabilities allowing the separation between the business logic (industrial services) and the technical logic (the way these services are invoked and composed). Interoperability can be assured, at the technical level, by choosing a platform of agnostic middleware able to mediate between distributed information systems and manage the interaction between disparate applications across heterogeneous computing platforms. Yet, at the business level, collaboration requires rethinking involved enterprises organisation and information systems simultaneously in order to obtain a "lean" organization allowing "on-demand" reconfigurations of production processes by integrating production constraints into the information system i.e. taking into account the way business processes are related to the way the manufacturing enterprise transforms raw materials into products, such as its production process.

2. Integrate and bring consistency to different partners' security strategies; in such complex environment the architecture dynamic should be able to tackle failures and unexpected risks. Consideration of security requirements of all partners and the protection of data and services stemming from various sources requires a flexible security policy framework. Such a framework must assure awareness of complex situations through intelligent security policy composition and verification mechanisms. The mechanism to be used should enable a compromise of the missions, goals and preferences of involved parties and enforces context-aware security policies in order to ensure satisfactory end-to-end security enforcement.

In the following section, we will sketch our methodology to resolve these two issues.

8.3 Towards A Production-Oriented Urbanization Strategy

To resolve the first issue we propose an urbanization strategy which emphasizes industrial organization to determine which industrial services could be compound in a consistent manner (by following the production logic.)

Our objective is to identify industrial "services" "which will be published and offered in such a way to allow the composition, in other words to allow organizing logical groups of production resources (production islands) and manage the business flows between these islands, and integrate all services required to complete a final product in response to customers' orders "on demand" (define a production system which enables building dynamic and incremental processes using a "bottom-up " logic)

To do this, we propose to achieve a production resources re-clustering to find out "logical" groups of resources that match to published services. To this end, we relied on the IS urbanization principles in order to define our production system urbanization approach. Recall that information system urbanization consists of cutting out the IS into modules, between these modules we settle information exchange areas in such a way to allow decoupling different modules, so that they can evolve separately while maintaining their ability to interact with each other and with other ISs, and most importantly in our context, to ensure the ability to build and integrate subsystems from different origins [Sassoon 98].

Our approach is based on a resources-driven organization in order to take into consideration the context while building collaborative process dynamically. This led us to define a matrixial organizations i.e. an organization which takes into consideration the horizontal structure of the production system and not only the vertical (functional) structure of the enterprise.

For this, we identified a set of rules; these rules enable the decomposition and re-composition of the production system in order allow constructing composite services to achieve a final product in response to customers orders, for this reason, the organization of our urbanization approach is based on the production process. The resource group is identified and isolated on the basis of interdependencies analysis; we identify and isolate resource groups that maintain a significant level of interdependency. In this way we can delineate the responsibilities of each unit of production. To provide a sufficient level of visibility, we introduce the concept of business object as a set of resources that are involved in the production of a tangible product. The identified resources groups will provide the basis for collaborative industrial services construction.

Indeed, resource clustering allows the identification of "potential" industrial services, however, in order to exploit these services in a collaborative process (composite service) we need to describe and expose them in such a way to provide a good level of interoperability, to this end, we propose to associate functional properties with each service, in fact, we have chosen to define the concept of industrial service in association with enterprise's final products. So that the logical "modelling" of the services relies on two main aspects, namely, the specification of industrial production (product, material, business rules ...) which represent the functional properties of services, and, associated with non-functional properties (constraints on the Quality of Service, Security or Context.).

To facilitate services selection in an appropriate manner upon the composition process, it is important to accurately define and publish them in a directory. This directory should support the organization of inter-enterprise collaboration; our solution to this issue is based on two key points: the Composite Service Contact and the establishment of a supporting middleware.

The construction of a chain of services is achieved using a selection process which takes into consideration services' functional and non-functional properties. Constraints and preferences issued from functional and non-functional properties are incorporated into (Service Level Agreement or SLA), which allows building a set of contracts defining the global framework of the collaboration. The enactment of selected industrial services relies on a set of underlying middleware technological services charged of mediation and security tasks.

Clearly, at the operational level, the industrial services can not be plugged directly on the middleware. That is why a semantic level on the top of a service bus is added. This is done by the implementation SemEUsE platform architecture; the aim of this project is to elaborate a semantic based service infrastructure that provides the initial services required for business-

focused Service Oriented composition in order to provide high level, secure and guaranteed end-to-end quality-of-services. Another factor promoting the adoption of SemEUsE architecture is that it provides "functional-properties" based service publication and selection mechanisms, SemEUsE architecture is implemented over PEtALS[1] a highly distributed open source ESB.

8.4 Cross-Enterprises Security Integration

The aforementioned urbanisation strategy generates product islands in which production objects can be found. The identification of business objects solidifies the implementation of a dynamic resource-driven process construction based on customer demand. However, the service composition layer acts as a layer of abstraction and hides the details of the underlying implementation, and abstracts the user identity and context from the invoked applications as users and services can access partners services located on different systems across the service chain, and the underlying applications no longer have the necessary information they require to authorize or deny specific actions (requestor identity or context.). These constraints make it a difficult and opaque task to grant to the composite service "initiator" the overall functionality he needed. Therefore, collaborative enterprises using SOA paradigm need a single identity management and security policy infrastructure that governs the access to the different interfaces in a composite service in a way that provides the overall security context for the systems, services and applications. Additionally partners have preferences concerning security, as a result, contextual information and partners' preferences should be identified and propagated through the service chain.

To this end, our architecture is enabled with two features:

- Access control and security requirements and constraints are integrated in the services' descriptions; each service provider expresses regularisation about who is qualified to use the service, what type of identification is required, what is the privacy level that the service can offer. Such that services descriptions are enabled with "security properties" to guaranty access controls.
- A context acquisition and processing capabilities, this allows identification and propagation of resources' and requestors' contexts enabling an efficient context-aware adaptation of services orchestration and composition (late binding), this is achieved by the integration of Attributes Based Access Control (ABAC) and the Role Based

[1] OW2, PEtALS, Open Source ESB: http://petals.objectweb.org/

Access Control (RBAC) to enable expressing the ever changing context of service requestors (partners) including their identity and preferences; depending on requestor's role only to grant or deny access to resources reduces critically the dynamicity of service composition and enactment, necessary for collaboration scenarios as role based access control is initially proposed to adapt to relatively stable organisations structures.

To assure the aforementioned features, the proposed architecture includes "signing" contractual agreements to govern service enactment without forcing participants to hand over the control of their resources, internal policies and internal organization of roles.

To enable contract establishment services descriptions are enabled with semantic security properties to guaranty matching between requestor's security requirements and preferences and those of the service providers. Security requirement are associated to the different services as non-functional properties, which are used to establish a Quality of protection agreement embedded in the SLA. Depending on the data type, the data owner can define security requirements regarding both the way the data is stored, the way access authorizations are given and the way the data is transferred. Role-based access authorization is used in a major way and nominative access control. Security preferences are expressed both by service providers and service customers. These preferences are used to define security requirements attached to functional property. By confronting user and services QoP requirements, convenient services can be selected. A filtering process is then achieved to select services of which the authorization policy allows the requesting user to invoke these services.

We validate our concepts and models by implementing them within PEtALS ESB, an open source and highly distributed ESB, supervised by Dragon a high performance SOA Governance solution[1] (Figure 20).

[1] OW2 Consortium, Dragon SOA Governance Solution: http://dragon.ow2.org

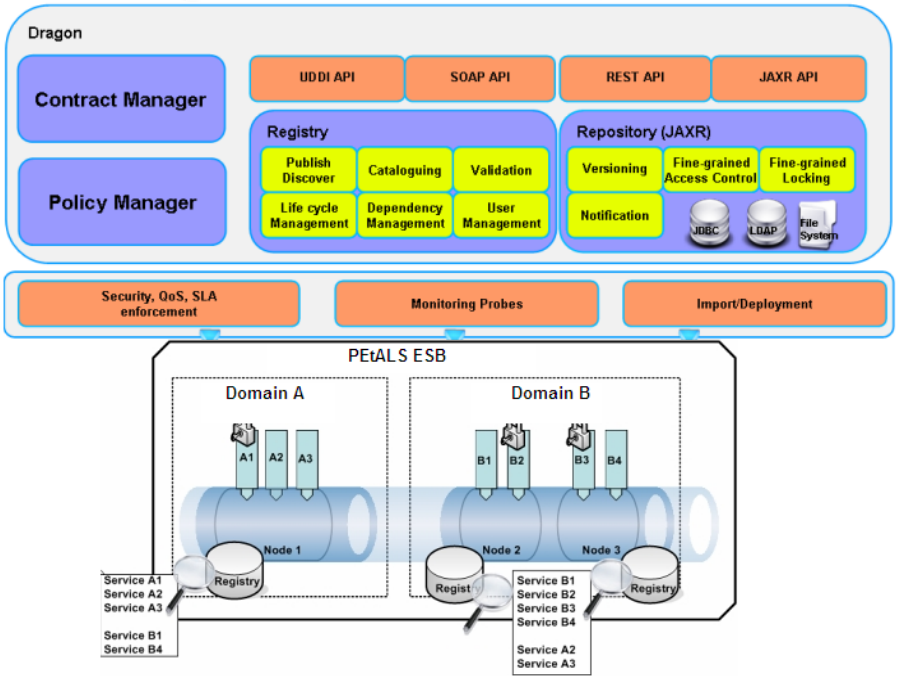

Figure 20 PEtALS ESB Supervised by Dragon

Integrated to SemEUsE1 framework, we added a security component implemented both in the technical and in the semantic layer. The security service intercepts all exchanged messages and processed them at the semantic layer whose the role is to separate the authorization process from the authentication process (in respect to OASIS SOA reference model), as well as to achieve the matching between QoP requirement expressed by the user and the related authorization policy expressed by the service provider. While the technical Layer enforces security constraints, it supports the semantic layer in terms of tools and libraries to encrypt messages establish authentication mechanism and manage identities.

1 - SemEUsE, ANR France, SemEUsE Architecture: www.semeuse.org

Chapter 9

9 New Urbanism Paradigm

9.1 Introduction

One of the principal objectives of the urbanization is to have an evolutionary and adaptive system to face continuous environmental changes. In the production context, an urbanized enterprise is an enterprise which considers the nature of a product and its lifecycle and the degree of predictability of the demand as the principal factors of its organization and the definition of production means. The urbanization of production system (PS) is depicted in a spatial organization or regrouping of small production units (PU) producing similar products which maintain between themselves intense complementary relations, while preserving the capacity to communicate and to orchestrate their actions with the remainder of the system.

In the production urbanization context, the identification of production process inter-dependencies is required in order to have a better understanding of each change's impact on the related process. After identifying the relation between the processes, we can identify the target situation. In the next we will describe our production oriented urbanization strategy.

9.2 Towards Production Oriented Urbanisation Strategy

A production system is made up of multiple interdependent objects. Any change of an object has an impact on related and dependent objects. To obtain good production process management, the resources participating in the various production sectors must be identified to rationalize relations and manage interdependencies. Interdependence represents a constraint for evolution management translated by the capacity to maintain coherence between different production system islands or group of objects after the changes or evolutions. The idea consists of identifying the objects used in production processes and analyzing the correspondences between the various processes to quantify the dependencies between them.

To attain this objective, the dependences between the processes are identified through the appearance of four dependency components: activities, and materials and a production object.

1. Dependencies through activities: by which two products share exactly the same activity. An activity is a work performed by people, equipment, technologies or other facilities.

2. Equipment and machine dependencies occur when two processes use the same machine or equipment.

3. Dependencies by materials, such as material resource sharing, this kind of dependency arises when two products use the same material resource.

4. Complex dependencies or dependencies by a production object stems from the situation in which the existing and potential business objects in an enterprise are identified. The idea is to identify the lower-level production components which could bring about a change in product functionality through changing one or more of its elements. A production object is a logical group of competencies, technologies, material resources and machines which provides a specific added value. Production objects at the lowest description level will achieve one or more sub-goals in the lowest level of the product decomposition diagram. In this type of dependency, two forms can be distinguished:

   - Production object sharing: by which two processes handle one or more identical production objects.

   - Event-driven dependencies occur in the case where a production object serves as a trigger event between two processes.

Our aim is to define a conjoint production and information system urbanisation strategy. This suggests the alignment of production system organisation and processes with the information system urbanisation initiative instead of basing it on a pure informational view only. To achieve this, information system urbanisation rules should first be explored and analyzed while establishing the production system's particularity in areas such as store, competence, and material flow management. A set of rules allowing the alignment of information and production system urbanisation initiatives should then be extracted.

9.2.1 New Urbanization Rules

Production enterprises have complex material flow issues due to several factors including an enterprise's global strategy, the nature of products, urgency in client demand, product diversity and customization, product implication level in the various production sectors and, finally, local constraints related to each production sector such as local charge and machine capacity. Additionally, production processes are very complex and involve well-trained human resource competencies and the use of specialized and sophisticated devices and

equipment for each product category. The production business sector requires an organizational approach which considers these constraints. Such an approach allows the various production processes and their resulted flow and involved objects to be managed and ensures the accurate sharing of human and material resources.

We introduce here a conversion methodology for traditional production systems of the organizational type such as line production, homogeneous workshops, and cellular systems. A new organization is privileged to assure the following objectives:

- Facilitation of the identification and smoothing of each product's value stream.
- Facilitation of the change management in the production system resulted by the introduction of new products or the modification of existing ones.
- Facilitation of the choice of control mechanism for each part of the production system.
- Facilitation of the production of flow leveling.
- Improvement of the redeployment of storage point and stock reduction.
- Improvement of workforce reattribution and training.
- Improvement of the redeployment of production machine and tools.
- Reduction of parts storage between operations.
- Reduction of useless transformation stages.
- Reduction of useless labor movements.

A production system urbanization initiative must be oriented to satisfy the following principles and sub-objectives:

- Production Autonomy supposes that the production system which serves each product should be as independent as possible.
- Regrouping according to product similarity is related to three essential points:
- Identification and sharing consolidation production units (PU whose products are essential for different PUs).
- Identification and sharing of consolidation storage points and materials essential for different PUs).
- Identification and sharing of common tools and resources used by various PUs).
- Complementarities and Interdependencies should be examined to determine the level of dependency between units.

According to [Sassoon 98], a system information urbanization initiative should satisfy the following rules:

1. Affiliation: in the hierarchical decomposing of an information system, each component belongs to one and only one unit.
2. Autonomy: no dependency should exist between units; a unit can deal with an Event from start to end, without the need for treatment in another unit.
3. 3-Asynchronism: a unit can handle an event and send the result without waiting for the response from the receiver.
4. Data Normalization: the results produced by each unit are based on recognized standards.
5. Single Exit Point: the information in each unit is sent by a single point.
6. Data Ownership: each unit is responsible for its own data; updates of these data are carried out by each individual unit.
7. Passage Point: units communicate indirectly via an Entity which is responsible for data flow management.

These rules should be re-exanimate in order to adapt them in such a way to derive new set of urbanization rules allowing a conjoint information and production system urbanization strategy, in other words an urbanization strategy which takes into account the particularity of production enterprise.

Production enterprises serve different types of customers and use various types of distribution and sales channels. This introduces both heterogeneity and complexity into global and local production management strategies. Each production island is forced to serve multiple clients with varying levels of urgency. An immediate need imposes a push flow management strategy; a differed need requires a pull or hydride flow management strategy. This results in an increased overhead and production management difficulties. In order to handle this complexity, we have identified a set of urbanization rules with which production system must comply, this rules allow delaminating responsibility of each production unit, production island autonomy, interoperability between production islands and simplification of flow control and production management.

By analysing the urbanization rules mentioned above we can note that the first four IS urbanization rules are applicable and help to achieve the stated objective of delaminating

responsibility of each production unit as well as simplifying flow control and production management. However, the fifth and sixth rule of a single exit and passage point seem to impose unnecessary efforts and coasts due to material flow characteristics. As each production island may deal with several other islands and several production flows could exist simultaneously, conflicts could be created in the case of a single physical passage and exit points. In addition, no updating operation could be carried out on products which could be modified. Any modification achieved will change the product's nature and may require another competency or additional components. The product would thus be transformed into another one. Therefore, we can conclude that the seventh rule, that of ownership cannot be applied in the production context. Instead, new rules should be defined to regulate product modification and transformation. To satisfy these constraints, a new set of rules can be established:

a) Membership: in the organization of units or islands, an island at the lower level of the groups belongs to only one island in the next level.

b) Self containing: there is no dependency between the islands. A unit can carry out a product from start to end, without the need for treatments carried out in other units.

c) The unit that offers a product is responsible for its quality.

d) A unit can produce a product and send it without waiting for recipient response.

e) Production Normalization: the products carried out by each island comply with recognized standards and norms.

f) Each block has its own unit of flow management.

g) Each block has its own unit of control.

h) Each block has is own unit of resource management.

i) - Each object in the system has a single reference resulting from the nature of this object.

j) - any transformation in the property of a product is entrusted to a unit even if the age or physical location of this object is susceptible to change.

k) - Each operation achieved on a product in a unit is associated with a date.

Principles on which a production system urbanization initiative must be based have been discussed. A clear and simple regrouping of production units that gives the PS the desired

flexibility must now be specified. This grouping strategy must generate product islands in which production elements concerning a certain group of product can be found. Separating shared objects from non-shared objects helps to manage the evolution and change in the production system for two reasons:

1- Non-shared elements help to distinguish the product in question from the other products. Innovation in this particular product is often related to a change in these elements which could be subject to frequent change.

2- Shared elements are subject to complex management policies which come from different production and flow management strategies, whereas non-shared elements or elements concerning only one product are only subject to production and flow management strategies in relation to one product alone. Shared elements must thus be considered separately during the change process

To do this, the production system should be decomposed and then recomposed. In our approach, two types of entities can be distinguished including the Production Island and consolidation point. The production island is a space or workshop in which one or more products are carried out, while in this approach, a consolidation point groups competencies, storing points, tools and machines which serve or are used by multiple products. It only contains elements related to these products.

In the following section, we introduce the way by which Production Islands and consolidation points are identified.

9.2.2 Grouping of Production Resources, Business Objects Identification

The enterprise's resources are exploited in production processes. Human and automated resources are also included alongside material resources including raw materials, products and machines. A clear and simple regrouping of production objects that gives the production system the desired flexibility must be specified. This regrouping should help in the management of enterprise resources; it fails to provide a flexible and efficient mean to define the boundary of management responsibility at the operational level such as the production system. The production process involves multiple and heterogeneous resource classes as well as different flow management schemes (such as push, pull and hybrid). This introduces both heterogeneity and complexity into global and local production management strategies. To this end, a target situation should be sought. For example, storage points and machines can be relocated alongside human competences in the different units. Next, the units must be set out

in such a way as to minimize product transfer time between the different units by grouping units according to production complementation and resource sharing. It is necessary to:

1 - Determine the production process by determining the parts to put in the production system.

2 - Determine the “ideal” position of the units based on the analysis of resemblance between them.

3 - Consider geographical, technological and financial constraints. In the case of production systems, these constraints could strongly impact the urbanization strategy such as in the case of location deficiency, financial shortcoming or geographical distance. A compromise would then need to be established to resolve the conflict between the ideal situation and constraints to define the best feasible situation.

The principle will be to avoid the multiplication and overlapping of units in the company through a classification which makes it possible to extract the analogies between various products and gather the different resources using product similarity.

To satisfy these requirements, this approach consists in taking the final products, analyzing them to isolate the principal factors related to their manufacturing. Three essential factors are identified: competencies, raw materials and tools and machines. The number and nature of the factors that can be identified are strongly related to the production context of each company. The preceding three factors are basic or common factors for all the production companies regardless of their context.

The first stage must thus be to identify the principal products according to those which will be used to urbanize the PS. The decomposition according to the “Causes and Effects Diagram” or “Ishikawa Diagram” (Figure 21) can be used for the decomposition and graphical representation of objectives models by adapting it to the context. For example, this can be suggested by the decomposition and graphical representation of product models starting from production process of end products to the production process of elementary products (products carried out by only one elementary process).

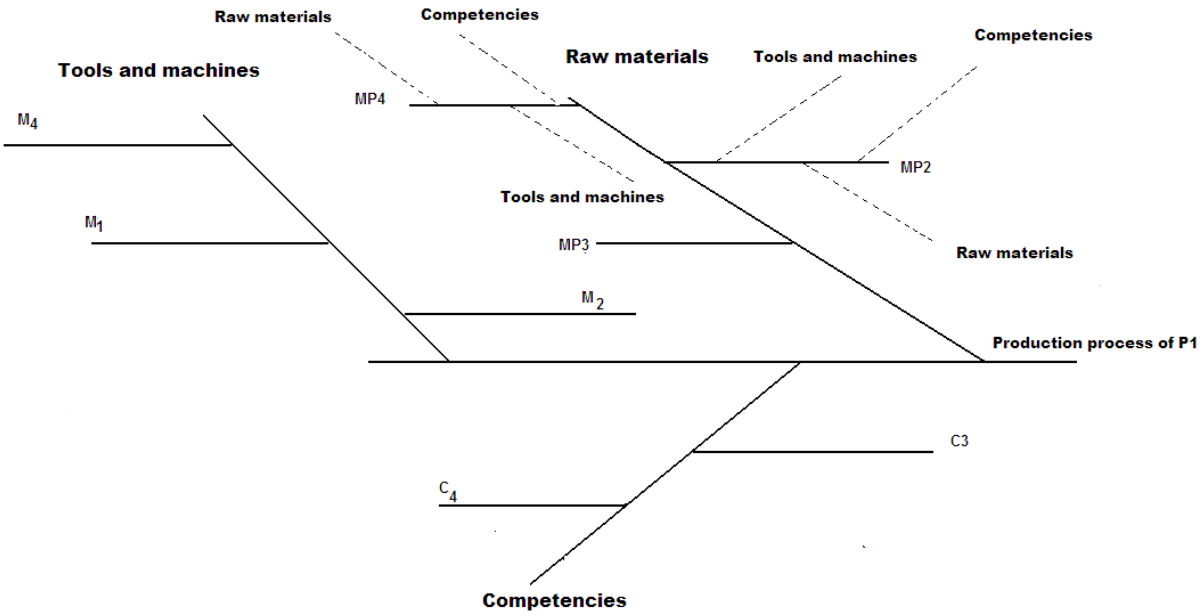

Figure 21 Product Decomposition Using Ishikawa

After carrying out this stage, elementary products must be classified according to selected factors to identify the closest product groups and form units according to a resemblance criterion.

A second stage will gather the units into larger zones according to the complementarities and sharing criteria until covering the entire production system.

In this approach, inspired by the McCormick method [Javel 04], we group the initial data in a matrix whose lines represent products and whose columns represent resources. All types of resources are included: the machine, M, human resources, C, and raw materials implied in the manufacturing processes, MP.

In fact, raw materials are included to specify the distribution of storage points in the various workshops. Tools belong to the same category of production machines and thus have similar natures. For each cell resulting from an intersection between a column and a line, a + is indicated if the object concerned is used to produce the product corresponding to this line, otherwise, a - is found (Table 1).

First Iteration	M1	M3	M2	M4	MP1	MP2	MP3	MP4	C1	C2	C3	C4	C5
P1	+	-	+	+	-	+	+	+	-	-	+	+	-
P2	+	+	+	+	+	+	+	-	-	+	+	+	-
P3	+	-	-	+	+	+	-	-	+	+	-	+	+

Second Iteration	M1	M4	M2	M3	MP2	MP1	MP3	MP4	C4	C2	C3	C1	C5
P1	+	+	+	-	+	-	+	+	+	-	+	-	-
P2	+	+	+	+	+	+	+	-	+	+	+	-	-
P3	+	+	-	-	+	+	-	-	+	+	-	+	+

Third Iteration	M1	M4	C4	MP2	M2	MP3	C3	MP1	C2	C5	M3	MP4	C1
P2	+	+	+	+	+	+	+	+	+	+	+	-	-
P1	+	+	+	+	+	+	+	-	-	-	-	+	-
P3	+	+	+	+	-	-	-	+	+	-	-	-	+

Table 1 Resources Grouping Approach

The second stage seeks to reorganize lines by bringing those which contain the biggest number of + closer to each other while placing the densest lines in the middle of the table. Then, the densest columns are moved forgetting the nature of the objects associated with them, either type M, C or MP.

Groups are formed by juxtaposing the columns having "+"s positioned on identical lines. Lastly, groups of columns that have more share objects are moved. These objects will be subjected to potentially different policies of management and control and for that reason; special attention is paid to them. Some objects use only one product. These objects are then subjected to management and control policies which only depend on this product. The following serves as an illustration (Figure 22).

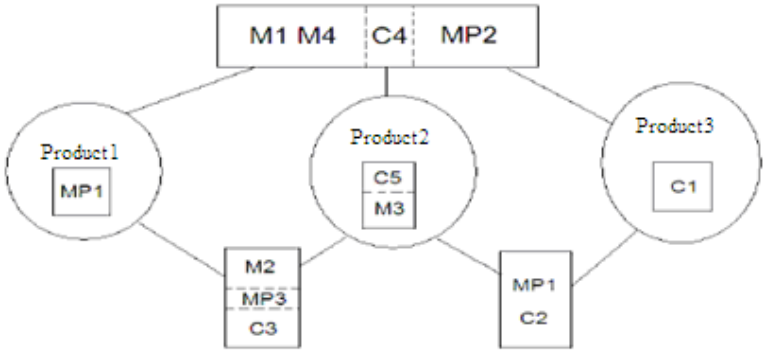

Figure 22 Production Island and Consolidation Point

As a result three production islands are shown (Fig. 22) (Product1, Product 2 and Product 3). Three consolidation points are also noted: (M1-M4-C4-MP2) for all product islands, (M2-MP3-C3) for Product 1 and Product 2, and (M-P1-C2) for Product 2 and Product 3.

Another example can be explored in which a product uses other products as components or sub-assemblies while sharing some production machines and competencies with the same or different product groups. For example, the product P4 uses both P1 and P2 as components while sharing with them MP2 and M1. Additionally, the product P5 uses P1 and P3 as components while sharing with other product MP3 and M1.

Using the same logic to reorganize this production system, the following result can be obtained, (Table 1) and (Figure 23):

	M1	M4	C4	MP2	M2	MP3	C3	MP1	C2	P2	C5	M3	C1	P1	P3	P4	P5
P2	+	+	+	+	+	+	+	+	+	-	+	+	-	-	-	-	-
P1	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-
P3	+	+	+	+	-	-	-	+	+	-	-	-	+	-	-	-	-
P4	+	-	-	+	-	-	-	-	-	+	-	-	-	-	+	-	-
P5	+	-	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-

Table 2 Example of a Product which Uses other products as Components or Sub-assemblies

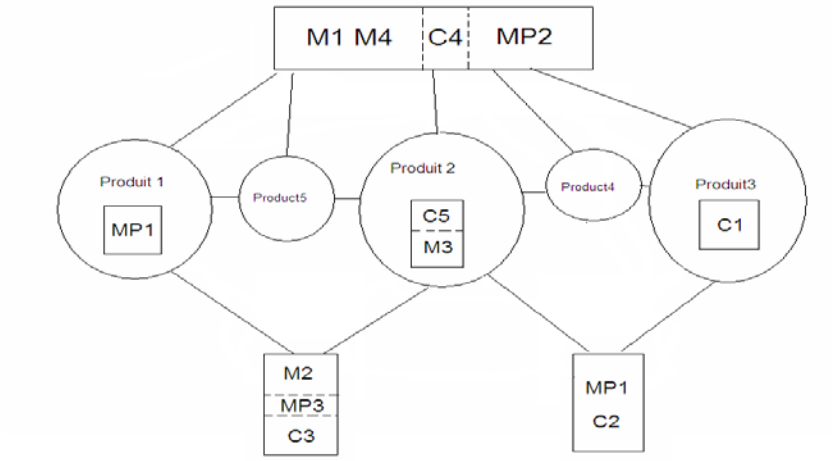

Figure 23 Example of a Product which Uses other Products as Components or Sub-assemblies

The aforementioned grouping strategy generates product islands in which production objects can be found. The identification of business objects solidifies the implementation of conjoint production and information systems urbanization strategy and results in industrial services identification basing on resource grouping, however, there is still a need to define a composition and orchestration logic which is able to assure a dynamic industrial services composition and enactment; in the following we will introduce a dynamic service-based production process orchestration based on customer demand.

9.3 New Orchestration Logic: a Resource Driven Orchestration

Although the introduced paradigm allows the formation of decoupled exposable and exploitable components each of which gathers a group of resources necessary to produce one or more of distinct sub-products, the involvement of the identified “industrial services” in dynamic collaborative processes which is able to produce “on demand” products requires the introduction of a production oriented orchestration logic which, beyond the informational aspects of the system- takes its material and human aspects into consideration.

A collaborative production enterprise must be modeled in such a way that takes into account, beyond the components of the enterprise itself, all interested actors and interactions with their environment.

This should be divided into the modeling of “why,” “what” and “how.” In other words, a global modeling of production systems requires the coherent integration of three views which are objective, static and dynamic views.

The production covers three elements in interaction (Figure 24): The customer expresses a need for a product and specifies its functionality. The product is produced using enterprise

processes and resources. The need is an objective or goal to be reached; it is the "why." It must be formulated in terms of requirements and not in terms of solutions. A Product is what will be provided to a customer to satisfy his or her needs by product functionality. A product is defined as a process output, the "what." A product can be tangible or intangible, such as a service. To be designed, a complex product is initially described and broken down into components through the Product Breakdown Structure (PBS). The Process comprises the "how" and consists of all finalized operations. A process can be divided into sub-processes, each of which is a sequence of inter-dependent activities linked by relations of temporal and/or objective precedence. An activity denotes the description of a work to be achieved and requires organizational specifications: who does what and when and, sometimes, on which objects?

The objective is to elaborate a functional proposition starting from client requirements based on two points:

- A specific preliminary demand which responds to client requirements.
- An evaluation of feasibility and profitability.

At this stage, requirements must be formalized by identifying a group of parameters which indicate the different values of this demand. In other words, client requirements should be explicitly followed by functional specifications.

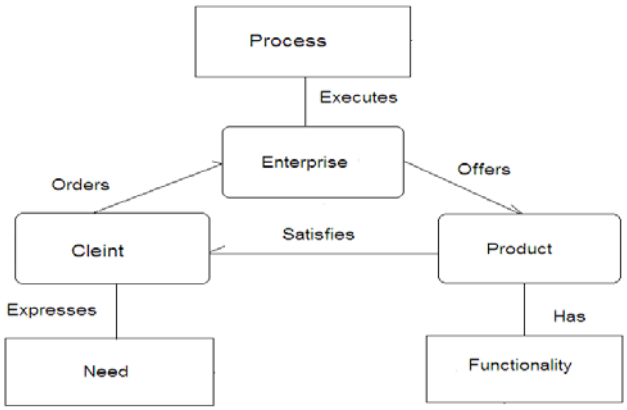

Figure 24 Production Elements and their relations

The second phase seeks to represent the final product through an objective view. In this phase, the final goal must be modelled by a hierarchy of represented objectives using the ISHIKAWA diagram, by breaking objectives into sub-objectives until reaching a level in which all of the objectives are realized by at least one operation. An appropriate limit of sub-

goal granularity will be reached when it is no longer possible to define desired sub-goals, and the only details remaining are actions. In this stage, the following rules must be respected:

- The same goal appears only once in the diagram.

- If an objective is divided into sub-goals whose list must be exhaustive, all sub-goals must cover the totality of the main goal's scope.

- A lower-level sub-goal should be able to be associated with one or more primitive production objects.

The third phase includes a search within the enterprise for the component with the appropriate functionality; this step should satisfy the following rules

- Minimizing the number of components to minimize interfacing.
- If a similar component with a similar but not identical functionality exists, an attempt should be made to adapt it to the new product while keeping the old one.
- Define the desired value of each component and find the most adapted available components for these values.

In this phase, a top-down analysis of the preliminary demand and a bottom-up construction of the functional representation of the final product are used. A top-down analysis often requires a decomposing phase of the desired result of goal to sub goals according to chosen axes related to the core process' nature.

The final phase elaborates a technical proposition on the basis of product functionality. It explains the technical specifications of product components. A technical proposition indicates appropriate components and operation sequences which contribute to the production of final products as well as their quality control. The list of instructions and component characteristics is exhaustive and includes the amount of time needed for their completion.

This new vision requires a rethinking of the definition of business processes to assure a bottom-up construction of business process construction and orchestration.

9.3.1 Industrial Service Oriented Process and Organization Model

According to [Crowston 98], the business process is defined as "sequences of goal-oriented actions undertaken by work units or business firms that repeat over time and which are measured in performance terms, such as time, resources or costs." The ISO 9000:2000 norm

gives a different definition of the business process whereby it is “a set of correlated and interactive activities which transforms the input elements into output elements.

These elements are material and/or informational objects.” Lastly the research carried out by [Harrison 93] defines the business process as: “any activity or group of activities that takes an input, adds value to it, and provides an output to an internal or external customer.” It is noted that these definitions consider the process according to a task-driven logic which is purely an algorithmic and rigid logic by which the process model consists of chains of explicit activities.

However, in the abovementioned continually changing context, the business process should be constructed dynamically using a generic process model and description so that it can be adapted to this non-static context. This leads to define the task and basic process element, using a resource-driven logic rather than a task-driven logic. With such a vision, the task is considered as a resource state and evolutions are managed by task activation conditions to achieve pre-defined goals which result from a change in the resource state and allow the activation of other tasks. The task post-condition denotes the state of the resource impacted by this task. Taking this definition into consideration, the process is considered the objective evolution of the resources’ state.

9.3.2 Incremental Process Organisation

In this context in which the organizational process is defined in an “ad-hoc” manner, the process should be redefined in terms of the goals to be achieved rather than what method should be used to achieve them. This reinforces the need for a resource-driven logic rather than a task-driven one. With this in mind, the following questions must be answered:

- What are the process objectives?
- How should the tasks be identified to form a complete process? (Taking into account the coordination between the identified tasks).
- Which task resource allocation will ensure that the process will act as desired to achieve the objective?
- At what point should the task be monitored?
- To respond to these questions, the notion of resource should be examined before taking interest in the generic evolution mechanism of the task.

9.3.2.1 Resources

The resources of a company participate in process execution. Human and automated resources are included in this notion alongside material resources including raw materials, products and machines.

The option to organize resources into groups each of which have similar static and dynamic specifications can facilitate resource management (Figure 25):

- Material Resources are decomposed into reusable resources that are consumable with supply guaranty or production resources and non-reusable resources that can be shared. For example, oil is a consumable resource whereas a machine is a production resource.
- Non-Material Resources are decomposed into two sub-groups including both passive resources such as data or required time for execution and active resources that are able to effect the environment such as actors or e-services.

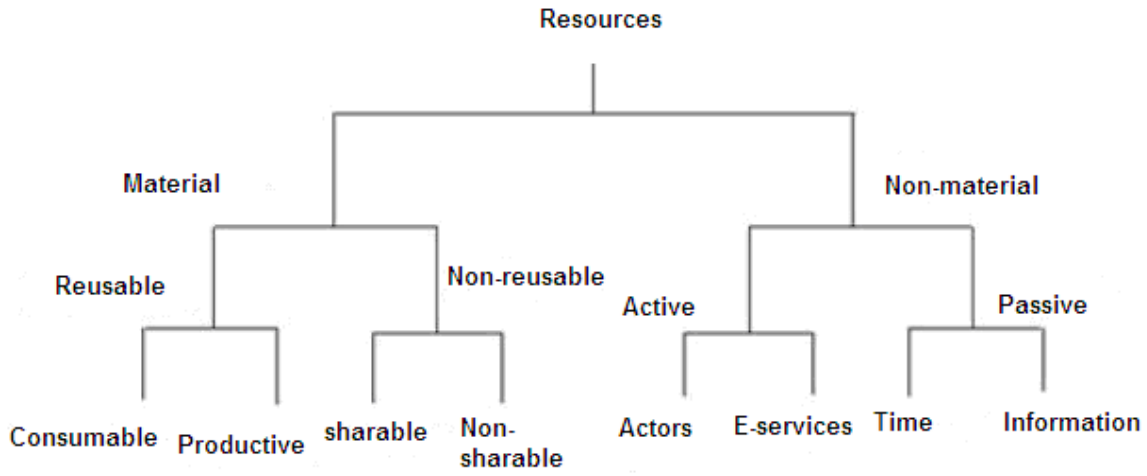

Figure 25 Topology of Resources

Thus, a resource entity reflects the static specification of the objects represented by this type as well as the dynamic attributes that represent the interface. The functionality and behavior provided by this object can be controlled and parameterized. The static, semantic and dynamic description of the object can be included in an XML document through an ontology-oriented approach.

9.3.2.2 Concepts

The achievement of this objective requires the definition of new concepts to relate the

organisational, operational and objective company views. Concepts to establish such relations must now be introduced.

- **Task:**

Autonomy constraints imply that a resource must be assigned and controlled or managed by the task which carries out the operation according to given objectives. Autonomy, however, does not mean isolation: each task should coordinate its objective and actions with other tasks to achieve the common objective and satisfy its own goals. These requirements require a definition of the task and related concepts. The task in this vision represents a generic concept. It could be assisted by a process or decomposed to sub-tasks. Here, no difference between task, sub-task and primitive task is made. A task is associated to a transition which controls the objective evolution of the resource.

- **Goal:**

In contrast with tasks, goals allow for synchronization. The goal describes the desired effects and objectives. It could be decomposed into sub goals which in turn could be decomposed into primitive goals. These goals, sub-goals and primitive goals are related to the task, sub-task and primitive task, respectively. The notion of measurable goals related to the execution state of a primitive task is introduced. This allows the process state and its behaviour to be monitored according to the task execution plan.

- **Plan:**

The task plan manages the resource behaviour interacting within and alongside other tasks. This plan is a document containing the generic structure of the task. It contains an abstract description of the required resource, the objectives to be achieved and events to be generated during the task execution. This is assured by including:

- The initial state; abstract description of the required resources and a final goal as well as a triggering event. These elements are regrouped in the precondition of the task.
- The in progress state is identified by measurable goals such as produced effects.
- The final state or post condition describes the expected state of the resource after achieving the task.

9.3.2.3 Relating the Organizational View with the Operational View

Human and automated resources are included in the larger category of company resources. These resources have a special importance because they give the enterprise its functional and operational capacity. They are, in essence, the resource through which the enterprise executes its processes and implements other resources.

Notions which allow the announced goals to be achieved must be introduced in order to define the process in terms of stated objectives with an abstract description of each resource's functionality. To do this, the capabilities and aptitudes of each active resource are noted whether it be a human or automated resource or a production machine. The idea is to exhaustively describe each machine's competency or application in terms of functionality. By such, we describe what each resource can do in term of activities. This takes into account an "action -verb- adjective-resource" grammatical schema. An activity can be a sequence of activities or can encapsulate other activities, a fact that recalls the distinction between complex and primitive activities and the notion of operation. An abstract description of each resource's functionality facilitates the search for and matching of suitable resources to construct the process dynamically and on demand.

By such, we propose using the notion of role inspired by the RBAC model used in the domain of access control in computer networks [Ferraiolo 95]. Hereafter we introduce an approach to combine an enterprise's organizational structure and business processes. Such model is service-oriented entity which includes the following components as illustrated in (Figure 26):

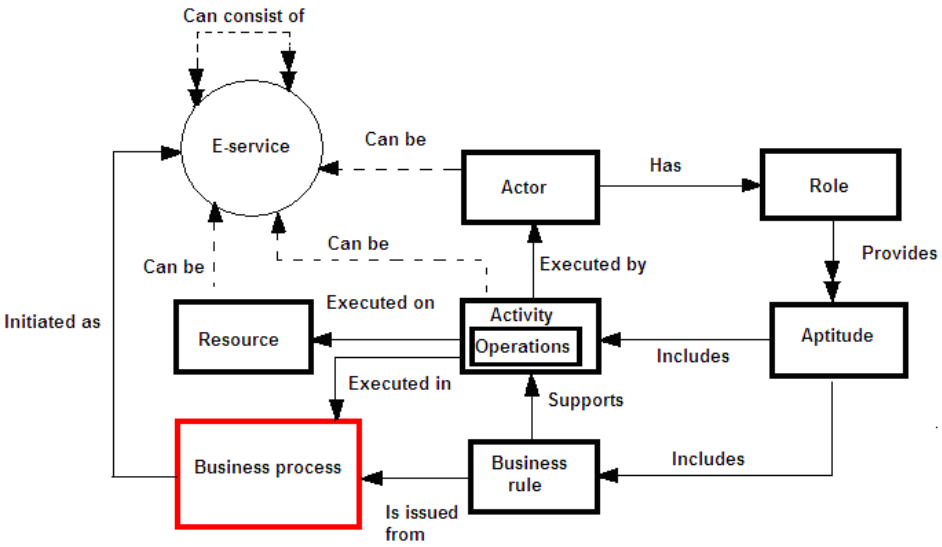

Figure 26 The Role and its Related Concepts

- Actor: designates a user or computational service. The actor initiates a request for a resource which could be tangible or intangible products or a third party service.
- Role: defines the engagement, permissions and aptitudes assigned to the actor.
- Aptitude: defines activities that can be associated with a particular role in terms of activities business rules and operations.
- Activity: identifies a functional feature. For example, Processing a client order..
- Business rules: define constraints on how to perform out activities.
- Operation: denotes independently designed atomic behaviour. For instance, we can define the activity of admission in the following operations: input patient insurance number, input patient's mate name, check for available rooms, schedule a room, and send the application to the rooms' service section.
- Process: consists of activities sharing a common objective.
- Session: integrates actors, roles and aptitudes and considered as set of interactions between users and resources.
- Event**:** an 'occurrence' which can prompt a change in the role state.
- E-service: services consist of a set of computational and atomic operations which provide benefits. Services are used in various contexts and have interrelations and dependencies; a service can be made of the aggregations of other services. Services interact via a well defined message-based interface.

9.3.2.4 Relating the Operational View with the Objective View

To relate these two views, a task generic evolution mechanism is proposed (Figure 27). A triggering event stemming from another process or external entity sparks the resource, satisfying the precondition. This initiates the associated treatment and progresses the resource state towards the final state or post-condition. A task can produce, consume, modify or borrow resources. Along with the execution of the task, the intermediate or measurable goal is produced. This facilitates monitoring of the process's overall progress. An ideal execution could be confused by an external event which might deactivate, hold or release the task.

This new, resource driven, process orchestration and enactment requires the ability of recognizing contexts and collecting their information to derive event-triggered actions and make context-responsive changes (Event Driven Architecture).

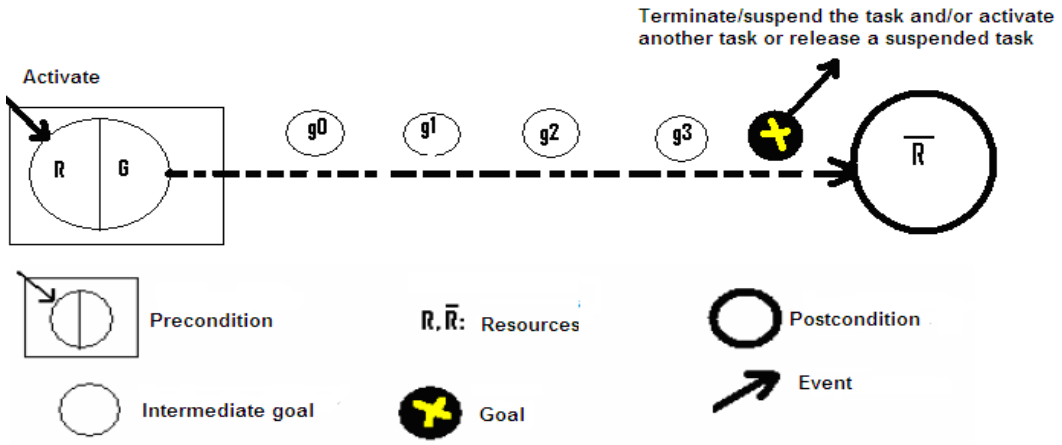

Figure 27 Task Generic Evolution Mechanism

The heterogeneous nature of service-based B2B paradigms and the complexity of representing a consistent and globally agreed-upon context model complicate this task. Each party will likely have its own perception and preferences about the context and how contextual changes should impact the process achievement. Additionally the exposed "industrial services" should be able to adapt to diverse execution contexts i.e. each service should be designed and described to be executable in diverse scenarios issued of the involved partners' contexts and preferences. To tackle this problem, we introduce a framework which is able to deal with context and partners preferences.

Clearly, depending on the concepts we have defied previously, mainly the dynamic -resource driven- orchestration, assigning and granting access to resources should consider the current state of the resources, enterprise's context and this of the involved partners. In fact the changes in the context represent the events which could result in triggering process execution. Here, the notion of context broadly refers to any information related to actors or resources that could impact the decision of assigning resources, as well as granting the access to these resources. Contextual information includes, for example, current resources state, resources availability, actors' identity, location, access time, current activities and preferences. As contextual information changes continuously, dynamic resources assignment and access control mechanisms and dynamic policy enforcement become essential, features for the dynamic orchestration. Policy here refers broadly to rules governing services behavior according to business needs, partner's requirements and agreements. In the following section we introduce our context-aware role-based resource assignment and access control model and provide mechanism to collect contextual information. The model is based on the concepts defined in our service oriented model.

9.3.3 Context Integration

As our model allows to set a collaborative organization via services composition, this requires the integration in heterogeneous services operating in various contexts and environments and having different requirements and constraints. A framework to support distributed and context aware services enactment, focusing on the integration of involved parties preferences, is needed.

In addition to catering to specifications of business policies and partners' preferences, the framework should be able to collect and generate context -based adaptation measures in cross-organizational scenarios, in order to provide a just-in-time process activation.

Service and actor context should be designed to facilitate its description, transmission, and processing in different situations. In our model a context is formally defined as follows:

- Definition1- A context, denoted by Conti, is an array of n values which represent a non-empty set of sub-contexts Conti= {c1, . . . , cn} where cj is represented by an array of m sub-contexts or states of different natures { s1,…,sm}. Each state represents a value of one element of the context such as time, location, identity, preferences. We propose to define a hierarchy of context of different abstraction levels. Each context in the lower level becomes a state in the upper level; the preliminary states are the lowest level of context where only one element of the context elements has a value: all the other elements are null. This representation of context has many advantages. The context hierarchy is easy to manipulate by machines and understand by users. It can be implemented as nested arrays. It also facilitates the matching of contextual information between heterogeneous applications. As context representations are application-dependent, each context instant may have different representations in different applications. Our representation allows the identification of the relations between heterogeneous representations of context instants, like inclusion relation, or intersection relation. It allows the identification of mutual context states in the lower level (i.e. preliminary states). It helps also in reasoning about inferring and matching of context related policies.

  Contextual location information, for example, are represented in the lower level of abstraction by three dimensions coordination (x, y, z). According to application needs the coordination metrics provide semantically useful data at the upper level of the context "hierarchy". In the case of the university information system, the

coordination metrics identify building number, floor number and room number. The identification of the intersection, inclusion or exclusion between two places represented differently could be difficult or even impossible, while it is easily identifiable in the lower level (i.e. preliminary states level).

Thus, in our model, contexts can be specified at different levels of granularity. The finest granularity of context is a state. In this perspective, we define three functions to treat each context element:

- Definition2- the *Context Recognizer* abstracts raw data from the context. These *recognizers* are essentially wrappers around underlying local state recognition service (i.e. sensors, RFID readers…etc.); they provide interfaces that deliver information to Context Interpreters.
- Definition3- Context Interpreters are responsible for abstracting low-level context (states) into a higher-level context. An interpreter can convert state information to another format or meaning. For example, an interpreter can convert a resource location into a country name.
- Definition4- the Context Collector collects information for relevant entities; an entity could be an actor or a resource. It holds information about an entity's environment and about the current activity or state of this entity. The context collection process is carried out by calling the respective services i.e. Context recognizers and Context Interpreters. Essentially, we have at least two class of Context collector i.e. Actors Context Collectors and Resource Context Collectors.

9.3.3.1 Context Reasoner

The Context Reasoner provides decision regarding the assignment of actor to a role depending on an activity within a particular session and with respect to a given context. The Context Reasoner is described in (Figure 28) and comprises the following functions.

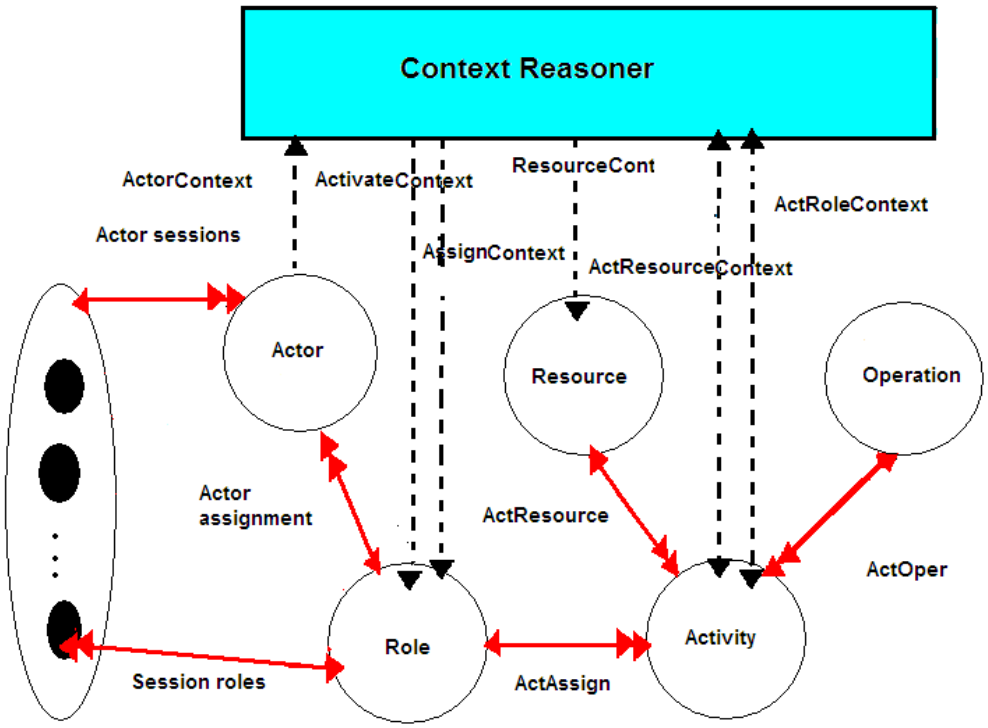

Figure 28 Context Reasoner and its Relation with our Model Constructs

9.3.3.2 Context Reasoner –Actor and Resource

By extending the location notion used in [Ray 06] taking into consideration our previous definitions. Our formalization of this relationship includes defining services that provide the context to which a user or resource belongs.

- Definition5- ActorContext (u) returns the context of an actor u. User context may include the identity of the organisation to which the user belongs. More generally, it may describe the organizational hierarchy under which the user exists.
- Definition6- ResourceContext (resource) captures context-relevant information about the target resource.

9.3.3.3 Context Reasoner -Roles

The assignment of user to a particular role is context-dependent. In our model, each role is associated with a set of possible contexts. Some roles can be activated only if the user is working in a specific context.

- Definition7- AssignContext(r) returns the set of contexts in which a given role r can be assigned. In the case that the assignment of some roles is not context-dependent. The AssignContext(r) returns all available contexts. Which means that

the user u can be at any context assigned the given role. The context of a user u must be contained within AssignContext(r) to be assigned the given role.

- Definition8- ActivateContext(r) returns the set of contexts in which a given role r is allowed to be activated. For roles with context-independent activation ActivateContext(r) equals allcontext. Before activating a role r for a user u, we must check whether the context of u is contained in ActivateContext(r).

9.3.3.4 Context Reasoner –Activities

Context-based access control should also make it possible to model real-world requirements where activity enactment is contingent upon the context. For example, an SP may restrict some activities execution to some resources in a particular situation, for example, data base backup in high network traffic to some customer groups. In our model, activities (aptitudes) are associated with roles, resources and operations.

- Definition9- ActAssign(act) returns the set of roles assigned to a given activity.
- Definition10- ActOper(act) returns the set of operations associated with a given activity
- Definition11- ActResource(act) returns the set of resources on which a given activity act can be performed. A user with a role r can perform an operation on a resource, if there is an activity act that assigned to the role r to perform the operation on the resource. To incorporate context aware access we associate an activity with two contexts: permissible role context and permissible resource context.
- Definition12- ActRoleContext(act,r) returns the set of permissible contexts for the role r associated with a given activity act.
- Definition13- ActResourceContext(act,resource) returns the set of permissible contexts for a given activity act to be performed on a given resource.

The defined concepts represent the main constructs of our framework; however a mechanism orchestrating their functionalities is needed in order to reach the overall functionality of the architecture i.e. dynamic service-based business process construction and orchestration.

9.3.4 Orchestration Mechanism Integrating the Described Concepts

Harmonizing the different concepts solidifies implementation of the announced goal which

might include a dynamic resource-driven process construction based on customer demand which integrates functional and non-functional service properties.

For this, two entities are introduced: the resource manager and the process initiator. The former has the role of organizing and exposing combinations of available resources including actors, service and materials which assure the demanded functionality and provide them to a client in the case of an agreement. Its role can be detailed through the following activities:

- Getting the descriptions of the managed objects and indexing these descriptions - Monitoring the managed objects in order to get their states
- Deducing decisions about the managed objects or assignment from the gathered information and resource management policy.
- Taking all the necessary actions to build a combination of resources to meet the demand or request.
- Redefining the generic behaviour of managed resources or roles.

The process initiator takes charge of negotiation with the other entities in order to get the necessary resources to execute the process while generating events related to the measurable goals mentioned in the task plan.

As noted from the precedent description of the two entities, it is recalled that they act as service provider in the SOA. Together, for example, they assure a part of the orchestration task.

In the resource allocation process, a triggering internal or external event will activate a process according to the following steps:

a. The process initiator sends a formalized request to the resource manager in a preliminary contract proposal describing the process in terms of resource functionality and terms and goals to be achieved.

b. The resource manager extracts the description of the demanded resources from the preliminary contract and then determines the combinations of resources which most closely match the request from among the managed resources and their attributes. In this case, the resource manager should have an inference motor to construct the most adequate available combinations. The most suitable technique in this case will be the backward changing inference motor starting from the last element of the process graph. For example, building the

minimum spanning tree of which the root is related to final goal to be achieved, the final product (backward changing inference motor starting from the final product) (Figure 29).

c. If the process initiator receives a negative response from the resource manager, it will try to externalise one or more parts of the process to be achieved in another enterprise. In this case, the process initiator must enrich the preliminary contract by the intermediate, measurable goals in order to integrate the events to be generated by the externalized parts and fit and synchronise the processes executed in the different enterprises.

d. The process initiator chooses the most adequate proposition according to its own criteria and then sends its choices. The resource manager allocates the chosen resources to the process initiator and declares these resources as activated.

e. The process initiator starts executing the process while generating the events related to the intermediate goals mentioned in the task plan.

f. The resource manager records the modification of the released resources upon achieving the task and removes the consumed resources.

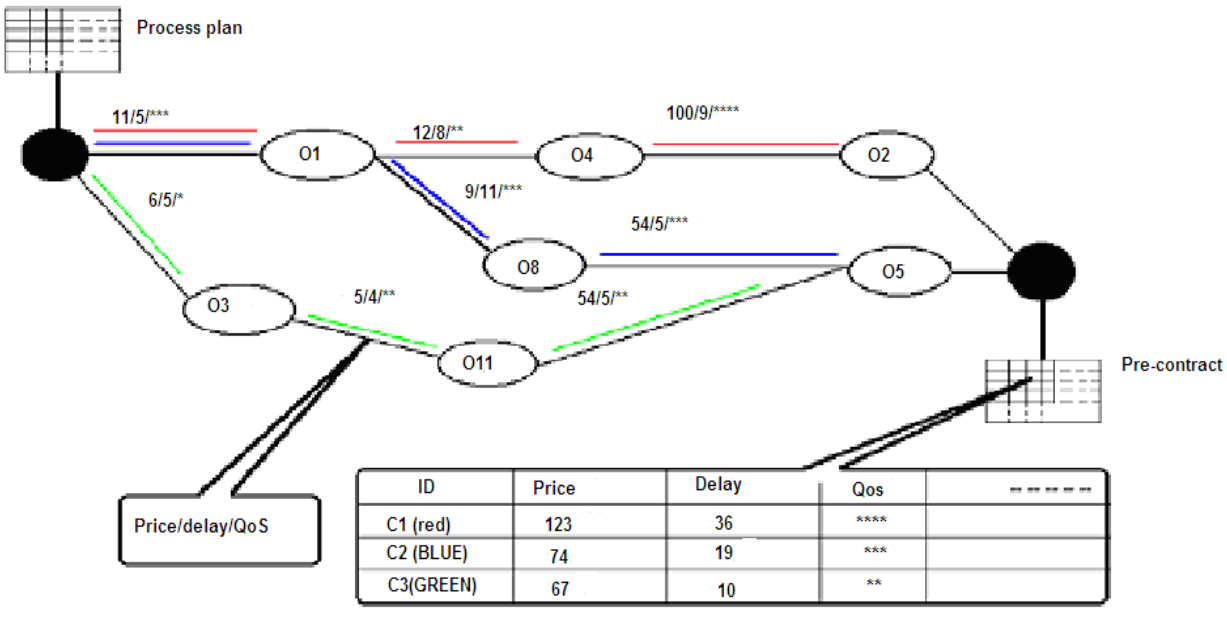

Figure 29 Contract Proposition Process

As mentioned in the above description of the process initiator and resource manager, both act as the service Orchestrator in the SOA. Together, they assure the orchestration task.

In order to allow process initiators to assume their tasks in collaborative scenarios where a collaborative process can nest many processes belonging to different partners, a new notion permitting a dynamic and composite process enactment should be defined. For this, we introduce the notion of composite role.

9.3.5 Collaborative Composite Business Services

The Composite Role (Figure 30) is an abstraction that encapsulates a collection of roles collaborating in order to accomplish a specific task or goal. It is assigned to the process initiator. The effective activities of an actor are derived from roles that the actor belongs to. The composite role is an aggregation of roles of different actors involved in the composite process. The composite process is initiated using a generic process model or template. The activities assigned to the composite role results in the union of those assigned role-based activities of all of its composing processes. A composite role structure is defined by the Role Graph which consists of the sub-roles of the elementary roles which represent the graph's nodes and the precondition-event combinations which represent the arcs. The transitions between two (or more) nodes are triggered when the related precondition-event combination is satisfied. The same precondition-event combination could result in the activation of many sub-roles which make it possible to represent the business process workflow. The Role Graph relies on concepts stemming from the infinite state machine, and the notion of inheritance, introduced by object-oriented paradigms.

The Composite Session consists in various simple sessions. During a composite session, roles are changed to match the process stage phase as well as the changing context. When a composite-process initiator wants to achieve an objective, it passes the request along with the corresponding graph. A principal session is then initiated with a start-up state; the session is assigned to a role containing all activities needed which is a combination of all of the activities associated to sub-roles indicated in the principal role graph. This process continues down to the finest level of role granularity in which roles are primitive and without a role graph. These roles correspond to primitive process initiators which carry out their associated activities in a simple session; the activities assigned to the roles are also activated, and the primitive process initiators enforce needed operations.

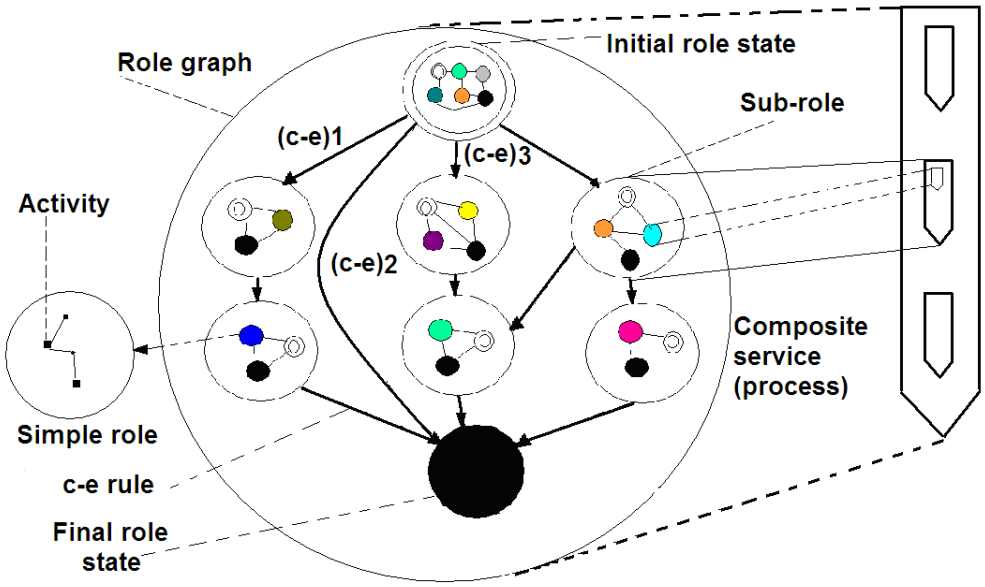

Figure 30 Composite Role

9.4 Conclusion

Establishing collaborative process starts by selecting partners, to enable such selection enterprises must define and publish, in an objective manner, what they can offer to potential partners, thus, the first stage of organising a manufacturing enterprise according to an "SOA" point of view is the identification of independent business process components, where each component is associated to a logical group of people, technologies, and resources which provide a specific added value, these identified "industrial services" could be offered to partners by describing and publishing them in particular directories. However, organising dynamic inter-enterprise service chains involves respecting the constraints imposed by the different "service offering" enterprises. Additionally composition process must offer consistent collaborative services, in other words, in addition to functional objectives; the composition process should satisfy the non-functional constraints imposed by the different partners, including customers, such as organisational and semantic constraints. In addition, partners' integration requires sharing and collaboration between parties traditionally discouraged even explicitly prohibited sharing their ideas. Consequently, partners usually are hostile to open their systems and they want to maintain their autonomy and privacy. Thus any collaborative scenario must provide a sufficient level of "securitisation" to all involved partners. Here, security goes beyond technical aspects, to cover partners' business model. Collaboration must take into consideration these aspects of "business" confidentiality and make distinction between private and shareable information and process regarding the implementation of activities within partners' business processes while maintaining well-

controlled levels of visibilities and coordination necessary for the achievement of the collaborative process.

Overcoming these challenges is much more critical than the technical problems, and thus requires a particular consideration at the time of development of a collaborative information system.

Chapter 10

10 Security for Dynamic Service Selection, Composition and Enactment

10.1 Introduction

In SOA paradigm, the selection of services and the interaction of the service requestors with service providers are governed by services interface descriptions. Discovery and selection of an appropriate service implementation is delegated to the orchestrator which matches the service description and the requestor's requirements. In addition to functional requirements, a service consumer specifies non-functional requirements and preferences. Thus, in addition to functional description, the service provider should annotate his or her services descriptions with non-functional offers and requirements. In the other side, Service consumers should be able to specify his/her non-functional requirements and preferences and apply them against service non-functional offers in order to select a convenient service or set of services to be composed, in this case automation of the access control is a key element for collaborative processes management since we must manage the sequence of services calls and actions specifications according to the specified functional model. To this end each participant should express regularisation about who is qualified to use the service, what type of identification is required, what is the privacy level that the service can offer. Such that services descriptions are enabled with security properties to guaranty access controls. Since the autonomy of participating organizations must be maintained; participation in a collaborative service should include “signing” contractual agreements in which partners adhere certain terms of Quality of Service (QoS) and accept responsibilities without being forced to hand over the control of their resources, internal policies and internal organization of roles, this is usually assured in term of Service Level Agreements (SLA). The problem here is that services belonging to different providers typically do not employ common policy semantics, to enable semantic matching for services security policies and consumers requirements, both during the selection and SLA initiation phases, security policies descriptions and requestor requirements should share a common vocabularies specified in security ontologies, and hence Orchestration and SLA should be enriched with semantics capabilities to carry out matching and to express the agreed Quality of Protection terms respectively.

In the following we will discuss these issues; first we will present our solution.

10.2 Semantic Annotation & Quality of Protection Enabled Services Selection

Typically, the service requestor expresses his or her needs and requirements by defining goals to be reached using different notations such as policies, natural languages or business values whereas security-related procedures and countermeasures of the service provider can be expressed using security policies. Providers thus should express aspects related to service security by annotating services description with their security policies and choices.
Both security policy descriptions and requestor requirements should share common terms specified in security ontologies to enable semantic matching for security policies and goals. Security ontologies should be defined in terms of two elements: the service security ontology, requestor security goal ontology, security policy ontology.

Security preferences are expressed both by service providers and service requester in order to precise their rights and duties. Providers require rights:

- Define which customer can access to information repositories (i.e. role-based access control)
- Define authentication mechanisms and access policy.
- Define the protection level required while exchanging confidential information.

On the other hands, providers' duties can be expressed as follow:

- First, providers must integrate legal requirements while defining their security policies. These legal constraints are defined according to service / information types and access rights may be applied to some requestor role and context (for example, medical staff members have restricted access to medical data according to their roles and contexts).
- Second, when retrieving user information required during the authentication process or necessary for the business logic enactment (i.e. credit card number for financial transaction) providers must protect user information during the exchange to avoid interceptions, and securely storing them in order to avoid unauthorised access.

Service consumers have rights to express:

- Preferences to describe their profiles (i.e. personal information use)
- Trust levels they hold regarding their providers

On the other hand service consumers' duties can be defined as

- First they must provide “authenticated” information (i.e. certification authorities are not commonly used to warranty these information)
- Second, they have to properly use services and information provided by their providers with respect to the agreement. Unauthorised behaviours can be registered and used as tracking elements to identify potential attacks.

These preferences are used to define security requirements attached to functional property or to operations embedded in a service. In order to formalise the different requirements, we need an ontology, to this end we propose an ontology, it is structured according to the different protection requirements:

- Transport security: this part is related to the communication infrastructure. Different requirements can be set
    - Either a secured infrastructure is implemented thanks to network or host mechanisms. This secured infrastructure is set before acceding to the service architecture.
    - Or secured communication protocols are used thanks to net security protocols. These protocols are set while communicating messages between nodes
- Message security: this part is divided into two parts:
    - In order to manage access control, we propose to define there authentication and authorisation properties.
    - In order to avoid message repudiation and to control the message identity, digital signature is applied on the message content
- Data security: this part is devoted to the protection of the stored data. This may lead to store encrypted data and to the addition of digital signature on the stored data.
- Environmental security: this last part is mostly devoted to the “process” organisation. It consists in defining access control (i.e. on credential management). This part is related to the credential sub-ontology.
- Authorisation policy: this part refers to access control. As services have to be composed into different service chain, different authorisation policies can be set, from the simplest one to the more constrained one:
    - Access control based to the calling service

- Access control based on the human user who invokes the service chain. This access control can be expressed thanks user's role and attributes including his or her identity and context.
- Access control based on hybrid invocation (calling service and human user).
- Access control based on the human user who invokes the service chain and on the upstream invoked services.

As set out in our ontology, we define three classes of QoP requirements:

- Concepts used to describe the infrastructure: these components deal with the certified hardware and software infrastructure components provided by the distributed infrastructure. They include:
  - Transport related components as private net, VPN or opened Internet
  - Data security components: as the integration of Firewalls, safe host, DB encryption…
- Concepts related to the security strategy: they are used to describe the data security strategy as the description of personal data storage delay and motivation
- Concepts used to define the security concepts that should be added to fulfil the security level required by the QoP agreement. These components include:
  - Transport-related concepts: these components are used to define the transport protocol that should be used between ESB nodes
  - Message-related concepts: these components are used to define the message integrity strategy (namely signed or not)
  - Access control-oriented concepts: these components are related to the SSO organisation. They are split into two parts
    - Authentication management concepts: this part includes the required authentication information (identity based, IP based, role based)
    - Authorisation strategy concepts: this is used to define the part of the service chain to be considered in the authorisation process.

For simplicity and as a first step we define Security requirements associated to 5 basic security objectives:

- The confidentiality objective is related to both message communication and to access control,
- The integrity objective is mostly associated to the message communication
- The non-repudiation objective: this last service is hardly connected to the “technical” QoS parameters.
- The anonymity objective.
- The openness objective.

Security concepts and objectives are integrated in the registry (user and service registries) in order to implement Quality of Protection agreement.

To this end we have defined a security ontology in which we can find two sub-ontologies, Security Concepts Ontology (Figure 31) and Security Objectives Ontology (Figure 32). These two sub-ontologies are related by means of relations, in this way each security concept can satisfy one or more security objective (Figure 33).

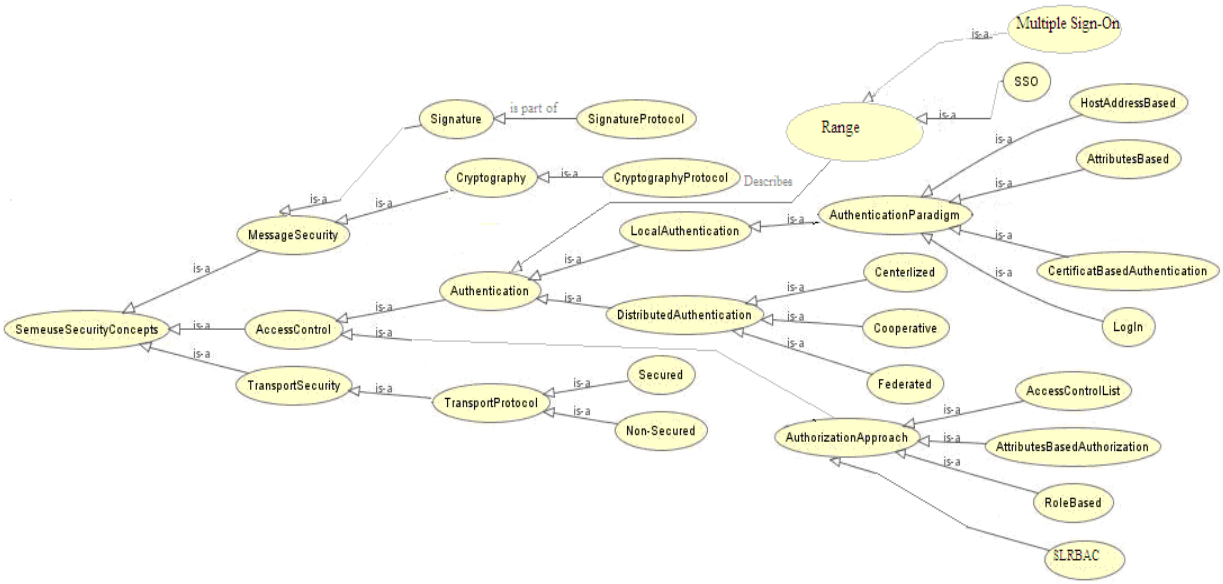

Figure 31 Security Concepts Ontology

On the provide side service security capabilities are described using security concepts ontology while consumers requirements are expressed using security objectives sub-ontology. Services’ policy to consumer needs matching and verification process should satisfy the requirements of composite services. In the following we explain how the process of security policy discovery could be established:

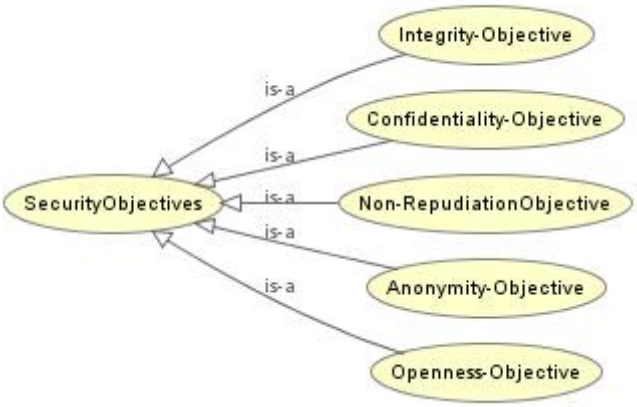

Figure 32 Security Objectives Ontology

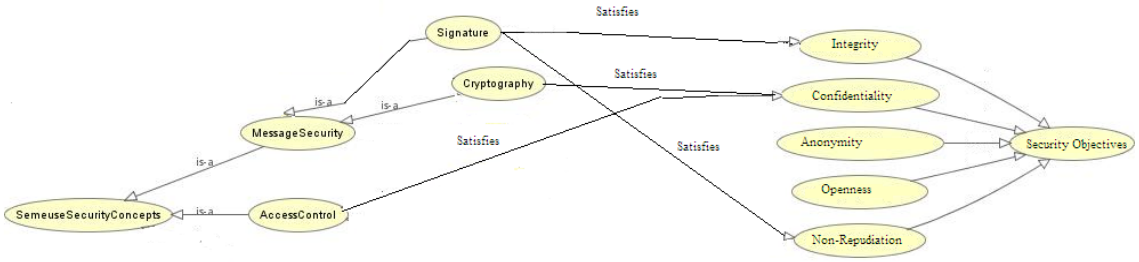

Figure 33 Example of the Relations between the two Sub-ontologies

Firstly, the requestor uses the security objectives ontology in his or her request query. After selecting the set of services which could satisfy functional specifications, the discovery process consists in comparing the requestor with the security policy available in the semantic registry. The goal ontology makes it possible to search services which have security policies similar to requestor requirements. If the requestor's requirements do not correspond exactly to any available services a negotiation phase should be established with the services which have the most similar security policy, this phase uses policy refinement to meet the requestor's requirement. Finally and after choosing the appropriate services, the agreed QoP-related terms are semantically transformed using the security ontology and included in an agreement using WS-Agreement specifications. In our framework the user preferences are represented as context elements, according to the requestor nature, these preferences could be expressed in two forms:

- If the requestor is a service stemming from other systems (or sub-systems), preferences are expressed as security policies which are stored in meta-data files, the meta-data files are referenced in the service description file. For instance, a security

policy can be expressed and stored in WS-SecurityPolicy files which can be referenced in the WSDL documents (Web Service Description Language) using WS-MetadataExchange.

- If the requestor is a human actor, the user application should offer user interfaces providing an exhaustive set of the security objectives expressed in natural languages. The user can express his or her preferences by choosing a sub-set of these options. The chosen preferences can be then matched to the convenient security policies by the underling middleware e.g. ESB, using its semantic capabilities.

10.3 Modelling Contextual User Preferences

According to OASIS reference architecture, users are associated with a set of personal data, consequently, they have to define their own security strategy regarding these personal data. Depending on the context, a human user can play different roles and can have several identities. According to the context, a user should be able to define different security strategies on his or her personal data.

Human end-user information can be stored in a particular registry that can be partly built form a corporate registry. The data model we propose is described in the figure below (Figure 34). The user is defined thanks to his or her main identity (used as a root point) as a subject in a user registry. Each user can play different roles and each role is used as a key to allow or refuse access control to a set of personal data. Each role can also be related to a secondary identity and to contextual information describing the motivation (leisure, professional…), the time period associated to the context (if needed) and the trusted communities that can be invoked in this context.

According to his or her objective and context, a user or a service can play different roles, use several identities and define different QoP sets of requirements according to role, identity and/or context. In our case, services include a set of operations and functional properties that can be either produced or consumed by the service. These functional properties are taken either from the provider legacy data or from the consumer personal data.

Depending on the data type, the data owner can define security requirements regarding the way the data is stored, and the way authorizations are given and transferred. Role-based access authorization is used in a major way and nominative access control. Security preferences are expressed both by service providers and service customers. These preferences are used to define security requirements attached to functional property.

As services are autonomous units that can be composed in different contexts, security requirements are described for both functional properties and operation contained in a service.

The global context is defined by trust communities. A “default” community is used to gather all the unidentified partners. Trust relationships between customers and providers define the current trust level so preferences patterns can be deduced. Trusted communities are defined as groups of actors, for instance an. end-user, a service, or even a chain of services invoked by a user.

The community is defined and owned by an actor who gathers a set of other actors that can be associated to similar security requirements. Trust policy describes trust relationships between actors and defines the way communities can be extended.

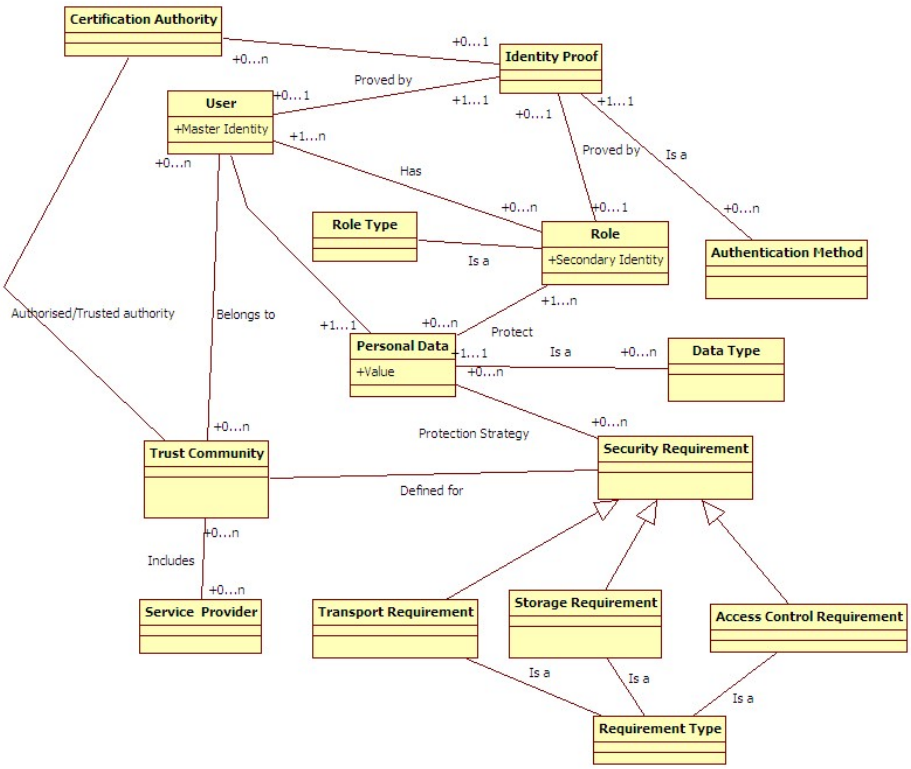

Figure 34 End-user Description

10.4 Modelling Services Security Requirements

Services are seen as a set of operations and functional properties. These functional properties either belong to the service provider or are taken from a user environment.

Security preferences are expressed both by service providers and service customers. These preferences are used to define security requirements attached to functional property. As

services are autonomous units that can be composed in different contexts, security requirements are described for both functional properties and operation contained in a service. The global context is defined by trust communities. A “default” community is used to gather all the unidentified partners.

Services include a set of operations and functional properties that can be either produced or consumed by the service. These functional properties are taken either from the provider’s legacy data or from the consumer’s personal data. Depending on the data type, the data owner can define security requirements regarding both the way the data is stored, the way access authorisations are given and the way the data is transferred. Role-based access authorisation is used in a major way and nominative access control can also be implemented for very restricted access.

Trust relationships between customers and providers are used to define the current trust level so that preferences patterns can be deduced (Figure 35).

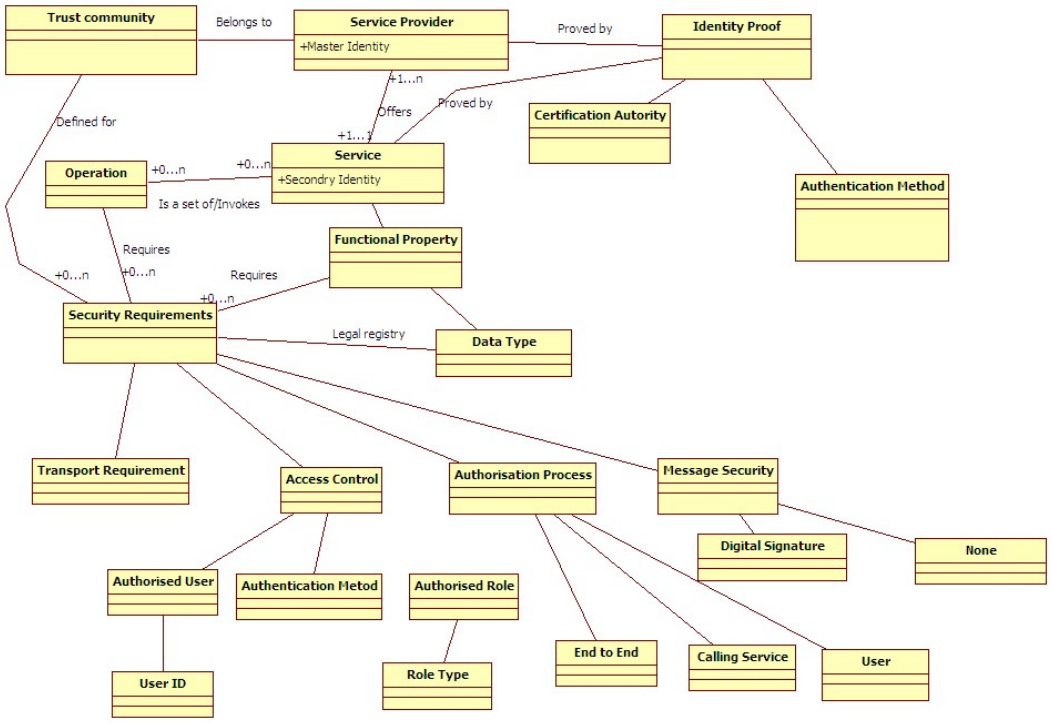

Figure 35 Security preference class diagram

10.5 Using Annotation to Set Security Preferences

While setting opened organisations as a service-based one, the major point consists in organising a common strategy, taking into account the requirements of each entity. Consequently, modelling approaches should supply semantic elements to describe security

components, so that “security tags” can be associated to information to identify their security level (black/grey/ white), to define secured exchange channels or authorisation requirements to take into account the different security constraints while specifying a service. For this purpose, we propose to consider different communities and to identify security sensitivity for both data and operation included in a service.

The method we propose consists in 3 steps based on the basic security services:

First, the confidentiality service aims to protect data both in the information system and in the communication system. Organising data confidentiality involves defining access rights depending on the user authentication and using cryptographic services to protect data while they are stored and exchanged. This leads to white, grey and black data organisation.

Then, the integrity service must protect the system against any attack aiming at modifying data. This involves using electronic signature to authenticate both the user and the content as well as using antivirus or intrusion detection systems to protect the information from malicious modifications.

At last, the availability service has to maintain the information system service’s operationality and availability. This service is related to the QoS parameters and also involves infrastructure-oriented components such as firewalls or intrusion detection systems, whereas replicated architectures can answer to high-availability constraints.

Building a common security strategy involves orchestrating conveniently the different services required by the various entities. To reach this goal, we propose some rules related to the information “value” (i.e. white, grey or black) to combine the security components:

- **Rule #1:** « white » data will not need extra checks, « grey » data will be checked according to access authorisations (identification and authentication) whereas « black » data is related to checking processes to verify that they fit access rights, confidentiality and integrity policies. These services involve identification and non-repudiation functions, processes activities will be logged and sensitive information will be encrypted.
- **Rule #2:** exchanges between elements belonging to a sensitive category (i.e. grey or black) will be secured (i.e. these exchanges will have to be authorized, to remain confidential, and to keep their integrity).
- **Rule #3:** we will consider that enterprise internal processes are black data by default.

This global policy is then super-imposed to the service model by defining new symbols (see (Figure 36), (Figure 37) and (Figure 38) show examples of security annotation for use case “Firefighter” see appendix). First, the different security levels (white / grey / black) are used to tag the different objects (data and processes), this allows verifying the global consistency of

the security requirements (for example, a process can not be “white” while processing black data…). Proceses having high QoP garanty are considered as black, while processes having low level of QoP garanty are considered as wihte.

Figure (38) shows a use case in which processes and data are annotated according to the aforementioned annotation; for clarity we will symbolize white objects (being data or processes) by”+”, while grey objects by “-“ and black objects by “#” . From this figure we can note for instance that “*Collect Media Info.*” Process has a low level QoP (white) thus it can only process the non-sensitive data (white) while attribution process has a high QoP and thus it can process the high sensitive data (black).

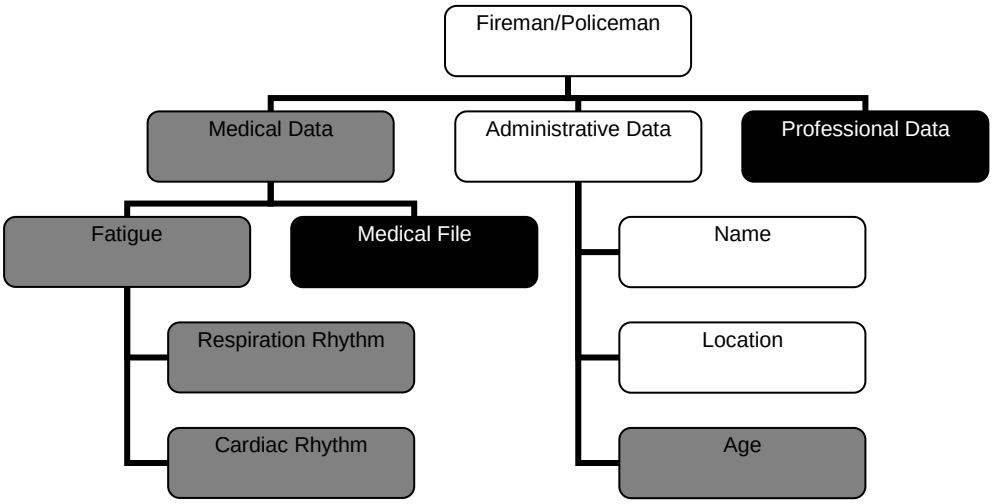

Figure 36 Data Annotation for Fireman and Policeman Personal Data

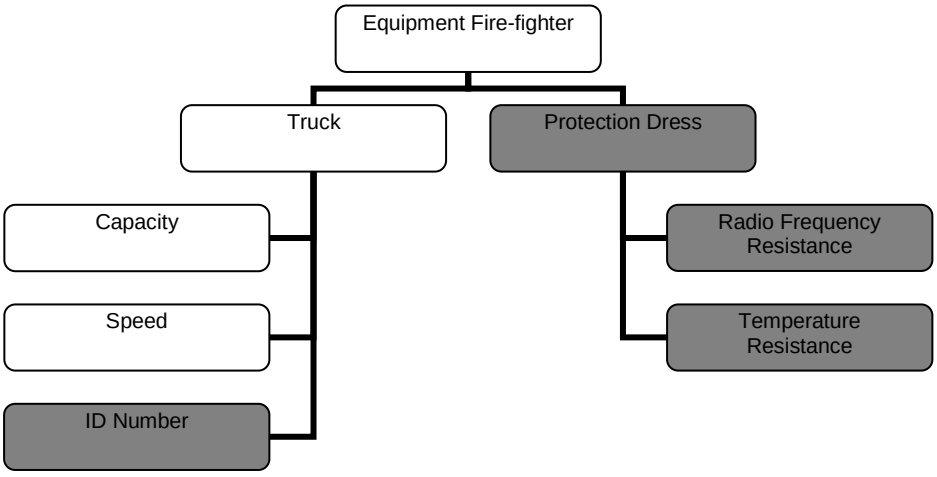

Figure 37 Data Annotation for Fire-fighter Data

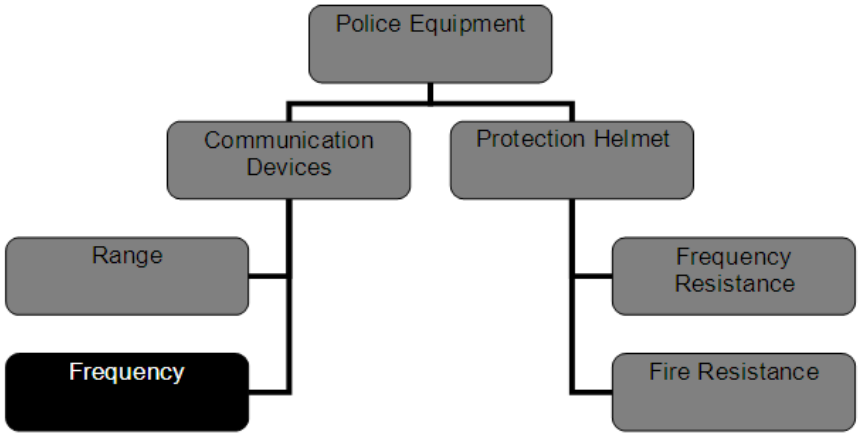

Figure 38 Data Annotation for Policeman Equipments

In our context, collaborative businesses implement their process based on service-centric architectures. Although services security standards, building on SOAP, WSDL, and UDDI, provide some guidance for the integration of security into collaborative process, they remain very close to the technical level and unintelligible to the business level. In c-business security requirements are governed in regulations of public law, business rules and agreements the implementation of security requirements has to guarantee a high level of correctness. This means that security aspects must not be viewed as isolated aspects at the implementation level, but must be integrated into all phases of the development process.

Domain experts must be able to think of security when modelling their own organization's internal workflows as well as those crossing the domain boundaries.

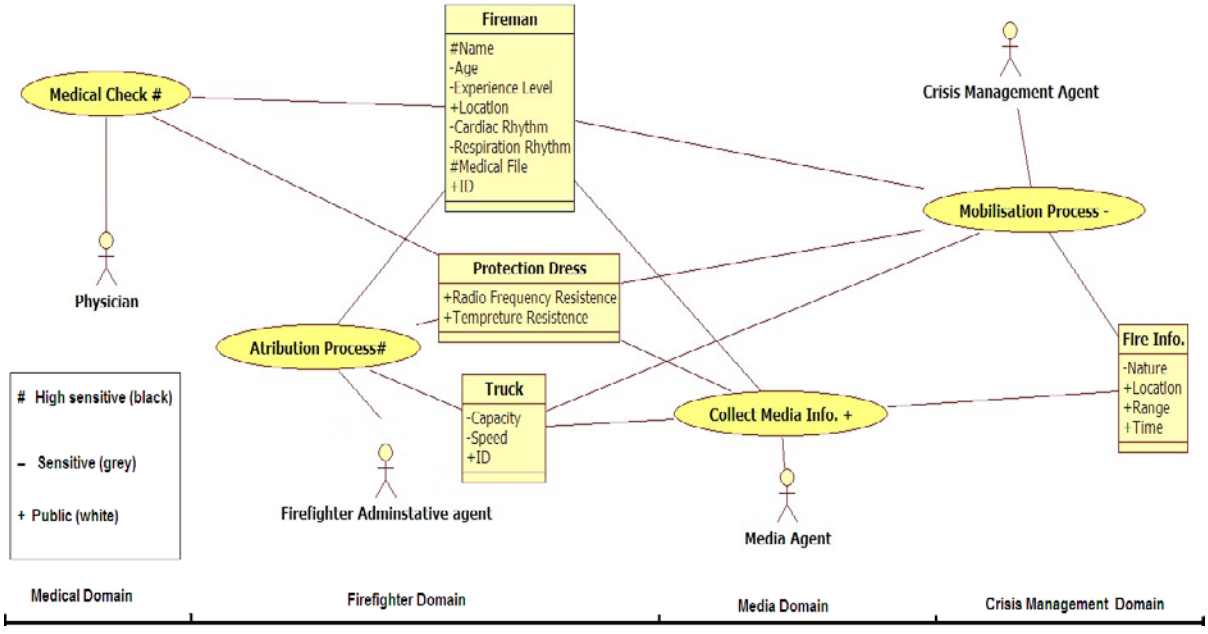

Figure 39 Example Process and data Annotation

These requirements create a strong need for methodologies and tools that support the design and management of secure inter-organizational workflows at a high level of abstraction.
Based on the context modelling approach, we introduce the security policy model focusing on ad-hoc business process and organizational workflow definition and considering the execution context in terms of users' roles activities and customer preferences. The security policy distinguishes between internal and external users of the system. (Figure 39) illustrates a simplified view of the architecture integrating different elements such as services Security Policy Description, Context Collector service, Context Reasoner, Policies Interpreter, Policy Conflict Resolution and Policy Enforcement Point.

10.6 Services Security Policy Description

In order to design collaborative services, security policies should be designed in terms of values such as goals and requirements. The policy model may be further refined regarding the context and technology-dependent artefacts [Zhou 06]. The policy model might include multi-layer abstraction elements that can be mapped to technical implementations to generate policies that conform to a particular platform. Designing security policies with declarative statements and publishing them in repository consistently maintain them available to enterprise applications and beyond the enterprise boundary.

The question will be how to express security requirements as generic policies used by various enforcement points. The problem consists of designing generic security based on process requirements and organisational constraints and then transforming them to security policies integrated in process model and business policies.

To get benefits of declarative security, two conditions should be satisfied; firstly, the language for expressing policies should be based on standard technologies. This helps in creating runtime platforms that can read the security policy for a service and enforce it. Secondly, the protocol to dispatch security policy should also be based on standard technology. To combine security policies and business policies in order to provide coherent and integrated security enforcement we propose the following guidelines:

1. Identify business requirements and constraints from the enterprise's organisation and processes models and business rules.
2. Identify business and security rules from business requirements and constraints using one of the methods indicated in the state of the art.

3. Transform each of these rules into condition and triggering event clause.
4. Classify derived rules into four categories:

    - Functional business clauses.
    - Functional security clauses.
    - Exceptions and their exception handling clauses.
    - Exceptions without exception handling clauses: an accepted risk due to some constraint.

5. Analysis of event combinations to identify overlapping rules and eliminate potential conflicts.
6. Classify each category of rules into two sub-classes:

    - Soft Rules can be modified to satisfy customer preferences.
    - Hard Rules cannot be modified.

10.6.1 Policies Interpreter

The policies interpreter component indentifies which services are suitable for a customer by interpreting the corresponding customer requirements. The interpretation process considers information regarding the user's current context and the customer's identity. This service uses ontology to get a common comprehension of service and customer policies. It transfers of customer policy descriptions to the Policy Conflict resolution component (Figure 40).

10.6.2 Policy Conflict Resolution

Conflicts arise when the customer activates his or her own policy whereas each service has its own policy. The policy conflict resolution component handles conflicts according to the conflicting policy type. For example, a soft rule can be modified. If, in the case of a hard rule a solution has not been found, a negotiation phase could be established.

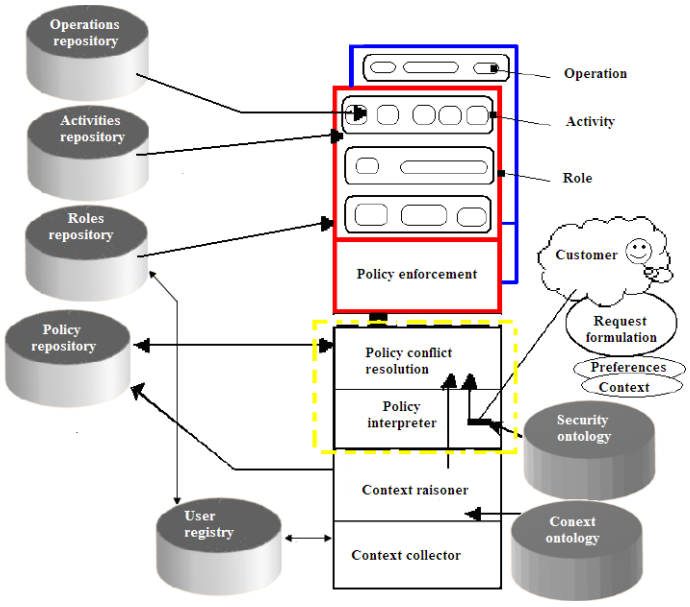

Figure 40 Security Policy and Context Integration Components

10.7 Conclusion

In the previous chapters we presented a conceptual architecture of context-aware security policy SOA architecture thought to govern the execution of collaborative industrial services which have been identified by a new, production-oriented urbanism strategy. Our modelling constructs make it possible to represent an adaptable organisation and business processes conjointly with the access control. The proposed architecture is based on a context-aware role-based access control model which integrates resources descriptions into service interface, human and active resources are integrated via their roles and attributes. The approach allows taking into consideration service consumer context and security preferences making it possible for the ad-hoc business process, workflow definitions and execution to fit the new B2B and B2C service-based paradigm. The notions of composite role, role graph and composite session are thought to be adaptable to our model to fit the composition service constraints. Collaborative ability is insured by many details, including ontology engineering to support mutual description of fundamental security elements for clients and services; however, at the enactment time, other problems appear. In fact, managing services calls, in a chain of services, is constrained by local authentication and authorization decisions; requestor's identity an credentials should be verified each time it tries to access a new service, the problem is that a requestor could have many identities related to different services providers and domains, and authorized actions granted on each service may vary according to access policy at service provider side. The challenge remains how to establish a chain of service in a dynamic manner where sets of fine grained distributed authentication and

authorization policies defining who is authorized, what mechanism of authentication is used to verify identity and what end users are allowed to access/use on local resources can be supported across various independent domains. One common solution is the establishment of federations, whereby a federation is group of organizations which agree to adopt common policies and technical standards to provide a common infrastructure for managing cross-domains access to resources and services in a uniform way [Robiette 05]. In the following section we present our architectural elaboration for detailed specifications for ad-hoc semantic security based composition. Finally, we present an architectural improvement for federated SSO authentication and authorization mechanisms allowing binding rights to requestor's identity across multiple domains.

Chapter 11

11 Service Security Architecture Implementation

11.1 Introduction

Business owners and end users must be able to fine-tune the security according to their preferences. The current solutions to express these preferences and requirements have a technical low-level tendency and are not user friendly. The ability to express High-level semantic security preferences and requirements is essential. Inversely, the ability to compose services within and across the enterprises boundaries to offer new and more coarse-grained business functionalities is one of the most important features of SOA. This involves a number of business partners each of them has his own security requirements expressed and enforced with diverse vocabularies and manners which augments the complexity of the security management at the platform level.

Additionally, the platform supporting the services enactment should be able to dynamically re-configure security to be able to respond to changes in the user and execution context.

Thus a framework to secure services architecture needs to be simple, flexible, and semantically rich to facilitate service discovery, composition and enactment and should offer the ability to express services requestors and provider security preferences and requirements.

One approach to meet the security requirements and complexity sketched above is to extend the ESB architecture with an extra layer that hides the complexity and heterogeneity and enables customized and consistence security definition and enforcement. We call such ESB semantic and context-aware security policy-driven ESB.

Enterprise Service Bus (ESB) is our architecture's engine. An ESB provides the features with which SOA may be implemented. Based on recognized standards, it provides fundamental services for higher level architectures via an event-driven and standards-based messaging engine. In our implementation we build our architecture within and over PEtALS components and nodes. PEtALS is a highly distributed, open source Service Bus. The PEtALS project is led by eBM Websourcing and developed in collaboration with OW2 consortium.

Our security architecture is a part of SemEUsE project architecture (Figure 41), the aim of which is to elaborate a semantic based service infrastructure that provides the initial services required for business-focused Service Oriented applications in order to provide high level,

secure and guaranteed end-to-end quality-of-services for all actors; from service providers to end users.

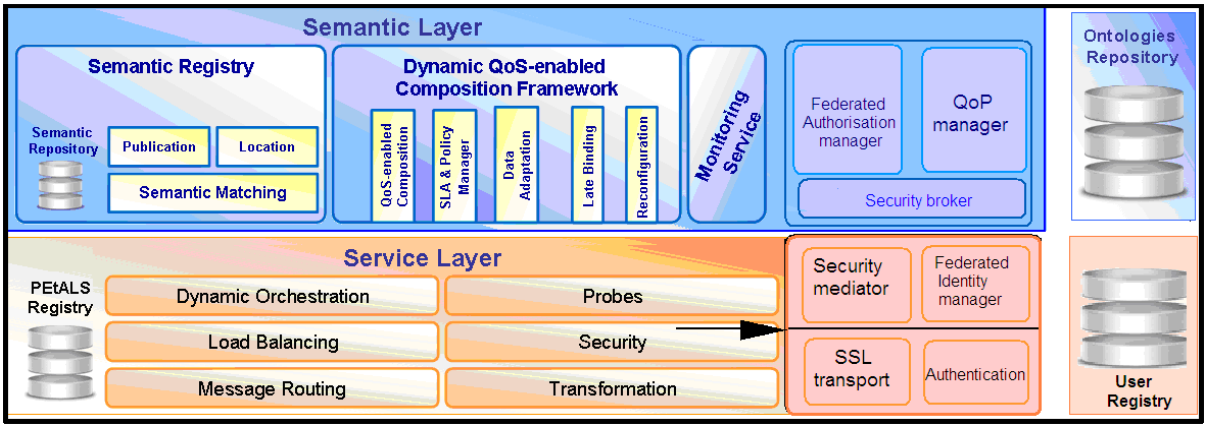

Figure 41 SemEUsE Security Architecture

11.2 Quality of Protection Filtering of Services

Services are applications wrapped by well defined interfaces and can be invoked using standard mechanisms and protocols. Services are exposed in special directories, for instance UDDI directories. In these directories services are described in terms of their capabilities by describing their authorized operations along with the input and output messages for each operation. Such definitions are impeded in WSDL documents. However, a WSDL document includes the functional aspects of a service without addressing its non-functional aspects, namely QoS and security characteristics.

To express service level security preference we need to join a security policy to the service (WSDL) document. Policy references could be made via *policyreference* element as defined in WS-SecurityPolicy specifications or could be embedded in the WSDL document using policy expression (*wsdl:definitions*). Nevertheless, as we saw so far in the state of the art, WS-SecurityPolicy falls short in providing details for a full implementation of security. To fill this gap we propose to extend WSDL by the description of QoP characteristics. The extension is carried out using our semantic security annotation exposed in (10.2). The WSDL document therefore becomes QoP-enabled WSDL. The obtained QoP-WSDL can be further used to filter services according to the level of protection they offer and to extend service level agreements (SLA) by QoP-oriented characteristics. Providers then publish QoP-enabled SLA templates along with WSDL document of their services. The service client searches and finds the appropriate service, retrieving the QoP enabled WSDL and SLA for the service. These will be further used to formalise and finalize the SLA, for instance by a negotiation phase.

(Table 3) shows the non-functional characteristics offered by services (taken from Use case , see appendix) augmented by QoP constraints:

- Services S1, and S5: the offered contracts exhibit equivalent or stronger QoS as well as QoP constraints than the one required at the process level (according to the sensitivity of the information transmitted). These services are not filtered.
- S2 in the other hand has non-secure transport protocol as well as a weak authentication mechanism and thus weaker QoP constraints than those required (according to the sensitivity of the information transmitted.) This service will be filtered.
- Service S3: although this service exhibits a suitable quality of protection level, the offered contract exhibits a weaker physical strain constraint than the one required at the process level. Consequently, this service will be filtered.

S1: non-functional contract	**S2: non-functional contract**
60°C <= Fire resistance <= 90°C Physical strain <= 70 PPM[1] 2Ghz < Radio frequency <= 2.5 Ghz **Authentication: Kerberos Protocol** **Transport: HTTPS**	60°C <= Fire resistance <= 90°C 20 <= physical strain <= 70 PPM 2Ghz < Radio frequency <= 3Ghz **Authentication: Login** **Transport: HTTP non-secured**
S3: non-functional contract	**S5: non-functional contract**
60°C <= Fire resistance <= 90°C Physical strain <= 100 PPM 2Ghz < Radio frequency <= 2.5 Ghz **Authentication: Kerberos Protocol** **Transport: HTTPS**	60°C <= Fire resistance <= 90°C physical strain <= 70 PPM 2Ghz < Radio frequency <= 2.5 Ghz **Authentication: SAML** **Transport: HTTPS**

Table 3 Non-functional Characteristics Offered by Services, Taken from Use case

For instance the document describing the security policy of S5 could include the following lines (Figure 42):

Lines 1 to 12 assert binding to be transport layer security and HTTPS-based point-to-point secured channel. While lines 14 to 19 specify that we need a SAML 1. 0 token as defined in the SAML Token Profile 1.0 for WS-Security.

As far as the QoP matching part is concerned, we'll use the QML based matching process between the User QoP part of the WS-Agreement and the provider QoP part of the WS-Agreement.

```
1.  <wsp:Policy xmlns:wsp=".../policy">
2.  <sp:TransportBinding
3.  xmlns:sp=".../securitypolicy">
4.  <wsp:Policy>
5.  <sp:TransportToken>
6.  <wsp:Policy>
7.  <sp:HttpsToken/>
8.  </wsp:Policy>
9.  </sp:TransportToken>
10. </wsp:Policy>
11. </sp:TransportBinding>
12. </wsp:Policy>
13. ...
14. <sp:SupportingTokens>
15. <wsp:Policy>
16. <sp:WssSamlV10Token10
17. sp:IncludeToken="... " />
18. </wsp:Policy>
19. </sp:SupportingTokens>
```

Figure 42 Snapshot of WS-SecurityPolicy of S5

This strategy involves defining ordering relationships between security means. These relationships are defined as below:

- Transport security: private net > VPN (IPSec or SSL) > Internet + SSL > Internet
- Message security: Signed > encrypted > none
- Data security:
  - Motivation: Legal = Transaction > personal use > commercial > none
  - Delay: None > Legal > Chosen
- Credential management
  - Authorisation: End to end > User + calling service > user > calling service > none
  - Authentication:
    - Role based: RBAC certificate > IP Domain > IP Address
    - Identity based: X509 Certificate > Kerberos > 1 Time password > login/password

We propose to set QoP terms in WS-Agreement by mean of *GuaranteeTerms*, *service level objectives* (SLO), and their related *business values:*

- Guarantee terms are an extensible and interoperable means to author and convey the nature of an agreement [Andrieux 07]. The Guarantee Term itself occurs where implementers can specify domain specific terms. These terms can be used to convey meaning during a negotiation exchange or afterwards to characterize an

existing agreement. We define an agreement document type and a set of Guarantee term types that must be used both in creations of an agreement service and in expressing this agreement in an agreement document. Externally defined Guarantee languages may be used to extend WS-Agreement behaviours in our case we will terminology to represent “security” terms.

- Service Level Objective: an assertion expressed over service descriptions. Each service level objective may have a list of business values attached to it, representing different value aspects of this objective [Andrieux O7]. Both agreement initiator and agreement responder may specify business values. The business value is intended to represent the strength of an agreement in domain-specific terms. In general, business value is an assertion representing a value aspect of a service level objective attached to the service that is the subject of the agreement. The value may be specified in terms of domain-specific qualities such as importance, cost and others.

In the following example we show an agreement (using NRL-SO ontology for specifying security terms) applied to the S5 service to set the obligation on the service provider to provide SAML authentication protocol AND XML-encryption to encrypt the message.

<wsag:GuaranteeTerm

wsag:Name="SecurityProtocol" wsag:Obligated=”ServiceProvider”>

wsag:ServiceScope="S5">

<wsag:ServiceLevelObjective>

Must_Exist (AuthenticationProtocol_And_EncryptionProtocol)

</wsag:ServiceLevelObjective>

<wsag:BusinessValueList>

<wsag:Preference>

…….

<wsag: AuthenticationProtocol >saml</wsag: AuthenticationProtocol t>

<wsag: EncryptionProtocol>XML-Enc</wsag:EncryptionProtocol >……..

</wsag:Preference>

</wsag:BusinessValueList>

</wsag:GuaranteeTerm>

11.3 Security Services

In order to improve dependability, the security service is designed as an external part, able to be connected either to the ESB or to the governance environment. It implements the necessary toolset to support security patterns organised to support security requirements:

- Confidentiality involves providing access control features as well as a secured transport.
- Integrity involves signing and authenticating messages. This feature also supports the non repudiation service as messages are signed by the service which has sent it.

Security requirements are expressed by the service provider (while defining semantic security annotation and building QoP agreement) and by the service customer while initiating his request. In order to simplify the security policy description security patterns are built as answers to security objectives defined thanks to a simplified ontology (Figure 32).

These patterns are then used to set an adapted XML parameter file which will be used by the Security Service Anatomy security mediator to coordinate and compose the convenient security services.

The security architecture is designed in light of dependability requirements. This leads us to define the security service as a pluggable external service (Figure 43). It is split into semantic and service layers. The semantic layer is in charge of taking into account security preferences and integrating the community model so that different security strategies can be managed according to the context, while the service layer includes the security mediation (federated identity manager, message security and secure transport.) to support end-to-end technological security services composition and integration in the service chain.

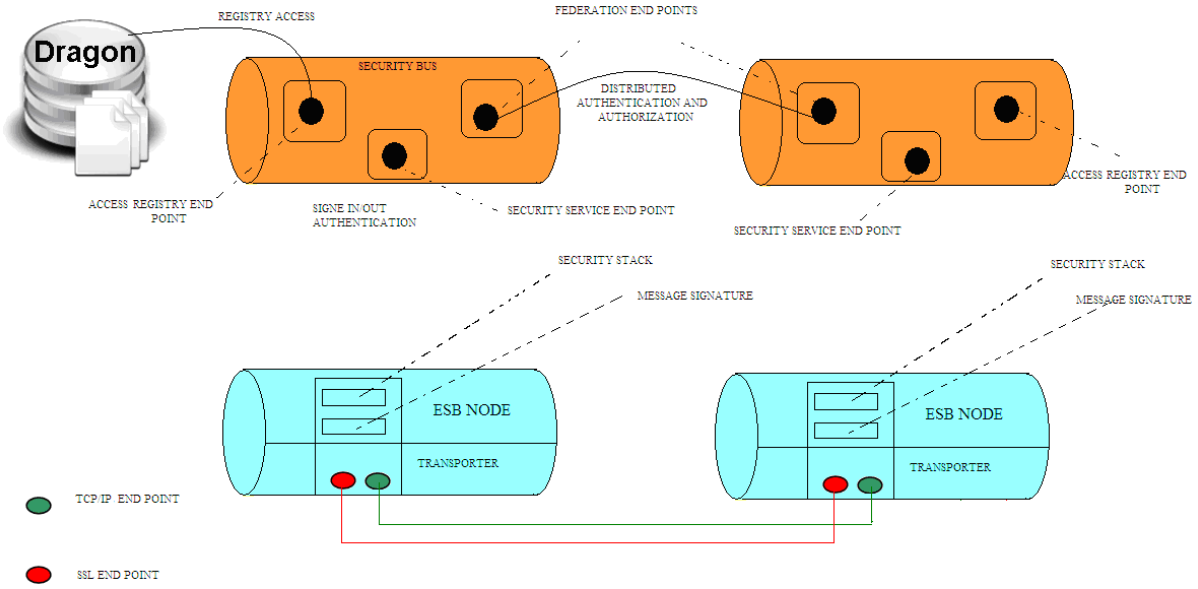

Figure 43 Security Service as a Pluggable External Service

The dependability requirements lead us to design the security architecture in a distributed way, split among ESB and governance nodes. This distributed architecture is based on a federation manager used to support collaborative authentication and authorisation. Trusted nodes communities are organised in federations so that distributed decisions can be supported.

The semantic level includes three components (Figure 44):

- The **QoP manager** is in charge of orchestrating the security components: first it has to activate the QoP matcher, then it has to extract the different actors roles and control that they belong to the convenient access list (generic authorisation process). Lastly, this security agent will activate the service level security mediator in charge of composing the technical security component.
- The **federated authorisation manager** has to extract the convenient authorisation strategy from the QoP contract before launching the authorisation process on the different sites involved in the service chain.
- The **security broker** is used to integrate security requirements while composing a service chain. It supports security integration using a distributed mediation system in which the security policy specification of different services provider may differ, i.e. enforcing security over semantically heterogeneous service provider taking into consideration the requestor requirements and preferences concerning the security. This security broker will manage the security parameters so that technological security component will be instantiated conveniently at the service layer. As said previously, security preferences are organised in different topics:

- Transport security: this part allows the integration of secured communication protocols to communicate safely on unsafe infrastructure
- Credential management: this part deals with the authentication and authorisation processes. Access control policy can be either identity-based or role-based.
- Data security: this part is related to the provider information system infrastructure.

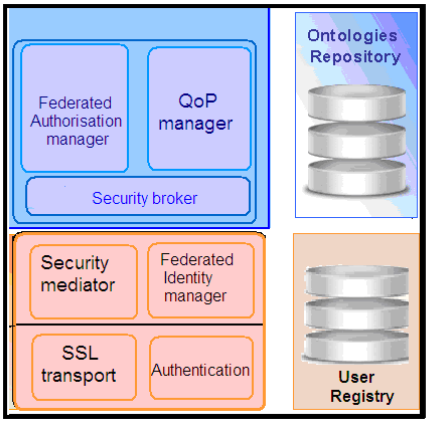

Figure 44 Security Architecture

The service layer is split according to 2 sub-levels (Figure 45):

- **Platform-Independent Component** implements the global view on authentication (implemented thanks to a federated identity manager (Single Sign On component) and transport management (used to describe process and deals with the trust chain constraint (i.e. sending and requesting user preferences).
- **Platform-Dependent Component** integrates platform-based certification information, to bind the interface with the convenient communication protocol. It also manages the implemented authentication frameworks and sends the convenient token / login information to the sign-on system embedded in the bus layer.

The platform independent components are coordinated thanks to the security mediator. This mediator is activated by the security agent. It is in charge of analysing the Quality of Protection Agreement and will activate the transport and SingleSingOn component.

The transport component consists in tagging the messages to define that secured communication must be implemented. This tag will be analysed and used to route the message to the convenient communication software (via SSL or not).

The SingleSignOn component is organised in realms. One realm is attached to each PEtALS

node. In each realm, the SSO repository includes both realm authentication information (including trusted 3rd party references) and user information.

As a person can be associated to several roles, different authentication information can be stored. Connection between realms is set thanks to the Realm federation component. This component is used to check if the user is already registered or not in a referenced realm. Trust relation between realms is used to identify potential and trusted partners to exchange actors' information with. Once registered in the SSO directory, a token is sent back. By this way, identity information can be stored and resent according to the needs. Of course, users can also "disconnect" and delete personal information thanks to a "signoff" process.

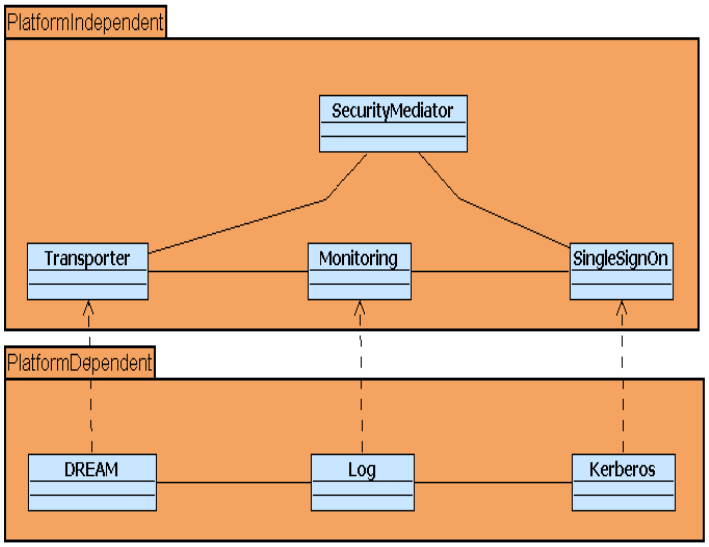

Figure 45 Organisation of the Service Level Security

11.3.1 Transport Security

The aim of the SSL integration is offering a end-to-end SSL solution between clients and providers who are not necessarily on the same or on a PEtALS node. To achieve this goal we have decided to integrated SSL encryption at two levels of PEtALS:

- Between PEtALS nodes: to achieve this goal we decide to add components to DREAM library.
- Between the outside and PEtALS nodes: to achieve this goal we decide to modify the SOAP binding component.

With these levels we can achieve our main goal which is offering tools to support an end to end SSL encryption solution. Indeed communications between clients and providers who are on the same PEtALS node do not need to be secured because they are on a same computer and they don't need to use external network to communicate. Communications between

clients and providers who are not on a same PEtALS node are now secured thanks to the integration of SSL encryption between PEtALS nodes. Communications between clients and providers who are necessarily on a PEtALS node are secured due to the integration of the SSL encryption between the outside and a PEtALS node.

As PEtALS transport component is based on DREAM, we integrate the SSL encryption between PEtALS nodes by modifying both the DREAM library and PEtALS.

As far as DREAM is concerned, we add five components to the library,

- **SSLProtocolImpl** which represents an SSL encrypted communication.
- **SSLProtocol** which encompasses the SSLProtocolImpl component.
- **ChannelOutSSLStack** which represents an output SSL communication.
- **ChannelInOutSSLStack** which represents an input/output SSL communication.
- **ProtocolChunkRouter** which represents a router based on **ProtocolChunk** chunks. This router selects the output corresponding to the protocol name in the **ProtocolChunk** chunk.

All these modifications have been made as modules so the DREAM architecture hasn't been modified and compatibility with older version is guaranteed.

Then, in PEtALS we modify the following components,

- **DreamWrapper** component, this component now looks for SSL property on exchanges to determine if they must be sent using SSL (adding a ProtocolChunk with SSL tag and selecting the correct port) or not (adding an ProtocolChunk with TCPIP tag and selecting the correct port).
- **DreamTransportProtocol** component, we added a **ProtocolChunkRouter** to determine the protocol (SSL or not) to use for the communication. This component selects the appropriate component for output communication.

Lastly, we have modified PEtALS to be allowed to add the DREAM SSL port in *topology.xml* file. We add a new Router module (SecurityModule) to PEtALS; this module will be the binding point between PEtALS and all security mechanisms we will develop.

All these modifications have a small influence in PEtALS architecture and will not alter compatibility with older versions.

Encryption parameters are set up thanks to the addition of *keystore* elements in the *topology.xml* file (Figure 46).

```
...
<tns:sub-domain name="subdomain1">
        ...
    <tns:container name="0">
        <tns:dreamservice>
            ...
            <tns:SSL>
                <tns:port>The SSL port</tns:port>
                <tns:keystore>Path to the keystore file</tns:keystore>
                <tns:keystore-password>Keystore password</tns:keystore-password>
                <tns:key-password>The key password</tns:key-password>
                <tns:truststore>Path to the truststore file</tns:truststore>
                <tns:truststore-password>Truststore password</tns:truststore-password>
            </tns:SSL>
        </tns:dreamservice>
    </tns:container>
</tns:sub-domain>
...
```

Figure 46 A Snapshot from *topology.xml* File Description

Note that in general *keystore* file and *truststore* file are in the same file, so in *truststore* properties you have to set same properties as *keystore* properties.

In order to support an end-to-end secured transport, binding components have also to use a secured transport stack (namely a SSL stack). To set up SSL encryption in SOAP binding component we just have to add the following lines in *jbi.xml* file (Figure 47).

```
<jbi:jbi version="1.0" xmlns:jbi="http://java.sun.com/xml/ns/jbi"
        xmlns:petalsCDK="http://petals.ow2.org/components/extensions/version-4.0"
        xmlns:soap="http://petals.ow2.org/components/soap/version-3.1">
            <jbi:component type="binding-component"
                bootstrap-class-loader-delegation="parent-first">
            ...
            <soap:ssl-port>The SSL port</soap:ssl-port>
            <soap:keystore>Path to the the keystore file</soap:keystore>
            <soap:keystore-password>Keystore password</soap:keystore-password>
            <soap:key-password>The key password</soap:key-password>
            <soap:truststore>Path to the truststore file</soap:truststore>
            <soap:truststore-password>Truststore password</soap:truststore-password>
        </jbi:component>
    </jbi:jbi>
```

Figure 47 Integration of a Secured Transport in the Binding Component by Altering jbi.xml file

11.3.2 Message Security

Message security requires different security services:

- Confidentiality: this service is achieved thanks to a secured transport
- Message integrity: this involves that the message content must be authenticated by the sender and checked by the target service
- Source authentication: the target service must be sure that the message has been really sent by the source service
- Non repudiation: once sent, the source should not repudiate the exchange.

In order to address these requirements, we add a message signature mechanism. This signature mechanism is implemented in the following way: depending on the signature tag, the signing service signs or not the exchange with the source container's private key and put the signature as a part of the exchange properties. In order to allow the destination to check if the message signature, the source Id is also added so that the target service can retrieve the source public key in the trusted key store file in a similar way as for the SSL encryption parameters (see the addition of keystore elements in the topology.xml file (Figure 46)).

This signature is associated to a hashed part of the message and is logged in both source and target log files so that this mechanism implements source authentication, message integrity and message non repudiation services.

11.3.3 Federative Authentication and Authorisation Service

Federative SSO enables creating composite services and securely sharing distributed resources by adopting a cross-organisation authentication mechanism as single set of user security credentials are sufficient to allow access to a multitude of federated resources across the ESB. It allows in the same time local enforcement of access control policies permitting establishing a composite service in a dynamic manner where sets of fine grained distributed security authorization policies of independent organisations.

In our architecture the distributed security service manager is in charge of the authentication and credential management (Figure 48). This system is organised in a distributed manner: each node belonging to the trusted community (a node can be associated to a PEtALS or a DRAGON instance) represents a realm and is associated to an instance of the security service. Each instance includes the following components:

- The federation manager manages the nodes community. It has to manage the community topology, send and process interactions with the other realms belonging to the same community.
- The authentication service implements the Single Sign On system. It includes and Identity provider and authentication service as well as a token manager in charge of generating the convenient tokens after the user has sign-in.
- The attribute manager is in charge of getting the convenient information from a user registry.
- The credential management service has to manage the interface with the service repository (implemented thanks to DRAGON in our case). It has to compose and orchestrate the authorisation process including the service authorisation policy acquisition, the service customer authentication and its convenient attributes collection before setting the matching process.

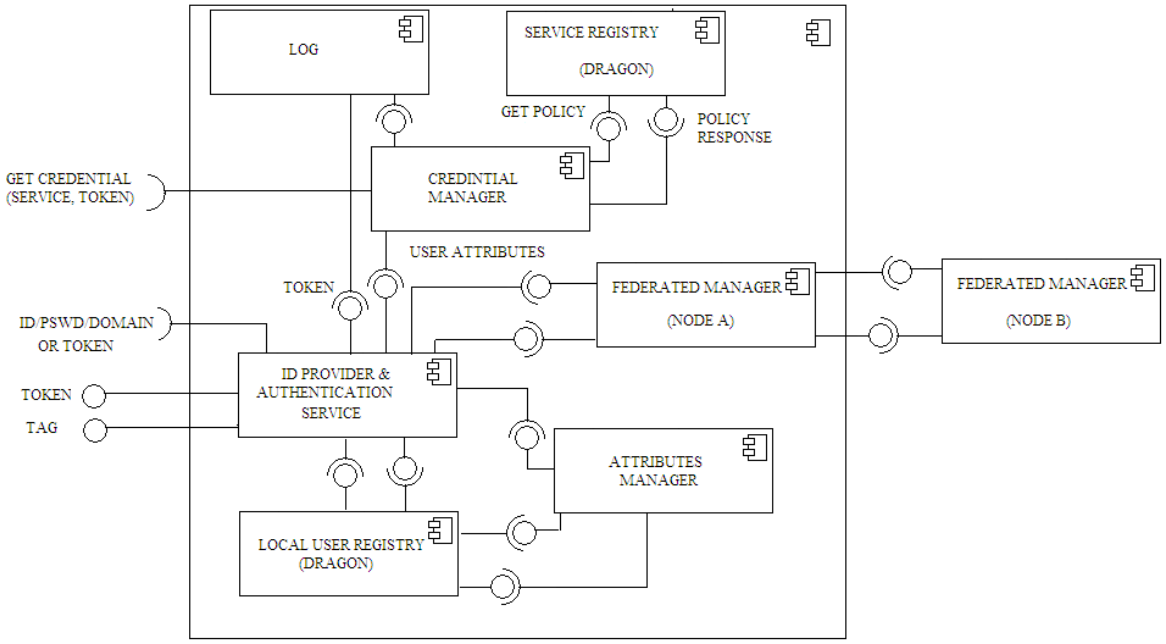

Figure 48 Security Component Organisation

11.3.3.1 User Registry

In our architecture, the user is associated with a service storing its private information. As proposed in the OASIS reference architecture, a user is associated to a main identity and can use several identifiers (seen as secondary identity information) depending on the trusted community it is involved in. The user also owns different attributes and roles that can be used to set access authorisation to a service or to a given piece of information (credential

management). Each sub-identity involves the necessary set of roles for service execution. In this case, user sub-roles could be considered as context attributes, and thus, could be propagated in conjunction with his or her preferences through the chain of services. This approach has two benefits. The master identity of the user can be hidden if necessary. For example, the elementary services enactment does not require user master identity disclosure whereas sub-identities are used according to the appropriate role of composite service enactment. Although this solution increases complexity of the system architecture due to the management of multiple user registries and identity mapping mechanisms, the benefits offered compensate the complex management. The second benefit allows the integration of Attributes Based Access Control (ABAC) and the Role Based Access Control (RBAC). ABAC is widely used in web-based B2C environments where the access control decision is taken based on requestor attributes instead of role-based model. The requestor attributes include information about the service consumer, for instance, in the case where the requestor is a human actor or a service initiated by a human actor, attributes could include full name, age, gender, preferences, context…etc.

Inversely, attributes-based access control requires service providers to express regularisation about who is qualified to use the service, what type of identification is required, what is the privacy level that the service can offer or the context in which a service can be initiated.. such that services descriptions are enabled with “non-functional properties” to guaranty access controls. In this way matching between requestor attributes and services “properties” becomes an important factor in service selection process, along with service functional properties. To enable such a paradigm a semantic mediation system should be established in the service middleware; the role of such a system is matching semantically heterogeneous requestor requirements and service non-functional properties.

Our architecture partly implements this model. Each user is associated to a main identity (defined as a “subject”) associated to the authentication data that will be used to sign in depending on the authentication library (login / password in our prototype). Then secondary identities (called “Principals”) are used to organise the different identifiers and the related attributes so that different roles (depending on the reference model) can be used to annotate semantically the user registry (Figure 49). In this way, authorisation models can be added in the service discovery process as user related functional properties selection.

11.3.3.2 Distributed Authentication System

In our architecture different authentication strategies can be used from the local authentication to a distributed single sign on system. The SingleSignOn architecture is organized in realms. One realm is attached to each PEtALS node. In each realm, the SSO repository includes both realm authentication information (including trusted 3rd party references) and user information.

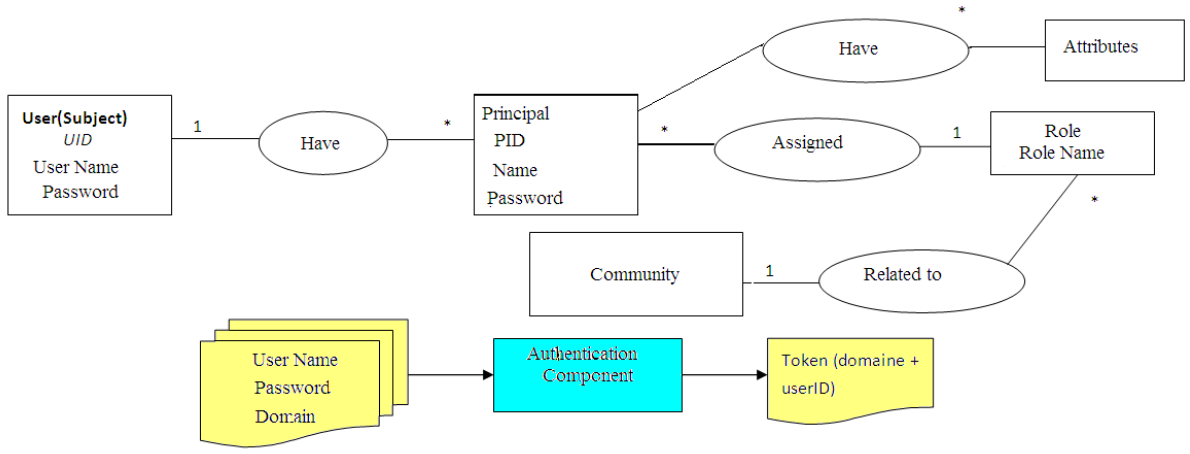

Figure 49 Internal User Registry Organisation

As a person can be associated with several roles, different authentication information can be stored. Connection between realms is set thanks to the federation manager component. This component is used to check if the user is already registered or not in a referenced realm. Trust relation between realms is used to identify potential and trusted partners to exchange actors' information with. Once registered in the SSO directory, a token is sent back). By this way, identity information can be stored and resent according to the needs. Of course, users can also "disconnect" and delete personal information thanks to a "signoff" process.

The authentication process ("sign-in" function) is achieved once the user opens a PEtALS or a Dragon Session. As in Shibboleth, this connection process includes a "WAYF" (Where Are You From) part. Consequently, the user has to provide its main identity information including authentication proof (the password in our case) and the realm identification (namely the node storing its personal information). This information is locally sent on the IdProvider. Depending on the domain the user belongs to, the IdProvider either processes the connection (Figure 50) or sends it to its federation manager which is in charge of routing it on the convenient node IdProvider via its local federation manager. (Figure 51)

Once logged in, a token is sent back to the requestor. This token consists of the user identity information (main id and domain id) and a signature achieved by the IdProvider which is a

connection proof. The connection information is also logged in (token and connection time stamp) in the log system so that it can be retrieved if necessary.

A Sign Out mechanism is also provided to allow the user to disconnect from the system. By this way, the connection token is removed from the active tokens registry. The disconnection event is also logged in (Figure 52).

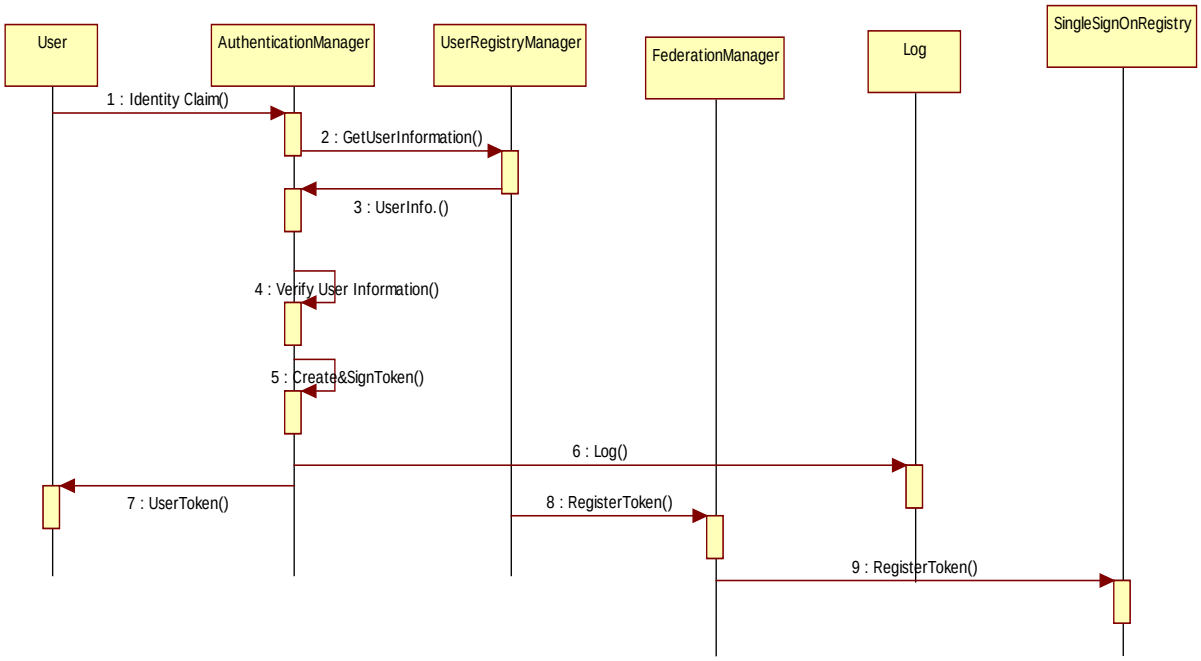

Figure 50 Local Authentication Process

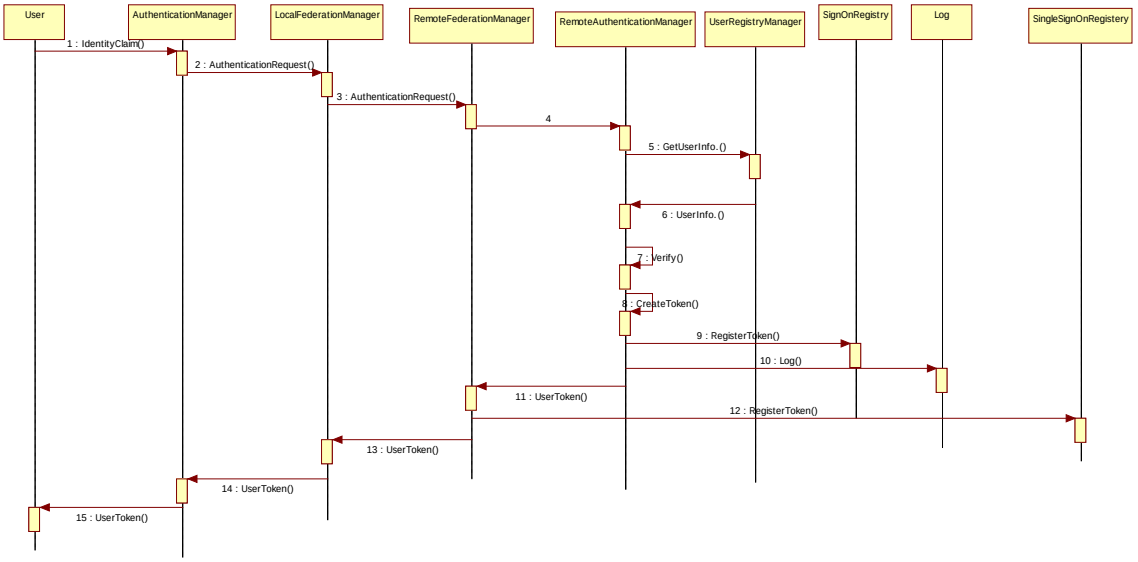

Figure 51 Distributed Authentication Process

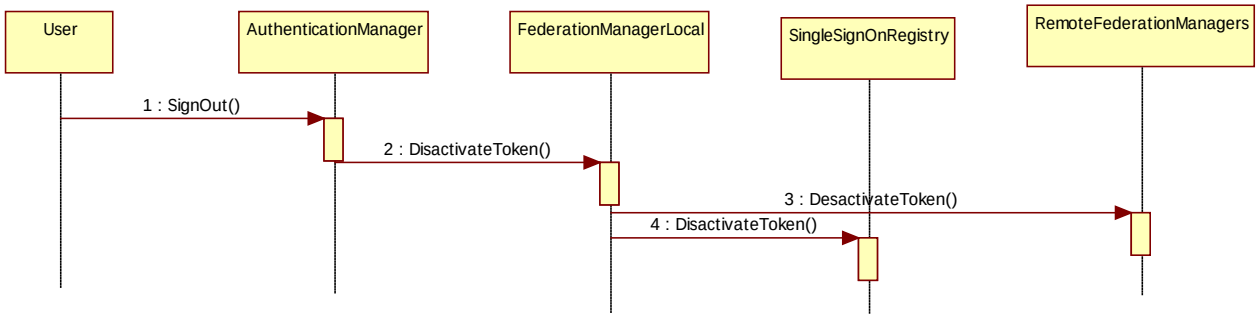

Figure 52 Sign Out Process

11.3.3.3 Credential Management

Different authorisation policies can be set to allow or not the service invocation. These policies can refer either to an access control list (based on user identifiers), a particular attribute or a role based access control. After the service has been discovered and before the service is invoked, the credential manager is used to log the authorisation. It has first to extract the convenient authorisation policy from the service registry. Depending on the authorisation strategy, the credential manager has to get the convenient attributes from the attribute manager (Figure 53). Thanks to the login token transmitted by the requestor, the credential manager invokes the attribute manager (directly for a locally authenticated user or via the federation manager service) to get the requested attributes before the matching process. If the user attributes match the access control criteria, a signed time-stamped access token (including the user identity, a time stamp and the requested attribute) is logged in and sent back to the requestor (Figure 54). The attribute collection process is also published so that services can invoke directly this service in order to get the requested attributes they need for their own internal authorisation processes.

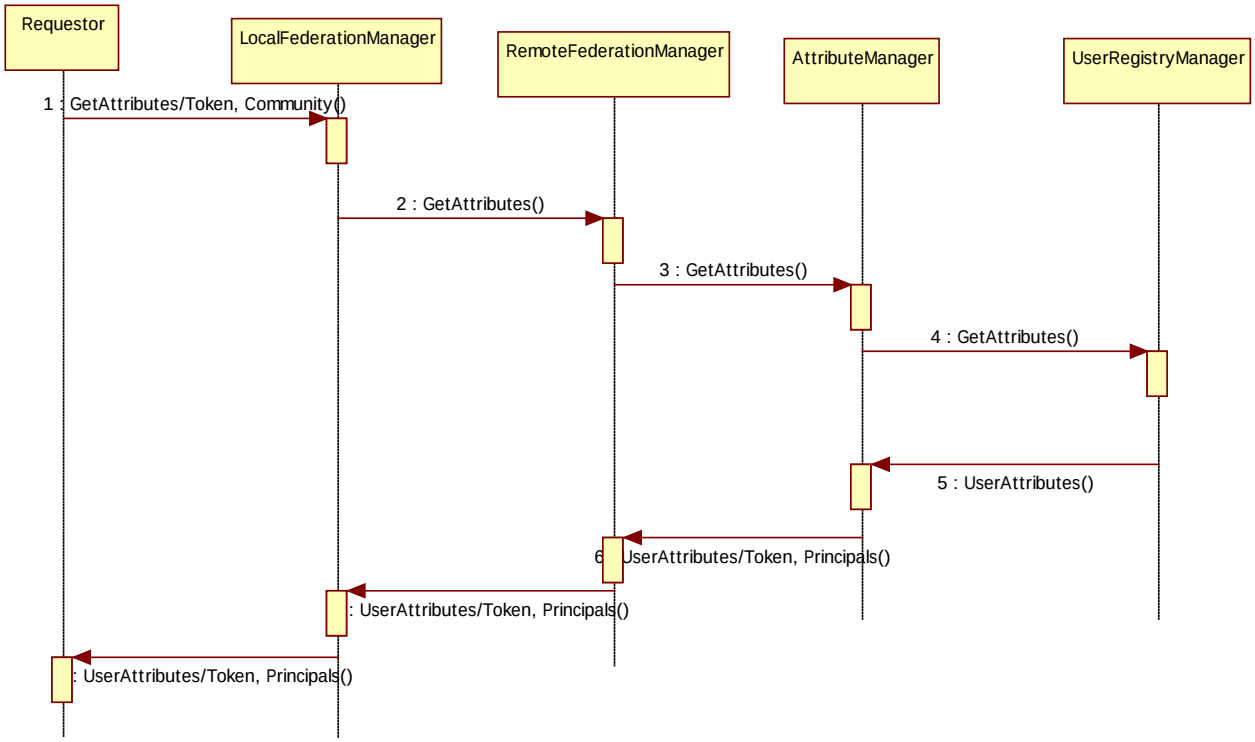

Figure 54 Attributes Collection Process

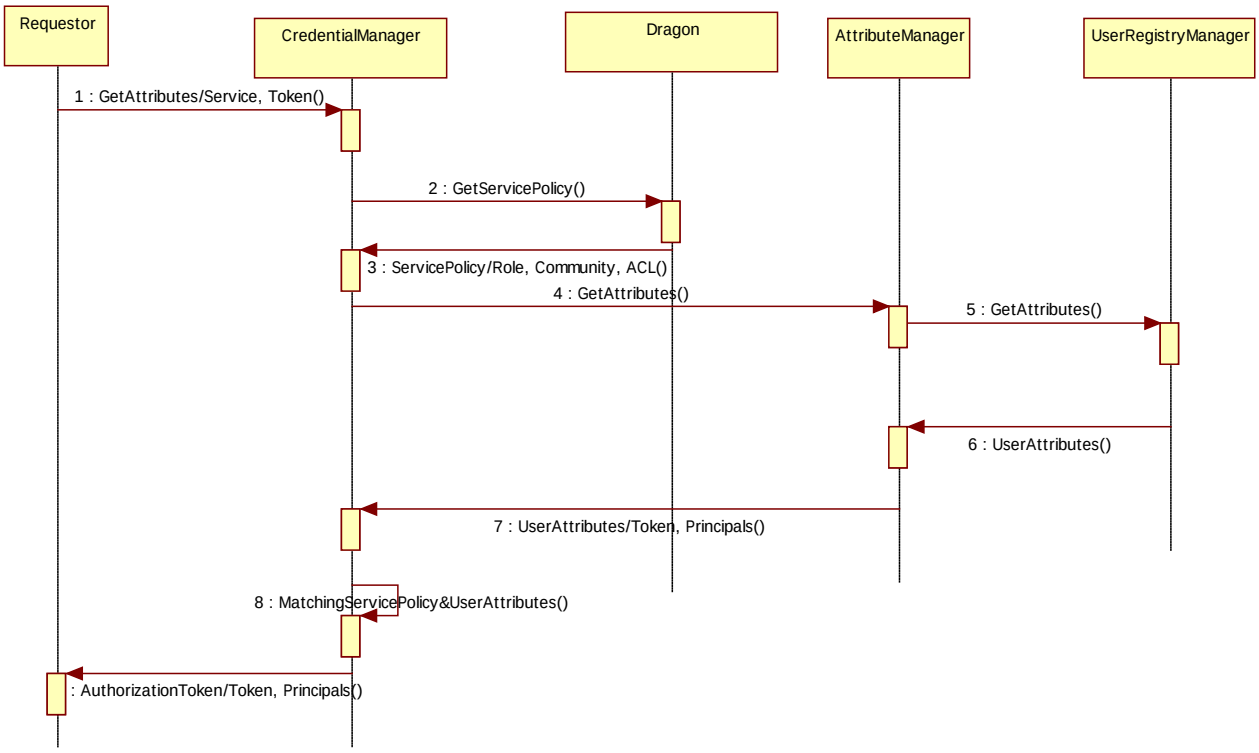

Figure 55 Distributed Authorisation Process

11.4 Integration of the Security Service Bus

The service security bus has been designed as an external component. As far as the semantic layer is concerned, this component is implemented as an independent service proposing different interfaces (Figure 55 and Table 4). As per the service layer, the security service has to be included in the node instance. As presented section (11.3), security patterns are used to set security tags (configured in a XML file used as a parameter for the service invocation) so that a convenient routing to secure/non secure mechanisms can be addressed in the bus, (Figure 56) presents globally the interaction between PEtALS and Dragon nodes with the security service at runtime.

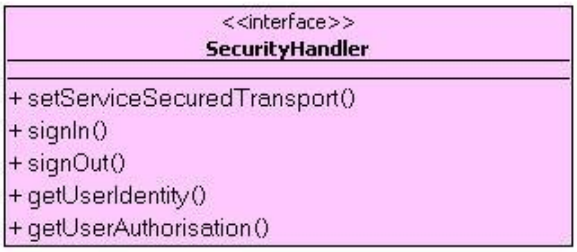

Figure 56 Security Service Interfaces

Interface	Short description	Operation(s)
SecurityHandler	Interface for selecting the convenient security services.	**SetServiceSecuredTransport(service)**: tags the secured transport metadata field in the service interface
AuthenticationManager	Requests and verifies user information from UserRegistryService/or FederationManager.	**SignIn(User, Token):** authenticates the user and import its personal data in the internal SSO directory
SignOutModule	Manages SSO registry, add or erase tokens	**SignOut(User, Token):** exports authenticated user personal data to the user registry and remove it from the SSO repository
FederationManager	Communicates user token obtained from other domains' federation manager.	**GetUserIdentity(Token, authenticationtype):** retrieves the convenient user authentication information for the registered user identified by the token in the SSO repository.
UserRegistryService	Communicates user information to requesting services.	**GetUserInformation, GetUserAttributes:** retrieve user information and user attributes and send them to requesting services.
CredintialManager	Manages authorization process by matching service policy and user attributes.	**GetUserAuthorisation(Token, service):** retrieves the service authorization parameters and control the matching with the properties associated to the registered user (identified by the token).

Table 4 Security Service Interface Specification

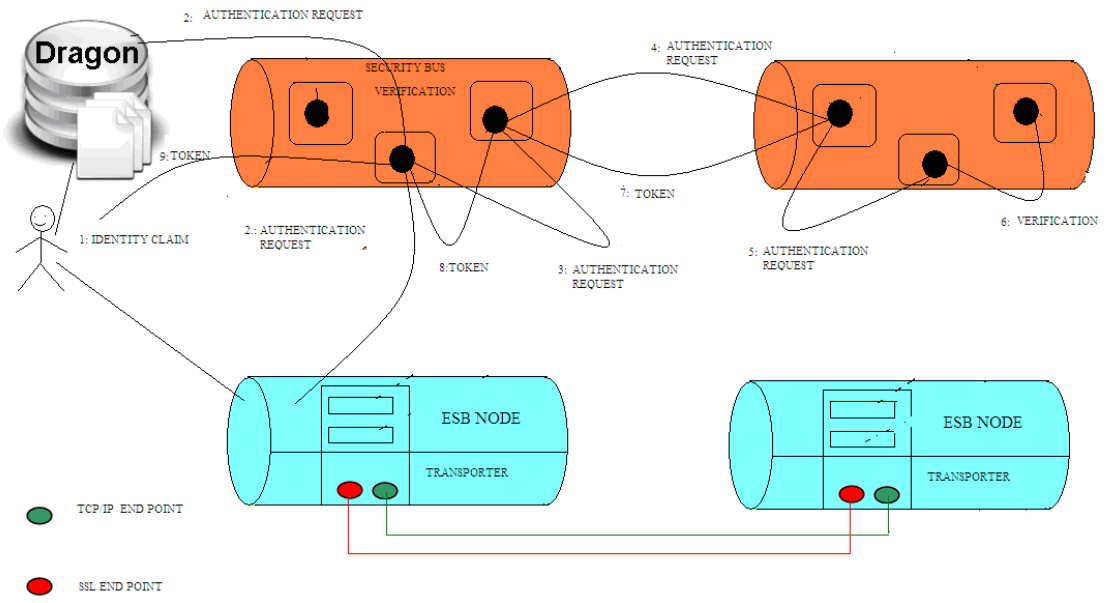

Figure 57 Interactions between PEtALS and Dragon Nodes with the Security Service at Runtime

11.5 Conclusion

Business owners and end users must be able to fine-tune security according to their preferences. The current solutions to express these preferences and requirements have a technical low-level tendency and are not user friendly. The ability to express High-level semantic security preferences and requirements is necessary. Inversely, the ability to compose services within and across the enterprises boundaries to offer new and more coarse-grained business functionalities is one of the most important features of SOA. This involves a number of business partners each of whom has his or her authentication and authorization mechanisms and security requirements expressed and enforced with diverse manners which augments the complexity of the security management at the platform level.

Additionally, the platform supporting the services enactment should be able to dynamically re-configure security to follow context evolution.

In this chapter we introduced a simple, flexible, and semantic framework to secure services architecture along with the implementation of mechanisms to support service discovery, composition and enactment, the implementation extends PEtALS ESB with an extra layer that hides complexity and heterogeneity and enables customized and consistence security definition and authorization enforcement.

As a result, a semantic-based service infrastructure that provides the initial services required to secure and guarantee an end-to-end quality-of-protection services composition has been introduced.

Chapter 12

12 Conclusion and Perspectives

In this work we proposed a Service-Oriented production and information system urbanization strategy and its related model of components to meet agility constraints for both the information system and for the production system organization with special emphasis on the collaboration openness. The concepts we identified provide a dynamic framework for controlling Enterprise Resources taking into account the production context and environment. We have introduced a cross-layer resource-driven approach ensuring the identification, modelling specifications and integration of production objects in form of industrial services in such a way to allow seeing production system as a set of services some of which will be exposed to be used by actors outside of the enterprise. Basing in the components introduced in the urbanization framework we presented architecture thought to govern the execution of composite services taking into account the requirements of partners involved, context and security policy. Our Role-based modelling constructs make it possible to represent organisation and business processes conjointly with the access control. The proposed architecture is based on a context-aware role-based access control model which integrates the different actors through their roles to take into consideration the changing context and security preferences making it possible for the ad-hoc collaborative business process definitions and execution to fit the new B2B service-oriented paradigm. The notions of composite role, role graph and composite session are thought to be adaptable to our model to fit the service composition constraints, that is, it provides a just-in-time role and permissions activation.

The construction of a chain of services is achieved using a selection process which takes into consideration services' functional and non-functional properties. Constraints and preferences issued from functional and non-functional properties are incorporated into the Service Level Agreement. At the implementation level, we have implemented the introduced architecture using Enterprise Service Bus (ESB). In our implementation we build our architecture within and over PEtALS components and nodes. PEtALS is a highly distributed, open source Service Bus. We have introduced a semantic level on the top of a PEtALS bus to cope with "on the fly" composition of services. Further, we introduced a dynamic federated identity management and advanced authorization mechanism implemented a highly distributed service-oriented middleware.

As a result a semantic-based service infrastructure that provides the initial services required to secure and guarantee an end-to-end quality-of-protection services composition has been introduced.

My future work will focus on the definition of a methodology to improve business object identification thanks to artificial intelligence techniques in order to Support a "bottom-up" process composition framework via the integration and updating of "best practices" gained from previous local and collaboration scenarios as basic rules to guide the composition system using capitalization of current practices to extract re-usable patterns, as well as on the integration of security policy requirements in business objects enactment. The aim is to elaborate a framework allowing building flexible collaborative business process based on declarative publishable binding-based rules along with their operational semantics matching methodology with the aim of meeting a distributed collaborative business process design and implementation to satisfy flexibility and adaptability simultaneously with reusability and conformity verification constraints. Furthermore, the framework should provide detailed specifications for an ad-hoc policy-based composition based on best security practices, i.e. security patterns.

There are many possible details that could inspire further work in this area, including:

- Ontology engineering to support mutual description and continuous enhancement and adaptation of fundamental business patterns as well as security elements for clients and services, in such a way to assure composing dynamically collaborative business processes using "best practices."
- Architectural improvement for content protection-oriented authentication mechanisms such as the ability to bind content and rights to a requestor's identity.
- Specification of a language for high-level business and security adjustment with user-friendly editors to allow non-experts to fine-tune or express their preferences according to their requirements.
- A pattern reasoning engine allows the validation of business and security patterns of the various partners involved by detecting inconsistencies or identifying equivalence, inclusion, incompatibility and conflict relationships across different sets of patterns.

References

[Abele 09] Abele, E. et al., Apache Online Documentation, Authentication, Authorization, and Access Control, available at: http://httpd.apache.org/docs-project/, last visit 1/11/2009.

[Abowd 99] Abowd, G., Dey, A., Brown, P., Davies, N., Smith, M., Steggles, P., Towards a Better Understanding of Context and Context-awareness, proc. HUC'99, Lecture Notes in Computer Science, Vol. 1707, (Springer Berlin) p.p. 304-307, 1999.

[AFGI 91] AFGI, Dictionnaire Anglais-Français, Français-Anglais des termes de gestion industrielle, et leurs définitions, AFGI, 1991.

[AFNOR 90] AFNOR, Norme française NF-X 50-150, Gestion de la production : Analyse de la chaîne de valeur, 1990, available at www.afnor.org, last visit 1/11/2009.

[AFNOR 91] AFNOR X50-310- Norme Française NF X50-310, Organisation et Gestion de la Production Industrielle, Concepts Fondamentaux de Gestion de Production, 1991.

[Agrawal 05] Agrawal, R., Kiernan, J., Srikant, R. & Xu,Y., Xpref :A Preference Language for P3P, Computer Networks Vol. 48, Elsevier B.V, p.p. 809-827, 2005.

[Alam 06] Alam, M., Breu, R. & Hafner, M., Modelling Permissions in a (U/X)ML World, proc. ARES 2006, IEEE Computer Society, Washington, p.p. 685 – 692, 2006.

[Alberts 01] Alberts, C. & . Dorofee, A., Octave Threats Profile. CERT White paper, 2001 available at: www.cert.org/archive/pdf/OCTAVEthreatProfiles.pdf, last visit 1/11/2009.

[Anderson 05] Anderson, A., Multiple Resource Profile of eXtensible Access Control Markup Language (XACML) v2.0, OASIS Standard, 2005.

[Andrieux 07] Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S. & Xu. M., Web Services Agreement Specification. Open Grid Forum (OGF), Grid Resource Allocation Agreement Protocol (GRAAP) WG, March 2007, available at: http://forge.gridforum.org/sf/projects/graap-wg, last visit 1/11/2009.

[Antonelli 99] Antonelli, M., Building Information Systems to Integrate the Manufacturing Supply Chain, Master Report, Massachusetts Institute of Technology, USA, 1999.

[ASQ] ASQ, Glossary of American Society for Quality, available at: www.asq.org/abtquality/glossary.cgi, last visit 1/11/2009.

[Aura 99] Aura, T., Distributed Access-Rights Management with Delegation Certificates Lecture Notes in Computer Science Springer Berlin / Heidelberg Volume 1603/199, , p.p. 211-23, 1999.

[Blackwell 98] Blackwell Encyclopedic Dictionary of Operations Management, , 272 pages, Blackwell Publishing, 1997.

[Biennier 05] Biennier, F. & Mathieu, H., Security Management: Technical Solutions v.s Global BPR Investment, Schedae informatica, vol. 14, p.p.13-34, Krakow, Poland, 2005.

[Blondel 97] Blondel, F., Gestion de la Production, 486 P, Paris, Dunod, , 1997.

[Brende 05] Brende, C., Buche, N., Delayre, S., Mocaër, A. & Nevers, J., SOA et urbanisme, Le rôle des Architectures Orientées Services dans l'alignement métier des Systèmes d'Information, Unilog Management White paper, 2005.

[Cahill 08] Cahill, C., Canales, C., Le Van Gong, H. A., Madsen, P., Maler, E., Whitehead, G., Liberty Alliance Web Services Framework: A Technical Overview, Version 1.0, Liberty Alliance Project, 2008.

[CCO 04] CCO, Common Criteria Organisation, Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and General Model Version 2.2, 2004, available at www.hds.com/fr/solutions/storage.../data-security/cc.html, last visit 1/11/2009.

[Chappell 04] Chappell, D. A., Enterprise Service Bus., 288 p, O'reilly Media, 2004

[Chapron 04] Chapron, J : L'urbanisme Organisationnel, Méthodes et Aide à la Décision pour Piloter l'Evolution de Système d'Information, Rapport de Thèse, Ecole Nationale Supérieure des Mines de Saint-Etienne, France, 2004.

[Chen 07] Chen, T.Y., Chen, Y.M, Wang, C.B., Chu, H.C. & Yang, H., Secure Resource Sharing on Cross-organization Using a Novel Trust Method, Robotics and Computer-Integrated Manufacturing Vol. 23, p.p. 421-435 Pergamon Press, USA, 2007.

[Choppy 04] Choppy, C. & Heisel, M., Une Approche à Base de Patrons pour la Spécification et le Développement de Systèmes d'Information, Report, Laboratoire d'Informatique de Paris-Nord, LIPN, 2004.

[Christensen 01] Christensen, E., Curbera, F., Meredith, G.& Weerawarana, S.,Web Services Description Language (WSDL) 1.1, W3C Note, 2001, available at: www.w3.org/TR/wsdl, last visit: 1/11/2009.

[Chu] Chu, C., 5S Home page, School of Information Sciences and Technology, The Pennsylvania State University, USA, http://net1.ist.psu.edu/chu/wcm/5s/5s.html, last visit: 1/11/2009.

[Clement 04] Clement, L., Hately, A., von Riegen, C., Rogers, T., et al. OASIS, Organization for the Advancement of Structured Information Standards, UDDI Spec Technical Committee Draft, 2004.

[Covington 01] Covington, J. M., Moyer, J. M. & Mustaque, A., Generalized Role-Based Access Control for Securing Future Applications, Proc. ICDCS'01, p.p. 391-401, IEEE computer Society, 2001.

[Cortada 95] Cortada, J.W. & Woods, J. A., Encyclopedia of Science & Technology – 8th Edition, New York: McGraw-Hill, 1995.

[Courtois 03] Courtois, A., Martin-Bonnefous, C., Pillet, M., Gestion de production, 454 p, Paris 2003.

[Cranor 03] Cranor, L., P3P: Making Privacy Policies More Useful, IEEE Security and Privacy journal, Vol. 1, IEEE press Security and Privacy, p.p. 50-55, 2003.

[Crowston 98] Crowston, K. & Osborn, C., A Coordination Theory Approach to Process Description and Redesign, IBM Systems Journal, 37(2), p.p. 227-245, 1998.

[Delen 03] Delen, D., Benjamin, P., Towards a Truly Integrated Enterprise Modelling and Analysis Environment, Computers in Industry, volume 51 , Issue 3, P.P. 257 - 268 , 2003.

[Dierks 08] Dierks, T.& Rescorla, E., Internet Engineering Task Force, Network Working Group, The Transport Layer Security (TLS) Protocol Version 1.2, 2008.

[Duncan 95] Duncan, W. L., Total Quality: Key Terms and Concepts, 192 P., AMACOM, New York, 1995.

[Eom 07] Eom, J., Park,S., Han, Y. & Chung, T., Risk Assessment Method Based on Business Process-Oriented Asset Evaluation for Information System Security, proc. ICCS 2007, Lecture Notes in Computer Science, Vol. 4489, p.p 1024-1031, Springer, Berlin, 2007.

[Eveaere 99] Eveaere, C., Management de la flexibilité, Economica, 203pages, Dound, Paris, 1999.

[Farlex 09] Farlex, The Free Dictionary, available at: http://www.thefreedictionary.com, last visit: 111/2009.

[Ferraiolo 95] Ferraiolo, D., Cugini,J., Kuhn, R., Role Based Access Control: Features and Motivations, Proc. Annual Computer Security Applications Conference, p.p.554-563, IEEE Computer Society Press, Washington, 1995.

[Fontanil 99] Fontanil, F., Intégration d'Outils de Simulation et d'Optimisation pour le Pilotage d'une Ligne d'Assemblage Multi-produit à Transfert Asynchrone, Ph. D. Dissertation, University of Paris 13, 1999.

[Forder 07] Forder, R. & Duffy, M., TOGAF to MODAF Mapping, Crown Copyright, Document Number C370-EP-01, Issue 2.0, 2007.

[Fox 95] Fox, M., Barbuceanu, M. & Gruninger, M.: An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour, Fourth Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises, p.p. 11-15, West Virginia

University, , 1995.

[Gadomski 98] Gadomski, A.M., Balducelli, C., Bologna S. & DiCostanzo, G., Integrated Parellel Bottom-up and Top-down Approach. Proc. of The International Emergency Management Society's Fifth Annual Conference (TIEMS 98), p.p.19-22, USA 1998.

[Gamma 95] Gamma, E., Helm, R., Johnson, R., Vissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, 416 p., Addison-Wesley, 1995.

[Gou 03] Gou, H., Huang, B., Liu, W., Li, X., A Framework for Virtual Enterprise Operation Management, Elsevier, Computers in Industry, Volume 50 , Issue 3, P.P: 333 – 352, 2003.

[Gunn 07] Gunn, R. A., Matrix Management Success: Method Not Magic, 161 p., Infinity Publishing, 2007.

[Graebsch 05] Graebsch, M., Information and Communication in Lean Product Development, Ph.D. Dissertation, Technical University of Munich, 2005.

[Gross 03] Gross, T., Security Analysis of the SAML Single Sign-on, Browser/Artifact Profile, proc. ACSAC2003, p.p. 298- 307, IEEE Computer Society, Washington, 2003.

[Haas 07] Haas, H., Hurley, O., Karmarkar, A., Mischkinsky, J., Jones, M., Thompson, L. & Martin, R., SOAP Version 1.2 Specification Assertions and Test Collection, Second Edition, W3C Recommendation, 2007, available at www.w3.org/TR/.../REC-soap12-testcollection-20070427/, last visit: 1/11/2009.

[Han 06] Han, J. & Khan, K., Security-oriented Negotiation for Service Composition and Evolution, Proc. APSEC06, p.p. 71-78, IEEE Computer Society, Washington, 2006.

[Harrison 93] Harrison, D. B. & Pratt, M. D., A Methodology for Reengineering Business, Planning Review, England, 21(2): p.p. 6-11, 1993.

[Hay88] Hay, E. J, The Just-In-Time Break through – Implementing The New Manufacturing Basics, 257 p., John Wiley & Sons, New York, 1988.

[Herault 05] Herault, C., Thomas, G., & Lalanda, P.: Mediation and Enterprise Service Bus, a Position Paper, First International Workshop on Mediation in Semantic Web Services p.p. 67-80, Amsterdame, 2005.

[Hopp 95] Hopp, W. J. & Spearman, M. L., Factory Physics, Foundations of Manufacturing Management, 600 pages, Richard D Irwin, 1995.

[HPM] HPM, Kaizen Blitz - Two Words that Spell Success, High Performance Manufacturing Consulting Inc, www.hpmconsulting.com/, last visit 1/11/2009.

[Hutter 06] Hutter, D., Volkamer, M., Information Flow Control to Secure Dynamic Web Service Composition, proc. SPC 2006, Lecture Notes in Computer Science, Vol. 3934, p.p.196-210, Springer, Berlin, 2006.

[IBM 02] IBM & Microsoft, Security in a Web Services World: A Proposed Architecture and Roadmap, A Joint Security Whitepaper, Version 1.0, 2002 available at: download.boulder.ibm.com/ibmdl/pub/software/dw /library/ws-secmap.pdf, last visit 1/11/2009.

[IFIP 99] IFIP–IFAC Task Force on Architectures for Enterprise Integration Generalised Enterprise Reference Architecture and Methodology, Version 1.6.3, 1999. http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html, last visit 1/11/2009.

[Imai 86] Imai, M., KAIZEN: The Key to Japan's Competitive Success, 260 P., McGraw-Hill; New York, 1986.

[ISO 00] ISO, International Standardisation Organisation, Code of Practice for Information Security Management Information Technology: ISO/IEC 17799:2000 Standard, available at: www.iso.org/iso/catalogue_detail?csnumber=33441, last visit 1/11/2009.

[ISO9000] ISO9000/2000, International Standardisation Organisation, www.iso.org/iso/fr/

[Jackson 00] Jackson, M., Problem Frames: Analyzing and Structuring Software Development problems, 416 p., Addison-Wesley, 2000.

[Javel 04] Javel, G., organisation et gestion de production, 520 p., Dunod, 2004.

[Joiner 94] Joiner, B., Fourth Generation Management, , 289 p., McGraw-Hill, New York, 1994.

[Kearney 07] Kearney, P., Brügger, L., A Risk-driven Security Analysis Method and Modelling Language, British Telecom Technology Journal Vol. 25, p.p.141-153, Springer, 2007.

[Kim 02] Kim, S. & Jang, K.J., Designing Performance Analysis and IDEF0 for Enterprise Modelling in BPR, Elsevier, International Journal of Production Economics, Volume 76, Number 2, 21, pp. 121-133, 2002.

[Kanneganti 07] Kanneganti, R. & Chodavarapu, P. A., SOA Security, 512 p., Manning Publications, 2007.

[Koubarakis 02] Koubarakis, M., Plexousakis, D., A Formal Framework for Business Process Modelling and Design, Journal of Information Systems, Volume 27 , Issue 5 , P.P: 299 – 319, Elsevier,2002.

[Lawrence 06] Lawrence, K., Kaler, C., Nadalin, A., Goodner, M., Gudgin, M., Barbir, A. & Granqvist, N. H., WS-SecurityPolicy 1.2 Committee Draft 01, Organization for the Advancement of

Structured Information Standards, OISIS, 2006, available at docs.oasis-open.org/ws.../ws-securitypolicy/.../ws-securitypolicy-1.2-spec-cd-01.pdf, last visit 1/11/2009.

[L'encyclopédie] L'encyclopédie e-Business, le Journal du Net: available at www.journaldunet.com/encyclopedie/definition/325/51/20/business_process_management_system.shtml, last visit 1/11/2009.

[Le Roux 04] Le Roux, B., Desbertrand, L., Guerif, P., Tang, X.: Urbanisation et Modernisation de SI, 214 p., Hermes Science Publications, Paris, 2004.

[Li 02] Li, D., Hu,S., Bai,S., A uniform model for authorization and access control in enterprise information platform. Proc. EDCIS2002, Lecture Notes in Computer Science, vol. 2480, p.p.180-192, Springer Berlin 2002.

[Li 94] Li, H. & Williams, T. J., A Formalisation and Extension of the Purdure Enterprise Reference Architecture and Purdure Methodology, Report Number 158, Published as Part of Engineering Experiment Station Bulletin 143 Series, Purdue Laboratory for Applied Industrial Control, School of Engineering, Purdue University, USA, l994.

[Longépé 04] Longépé, C.: Le Projet d'Urbanisation du S.I, Second Edition,284 p., Dunod, Paris, 2004.

[Lossent, L.] Lossent, L. & Gzara, L., Système de Gestion des Données Technique et Wokflow associé, Université de Cuces, Nancy, France, available at www.cyber.uhp-nancy.fr/.../cours_2_1_1.html, last visit : 1/11/2009.

[Madsen 05] Madsen. P., Maler, E., Wisniewski, T., Nadalin E.T. , Cantor, S., Hodges, J. & Mishra, P., Security Assertion Markup Language (SAML) Executive Overview, version 2.0, Organization for the Advancement of Structured Information Standards, 2005, available at www.oasis-open.org/...php/.../sstc-saml-tech-overview-2%200-draft-10.pdf, last visit 1/11/2009.

[Madsen 08] Madsen, P., Itoh, H., Ishikawa, K. & Arisa, F., Liberty ID-WSF Multi-Device SSO Deployment Guide, Version: 1.0-02, Liberty Alliance Project, 2008, available at: www.projectliberty.org/liberty/.../draft-liberty-idwsf-mdsso-deployguide-v1.0-02.pdf, last visit 1/11/2009.

[McCabe 08] McCabe, F. G., Estefan, J. A., Laskey, K., McCabe, F.G., Thornton, D., Reference Architecture for Service Oriented Architecture Version 1.0, Public Review Draft 1, OISIS, Organization for the Advancement of Structured Information Standards, OISIS, 2008, available at: docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf, last visit 1/11/2009.

[Mahoué 01] Mahoué, F., The E-World as an Enabler to Lean, Master Dissertation, Massachusetts Institute of Technology, 2001.

[Mayer 95] Mayer, R. J., Menzel, C. P., Painter, M. K., deWitte P. S., Blinn, T. & Perakath, B.,

Information Integration for Concurrent Engineering IDEF3 Process Description Capture Method Report, University Drive East College, Texas, 1995, available at: www.idef.com/pdf/Idef3_fn.pdf, last visit 1/11/2009.

[Mertins 05] Mertins, K., Jochem, R., Architectures, Methods and Tools for Enterprise Engineering, International Journal of Production Economics, Volume: 98, p.p. 179-188, Elsevier, 2005.

[MIT 2009] MIT, Massachusetts Institute of Technology, Kerberos: The Network Authentication Protocol, 2009, available at http://web.mit.edu/Kerberos/, last visit 1/11/2009.

[Monden 97] Monden, Y., Toyoto Production System – An Integrated Approach to Just-In-Time, 480 P, Georgia, Engineering &Management Press, Norcross, 1997.

[Monden 93] Monden, Y., Toyota Management System - Linking the Seven Key Functional Areas, Portland 222 p., Productivity Press, Oregon, 1993.

[Mondon 05] Mondon, C., Supply Chain Management en PMI : Le chaînon manquant, 378 p., AFNOR, 2005.

[NATO 07] NATO Consultation Command and Control Board, NATO Architecture Framework Version 3, 2007, available at: www.mod.uk/DefenceInternet/.../MODAF/, last visit 1/11/2009.

[Nickulland 05] Nickulland, D., Connor, M., MacKenzie, C., Watson, B., Cowan, M. & Chase, E.: Service Oriented Architecture Whitepaper, Adobe Systems Inc. 2005, available at: www.adobe.com/.../pdfs/Services_Oriented_Architecture_from_Adobe.pdf, last visit 1/11/2009.

[Nightingale 98] Nightingale,D. S., Lean Aerospace Initiative, Massachusetts Institute of Technology, USA, 1998, available at: lean.mit.edu, last visit 1/11/2009.

[NIST 93] NIST, National Institute of Standards and Technology, IDEFØ a Standard for Function Modeling, 183 p., FIPS Publication, 1993.

[OMB 07] OMB, FEA Consolidated Reference Model Document, Version 2.3, The White House Office of Management and Budget, 2007, Available at: http://www.whitehouse.gov/omb/e-gov/fea/, last visit 1/11/2009.

[Preibusch 06] Preibusch, S., Privacy Negotiations with P3P, Proc. W3C Privacy Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement,Ispra, Italy, 2006, available at: www.w3.org/2006/07/privacy-ws/papers/24-preibusch-negotiation-p3p/, last visit 1/11/2009.

[Rafiy 07] Rafiy, S., Dawn, J. & Peter, B, Management of Users Privacy Preferences in Context, proc. IRI 2007, , p.p. 91-97, IEEE Computer Society, Washington2007.

[Ray 06] Ray, I., Kumar, M. & Yu, L., LRBAC: A Location-Aware Role-Based Access Control, Proc.ICISS2006, Lecture Notes in Computer Science Vol. 4332, p.p. 147- 161, Springer, Berlin, 2006.

[Razmerita 05] Razmerita, L., Services Contextualisés pour Utilisateurs et la Modélisation des Utilisateurs à base d'Ontologie: Défis et Perspectives, proc. EGC'2005 Workshop, Paris, 2005, available at: www-sop.inria.fr/acacia/.../Liana.../RazmeritaAtelierMUf.pdf, last visit 1/11/2009.

[Reithofe 97] Reithofe, W. & Naeger, G., Bottom-up Planning Approaches in Enterprise Modelling – the Need and the State of The Art, Computers in Industry, Volume 33, Issue 2-3, P.P: 223 -235, Elsevier,1997.

[Rembold 98] Rembold, U. & Janusz, B., The Role of Models in Future Entreprise, Annual Reviews in Control, issue 22, P.P: 73-83, Elsevier, 1998.

[Robiette 05] Robiette, A. & Morrow, T., Innovation in the use of ICT in Education and Research, Blueprint for a JISC Production Federation, JISC Development Group; Version 1.1, 2005, available at: www.jisc.ac.uk, last visit 1/11/2009.

[Rother 99] Rother, M., Shook, J.: Learning to See Value Stream Mapping to Add Value and Eliminate Muda, 102 p., The Lean Enterprise Institute, Brookline MA, 1999.

[Sassoon 98] Sassoon, J., Urbanisation des Systèmes d'Information, 122p, Hermes Science, Paris, 1998.

[Scavo 05] Scavo, T. & Cantor, S., Shibboleth Architecture, Technical Overview, Working Draft 02, 2005, available at: shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf last visit 1/11/2009.

[Seitz 03] Seitz, T.A., Lean Enterprise Integration: A New Framework for Small Businesses, Master Report, Massachusetts Institute of Technology, USA, 2003.

[Shah 03] Shah, R. & Ward, P.T, Lean manufacturing: Context, Practice Bundles, and Performance, Elsevier, Journal of Operations Management, Volume 21, pp. 129-149, 2003.

[Simon 95] Simon, K. A., From Structure to Process: A Vision of a Process-Based Organization, Proc. ENTER95 Conference, Springer Verlag, Innsbruck, Austria.Innsbruck, 1995.

[South95] South, D.W., Encyclopedic Dictionary of Industrial Automation and Computer Control, 352 p Prentice Hall, New Jersey: 1995,.

[Sun 08] SunMicrosystems, Sun Java System Directory Server Enterprise Edition 6.3 Reference, 2008, available at: docs.sun.com/coll/1224.4, last visit 1/11/2009.

[Suresh 98] Suresh, N. C. & Kay, J. M., Group Technology and Cellular Manufacturing: State of the Art

Synthesis of Research and Practice, 568 p., Springer, 1998,.

[Syukur 04] Syukur, E., Loke, S. & Stanski, P., A Policy Based Framework for Context Aware Ubiquitous Services, Proc. EUC 2004, Lecture Notes in Computer Science, Vol. 3207, p.p.346-355, Springer, Berlin, 2004.

[Szyperski 97] Szyperski, C., Component Software - Beyond Object Oriented Programming, 411 p., Addison Wesley, 1997.

[Tajiri 92] Tajiri, M.& Gotoh, F., TPM Implementation, 220 pages, McGraw-Hill, New York, 1992.

[Trabelsi 07] Trabelsi, S., Gomez, L., Roudie, Y., Context Aware Security Policy for the Service Discovery, Proc. AINA07, p.p. 477-482, IEEE Computer Society, Washington, 2007.

[URBA-SI 02] URBA-SI, Pratiques de l'Urbanisme des Systèmes d'Information en Entreprises, 184 p., Publibook, 2002.

[USDoD 04] USDoD, Department of Defence of USA, NRL Security Ontology, 2004, available at: chacs.nrl.navy.mil/projects/4SEA/ontology.html, last visit:1/11/2009.

[USDoD 07] USDoD, DoD Architecture Framework Version 1.5, Volume I: Definitions and Guidelines, 2007, available at www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf, last visit 1/11/2009.

[USDoD 85] USDoD, United States Department of Defence, Trusted Computer Security Evaluation Criteria Orange Book, Report19855200.28-STD, 1999, available at csrc.nist.gov/publications/history/dod85.pdf, last visit 1/11/2009.

[Vallespir 92] Vallespir, B., Zanettin, M. & Chen, D., GIM, GRAI: A Methodology for Designing CIM Systems, Version 1.0, Unnumbered Report, LAP/GRAI, University Bordeaux 1, Bordeaux, France, 1992.

[Vanhaver. 99] Vanhaverbeke, W. & Torremans, H., Organizational Structure in Process-Based Organization, Knowledge and Process Management Review, Volume 6, p.p. 41-52, Wiley Interscience, 1999.

[W3C 03] P3P, Platform for Privacy Preferences, Project of W3C, Enabling Smarter Privacy Tools for the Web, available at http://www.w3.org/P3P/, last visit: 1/11/2009.

[W3C 02] W3C Working Group, A P3P Preference Exchange Language, Working Draft , version1.0, 2002, available at:http://www.w3.org/TR/2002/WD-P3P-preferences- 20020415/, last visit: 1/11/2009.

[W3C 04] W3C Working Group, P3P Using the Semantic Web (OWL Ontology, RDF Policy and RDQL Rules), 2004, available at: www.w3.org/P3P/2004/040920_p3p-sw.html, last visit: 1/11/2009.

[Wainer 07] Wainer, J., Kumar, A. & Barthelmess, P., DWRBAC: A Formal Security Model of Delegation and Revocation in Workflow Systems, Information Systems, Vol. 32, p.p.365-384, Elsevier Science, Oxford, 2007.

[Wason 05] Wason, T., Cantor ,S., Hodges, J., Kemp, J. & Thompson, P., Liberty ID-FF Architecture Overview, Version: 1.2-errata-v1.0, Liberty Alliance Project, 2005.

[Weber 05] Weber, B., Reichert, M., Wild, W. & Rinderle, S., Balancing Flexibility and Security in Adaptive Process Management, proc. CoopIS, DOA, and ODBASE 2005, Lecture Notes in Computer Science, Vol. 3760 p.p. 59-76, Springer, 2005.

[Williams 95] Williams, T. J.& Rathwell, G. A., Use of the Purdue Enterprise Reference Architecture and Methodology in industry (the Fluor Daniel Example), 1995, available at: http://www.pera.net/, available at: 1/11/2009.

[Womack 03] Womack, P. & Jones, T., Lean Thinking – Banish Waste and Create Wealth in Your Corporation, 384 p, Free Press, USA, 2003,.

[Yee 05] Yee, G., Korba, L., Negotiated Security Policies for E-Services and Web Services, Proc. ICWS 2005, IEEE Computer Society, Washington, 612-619, 2005.

[Yialelis 96] Yialelis, N., Lupu, E. & Sloman M., Role-BasedSecurity for Distributed Object Systems, Proc. ICE'96, IEEE Computer Society, Washington, p.p. 80–85, 1996.

[Zhang 06] Zhang G. & Parashar, M., Dynamic context aware access control for grid application, Proc. Of GRID'XY, Computers & Security, Volume 25, p.p. 507-521, SienceDirect, 2006.

[Zhou 06] Zhou, X., Tsai, W., Wei, X., Chen, Y. & Xiao, B., Policy Infrastructure for Verification and Control of Service Collaboration, proc. ICEBE2006, p.p. 307-314, IEEE Computer Society, Washington, , 2006.

Appendixes

1 Use Case: Fire-fighting 1

1.1 General Context

This use case has been taken from SemEUsE's use cases and has been extended to demonstrate our propositions concerning QoP annotations of the services. The use case binds the general notion of physical resources available in the physical world to Web Services. In fact, every manageable fire-fighting mean (Truck, Fire Fighter, Policeman,…) is associated to a corresponding Web service in the system. Its interface allows sending orders to the associated piece of equipment or personnel, and its service offer advertises both functional and non-functional characteristics (e.g. sensors on the physiological state of a fireman and physical state of its equipment).

The use case especially illustrates, at the business level, the importance of:

- Transversal coordination between the various stakeholders, which translates to:
- The dynamic and potentially optimal, use of all the equipments made available by various stakeholders within the framework of a general coordination.
- Interoperability need between heterogeneous information systems that weren't build to this end.
- Taking into account the Quality of Service (QoS) associated to the available equipments and personnel.

  The separation between:

  - The specification, before the crisis, of business processes corresponding to usual best practices for solving this type of situation.
  - The execution, during intervention, of these business processes according to the available stakeholders or equipments (Web Services) that were discovered dynamically in-situ and selected according to their functional and non-functional characteristics (i.e. QoS).

[1] This use case is an extended version of a use case defined by Pierre Châtel, Thales Communications, 1 à 5 avenue Carnot – 91883 Massy Cedex - France

The general context of this use case being civilian crisis management, the use case will focus on fire-fighting scenarios. For instance, following a major fire in a sensitive inhabited area, different stakeholders are brought to cooperate in order to carry out relief operations successfully.

1.2 Actors

Several professions are concerned by this use case, but there are two main categories of actors:

- Category 1: Domain experts and engineers that are involved in setting up the system before crisis (and, as such, before system runtime).
- Category 2: Firefighters, law enforcement authorities, health and medical services that come to participate during crisis (and, as such, during system runtime). In fact, the key to the correct resolution of this crisis lies mostly in an enduring information sharing between the different stakeholders when conducting joint operation phases.

These actors are detailed in the following table:

Actor	Type	Localization	Role
Domain Expert (Cat. 1)	Human	N/A	Establish or reuse domain ontologies with input from other actors.
Engineer (Cat. 1)	Human	N/A	Foster SemEUsE usage by specifying Web Services and Business Process with input from other actors.
Firefighters (Cat. 2)	Service	Barracks / Scene	Extinguishing fires in order to safeguard property and persons. Also to give first aid if necessary.
Law enforcement authorities (Cat. 2)	Service	Barracks / Scene	Maintain order and secure site
Health and Medical Services (Cat. 2)	Service	Hospital / Scene	Help injured people.

1.3 Scenario 1: Definition of a Fire-fighting Service Offer for Publication into SemEUsE

1.3.1 Context

When adopting the SemEUsE platform in a fire department, the first task that needs to be carried out is to ensure the presence of the equipment (trucks, helicopters, CCTV, weather services,…) in the form of Web Services that can be interrogated or controlled remotely.

In this scenario we will see how to integrate one particular piece of equipment, a fire-fighter tank truck into the SemEUsE platform. Once the technical constraints associated with the deployment of this new technology have been settled, the formal specification of Web Service offer is needed. The following actions need to be undertaken for this specification to be complete.

1.3.2 Description

This scenario includes the specification of all the following characteristics of the service before entering the crisis-management phase:

- Specification of the technical characteristics of the service at the syntactic level (protocols, methods, input/outputs …)
- Specification of the functional properties of the service in the form of semantic annotations on the syntactic service offer.
- Specification of the non-functional properties of the service in the form of guaranteed numerical QoS bounds in a QML-inspired formalism.

1.3.3 Steps

Step	Actor	Event
1	Domain Expert	Writes or reuse domain ontology(ies) that will be used for semantic annotation.
2	Engineer	Writes the technical (syntactic) specification of the tank truck Web Service.
3	Engineer	Defines the functional properties of the tank truck Web Service by annotating the previously established technical specification with semantic concepts extracted from domain ontologies.
4	Engineer	Defines the non-functional properties of the tank truck Web Service by translating its Quality of Service specification into formal QoS bounds.

1.3.3.1 Details on Step 3

In step 3, an engineer semantically annotates the syntactic service offer that has already been defined back in step 2. This annotation is made at two different levels:

- First, at the service level. In the SAWSDL offer, an annotation is put on the interface of the service that allows an engineer to link the service to a defining semantic concept tank trunk extracted from a domain ontology (see. Figure 57).
- Then, at the method level. This tank truck service offers one method to instruct a geographic destination, and another to indicate a location to irrigate. Both will be annotated with semantic concepts extracted from a domain ontology (go_to, location, irrigate, location, volume_liter)
    - *go_to(location) :* status

```
<wsdl:operation name="go_to" sawsdl:modelReference=
"http://www.thalesgroup.com/ontologies/firefighting#go_to">
  <wsdl:input element="location" />
  <wsdl:output element="status" />
</wsdl:operation>
```

    - *irrigate(location, volume_liter) :* status

```
<wsdl:operation name=" irrigate " sawsdl:modelReference=
"http://www.thalesgroup.com/ontologies/ firefighting #irrigate">
  <wsdl:input element="location" />
  <wsdl:input element="volume_liter" />
  <wsdl:output element="status" />
</wsdl:operation>
```

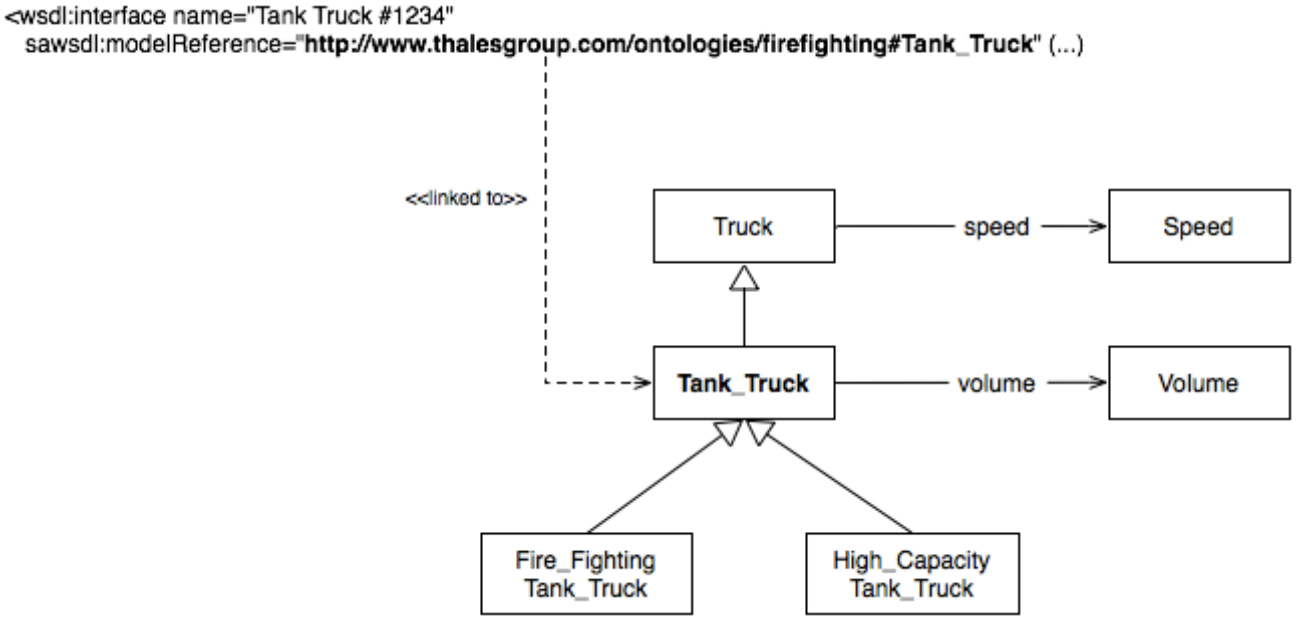

Figure 58 – Semantic Annotation of a Web Service Interface

1.3.3.2 Details on Step 4

In step 4, an engineer indicates the QoS contract offered by the service and its methods. These characteristics may include business or technical considerations. In the following list, we give the various contracted QoS properties in their human readable and formalized forms:

- Business: “The tank truck has a maximum capacity of 4000 liters of water and a minimum of 1000 liters”.

- 1000 Liters <= tank_truck.capacity <= 4000 Liters

- Business: “The tank truck will maintain its speed between 150 and 200 Km/h”
  - 150 Km/h <= tank_truck.speed <= 200 Km/h

- Technical: “The service will maintain a response time below 20 ns (not included)”
  - tank_truck.response_time < 20 ns.

- Technical: “The processing time of the irrigate command will not go over 10ns“
  - tank_truck.irrigate.processing_time <= 10 ns

1.4 Scenario 2: Definition of a Fire-fighting Process for Execution

1.4.1 Context

When adopting the SemEUsE platform in a fire department, the first task that needed to be carried out was to ensure equipment and stakeholder presence in the form of Web Services. This has been illustrated by the previous scenario.

It is also mandatory to establish the processes that will use the available services at runtime. In this scenario we focus on a particular fire-fighting process defined as a BPEL business process that has been extended to take into account the functional and non-functional requirements of the service consumer.

1.4.2 Description

Thanks to the targeted use of business semantics, the aim is to significantly reducing the coupling between service requests and offers. Even without knowing precisely, at the time of writing a process, which services will be available at runtime, one is able to express unambiguously its functional and non-functional requirements through the available domain ontologies.

This scenario illustrates the following features of our framework:

- Functional requirements specification, using the semantic specification of the domain in the form of ontologies.
- Non-functional (QoS) requirements specification, which will be used to optimize the fire-fighting operation.
- Non-functional user preferences specification, in order to allow the non-functional selection of "optimal" service(s) at runtime.

1.4.3 Steps

Step	Actor	Event
1	Engineer	Writes the syntactic outline of the fire-fighting BPEL process. This include the various steps of the process, its branches, used data-types and so on…
2	Engineer	Defines the functional requirements of each service call in the fire-fighting process using concepts extracted from domain ontologies.
3	Engineer	Defines the non-functional requirements (if any) of each service call in the fire-fighting process
4	Engineer	Defines the non-functional user preferences (if any) for each service call in the fire-fighting process.

1.4.3.1 Details on Steps 1 to 3

After steps 1 to 3 completion, the following fire-fighting process is obtained (see Figure 58). It indicates some of the necessary steps involved in the management of this particular crisis:

I. Delimit fire perimeter.

II. Circle the area to protect lives and limit access to the site.

III. Prevent fire propagation by cutting vegetation around the perimeter in order to establish a clean area.

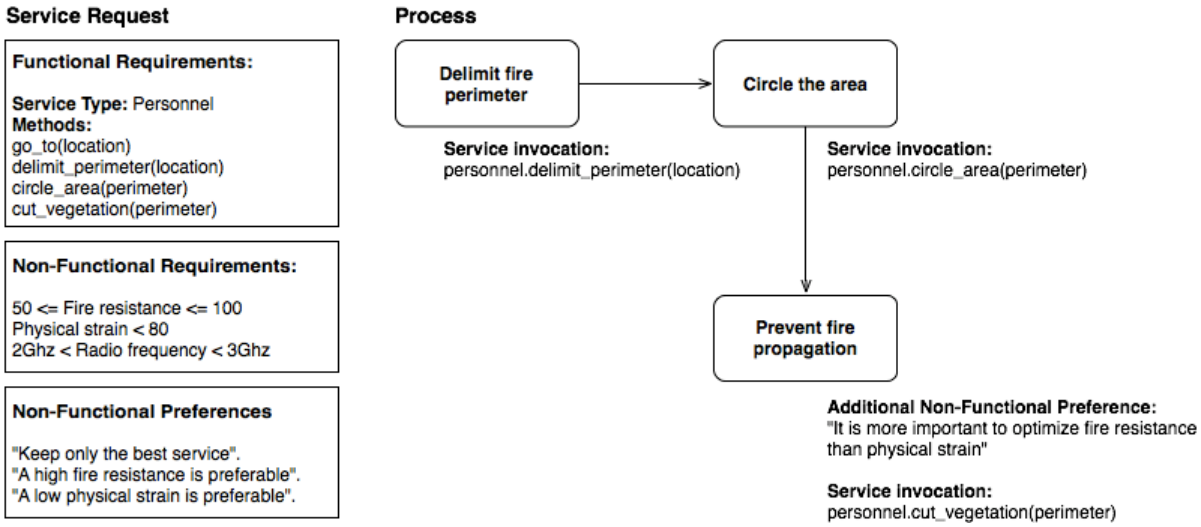

Figure 59 – Fire Fighting Process

For readability purposes, this scenario defines a single service request for use in all the fire-fighting tasks of the process. After step 2 and 3 we obtain the following requirements:

- Functional requirements: by specifying the required service type (Personnel) and its methods they enable functional filtering of Web Services. All of these requirements are specified semantically using ontological concepts. As such, the coupling between service requests and offers has been significantly reduced, since it's no longer established at the syntactic level.
- Non-functional (QoS) requirements: they allow the non-functional filtering of Web Services on the basis of their registry-stored quality of service contracts. These requirements define bounds on the various looked-for QoS properties.

1.4.3.2 Details on Step 4

Step 4 is the specification of non-functional preferences by the user. These preferences define the relative importance attached to QoS dimensions and values in order to be able to select, at runtime, the "optimal" service(s) for each invocation.

In (Figure 58), preferences are defined in an informal way. In fact, they are compiled in a formalism derived from CP-Net (Boutilier, Brafman et al. 2004) named LCP-Net (Châtel, Truck et al. 2008) as seen in (Figure 59). The semantic of this particular model is defined in the following:

- The "Fire resistance" QoS dimensions is symbolized by node R, and the "Physical strain" dimension by node P.

- The independent preference of a high fire resistance is expressed in the table associated to node R. RL , RM et RH correspond to the “Low Resistance”, “Medium Resistance” and “High Resistance” linguistic terms semantically defined by fuzzy subsets of the fire resistance value domain.
- The same fuzzyfication principle is used for the preference on Physical strain.
- The preference domain is also partitioned because it simplifies greatly its expression for the user. In both tables, the linguistic terms Low, Average and High designate preference degrees given by the user to each domain value.
- The i-arc between nodes R and P allows expressing the relative preference between the two QoS dimensions. It states formally that choosing a service with a higher fire resistance is more important than trying to optimize its physical strain.

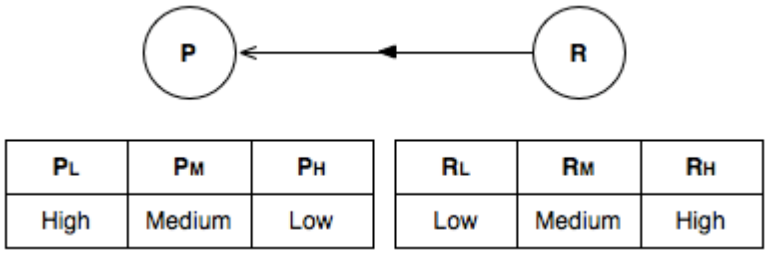

Figure 60 - LCP-Net Preference Model

1.5 Using Annotation to Set Security Preferences

While setting opened organisations as a service-based one, the major point consists in organising a common strategy, taking into account the requirements of each entity. Consequently, modelling approaches should supply semantic elements to describe security components, so that “security tags” can be associated to information to identify their security level (black/grey/ white), to define secured exchange channels, authorisation requirements,… to take into account the different security constraints while specifying a service. For this purpose, we propose to consider different communities and to identify security sensitivity for both data and operation included in a service.

The method we propose consists in 3 steps based on the basic security services:

First, the confidentiality service aims to protect data both in the information system and in the communication system. Organising data confidentiality involves defining access rights depending on the user authentication and using cryptographic services to protect data while they are stored and exchanged. This leads to white, grey and black data organisation.

Then, the integrity service must protect the system against any attack aiming at modifying data. This involves using electronic signature to authenticate both the user and the content as well as using antivirus or intrusion detection systems to protect the information from malicious modifications.

At last, the availability service has to maintain information system services operational and available. This service is related to the QoS parameters and involves also infrastructure oriented components such as firewalls or intrusion detection systems, whereas replicated architectures can answer to high-availability constraints.

Building a common security strategy involves to orchestrate conveniently the different services required by the different entities. To reach this goal, we propose few rules, related to the information "value" (i.e. white, grey or black) to combine the security components:

- **Rule #1:** « white » data will not need extra checks, « grey » data will be checked according to access authorisations (identification and authentication) whereas « black » data are related to checking processes to verify that they fit access rights, confidentiality and integrity policies. These services involve identification and non-repudiation functions, processes activities will be logged and sensitive information will be encrypted.
- **Rule #2:** exchanges between elements belonging to a sensitive category (i.e. grey or black) will be secured (i.e. these exchanges will have to be authorized, to remain confidential, and to keep their integrity).
- **Rule #3:** we will consider that enterprise internal processes are black data by default.

This global policy is then super-imposed to the service model by defining new symbols (see e (Figure 60) (Figure 61), and (Figure 62) show examples of security annotation for use case "Firefighter"). First, the different security levels (white / grey / black) are used to tag the different objects (data and processes), this allows verifying the global consistency of the security requirements (for example, a process can not be "white" while processing black data…). Proceses having high QoP garanty are considered as black, while processes having low level of QoP garanty are considered as wihte.

(Figure 63) shows a use case in which processes and data are annotated according to the aforementioned annotation; for clarity we will symbolize white objects (being data or processes) by"+", while grey objects by "-" and black objects by "#" . From this figure we can note for instance that "Collect Media Info." Process has a low level QoP (white) thus it can only process the non-sensitive data (white) while attribution process has a high QoP and thus it can process the high sensitive data (black).

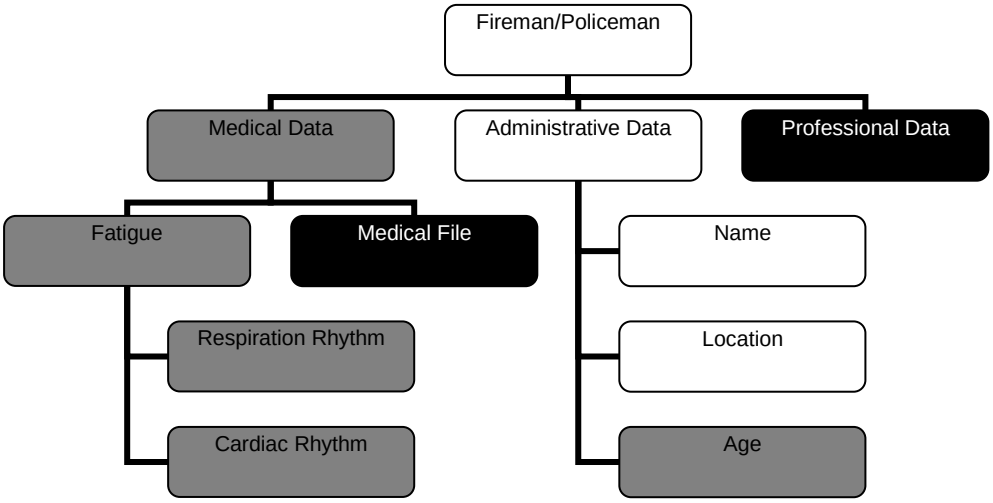

Figure 61 Data Annotation for Fireman and Policeman Personal Data

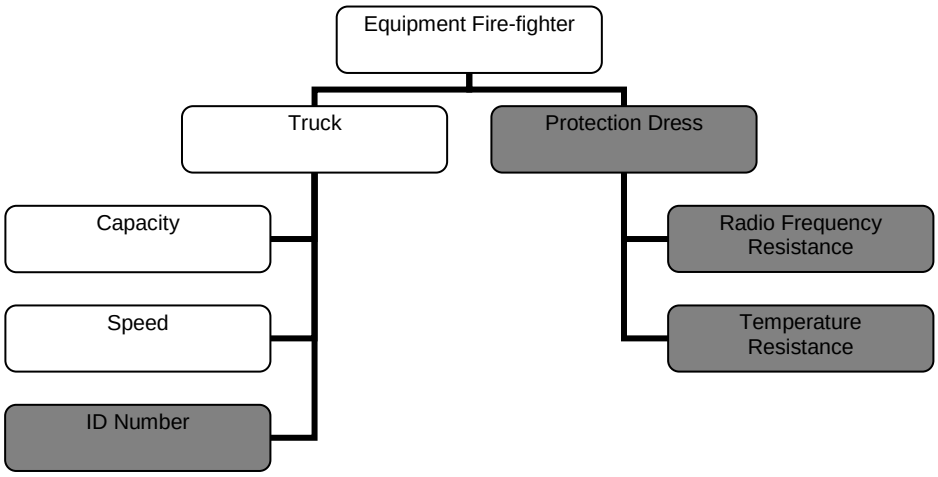

Figure 62 Data Annotation for Fire-fighter Data

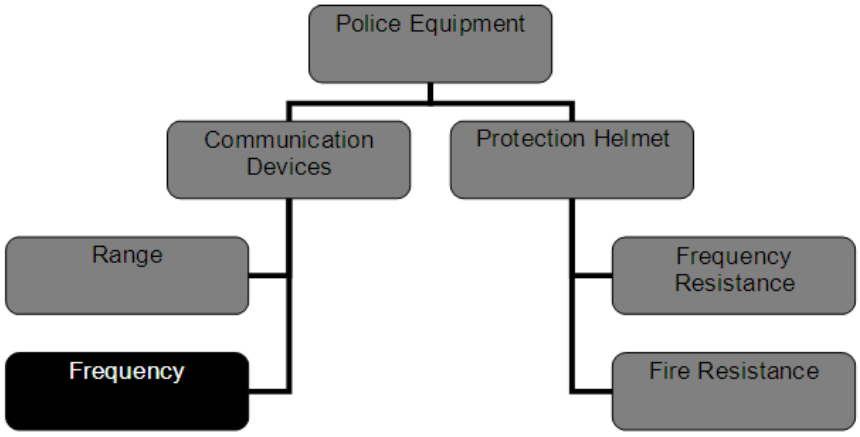

Figure 63 Data Annotation for Policeman Equipment

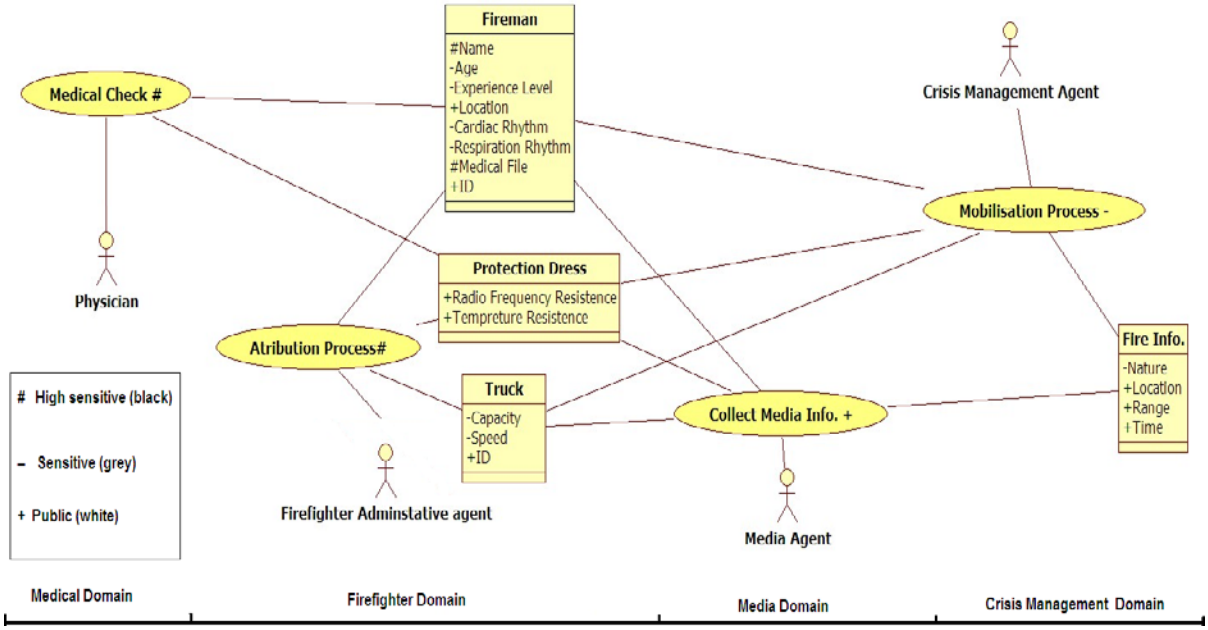

Figure 64 Example Process and Data Annotation

1.6 Scenario 3: Process Execution during Fire-fighting

1.6.1 Context

This scenario illustrates the following features of our framework through the orchestration of fire-fighters and policemen Web services at runtime:

- Functional filtering of services when entering process.
- Non-functional filtering of services when entering process.
- Non-functional selection of optimal service(s) at runtime using instantaneous QoS values of monitored services and user preferences.

We should mention that in order to retain concision, the focus will be on the execution of a specific task of the previously established fire-fighting process: the “Prevent fire propagation” service invocation step.

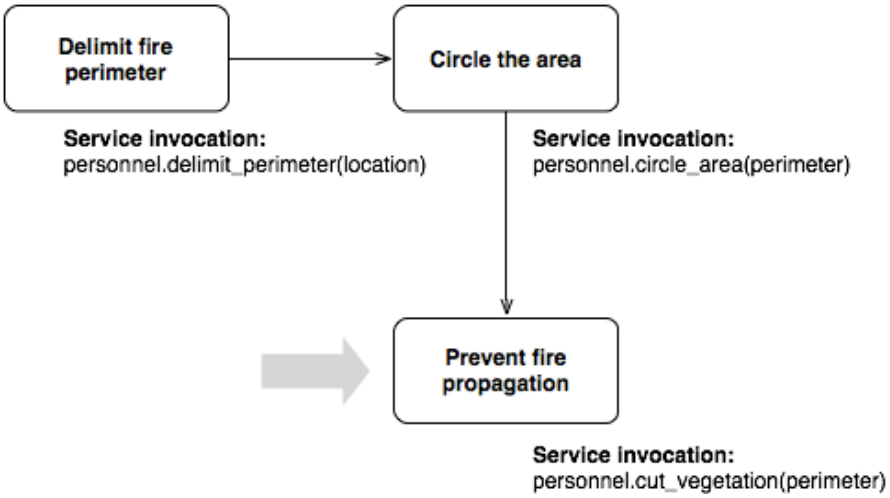

Figure 65 - Prevent Fire Propagation

1.6.2 Sub Use Case 3.1: Functional Filtering of Services

1.6.2.1Description

Functional filtering is initiated right from the beginning of the fire-fighting process execution, before entering the “Prevent fire propagation” step, in order to prevent impairing slowdowns at invocation time.

Functional requirements are gathered and services filtered from the available registry, asynchronously to the execution of the process. As such, when reaching the specific service invocation we are interested in, a local list of pre-filtered services should already be available. If this list is empty, or if no service is reachable, a fallback call to the registry is made in a synchronous fashion (i.e process execution is halted as long as no service can be reached or cancelled if there is no matching service) in order to reiterate service filtering.

Figure 65 presents some of the ontological concepts that are used in the Web process’ requirements. The Personnel concept being a super-concept of Fire Fighter and Policeman, it allows designating both in a unified fashion. It has been used in the process functional requirements because both can complete most of the tasks without distinction.

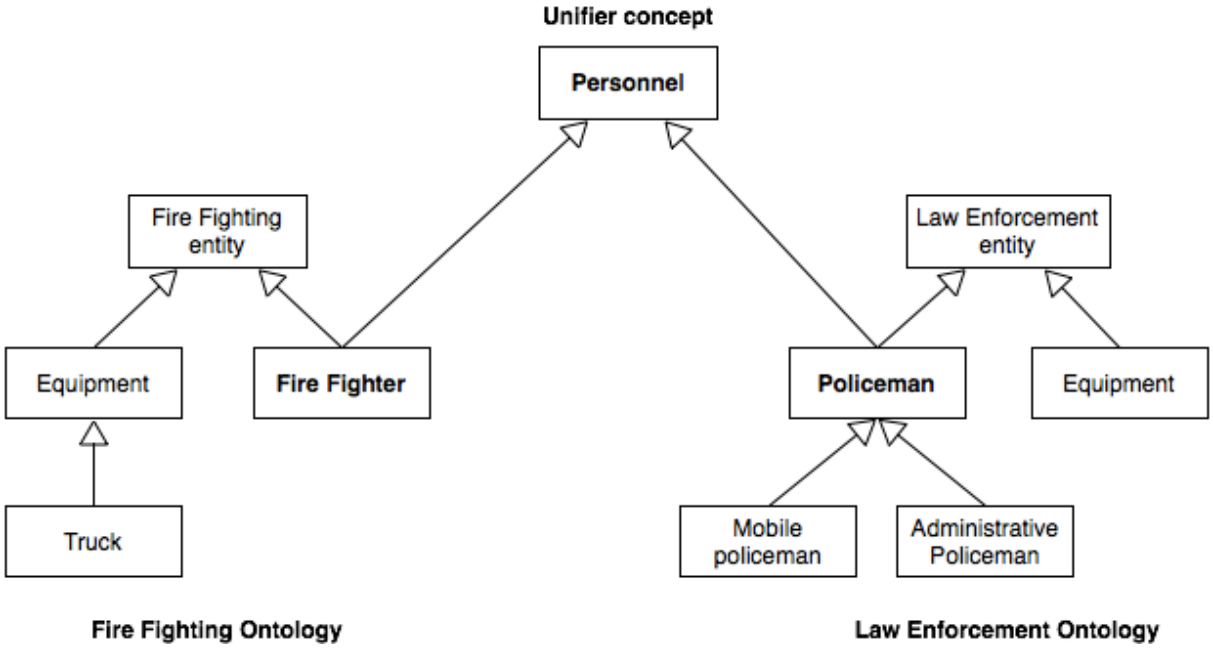

Figure 66 – Ontological Excerpt of Fire Fighting and Law Enforcement Domains

The following table shows the functional service offers available in the registry when initiating functional filtering of services. It presents a high-level representation of the stored service offers: methods and service categories are described using ontological concepts (terms in italic are ontology classes).

S1: functional offer	**S2: functional offer**
Service category: *fire-fighter* Methods: *go_to(location),* *delimit_perimeter(location),* *circle_area(perimeter),* *cut_vegetation(perimeter)*	Service category: *policeman* Methods: *go_to(location),* *delimit_perimeter(location),* *circle_area(perimeter),* *cut_vegetation(perimeter)* *serve(),* *protect()*
S3: functional offer	**S4: functional offer**
Service category: *fire-fighter* Methods: *go_to(location),* *delimit_perimeter(location),* *circle_area(perimeter),* *cut_vegetation(perimeter)*	Service category: *administrative policeman* Methods: *write_report(),* *write_fine()*
S5: functional offer	

Service category: *fire-fighter* Methods: *go_to(location),* *delimit_perimeter(location),* *circle_area(perimeter),* *cut_vegetation(perimeter)*

1.6.2.2 Steps

Step	Actors	Event
1	Fire-fighter S1 Policeman S2 Fire-fighter S3 Policeman S4 Fire-fighter S5	Functional filtering of the available services. Due to the requirements specified at process level, S4 is removed from the list of matching services.

1.6.2.3 Details on Step 1

Due to the functional requirements specified at process level, S4 is removed from the list of matching services and the other services are preserved.

- Services S1, S2, S3 and S5: categorized by a subclass of the one used in the functional requirements and possess the same semantic annotations for their methods than in the requirements: not filtered
- Service S4: categorized by a subclass of the one used in the requirements but does not possesses the same semantic annotations for its methods than in the requirements: filtered

1.6.3 Sub use case 3.2: Non-Functional Filtering of Services

1.6.3.1 Description

Non-Functional filtering of services is initiated in the footsteps of functional filtering also asynchronously to process execution, before Web Service selection and effective invocation. As such it will filter the set of services provided by the previous functional filter.

This phase is grounded on the non-functional constraints defined in the Web process and the various non-functional contracts statically provided by the services and stored inside the registry. Its goal is to make sure, before invocation time, that the services will be able to provide a Quality of Service, including Quality of Protection, located inside the required bounds, but not to enforce it.

1.6.3.2 Details on Step 1

To express service level security preference we need to join a security policy to the service (WSDL) document. Policy references could be made via policyreference element as defined in security policy specifications or could be embedded in the WSDL document using policy expression (*wsdl:definitions*). Nevertheless, as we saw so far in the state of the art, WS-SecurityPolicy falls short in providing details for a full implementation of security. To fill this gap we propose to extend WSDL by the description of QoP characteristics. The extension is carried out using our semantic security annotation exposed in (8.2) in this way the WSDL document becomes QoP-enabled WSDL. The obtained QoP-WSDL can be further used to filter services according to the level of protection they offer and to extend service level agreements (SLA) by QoP-oriented characteristics. Providers then publish QoP-enabled SLA templates along with WSDL document of their services. The service client searches and finds the appropriate service, retrieving the QoP enabled WSDL and SLA for the service. These will be further used to formalise and finalize the SLA, for instance by a negotiation phase.

(Table 5) shows the non-functional characteristics offered by services extended with QoP constraints (described using ontological concepts (terms in red are ontology classes) :

- Services S1, and S5: the offered contracts exhibit equivalent or stronger QoS as well as QoP constraints than the one required at the process level (according to the sensitivity of the information transmitted). These services are not filtered.
- S2 in the other hand has non-secure transport protocol as well as a weak authentication mechanism and thus weaker QoP constraints than those required (according to the sensitivity of the information transmitted) this service will be filtered.
- Service S3: although this service exhibits a suitable quality of protection level, the offered contract exhibits a weaker physical strain constraint than the one required at the process level, consequently, this service will be filtered.

S1: non-functional contract	**S2: non-functional contract**
60°C <= Fire resistance <= 90°C	60°C <= Fire resistance <= 90°C
Physical strain <= 70 PPM[1]	20 <= physical strain <= 70 PPM
2Ghz < Radio frequency <= 2.5 Ghz	2Ghz < Radio frequency <= 3Ghz
Authentication: Kerberos Protocol	**Authentication: Login**
TransportProtocol: HTTPS	**TransportProtocol: HTTP non-secured**

S3: non-functional contract	S5: non-functional contract
60°C <= Fire resistance <= 90°C Physical strain <= 100 PPM 2Ghz < Radio frequency <= 2.5 Ghz **Authentication: Kerberos Protocol** **TransportProtocol: HTTPS**	60°C <= Fire resistance <= 90°C physical strain <= 70 PPM 2Ghz < Radio frequency <= 2.5 Ghz **Authentication: SAML** **TransportProtocol: HTTPS**

Table 5 Non-functional Characteristics Offered by Services, Taken from Use case

2 WSDL of User Registry Service Interface

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:ax233="http://principal.component/xsd" xmlns:ns1="http://org.apache.axis2/xsd"
    xmlns:ax231="http://component/xsd"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xsd="http://registry.user.ws"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    targetNamespace="http://registry.user.ws">
- <wsdl:types>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://principal.component/xsd">
- <xs:complexType name="UserPrincipal">
- <xs:sequence>
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="pID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="password" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="rolePrincipal" nillable="true" type="ax233:Role" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="Role">
- <xs:sequence>
  <xs:element minOccurs="0" name="communityPrincipal" nillable="true" type="ax233:Community" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="roleName" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="Community">
- <xs:sequence>
  <xs:element minOccurs="0" name="communityName" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
- <xs:schema xmlns:ax232="http://component/xsd" xmlns:ax234="http://principal.component/xsd"
    attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://registry.user.ws">
  <xs:import namespace="http://component/xsd" />
  <xs:import namespace="http://principal.component/xsd" />
- <xs:element name="GetUserByID">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetUserByIDResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="ax231:UserInfo" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAllUserPrincipalsByID">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAllUserPrincipalsByIDResponse">
- <xs:complexType>
```

```
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true"
      type="ax234:UserPrincipal" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:schema>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://component/xsd">
- <xs:complexType name="UserInfo">
- <xs:sequence>
  <xs:element minOccurs="0" name="domain" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element maxOccurs="unbounded" minOccurs="0" name="password" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
    </wsdl:types>
- <wsdl:message name="GetUserByIDRequest">
  <wsdl:part name="parameters" element="xsd:GetUserByID" />
    </wsdl:message>
- <wsdl:message name="GetUserByIDResponse">
  <wsdl:part name="parameters" element="xsd:GetUserByIDResponse" />
    </wsdl:message>
- <wsdl:message name="GetAllUserPrincipalsByIDRequest">
  <wsdl:part name="parameters" element="xsd:GetAllUserPrincipalsByID" />
    </wsdl:message>
- <wsdl:message name="GetAllUserPrincipalsByIDResponse">
  <wsdl:part name="parameters" element="xsd:GetAllUserPrincipalsByIDResponse" />
    </wsdl:message>
- <wsdl:portType name="UserRegistryPortType">
- <wsdl:operation name="GetUserByID">
  <wsdl:input message="xsd:GetUserByIDRequest" wsaw:Action="urn:GetUserByID" />
  <wsdl:output message="xsd:GetUserByIDResponse" wsaw:Action="urn:GetUserByIDResponse" />
    </wsdl:operation>
- <wsdl:operation name="GetAllUserPrincipalsByID">
  <wsdl:input message="xsd:GetAllUserPrincipalsByIDRequest"
      wsaw:Action="urn:GetAllUserPrincipalsByID" />
  <wsdl:output message="xsd:GetAllUserPrincipalsByIDResponse"
      wsaw:Action="urn:GetAllUserPrincipalsByIDResponse" />
    </wsdl:operation>
    </wsdl:portType>
- <wsdl:binding name="UserRegistrySoap11Binding" type="xsd:UserRegistryPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="GetUserByID">
  <soap:operation soapAction="urn:GetUserByID" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetAllUserPrincipalsByID">
  <soap:operation soapAction="urn:GetAllUserPrincipalsByID" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="UserRegistrySoap12Binding" type="xsd:UserRegistryPortType">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="GetUserByID">
```

```
<soap12:operation soapAction="urn:GetUserByID" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="GetAllUserPrincipalsByID">
<soap12:operation soapAction="urn:GetAllUserPrincipalsByID" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:binding name="UserRegistryHttpBinding" type="xsd:UserRegistryPortType">
<http:binding verb="POST" />
- <wsdl:operation name="GetUserByID">
<http:operation location="UserRegistry/GetUserByID" />
- <wsdl:input>
<mime:content type="text/xml" part="GetUserByID" />
</wsdl:input>
- <wsdl:output>
<mime:content type="text/xml" part="GetUserByID" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="GetAllUserPrincipalsByID">
<http:operation location="UserRegistry/GetAllUserPrincipalsByID" />
- <wsdl:input>
<mime:content type="text/xml" part="GetAllUserPrincipalsByID" />
</wsdl:input>
- <wsdl:output>
<mime:content type="text/xml" part="GetAllUserPrincipalsByID" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="UserRegistry">
- <wsdl:port name="UserRegistryHttpSoap11Endpoint" binding="xsd:UserRegistrySoap11Binding">
<soap:address location="http://localhost:8080/axis2/services/UserRegistry" />
</wsdl:port>
- <wsdl:port name="UserRegistryHttpSoap12Endpoint" binding="xsd:UserRegistrySoap12Binding">
<soap12:address location="http://localhost:8080/axis2/services/UserRegistry" />
</wsdl:port>
- <wsdl:port name="UserRegistryHttpEndpoint" binding="xsd:UserRegistryHttpBinding">
<http:address location="http://localhost:8080/axis2/services/UserRegistry" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

3 WSDL of Attributes Manager Interface

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:ns1="http://org.apache.axis2/xsd"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:ax237="http://principal.component/xsd" xmlns:ax235="http://component/xsd"
    xmlns:xsd="http://attribute.ws" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" targetNamespace="http://attribute.ws">
- <wsdl:types>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://principal.component/xsd">
- <xs:complexType name="UserPrincipal">
- <xs:sequence>
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="pID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="password" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="rolePrincipal" nillable="true" type="ax237:Role" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="Role">
- <xs:sequence>
  <xs:element minOccurs="0" name="communityPrincipal" nillable="true" type="ax237:Community" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="roleName" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="Community">
- <xs:sequence>
  <xs:element minOccurs="0" name="communityName" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://component/xsd">
- <xs:complexType name="UserToken">
- <xs:sequence>
  <xs:element minOccurs="0" name="domain" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="tag" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
- <xs:schema xmlns:ax236="http://component/xsd" xmlns:ax238="http://principal.component/xsd"
    attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://attribute.ws">
  <xs:import namespace="http://component/xsd" />
  <xs:import namespace="http://principal.component/xsd" />
- <xs:element name="GetAttributeByCommunity">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="token" nillable="true" type="ax235:UserToken" />
  <xs:element minOccurs="0" name="communityName" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAttributeByCommunityResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true"
    type="ax237:UserPrincipal" />
```

```
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAttributeByRole">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="token" nillable="true" type="ax235:UserToken" />
  <xs:element minOccurs="0" name="roleName" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAttributeByRoleResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true"
      type="ax237:UserPrincipal" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:schema>
    </wsdl:types>
- <wsdl:message name="GetAttributeByCommunityRequest">
  <wsdl:part name="parameters" element="xsd:GetAttributeByCommunity" />
    </wsdl:message>
- <wsdl:message name="GetAttributeByCommunityResponse">
  <wsdl:part name="parameters" element="xsd:GetAttributeByCommunityResponse" />
    </wsdl:message>
- <wsdl:message name="GetAttributeByRoleRequest">
  <wsdl:part name="parameters" element="xsd:GetAttributeByRole" />
    </wsdl:message>
- <wsdl:message name="GetAttributeByRoleResponse">
  <wsdl:part name="parameters" element="xsd:GetAttributeByRoleResponse" />
    </wsdl:message>
- <wsdl:portType name="AttributeManagerPortType">
- <wsdl:operation name="GetAttributeByCommunity">
  <wsdl:input message="xsd:GetAttributeByCommunityRequest"
      wsaw:Action="urn:GetAttributeByCommunity" />
  <wsdl:output message="xsd:GetAttributeByCommunityResponse"
      wsaw:Action="urn:GetAttributeByCommunityResponse" />
    </wsdl:operation>
- <wsdl:operation name="GetAttributeByRole">
  <wsdl:input message="xsd:GetAttributeByRoleRequest" wsaw:Action="urn:GetAttributeByRole" />
  <wsdl:output message="xsd:GetAttributeByRoleResponse"
      wsaw:Action="urn:GetAttributeByRoleResponse" />
    </wsdl:operation>
    </wsdl:portType>
- <wsdl:binding name="AttributeManagerSoap11Binding" type="xsd:AttributeManagerPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="GetAttributeByCommunity">
  <soap:operation soapAction="urn:GetAttributeByCommunity" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetAttributeByRole">
  <soap:operation soapAction="urn:GetAttributeByRole" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
```

```
- <wsdl:binding name="AttributeManagerSoap12Binding" type="xsd:AttributeManagerPortType">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="GetAttributeByCommunity">
  <soap12:operation soapAction="urn:GetAttributeByCommunity" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetAttributeByRole">
  <soap12:operation soapAction="urn:GetAttributeByRole" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="AttributeManagerHttpBinding" type="xsd:AttributeManagerPortType">
  <http:binding verb="POST" />
- <wsdl:operation name="GetAttributeByCommunity">
  <http:operation location="AttributeManager/GetAttributeByCommunity" />
- <wsdl:input>
  <mime:content type="text/xml" part="GetAttributeByCommunity" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="GetAttributeByCommunity" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetAttributeByRole">
  <http:operation location="AttributeManager/GetAttributeByRole" />
- <wsdl:input>
  <mime:content type="text/xml" part="GetAttributeByRole" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="GetAttributeByRole" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:service name="AttributeManager">
- <wsdl:port name="AttributeManagerHttpSoap11Endpoint"
      binding="xsd:AttributeManagerSoap11Binding">
  <soap:address location="http://localhost:8080/axis2/services/AttributeManager" />
    </wsdl:port>
- <wsdl:port name="AttributeManagerHttpSoap12Endpoint"
      binding="xsd:AttributeManagerSoap12Binding">
  <soap12:address location="http://localhost:8080/axis2/services/AttributeManager" />
    </wsdl:port>
- <wsdl:port name="AttributeManagerHttpEndpoint" binding="xsd:AttributeManagerHttpBinding">
  <http:address location="http://localhost:8080/axis2/services/AttributeManager" />
    </wsdl:port>
    </wsdl:service>
    </wsdl:definitions>
```

4 WSDL of Credential Manager Service Interface

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:ns1="http://org.apache.axis2/xsd"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xsd="http://credential.ws"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ax241="http://component/xsd"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:ax243="http://principal.component/xsd" targetNamespace="http://credential.ws">
- <wsdl:types>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://principal.component/xsd">
- <xs:complexType name="UserPrincipal">
- <xs:sequence>
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="pID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="password" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="rolePrincipal" nillable="true" type="ax243:Role" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="Role">
- <xs:sequence>
  <xs:element minOccurs="0" name="communityPrincipal" nillable="true" type="ax243:Community" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="roleName" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="Community">
- <xs:sequence>
  <xs:element minOccurs="0" name="communityName" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
- <xs:schema xmlns:ax242="http://component/xsd" attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://credential.ws">
  <xs:import namespace="http://component/xsd" />
- <xs:element name="GetCredential">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="serviceID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="token" nillable="true" type="ax241:UserToken" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetCredentialResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="ax241:CredentialManagerResponse" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:schema>
- <xs:schema xmlns:ax244="http://principal.component/xsd" attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://component/xsd">
  <xs:import namespace="http://principal.component/xsd" />
- <xs:complexType name="UserToken">
- <xs:sequence>
  <xs:element minOccurs="0" name="domain" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="tag" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
```

```
- <xs:complexType name="CredentialManagerResponse">
- <xs:sequence>
  <xs:element minOccurs="0" name="credentialToken" nillable="true" type="ax241:CredentialToken" />
  <xs:element minOccurs="0" name="userPrincipal" nillable="true" type="ax243:UserPrincipal" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="CredentialToken">
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="authorizedOperations" nillable="true"
      type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
    </wsdl:types>
- <wsdl:message name="GetCredentialRequest">
  <wsdl:part name="parameters" element="xsd:GetCredential" />
    </wsdl:message>
- <wsdl:message name="GetCredentialResponse">
  <wsdl:part name="parameters" element="xsd:GetCredentialResponse" />
    </wsdl:message>
- <wsdl:portType name="CredentialManagerPortType">
- <wsdl:operation name="GetCredential">
  <wsdl:input message="xsd:GetCredentialRequest" wsaw:Action="urn:GetCredential" />
  <wsdl:output message="xsd:GetCredentialResponse" wsaw:Action="urn:GetCredentialResponse" />
    </wsdl:operation>
    </wsdl:portType>
- <wsdl:binding name="CredentialManagerSoap11Binding" type="xsd:CredentialManagerPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="GetCredential">
  <soap:operation soapAction="urn:GetCredential" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="CredentialManagerSoap12Binding" type="xsd:CredentialManagerPortType">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="GetCredential">
  <soap12:operation soapAction="urn:GetCredential" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="CredentialManagerHttpBinding" type="xsd:CredentialManagerPortType">
  <http:binding verb="POST" />
- <wsdl:operation name="GetCredential">
  <http:operation location="CredentialManager/GetCredential" />
- <wsdl:input>
  <mime:content type="text/xml" part="GetCredential" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="GetCredential" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:service name="CredentialManager">
- <wsdl:port name="CredentialManagerHttpSoap11Endpoint"
      binding="xsd:CredentialManagerSoap11Binding">
  <soap:address location="http://localhost:8080/axis2/services/CredentialManager" />
    </wsdl:port>
```

```
- <wsdl:port name="CredentialManagerHttpSoap12Endpoint"
      binding="xsd:CredentialManagerSoap12Binding">
  <soap12:address location="http://localhost:8080/axis2/services/CredentialManager" />
    </wsdl:port>
- <wsdl:port name="CredentialManagerHttpEndpoint" binding="xsd:CredentialManagerHttpBinding">
  <http:address location="http://localhost:8080/axis2/services/CredentialManager" />
    </wsdl:port>
    </wsdl:service>
    </wsdl:definitions>
```

5 WSDL of Domains Manager Service

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:ns1="http://org.apache.axis2/xsd"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xsd="http://registry.domain.ws"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    targetNamespace="http://registry.domain.ws">
- <wsdl:types>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://registry.domain.ws">
- <xs:element name="FindDistantFederationManagerURIByDomain">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="domain" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="FindDistantFederationManagerURIByDomainResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="FindAllDistantFederationManager">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="suffix" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="FindAllDistantFederationManagerResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:schema>
    </wsdl:types>
- <wsdl:message name="FindAllDistantFederationManagerRequest">
  <wsdl:part name="parameters" element="xsd:FindAllDistantFederationManager" />
    </wsdl:message>
- <wsdl:message name="FindAllDistantFederationManagerResponse">
  <wsdl:part name="parameters" element="xsd:FindAllDistantFederationManagerResponse" />
    </wsdl:message>
- <wsdl:message name="FindDistantFederationManagerURIByDomainRequest">
  <wsdl:part name="parameters" element="xsd:FindDistantFederationManagerURIByDomain" />
    </wsdl:message>
- <wsdl:message name="FindDistantFederationManagerURIByDomainResponse">
  <wsdl:part name="parameters" element="xsd:FindDistantFederationManagerURIByDomainResponse" />
    </wsdl:message>
- <wsdl:portType name="DomainRegistryPortType">
- <wsdl:operation name="FindAllDistantFederationManager">
  <wsdl:input message="xsd:FindAllDistantFederationManagerRequest"
      wsaw:Action="urn:FindAllDistantFederationManager" />
  <wsdl:output message="xsd:FindAllDistantFederationManagerResponse"
      wsaw:Action="urn:FindAllDistantFederationManagerResponse" />
    </wsdl:operation>
- <wsdl:operation name="FindDistantFederationManagerURIByDomain">
```

```
    <wsdl:input message="xsd:FindDistantFederationManagerURIByDomainRequest"
        wsaw:Action="urn:FindDistantFederationManagerURIByDomain" />
    <wsdl:output message="xsd:FindDistantFederationManagerURIByDomainResponse"
        wsaw:Action="urn:FindDistantFederationManagerURIByDomainResponse" />
      </wsdl:operation>
      </wsdl:portType>
- <wsdl:binding name="DomainRegistrySoap11Binding" type="xsd:DomainRegistryPortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="FindAllDistantFederationManager">
    <soap:operation soapAction="urn:FindAllDistantFederationManager" style="document" />
- <wsdl:input>
    <soap:body use="literal" />
      </wsdl:input>
- <wsdl:output>
    <soap:body use="literal" />
      </wsdl:output>
      </wsdl:operation>
- <wsdl:operation name="FindDistantFederationManagerURIByDomain">
    <soap:operation soapAction="urn:FindDistantFederationManagerURIByDomain" style="document" />
- <wsdl:input>
    <soap:body use="literal" />
      </wsdl:input>
- <wsdl:output>
    <soap:body use="literal" />
      </wsdl:output>
      </wsdl:operation>
      </wsdl:binding>
- <wsdl:binding name="DomainRegistrySoap12Binding" type="xsd:DomainRegistryPortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="FindAllDistantFederationManager">
    <soap12:operation soapAction="urn:FindAllDistantFederationManager" style="document" />
- <wsdl:input>
    <soap12:body use="literal" />
      </wsdl:input>
- <wsdl:output>
    <soap12:body use="literal" />
      </wsdl:output>
      </wsdl:operation>
- <wsdl:operation name="FindDistantFederationManagerURIByDomain">
    <soap12:operation soapAction="urn:FindDistantFederationManagerURIByDomain" style="document" />
- <wsdl:input>
    <soap12:body use="literal" />
      </wsdl:input>
- <wsdl:output>
    <soap12:body use="literal" />
      </wsdl:output>
      </wsdl:operation>
      </wsdl:binding>
- <wsdl:binding name="DomainRegistryHttpBinding" type="xsd:DomainRegistryPortType">
    <http:binding verb="POST" />
- <wsdl:operation name="FindAllDistantFederationManager">
    <http:operation location="DomainRegistry/FindAllDistantFederationManager" />
- <wsdl:input>
    <mime:content type="text/xml" part="FindAllDistantFederationManager" />
      </wsdl:input>
- <wsdl:output>
    <mime:content type="text/xml" part="FindAllDistantFederationManager" />
      </wsdl:output>
      </wsdl:operation>
- <wsdl:operation name="FindDistantFederationManagerURIByDomain">
    <http:operation location="DomainRegistry/FindDistantFederationManagerURIByDomain" />
- <wsdl:input>
    <mime:content type="text/xml" part="FindDistantFederationManagerURIByDomain" />
      </wsdl:input>
- <wsdl:output>
    <mime:content type="text/xml" part="FindDistantFederationManagerURIByDomain" />
      </wsdl:output>
```

```
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:service name="DomainRegistry">
- <wsdl:port name="DomainRegistryHttpSoap11Endpoint" binding="xsd:DomainRegistrySoap11Binding">
  <soap:address location="http://localhost:8080/axis2/services/DomainRegistry" />
    </wsdl:port>
- <wsdl:port name="DomainRegistryHttpSoap12Endpoint" binding="xsd:DomainRegistrySoap12Binding">
  <soap12:address location="http://localhost:8080/axis2/services/DomainRegistry" />
    </wsdl:port>
- <wsdl:port name="DomainRegistryHttpEndpoint" binding="xsd:DomainRegistryHttpBinding">
  <http:address location="http://localhost:8080/axis2/services/DomainRegistry" />
    </wsdl:port>
    </wsdl:service>
    </wsdl:definitions>
```

6 WSDL of Security Policy Service Interface

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:ns1="http://org.apache.axis2/xsd"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xsd="http://registry.service.ws"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    targetNamespace="http://registry.service.ws">
- <wsdl:types>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://registry.service.ws">
- <xs:element name="GetSecurityPolicyCommunity">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="serviceID" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetSecurityPolicyCommunityResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAuthorizedOperationsByCommunity">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="serviceID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="communityName" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAuthorizedOperationsByCommunityResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAuthorizedOperationsByPID">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="serviceID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="PID" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAuthorizedOperationsByPIDResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetAuthorizedOperationsByRole">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="serviceID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="roleName" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
```

```
    </xs:element>
- <xs:element name="GetAuthorizedOperationsByRoleResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetSecurityPolicyRole">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="serviceID" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="GetSecurityPolicyRoleResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:schema>
    </wsdl:types>
- <wsdl:message name="GetAuthorizedOperationsByRoleRequest">
  <wsdl:part name="parameters" element="xsd:GetAuthorizedOperationsByRole" />
    </wsdl:message>
- <wsdl:message name="GetAuthorizedOperationsByRoleResponse">
  <wsdl:part name="parameters" element="xsd:GetAuthorizedOperationsByRoleResponse" />
    </wsdl:message>
- <wsdl:message name="GetAuthorizedOperationsByCommunityRequest">
  <wsdl:part name="parameters" element="xsd:GetAuthorizedOperationsByCommunity" />
    </wsdl:message>
- <wsdl:message name="GetAuthorizedOperationsByCommunityResponse">
  <wsdl:part name="parameters" element="xsd:GetAuthorizedOperationsByCommunityResponse" />
    </wsdl:message>
- <wsdl:message name="GetSecurityPolicyRoleRequest">
  <wsdl:part name="parameters" element="xsd:GetSecurityPolicyRole" />
    </wsdl:message>
- <wsdl:message name="GetSecurityPolicyRoleResponse">
  <wsdl:part name="parameters" element="xsd:GetSecurityPolicyRoleResponse" />
    </wsdl:message>
- <wsdl:message name="GetAuthorizedOperationsByPIDRequest">
  <wsdl:part name="parameters" element="xsd:GetAuthorizedOperationsByPID" />
    </wsdl:message>
- <wsdl:message name="GetAuthorizedOperationsByPIDResponse">
  <wsdl:part name="parameters" element="xsd:GetAuthorizedOperationsByPIDResponse" />
    </wsdl:message>
- <wsdl:message name="GetSecurityPolicyCommunityRequest">
  <wsdl:part name="parameters" element="xsd:GetSecurityPolicyCommunity" />
    </wsdl:message>
- <wsdl:message name="GetSecurityPolicyCommunityResponse">
  <wsdl:part name="parameters" element="xsd:GetSecurityPolicyCommunityResponse" />
    </wsdl:message>
- <wsdl:portType name="SecurityPolicyPortType">
- <wsdl:operation name="GetAuthorizedOperationsByRole">
  <wsdl:input message="xsd:GetAuthorizedOperationsByRoleRequest"
      wsaw:Action="urn:GetAuthorizedOperationsByRole" />
  <wsdl:output message="xsd:GetAuthorizedOperationsByRoleResponse"
      wsaw:Action="urn:GetAuthorizedOperationsByRoleResponse" />
    </wsdl:operation>
- <wsdl:operation name="GetAuthorizedOperationsByCommunity">
  <wsdl:input message="xsd:GetAuthorizedOperationsByCommunityRequest"
      wsaw:Action="urn:GetAuthorizedOperationsByCommunity" />
  <wsdl:output message="xsd:GetAuthorizedOperationsByCommunityResponse"
      wsaw:Action="urn:GetAuthorizedOperationsByCommunityResponse" />
    </wsdl:operation>
```

```
- <wsdl:operation name="GetSecurityPolicyRole">
  <wsdl:input message="xsd:GetSecurityPolicyRoleRequest" wsaw:Action="urn:GetSecurityPolicyRole" />
  <wsdl:output message="xsd:GetSecurityPolicyRoleResponse"
      wsaw:Action="urn:GetSecurityPolicyRoleResponse" />
    </wsdl:operation>
- <wsdl:operation name="GetAuthorizedOperationsByPID">
  <wsdl:input message="xsd:GetAuthorizedOperationsByPIDRequest"
      wsaw:Action="urn:GetAuthorizedOperationsByPID" />
  <wsdl:output message="xsd:GetAuthorizedOperationsByPIDResponse"
      wsaw:Action="urn:GetAuthorizedOperationsByPIDResponse" />
    </wsdl:operation>
- <wsdl:operation name="GetSecurityPolicyCommunity">
  <wsdl:input message="xsd:GetSecurityPolicyCommunityRequest"
      wsaw:Action="urn:GetSecurityPolicyCommunity" />
  <wsdl:output message="xsd:GetSecurityPolicyCommunityResponse"
      wsaw:Action="urn:GetSecurityPolicyCommunityResponse" />
    </wsdl:operation>
    </wsdl:portType>
- <wsdl:binding name="SecurityPolicySoap11Binding" type="xsd:SecurityPolicyPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="GetAuthorizedOperationsByRole">
  <soap:operation soapAction="urn:GetAuthorizedOperationsByRole" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetAuthorizedOperationsByCommunity">
  <soap:operation soapAction="urn:GetAuthorizedOperationsByCommunity" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetSecurityPolicyRole">
  <soap:operation soapAction="urn:GetSecurityPolicyRole" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetAuthorizedOperationsByPID">
  <soap:operation soapAction="urn:GetAuthorizedOperationsByPID" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetSecurityPolicyCommunity">
  <soap:operation soapAction="urn:GetSecurityPolicyCommunity" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="SecurityPolicySoap12Binding" type="xsd:SecurityPolicyPortType">
```

```
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="GetAuthorizedOperationsByRole">
<soap12:operation soapAction="urn:GetAuthorizedOperationsByRole" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="GetAuthorizedOperationsByCommunity">
<soap12:operation soapAction="urn:GetAuthorizedOperationsByCommunity" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="GetSecurityPolicyRole">
<soap12:operation soapAction="urn:GetSecurityPolicyRole" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="GetAuthorizedOperationsByPID">
<soap12:operation soapAction="urn:GetAuthorizedOperationsByPID" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="GetSecurityPolicyCommunity">
<soap12:operation soapAction="urn:GetSecurityPolicyCommunity" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:binding name="SecurityPolicyHttpBinding" type="xsd:SecurityPolicyPortType">
<http:binding verb="POST" />
- <wsdl:operation name="GetAuthorizedOperationsByRole">
<http:operation location="SecurityPolicy/GetAuthorizedOperationsByRole" />
- <wsdl:input>
<mime:content type="text/xml" part="GetAuthorizedOperationsByRole" />
</wsdl:input>
- <wsdl:output>
<mime:content type="text/xml" part="GetAuthorizedOperationsByRole" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="GetAuthorizedOperationsByCommunity">
<http:operation location="SecurityPolicy/GetAuthorizedOperationsByCommunity" />
- <wsdl:input>
<mime:content type="text/xml" part="GetAuthorizedOperationsByCommunity" />
</wsdl:input>
- <wsdl:output>
<mime:content type="text/xml" part="GetAuthorizedOperationsByCommunity" />
</wsdl:output>
</wsdl:operation>
```

```
- <wsdl:operation name="GetSecurityPolicyRole">
  <http:operation location="SecurityPolicy/GetSecurityPolicyRole" />
- <wsdl:input>
  <mime:content type="text/xml" part="GetSecurityPolicyRole" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="GetSecurityPolicyRole" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetAuthorizedOperationsByPID">
  <http:operation location="SecurityPolicy/GetAuthorizedOperationsByPID" />
- <wsdl:input>
  <mime:content type="text/xml" part="GetAuthorizedOperationsByPID" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="GetAuthorizedOperationsByPID" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="GetSecurityPolicyCommunity">
  <http:operation location="SecurityPolicy/GetSecurityPolicyCommunity" />
- <wsdl:input>
  <mime:content type="text/xml" part="GetSecurityPolicyCommunity" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="GetSecurityPolicyCommunity" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:service name="SecurityPolicy">
- <wsdl:port name="SecurityPolicyHttpSoap11Endpoint" binding="xsd:SecurityPolicySoap11Binding">
  <soap:address location="http://localhost:8080/axis2/services/SecurityPolicy" />
    </wsdl:port>
- <wsdl:port name="SecurityPolicyHttpSoap12Endpoint" binding="xsd:SecurityPolicySoap12Binding">
  <soap12:address location="http://localhost:8080/axis2/services/SecurityPolicy" />
    </wsdl:port>
- <wsdl:port name="SecurityPolicyHttpEndpoint" binding="xsd:SecurityPolicyHttpBinding">
  <http:address location="http://localhost:8080/axis2/services/SecurityPolicy" />
    </wsdl:port>
    </wsdl:service>
    </wsdl:definitions>
```

7 WSDL of User Authentication Service

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ax29="http://component/xsd"
    xmlns:ns1="http://org.apache.axis2/xsd"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xsd="http://authentication.ws"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    targetNamespace="http://authentication.ws">
- <wsdl:types>
- <xs:schema xmlns:ax210="http://component/xsd" attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://authentication.ws">
  <xs:import namespace="http://component/xsd" />
- <xs:element name="AuthenticateUser">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="user" nillable="true" type="ax210:UserReq" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="AuthenticateUserResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="ax210:UserToken" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="Logout">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="tokenID" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:schema>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://component/xsd">
- <xs:complexType name="UserReq">
- <xs:sequence>
  <xs:element minOccurs="0" name="domain" nillable="true" type="xs:string" />
  <xs:element maxOccurs="unbounded" minOccurs="0" name="password" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="UserToken">
- <xs:sequence>
  <xs:element minOccurs="0" name="domain" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="tag" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
    </wsdl:types>
- <wsdl:message name="AuthenticateUserRequest">
  <wsdl:part name="parameters" element="xsd:AuthenticateUser" />
    </wsdl:message>
- <wsdl:message name="AuthenticateUserResponse">
  <wsdl:part name="parameters" element="xsd:AuthenticateUserResponse" />
    </wsdl:message>
- <wsdl:message name="LogoutRequest">
  <wsdl:part name="parameters" element="xsd:Logout" />
    </wsdl:message>
```

```
- <wsdl:portType name="UserAuthenticationPortType">
- <wsdl:operation name="AuthenticateUser">
  <wsdl:input message="xsd:AuthenticateUserRequest" wsaw:Action="urn:AuthenticateUser" />
  <wsdl:output message="xsd:AuthenticateUserResponse" wsaw:Action="urn:AuthenticateUserResponse" />
    </wsdl:operation>
- <wsdl:operation name="Logout">
  <wsdl:input message="xsd:LogoutRequest" wsaw:Action="urn:Logout" />
    </wsdl:operation>
    </wsdl:portType>
- <wsdl:binding name="UserAuthenticationSoap11Binding" type="xsd:UserAuthenticationPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="AuthenticateUser">
  <soap:operation soapAction="urn:AuthenticateUser" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="Logout">
  <soap:operation soapAction="urn:Logout" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="UserAuthenticationSoap12Binding" type="xsd:UserAuthenticationPortType">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="AuthenticateUser">
  <soap12:operation soapAction="urn:AuthenticateUser" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="Logout">
  <soap12:operation soapAction="urn:Logout" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="UserAuthenticationHttpBinding" type="xsd:UserAuthenticationPortType">
  <http:binding verb="POST" />
- <wsdl:operation name="AuthenticateUser">
  <http:operation location="UserAuthentication/AuthenticateUser" />
- <wsdl:input>
  <mime:content type="text/xml" part="AuthenticateUser" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="AuthenticateUser" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="Logout">
  <http:operation location="UserAuthentication/Logout" />
- <wsdl:input>
  <mime:content type="text/xml" part="Logout" />
    </wsdl:input>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:service name="UserAuthentication">
- <wsdl:port name="UserAuthenticationHttpSoap11Endpoint"
      binding="xsd:UserAuthenticationSoap11Binding">
  <soap:address location="http://localhost:8080/axis2/services/UserAuthentication" />
```

```
    </wsdl:port>
- <wsdl:port name="UserAuthenticationHttpSoap12Endpoint"
      binding="xsd:UserAuthenticationSoap12Binding">
  <soap12:address location="http://localhost:8080/axis2/services/UserAuthentication" />
    </wsdl:port>
- <wsdl:port name="UserAuthenticationHttpEndpoint" binding="xsd:UserAuthenticationHttpBinding">
  <http:address location="http://localhost:8080/axis2/services/UserAuthentication" />
    </wsdl:port>
    </wsdl:service>
    </wsdl:definitions>
```

8 WSDL of Federation Manager Service Interface

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ax213="http://principal.component/xsd"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:ax211="http://component/xsd"
    xmlns:xsd="http://federation.ws" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    targetNamespace="http://federation.ws">
- <wsdl:types>
- <xs:schema xmlns:ax212="http://component/xsd" attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://federation.ws">
  <xs:import namespace="http://component/xsd" />
- <xs:element name="DispatchGetCredentialRequest">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="serviceID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="token" nillable="true" type="ax211:UserToken" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="DispatchGetCredentialRequestResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="ax211:CredentialManagerResponse" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="DispatchLoginRequest">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="user" nillable="true" type="ax211:UserReq" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="DispatchLoginRequestResponse">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="ax211:UserToken" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="UpdateSSORegistry">
- <xs:complexType>
- <xs:sequence>
  <xs:element minOccurs="0" name="tokenID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="toLogin" type="xs:boolean" />
  <xs:element minOccurs="0" name="isFinal" type="xs:boolean" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:schema>
- <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://principal.component/xsd">
- <xs:complexType name="UserPrincipal">
- <xs:sequence>
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="pID" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="password" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="rolePrincipal" nillable="true" type="ax213:Role" />
    </xs:sequence>
    </xs:complexType>
```

```
- <xs:complexType name="Role">
- <xs:sequence>
  <xs:element minOccurs="0" name="communityPrincipal" nillable="true" type="ax213:Community" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="roleName" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="Community">
- <xs:sequence>
  <xs:element minOccurs="0" name="communityName" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="name" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
- <xs:schema xmlns:ax214="http://principal.component/xsd" attributeFormDefault="qualified"
     elementFormDefault="qualified" targetNamespace="http://component/xsd">
  <xs:import namespace="http://principal.component/xsd" />
- <xs:complexType name="UserToken">
- <xs:sequence>
  <xs:element minOccurs="0" name="domain" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="tag" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="CredentialManagerResponse">
- <xs:sequence>
  <xs:element minOccurs="0" name="credentialToken" nillable="true" type="ax211:CredentialToken" />
  <xs:element minOccurs="0" name="userPrincipal" nillable="true" type="ax213:UserPrincipal" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="CredentialToken">
- <xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="0" name="authorizedOperations" nillable="true"
     type="xs:string" />
    </xs:sequence>
    </xs:complexType>
- <xs:complexType name="UserReq">
- <xs:sequence>
  <xs:element minOccurs="0" name="domain" nillable="true" type="xs:string" />
  <xs:element maxOccurs="unbounded" minOccurs="0" name="password" nillable="true" type="xs:string" />
  <xs:element minOccurs="0" name="uid" nillable="true" type="xs:string" />
    </xs:sequence>
    </xs:complexType>
    </xs:schema>
    </wsdl:types>
- <wsdl:message name="UpdateSSORegistryRequest">
  <wsdl:part name="parameters" element="xsd:UpdateSSORegistry" />
    </wsdl:message>
- <wsdl:message name="DispatchGetCredentialRequestRequest">
  <wsdl:part name="parameters" element="xsd:DispatchGetCredentialRequest" />
    </wsdl:message>
- <wsdl:message name="DispatchGetCredentialRequestResponse">
  <wsdl:part name="parameters" element="xsd:DispatchGetCredentialRequestResponse" />
    </wsdl:message>
- <wsdl:message name="DispatchLoginRequestRequest">
  <wsdl:part name="parameters" element="xsd:DispatchLoginRequest" />
    </wsdl:message>
- <wsdl:message name="DispatchLoginRequestResponse">
  <wsdl:part name="parameters" element="xsd:DispatchLoginRequestResponse" />
    </wsdl:message>
- <wsdl:portType name="FederationManagerPortType">
- <wsdl:operation name="UpdateSSORegistry">
  <wsdl:input message="xsd:UpdateSSORegistryRequest" wsaw:Action="urn:UpdateSSORegistry" />
    </wsdl:operation>
- <wsdl:operation name="DispatchGetCredentialRequest">
  <wsdl:input message="xsd:DispatchGetCredentialRequestRequest"
     wsaw:Action="urn:DispatchGetCredentialRequest" />
```

```
  <wsdl:output message="xsd:DispatchGetCredentialRequestResponse"
      wsaw:Action="urn:DispatchGetCredentialRequestResponse" />
    </wsdl:operation>
- <wsdl:operation name="DispatchLoginRequest">
  <wsdl:input message="xsd:DispatchLoginRequestRequest" wsaw:Action="urn:DispatchLoginRequest" />
  <wsdl:output message="xsd:DispatchLoginRequestResponse"
      wsaw:Action="urn:DispatchLoginRequestResponse" />
    </wsdl:operation>
    </wsdl:portType>
- <wsdl:binding name="FederationManagerSoap11Binding" type="xsd:FederationManagerPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="UpdateSSORegistry">
  <soap:operation soapAction="urn:UpdateSSORegistry" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
    </wsdl:operation>
- <wsdl:operation name="DispatchGetCredentialRequest">
  <soap:operation soapAction="urn:DispatchGetCredentialRequest" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="DispatchLoginRequest">
  <soap:operation soapAction="urn:DispatchLoginRequest" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="FederationManagerSoap12Binding" type="xsd:FederationManagerPortType">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <wsdl:operation name="UpdateSSORegistry">
  <soap12:operation soapAction="urn:UpdateSSORegistry" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
    </wsdl:operation>
- <wsdl:operation name="DispatchGetCredentialRequest">
  <soap12:operation soapAction="urn:DispatchGetCredentialRequest" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="DispatchLoginRequest">
  <soap12:operation soapAction="urn:DispatchLoginRequest" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
    </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:binding name="FederationManagerHttpBinding" type="xsd:FederationManagerPortType">
  <http:binding verb="POST" />
- <wsdl:operation name="UpdateSSORegistry">
  <http:operation location="FederationManager/UpdateSSORegistry" />
```

```
- <wsdl:input>
  <mime:content type="text/xml" part="UpdateSSORegistry" />
    </wsdl:input>
    </wsdl:operation>
- <wsdl:operation name="DispatchGetCredentialRequest">
  <http:operation location="FederationManager/DispatchGetCredentialRequest" />
- <wsdl:input>
  <mime:content type="text/xml" part="DispatchGetCredentialRequest" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="DispatchGetCredentialRequest" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="DispatchLoginRequest">
  <http:operation location="FederationManager/DispatchLoginRequest" />
- <wsdl:input>
  <mime:content type="text/xml" part="DispatchLoginRequest" />
    </wsdl:input>
- <wsdl:output>
  <mime:content type="text/xml" part="DispatchLoginRequest" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
- <wsdl:service name="FederationManager">
- <wsdl:port name="FederationManagerHttpSoap11Endpoint"
     binding="xsd:FederationManagerSoap11Binding">
  <soap:address location="http://localhost:8080/axis2/services/FederationManager" />
    </wsdl:port>
- <wsdl:port name="FederationManagerHttpSoap12Endpoint"
     binding="xsd:FederationManagerSoap12Binding">
  <soap12:address location="http://localhost:8080/axis2/services/FederationManager" />
    </wsdl:port>
- <wsdl:port name="FederationManagerHttpEndpoint" binding="xsd:FederationManagerHttpBinding">
  <http:address location="http://localhost:8080/axis2/services/FederationManager" />
    </wsdl:port>
    </wsdl:service>
    </wsdl:definitions>
```

References of the Appendixes

Boutilier, C., R. I. Brafman, et al. (2004). "CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements." Journal of Artificial Intelligence Research 21: 135-191.

Brafman, R. and C. Domshlak (2002). "Introducing variable importance tradeoffs into CP-nets." Proceedings of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence: 69–76.

Châtel, P., I. Truck, et al. (2008). A linguistic approach for non-functional constraints in a semantic SOA environment. FLINS'08, Madrid, España.

FOLIO ADMINISTRATIF

THESE SOUTENUE DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

NOM : SLIMAN DATE de SOUTENANCE : 27/11/2009

Prénoms : LAYTH

TITRE : C-BUSINESS ET URBANISATION D'ENTREPRISE

NATURE : Doctorat Numéro d'ordre : 2009-ISAL-0099

Ecole doctorale : INFOMATHS

Spécialité : INFORMATIQUE

Cote B.I.U. - Lyon : T 50/210/19 / et bis CLASSE :

RESUME

Les évolutions permanentes du marché ont forcé la plupart des entreprises à se focaliser sur les processus liés à leur cœur de métier. Ce recentrage les conduit alors soit à externaliser certaines parties de leurs processus, soit former temporairement une association avec d'autres partenaires. Ces scénarios de collaboration imposent plusieurs contraintes sur la conception et l'organisation du système d'information à fin de le rendre facilement adaptable pour suivre les changements au niveau d'organisation.
Pour que le système d'information soit facilement adaptable il est possible de restructurer le système d'information en respectant les principes de l'urbanisation du système d'information couplé par une architecture orienté service, toute fois, cette organisation conduit à des systèmes assez rigides ne donnant pas réellement les capacités d'initier des processus collaboratifs. Or, la collaboration impose de prendre en compte les contraintes de sécurité car l'approche traditionnelle d'urbanisation ne prend pas en considération la possibilité de collaboration et forme des «îlots de sécurité » ce qui s'oppose à la nature transversale de la sécurité. En plus, dans un modèle orienté services, les applications distribuées sur plusieurs site ont peu ou pas de visibilité en matière de l'information nécessaires pour assurer la sécurité au nouveau globale.
C'est dans ce contexte que nous avons proposé d'adopter une démarche d'urbanisation d'entreprise qui promeut une organisation transversale du système de production de l'entreprise qui permet une construction incrémentale des processus collaboratifs. Nous sommes parvenus à spécifier un modèle de service industriel construit par regroupement de toutes les fonctions nécessaires autour de la fabrication du produit. Ensuite, nous nous somme proposé de construire un middleware supportant ces services industriels. Cela induit d'ajouter un niveau sémantique capable de gérer les propriétés fonctionnelles et non fonctionnelles (qualité de service et sécurité) aux bus de services traditionnels (ESB). Dans le cadre du projet ANR SEMEUSE visant à doter un ESB Open source (PETALS) d'un niveau sémantique, notre contribution a plus particulièrement portée sur la spécification et la mise en œuvre des composants permettant d'intégrer de manière contextuelle les politiques de sécurité.

MOTS-CLES : Urbanisation d'entreprise, Collaboration, Sécurité de Système d'Information, Entreprise Service Bus (ESB), SOA Sémantique.

Laboratoire (s) de recherche: LIESP
Directeur de thèse: Frédérique BIENNIER (Professeur à l'INSA de Lyon).
Président de jury : Jacques MALENFANT (Professeur à l'Université Pierre et Marie Curie, Paris).
Composition du jury : Danielle BOULANGER (Professeur à l'IAE – Université Jean Moulin Lyon 3).
Bruno VALLESPIR (Professeur à l'ENSEIRB, Université de Bordeaux).
Zensho NAKAO (Professeur à la Faculté d'ingénierie de l'Université de Ryukus- Japon).
Youakim BADR (Maître de Conférences à l'INSA de Lyon).

École Doctorale Informatique et Mathématiques

InfoMaths

Institut National des Sciences Appliquées de Lyon

Laboratoire d'Informatique pour l'Entreprise et les Systèmes de Production

C-Business et Urbanisation d'Entreprise

Thèse de Doctorat
(Spécialité Informatique)

Layth SLIMAN
(Ingénieur en Informatique)

Commission d'Examen

Jacques MALENFANT (Président, Professeur - Université Pierre et Marie Curie, Paris)

Danielle BOULANGER (Rapporteur, Professeur - Université Jean Moulin Lyon 3)

Bruno VALLESPIR (Rapporteur, Professeur - Université de Bordeaux)

Zensho NAKAO (Examinateur, Professeur - Université des Ryukyus- Japon)

Frédérique BIENNIER (Directrice de thèse, Professeur - INSA de Lyon)

Youakim BADR (Co-directeur, Maître de Conférences - INSA de Lyon)

Table de matières

Table des figures

Chapitre 1

1 Introduction

L'évolution actuelle du marché (customisation de masse, logique de globalisation, mutations technologiques…) conduit les entreprises à opérer des recentrages métier. Cette focalisation sur le cœur de métier s'accompagne d'une logique d'externalisation/collaboration entre entreprises conduisant à de nouvelles stratégies d'organisation qui intègrent une partie de l'environnement dans l'organisation propre de l'entreprise (alliances, entreprises virtuelles ou étendues, « supply chain »…) pour mieux intégrer la chaîne de valeur perçue par le client. Ceci implique agilité (i.e. la capacité de répondre aux changements structurels) et interopérabilité tant en ce qui concerne l'organisation de l'entreprise que les outils supports. Pour répondre à ces enjeux, plusieurs problèmes doivent être résolus :

- L'entreprise doit identifier son cœur de métier et ce qui peut être externalisé
- L'organisation effective de la production pour répondre aux besoins des clients est un processus essentiellement transversal. Il faut donc déterminer comment équilibrer les relations de pouvoir entre responsables fonctionnels et comment donner des objectifs adaptés aux acteurs impliqués pour garantir un partage des objectifs communs et assurer globalement un bon niveau de performance
- La mise en cohérence des contraintes liées à l'organisation ou aux technologies et celles liées aux aspects humains (compétences, culture) doit être réalisée lors de l'organisation de processus
- Pour ce qui concerne l'organisation d'une entreprise, l'orientation processus oblige de repenser l'organisation en partant des buts (liés à la raison d'être l'entreprise) pour identifier les processus que l'entreprise doit réaliser. Il faut ensuite déterminer processus par processus les compétences et les moyens nécessaire sans tenir compte de l'organisation actuelle de l'entreprise pour qu'on puisse donner à chaque processus les moyens dont il a besoin.

On peut résumer ces enjeux par le postulat suivant :
La structure d'organisation d'entreprise doit refléter constamment les activités à la fois selon la dimension verticale (fonctionnelle) pour prendre en considération la contrainte de cohérence métier et garantir que chaque métier dans l'organisation soit évalué par un responsable ou un expert de ce métier, et selon la dimension horizontale (exécution) pour

pouvoir construire les processus d'une manière dynamique et les exécuter d'une façon cohérente et efficace. De cette façon l'organisation aura une structure dynamique, capable de s'adapter aux différents contextes.

Outre la structure organisationnelle de l'entreprise qui supporte la constitution et l'exécution de ces processus, ces scénarios imposent plusieurs contraintes sur la conception et l'organisation du système d'information (SI) qui gère - dans le contexte actuel- la quasi-totalité des activités de l'entreprise et qui supporte et/ou exécute les processus de l'entreprise. Pour cela, il faut garantir l'agilité du système d'information. Ceci se traduit par la capacité de modifier ou étendre avec simplicité les fonctions rendues par le système d'information pour pouvoir suivre rapidement les évolutions fréquentes du métier de l'entreprise et celles de la technologie.

Or, les systèmes d'informations des entreprises sont traditionnellement conçus pour fonctionner dans un environnement assez fermé où l'interaction avec l'entourage est strictement définie (pour ce qui concerne les formes d'échanges d'information). En revanche, dans le contexte de la collaboration entre entreprises, les systèmes d'information des entreprises impliquées doivent être prêts à «inter-opérer» dans un processus collaboratif.

Pour répondre à ces contraintes, il faut restructurer le système d'information pour en faciliter l'évolution, augmenter le niveau d'interopérabilité technologique tout en protégeant le patrimoine informationnel. Pour cela, on peut recourir aux approches d'urbanisation du SI couplées aux technologies "orientées services".

L'urbanisation du système d'information d'une entreprise consiste à découper le SI en modules autonomes, de taille de plus en plus petite, de manière similaire à l'organisation de l'urbanisation à l'échelle d'une ville. Entre les modules se dessinent des zones d'échange d'informations qui permettent de découpler les différents modules pour qu'ils puissent évoluer séparément tout en conservant leur capacité à interagir avec le reste du système, pour pouvoir remplacer certains de ces sous-systèmes (contrainte d'interchangeabilité), pour garantir la capacité à construire et à intégrer des sous-systèmes d'origines diverses (contrainte d'intégration) et pour renforcer la capacité du SI d'interagir avec d'autres SI (contrainte d'interopérabilité). Dans l'approche traditionnelle la stratégie d'urbanisation est conçue autour de l'architecture fonctionnelle de l'entreprise sans prendre compte la logique de production (l'architecture métier). Cela rend difficile de redéfinir les processus de l'entreprise « à la demande » dans un scénario de collaboration en raison du cloisonnent inter-métiers issu de la démarche d'urbanisation. Ceci pouvait conduire à une certain rigidité voire à un manque de flexibilité au niveau de l'organisation du processus transversal. Pour pallier cette limite, il

importe donc de proposer une organisation « agile » des systèmes d'information et de l'organisation du processus afin de permettre des reconfigurations « à la demande ».

En outre, dans le contexte de processus collaboratif définis de manière « ad-hoc », la construction des processus dans une logique de contrôle « pilotée par les tâches » s'avère inadaptée car dans cette logique les processus des entreprises impliquées dans la collaboration doivent êtres définis à l'avance d'une façon purement algorithmique. Il serait donc souhaitable de reconstruire les processus dans une logique « pilotée par les ressources » pour arriver à construire les processus d'une façon incrémentale à partir des modèles abstraits. Cela permet aux entreprises impliquées de construire leurs processus selon les fonctionnalités des ressources disponibles.

Mettre en œuvre des processus collaboratifs commence par la sélection de partenaires. Pour permettre à une telle sélection, les entreprises doivent définir et publier, d'une manière objective, ce qu'elles peuvent offrir à ses partenaires potentiels. En outre, les entreprises doivent préciser de manière exhaustive les contraintes, les conditions et les contextes dans lesquelles leurs ressources peuvent être utilisées, et leurs processus peuvent être contrôlés et « monitorés », faute de quoi la collaboration à la demande sera difficile, voire impossible et des partenaires potentiels peuvent être éliminés à cause de l'ambiguïté de leurs offres. En conséquence, la «composition» dynamique des processus est assurée principalement par une meilleure description des offres ce qui engendre le besoin des outils adéquats permettant l'identification des offres disponibles et la vérification de leurs compatibilité vis-à-vis aux demandes, aussi bien pour leurs aspects fonctionnels, que pour les aspects non-fonctionnels (Qualité de Service, Qualité de Protection...).

D'autre part, la collaboration impose une réorganisation de l'entreprise afin de promouvoir des structures interopérables à tous les niveaux, technologique, organisationnel et au niveau business (affaire).

L'interopérabilité d'affaire implique de fournir la capacité de construire des processus de collaboration à partir de processus diversifié et spécialisés appartenant à des partenaires différents, indépendamment de leurs modèles d'affaires internes et de leur systèmes de décisions. Cela nécessite des mécanismes et des moyens pour orchestrer des processus distribués, des mécanismes du contrôle, de vérification et de coordination décentralisé.

L'interopérabilité organisationnelle impose d'adapter les structures organisationnelles pour parvenir à une organisation dynamique qui est capable de fournir des réponses adaptées et instantanées aux scénarios diversifiés de collaboration, ce qui implique une organisation

transversale (ou horizontale) à partir de la gestion de la relation client jusqu'à la gestion de la relation fournisseur au processus production.

L'interopérabilité technologique couvre deux aspects:

- Aspect technique: assurer que les données peuvent être transférées efficacement d'une application à l'autre (infrastructure de communication adaptée).
- Aspect sémantique: la signification des données transférées (le contenu du message) devrait être comprise de la même façon par l'expéditeur et le destinataire.

Au niveau technologique, il faut que le système d'information puisse être réorganisé. Pour cela on peut envisager l'approche service. La SOA « Service Oriented Architecture » est une approche de conception, d'implémentation de systèmes à partir de composants mettant en œuvre des fonctions spécifiques. Ces composants, appelés "Services", peuvent être distribués sur l'ensemble des sites et entreprises, puis reconfigurés en nouveaux processus métier si nécessaire.

La technologie SOA permet aux entreprises d'optimiser la réutilisation des composants existants, et de répondre plus rapidement aux évolutions voire à la construction à la demande du processus. L'avantage de l'architecture orientée services réside dans la liaison entre les processus métier et les processus informatique les supportant. Les facteurs clés de SOA sont les suivants:

- L'architecture SOA est conçue sur des standards Web permettant des mises en œuvre rentables à l'échelon mondial.
- Les services peuvent être combinés et offrent ainsi une bien meilleure souplesse par la création de nouvelles fonctions à la fois dans et entre les entreprises.
- L'approche SOA permet de créer des services qui supportent les processus métier et améliore la capacité à les externaliser et à les étendre aux partenaires.
- La technologie SOA permet de tirer partie des systèmes et processus existants, et ainsi de préserver, voire même d'optimiser les investissements existants.
- Les services prennent en compte les activités logiques. Ces services tiennent compte du concept de processus métier, et non pas des fonctionnalités ou API définies par les progiciels traditionnels.
- De nouveaux services peuvent être ajoutés ou créés sans que cela affecte les mises en œuvre de services actuelles.

Il est essentiel de composer les services collaboratifs d'une façon dynamique, adaptée au contexte et aux variations des besoins. La composition devrait être conduite de façon que : (I)

le service composé adhère aux politiques imposées par les organisations des entreprises offrant le service, (II) les services composants choisis soient compatibles entre eux, de telle sorte que la composition entière ait comme conséquence un service cohérent et que (III) les services composants choisis répondent le plus aux contraintes imposées par les exigences du client. Autrement dit, les services doivent « matcher » les politiques organisationnelles, les contraintes syntaxiques et sémantiques ainsi que les aspects non-fonctionnel de la production.

La mise en œuvre des services métier repose sur un système de gestion des processus métier (Business Process Management System) impliquant la gestion des processus. Il s'agit dans le contexte de la SOA d'un Orchestrateur qui gère l'état de chaque processus ainsi que les résultats intermédiaires. Il guide la construction l'exécution et la coordination de processus « collaboratifs » composés et complexes.

Outre une réorganisation de l'entreprise (selon les dimensions verticale et horizontale) la collaboration impose des contraintes sur l'organisation de production et sur l'organisation d'une politique de sécurité cohérente. En effet, la sélection d'un partenaire repose sur le niveau de confiance « perçu ». Ainsi, tout scénario de collaboration doit fournir un niveau suffisant de sécurité à tous les partenaires impliqués dans un processus collaboratif. Ce niveau dépend souvent des entreprises partenaires : lorsque des entreprises potentiellement concurrentes s'allient pour un projet de collaboration, les besoins en sécurité sont accrus du fait de la perception du risque de concurrence. L'analyse des besoins de sécurité liés à la collaboration doit prendre en compte ces aspects de «Business Security» et faire la distinction entre informations et processus privés d'une part et ceux qui sont partageables d'autre part tout en veillant à la mise en œuvre des activités au sein des entreprises partenaires et à la coordination nécessaires à la réalisation des processus collaboratifs. Ces besoins doivent ensuite être retranscrits dans l'architecture mise en œuvre pour assurer les fonctions de sécurité nécessaires.

Pour répondre à ces exigences, nous proposons notre solution conduisant à réorganiser l'entreprise d'une manière compatible avec l'organisation de service industrielle. Nous présenterons dans un premier temps notre modèle d'urbanisation qui prend en considération la particularité du contexte de production collaborative. Nous traiterons ensuite les aspects de sécurité qui représentent un souci majeur lorsque des entreprises différentes interagissent dans le cadre de collaborations. Enfin, pour mettre en œuvre notre stratégie d'urbanisation collaborative à base de services industriels, le choix du middleware et la plate-forme sécurisés supportant notre modèle sont exposés dans une troisième partie.

Chapitre 2

2 Etat de l'art

2.1 Modélisation d'entreprise

Dans le contexte de la collaboration interentreprises, la modélisation a un rôle très important dans l'analyse des réseaux complexes constitués par les différentes parties impliquées dans la réalisation du but commun. Il importe donc de considérer les différentes approches de modélisation d'entreprise.

Dans cette partie, nous présenterons un état de l'art portant sur les différentes méthodologies de modélisation d'entreprise ainsi que certaines approches de modélisation en particulier les approches liées à l'orientation processus et à l'amélioration et l'adaptation continues qui sont les points de départ pour des démarches d'urbanisation et de construction d'une entreprise collaborative.

2.1.1 Modélisation fonctionnelle

Dans l'approche de modélisation, on peut distinguer la fonctionnalité de l'entreprise (c à d la description des activités) et le comportement de l'entreprise (c à d le choix et l'ordre de l'exécution définissant quelles sont les activités à exécuter et dans quel ordre). L'activité représente le comportement des ressources de l'entreprise par contre les processus représente le comportement dynamique de l'entreprise elle-même les flux de contrôle (dans ce sens il faut prendre en compte le contexte de l'exécution : temps, coût, disponibilité des ressources).

Une activité contribue à la réalisation d'une tâche en transforment un état d'entré en état de sortie en utilisant les ressource de l'entreprise. Plusieurs méthodes et langages sont utilisables pour réaliser ces modélisations comme nous le présentons dans les paragraphes suivantes:

2.1.1.1 IDEF0 ou SADT

IDEF0 ou SADT [NIST 93] sont des langages de modélisation reposent sur deux primitives graphiques :

Les boites qui représentent les activités et les flèches qui représentent les relations, les flèches ne représentent aucunement des notions de casualités mais uniquement des relations de flux entre une activité source et une activité dentinaire, ces sont sur quatre types :

- Les entrés représentent les objets à traiter (subissant une transformation), qui sont des données et des matières.
- Les entrés de contrôles sont des informations (directives, procédures, règles, contraintes …) qui contraignent l'exécution de l'activité mais ne sont pas modifiables par l'activité.
- Les sorties représentent les objets (données, matières) produits ou modifiés par l'activité.
- Les mécanismes représentent les moyens (ressource humaines ou matérielles) nécessaires à l'exécution de l'activité.

2.1.1.2 Modélisation des processus opérationnels (IDEF3)

La méthode IDEF3 [Mayer 95] a été proposée pour palier les défauts d'IDEF0 en matière de modélisation de comportement de l'entreprise (des flux de contrôle). En effet, IDEF3 permet la description des processus opérationnels d'entreprise en utilisant une notation graphique pour modéliser le processus d'entreprise sous forme d'un enchaînement d'étapes, appelées unités de comportement (unit of behaviour) connectés par des boîtes de jonction et des liens.
Une telle représentation forme un diagramme appelé description de flux de contrôle du processus. On peut également adjoindre aux diagrammes de flux de contrôle des processus des diagrammes de transitions d'états des objets manipulés dans les étapes des processus. Ces diagrammes sont des graphes nommés OSTN (Object State Transition Network diagrammes). Les nœuds de ces digrammes sont des états d'objet représentés par des cercles. Les nœuds sont connectés par des flèches étiquetée par une ou plusieurs unités de comportement c à d les actions faisant passer l'objet d'un état à un autre.

2.1.2 Les architectures de référence

Il s'agit de fournir un cadre général et des points de repère aux utilisateurs en leur indiquant quels aspects de l'entreprise doivent être pris en compte, les relations qui existent entre eux et la terminologie communément admis dans le domaine. Parmi les plus connus on peut citer CIMOSA, GIM, PERA.

2.1.2.1 CIMOSA

CIMOSA [Li 94] est une architecture de modélisation pour construire des systèmes intégrés de production. Le cadre de modélisation de CIMOSA formalise trois principes fondamentaux pour la modélisation en suivant une structure à trois axes (Fig. 1) :

- L'axe de généricité : il se compose de trois nivaux, le niveau générique où sont définis les primitives de base du langage de modélisation, le niveau partiel contenant des modèles partiels c à d des structure prédéfinies et réutilisables pour un domaine d'application donné et le niveau particulier qui correspond aux modèles spécifiques de l'entreprise. Les niveaux générique et partiel constituent l'architecture de référence de CIMOSA alors que le niveau particulier correspond à l'architecture particulière d'une entreprise donnée.
- L'axe de dérivation permet de modéliser suivant trois niveaux : le niveau de définition de besoins permet l'écriture du cahier de charge dans le langage de l'utilisateur final, le niveau de spécification de conception permet de spécifier et d'analyser dans le détail des solutions répondant aux besoins exprimés et le niveau de description d'implémentation permet de décrire précisément l'implémentation des solutions retenues.

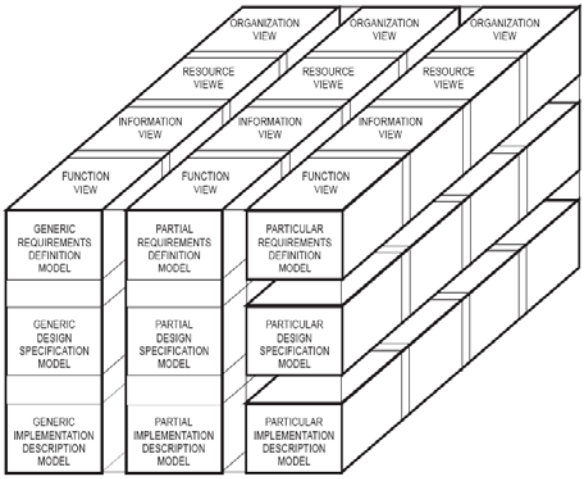

Figure 1 Framework de CIMOSA, tiré de [Li 94], page 119

- L'axe de génération définit quatre vues de modélisation permettant d'accéder au modèle en se focalisant sur certains aspects et en négligeant les autres pour gérer la complexité de modèle. Les vue ne sont pas des sous modèles indépendantes mais offrent des mécanismes pour accéder à certains aspects d'un même modèle. Les vues gérées par CIMOSA sont la vue fonctionnelle, la vue d'information, la vue d'organisation et la vue de ressources.

2.1.2.2 GRAI-GIM

GRAI-GIM [Vallespir 92] est une méthodologie de modélisation et d'analyse des systèmes de décision des entreprises de production, elle reprend les notions de vues et les niveaux d'abstraction comme CIMOSA. GIM considère quatre vues (Fig. 2): information (données et connaissance), décision (chaîne d'activités et centre de décision), physiques (ressource) et fonction (décomposition fonctionnelle). La démarche consiste à modéliser un système d'entreprise au niveau conceptuel (définition des besoins tels que perçu par les utilisateurs) puis au niveau structurel (définition d'une solution technologique validé par les utilisateurs) et enfin du niveau dit réalisation qui décrit l'implémentation du système conçu ou réorganisé en prenant en compte les aspects organisationnels, les aspects relatifs aux technologies de l'information et les aspects relatifs aux technologies industrielles. GIM ne fournit pas un langage de modélisation unique ni de constructs propres mais elle utilise les formalismes existantes. On notera que cette méthode privilégie la description des processus décisionnels en identifiant les acteurs impliqués et les différents horizons de décision (stratégique, tactique et opérationnelle).

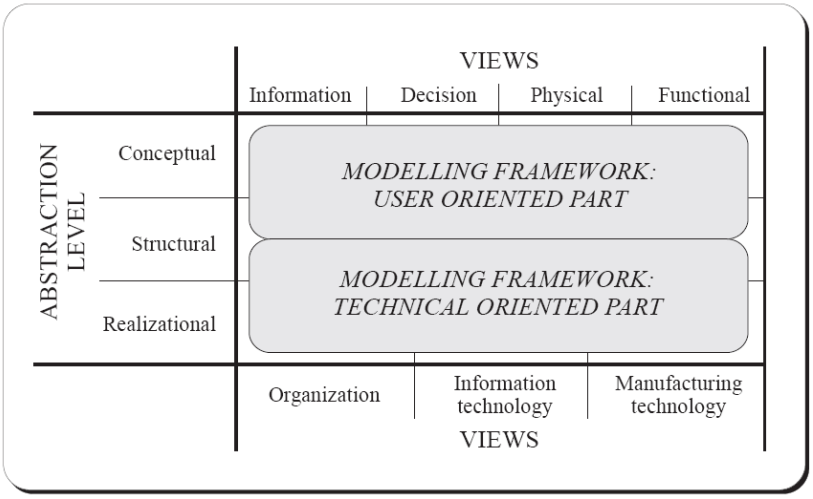

Figure 2 GIM Framework, tiré de [Li 94], page 159

2.1.2.3 PERA

PERA [Williams 95] est une méthodologie complète d'ingénierie des environnements industriels. Cette méthodologie définit toutes les phases de cycles de vie d'une entité industrielle selon cinq phases (Fig. 3) :

- Une phase de conceptualisation qui se compose de deux étapes : une étape dite d'identification du périmètre d'étude de l'entité industrielle et une étape de conception qui en définit la mission en termes de politiques concernant le produit et le système

opérationnel. Ceci donne naissance à deux branches de la méthodologie : une pour la partie opérative (machines et équipements) et une pour la partie information/commande (système d'information et de gestion).

- Une phase de définition pour chaque partie : cette phase définit les besoins(4,5), les tâches et modules nécessaires pour répondre à ces besoins (6,7) et fournit des diagrammes de flux pour l'entité (8,9).
- Une phase de conception : cette phase comporte deux étapes majeures une étape de conception fonctionnelle, et une étape de spécifications portant sur l'organisation des ressources humaines, du SI et architecture opérative.
- Une phase de d'installation et de construction : cette phase consiste à transformer les modèles issus de la conception détaillée en une implémentation effective (machine, systèmes informatique, système de logistique).
- Une phase opérationnelle et de la maintenance : correspond à l'utilisation effective de l'installation pendant laquelle celle-ci doit remplir sa mission telle que définie par la direction.

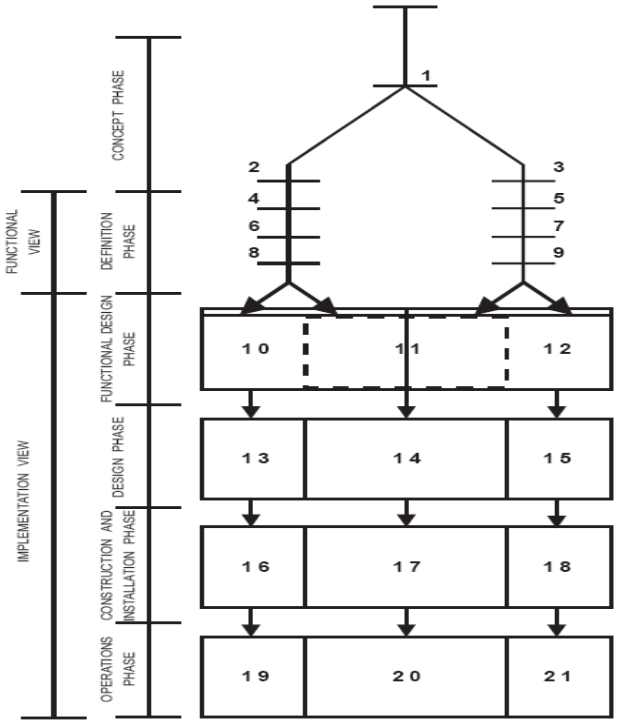

Figure 3 la structure de PERA, tiré de [Williams 95], page 101

2.1.2.4 GERAM

GERAM [IFIP 99] est une approche pragmatique fournissant un cadre généralisé pour décrire les composants requis dans tous les types du processus d'ingénierie d'entreprise et s'applique

dans différents contextes (comme la réingénierie complète, la fusion, la réorganisation, la formation d'entreprise virtuelle, l'intégration d'une chaîne d'approvisionnements, etc.) et les changements incrémentaux de toutes sortes.

GERAM est prévu pour faciliter l'intégration des méthodes de plusieurs disciplines comme les méthodes de génie industriel, de science de la gestion, d'automatique, de communication et de la technologie de l'information, c.-à-d permet leur utilisation combinée, au lieu de l'application isolée de ses méthodes (Fig. 4).

Un aspect important du cadre de GERAM est qu'il unifie les deux approches distinctes de les approches basées sur les modèles de processus métier et ceux qui sont basées sur le produit.

Un des composants les plus importants du cadre de GERAM est GERA (architecture de référence généralisée d'entreprise) qui inclue les concepts de base à employer dans l'ingénierie et l'intégration d'entreprise (par exemple, les entités d'entreprise, cycle de vie).

GERAM distingue les méthodologies pour l'ingénierie d'entreprise (EEMs) et les langages de modélisation (EMLs) qui sont employés par les méthodologies pour décrire et le modèle, la structure, le contenu et le comportement des entités d'entreprise. Ces langages permettent de modéliser aussi bien la partie humaine que les parties liées aux processus métier et aux technologies de support. Le processus de modélisation produit des modèles d'entreprise (SME) qui représentent l'ensemble ou une partie des opérations d'entreprise (fabrication, organisation et gestion, systèmes de commande et d'information…).

La méthodologie et les langages utilisés pour modéliser l'entreprise sont supportés par des outils d'ingénierie d'entreprise (EETs). La sémantique des langages de modélisation peut être définie par des ontologies, des métas modèles et des glossaires (concepts générique de modélisation d'entreprise (GEMCs)). Le processus de modélisation est amélioré en utilisant les modèles partiels (PEMs) qui sont les modèles réutilisables des rôles humains, des processus et des technologies. L'utilisation opérationnelle des modèles d'entreprise est supportée par des modules spécifiques (EMOs)qui fournissent des composants préfabriqués comme des profils de compétence humains pour certaines professions, ou des services d'infrastructure technologique, ou tout autre produit qui peut être employé comme composant dans l'exécution du système opérationnel (EOSs).

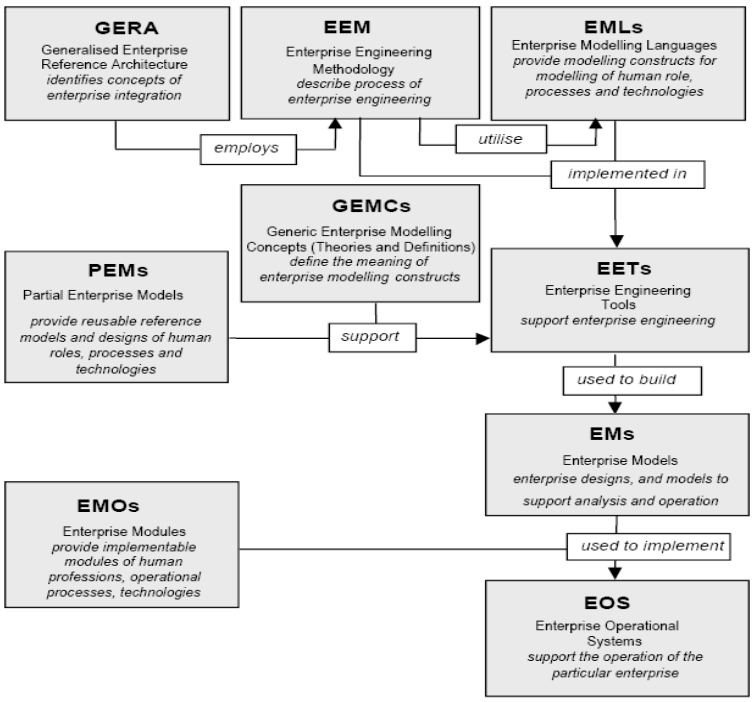

Figure 4 Les composants du GERAM, tiré de [IFIP 99], page 5

2.1.3 Autres approches de modélisation

Après avoir discuté les méthodes et *frameworks* de modélisation d'entreprise le plus connus il importe prendre en considération quelque travaux de recherche effectué dans des domaines liés au domaine de modélisation d'entreprise comme l'intégration d'entreprise, l'ingénierie et la réingénierie d'entreprise ,l'entreprise virtuelle, les ontologies…etc.

[Mertins 05] décrit un ensemble d'architectures, méthodes et outils pour l'Entreprise Engineering (EE) avant de présenter une méthodologie et un outil appelé Integrated Enterprise Modelling (IEM), qui supporte toutes les phases et aspects de l'ingénierie d'entreprise. La méthode utilise l'approche orientée objet pour décrire les informations et les fonctions des objets comme des vues sur un modèle simple de l'entreprise de fabrication.

Le modèle contient deux vues : une vue associé aux processus métier et une vue associé au modèle des informations.

Dans ce modèle, les processus de fabrication et toutes les activités qui sont en réalité liées à la production, sont décrits par des fonctions et des processus métiers.

La base pour le développement du modèle dans le contexte d'une entreprise individuelle est constituée par les classes « produit » et « ressource » et « ordre ». Les données et les fonctions exigées sont assignées à ces objets. Les relations entre les objets sont également déterminées dans cette phase.

Le résultat est que les tâches, l'organisation de processus, les données de l'entreprise, les équipements de production et tous les composants du système d'information sont décrits d'une façon compréhensible à n'importe quel niveau désiré de détail.

Le modèle de vue de processus métier porte sur les tâches et les processus qui sont exécutés sur les objets. Le modèle de vue sur les informations souligne les structures des objets. Les processus métiers et l'information associée sont décrits dans un modèle « cœur ». Les systèmes d'information, la structure d'organisation, les conditions de qualité constituent les vues d'utilisateur qui se relient au modèle « cœur ». Ceci permet d'évaluer des solutions alternatives, des modifications de processus ou d'organisation tout en prenant en considération les effets sur le contrôle, la qualité, système de support, la structure d'organisation et le profil de la qualification des acteurs.

Dans [Delen 03] les auteurs présentent une approche et une application qui adressent modélisation et l'analyse de l'entreprise. Dans ce travail, les auteurs définissent la modélisation de l'entreprise comme le processus permettant de capturer l'abstraction la plus riche possible de l'entreprise selon différentes perspectives (appelé Enterprise model set) de telle sorte que ces modèles d'entreprise puissent être employés pour une grande variété de scénarios dans l'analyse commerciale et l'aide à la décision. Cet ensemble comprend des différents types de modèles associés à un point de vue.

Dans [Koubarakis 02] les auteurs présentent une approche de modélisation de l'entreprise composée de cinq sous-modèles qui sont reliés ensemble et employés pour décrire d'une façon formelle différents aspects de l'entreprise:

- sous-modèle d'organisation : il décrit les acteurs de l'entreprise (humains et automatisés), leurs rôles, leurs responsabilités et leurs compétences, les acteurs sont capables d'exécuter certaines activités.
- sous-modèle d'objectifs et de buts : il décrit ce que l'entreprise et ses acteurs essayent de réaliser.
- sous-modèle de processus décrit comment réaliser les processus, les concepts principaux de ce sous-modèle sont : action, processus, rôle, acteur et but. Toutes les actions effectuées en tant qu'élément d'un processus sont exécutées dans le contexte d'un rôle d'organisation par un acteur assume ce rôle.
- sous-modèle de concepts : contient des informations sur les aspects non-intentionnels des entités de l'entreprise, de leurs rapports et leurs attributs. Les données d'entreprise font partie de ce sous-modèle.

- sous-modèle de contraintes, décrit les limites de ce que l'entreprise et ses composants peuvent faire. Il est employé pour exprimer des restrictions imposées à l'entreprise. Les contraintes peuvent être statique (c.-à-d., liés à une simple situation) ou dynamique (c.-à-d., liées à plusieurs situations).

Les méthodes présentées précédemment sont focalisées sur une entreprise isolée. Intégrer le « Collaborative Business » impose de définir des stratégies complémentaires permettant de se focaliser sur la modélisation des collaborations.

Dans [Rembold 98] on présente un outil basé sur CIMOSA pouvant être utilisé pour la conception des processus virtuels, c.-à-d. des processus composés d'activités exécutées par plusieurs partenaires.

On suppose que le modèle d'entreprise pour chaque partenaire potentiel de l'entreprise virtuelle est déjà existant. En outre, le produit fini objet de la collaboration est connu. Le problème est donc de choisir les partenaires pour que la production puisse être exécutée de façon optimale en prenant en considération certains critères.

Dans le modèle de l'entreprise membre, seule la description des activités, leurs entrées de fonctionnel et de contrôle et leurs sorties de fonctionnel est prises en considération. Les activités de toutes les entreprises sont alors stockées dans un « réservoir » commun. Un algorithme est employé pour produire des modèles des processus consacrés à la fabrication du produit désiré. Cette approche fonctionne en logique de flux tiré à partir du produit final désiré en remontant jusqu'à la première activité nécessaire.

Les auteurs de [Gou 03] développent un cadre pour la gestion des opérations de l'entreprise virtuelle. Ce cadre adopte un point de vue « processus métier » afin d'augmenter l'efficacité et l'agilité de production. Le modèle distribué de processus métier et le modèle de l'entreprise virtuelle elle-même sont établis dans ce cadre. Le premier fournit la base fonctionnelle des opérations de l'entreprise virtuelle, alors que le deuxième fournit sa base structurelle. De plus, les deux modèles sont intégrés. Par une telle séparation et intégration de modèles, la gestion de processus industriel des entreprises virtuelles est rendue possible et l'intégration d'affaires des entreprises virtuelles est réalisée. Les auteurs proposent une approche modélisation et d'analyse de processus distribué de l'entreprise virtuelle VE-DBP basée sur les diagrammes de séquence d'UML les réseaux de Pétri. Ceci permet de définir le modèle d'interaction du processus de l'entreprise virtuelle (VE-DBP) qui décrit les interactions entre les processus des entreprises membres (ME-SBP). Ce diagramme de séquence est ensuite mappé dans des réseaux de Pétri pour une phase de vérification. Une telle approche fournit des spécifications de modèles par étapes (à partir de la vue globale vers la vue locale) et fournit également une

méthode d'analyse des propriétés globales à partir de la perspective locale. Par conséquent, il peut adresser la complexité des spécifications et l'analyse de modèles liées au processus métier distribués de l'entreprise virtuelle.

Enfin, le projet TOVE [Fox 95] intègre une ontologie qui se concentre sur la structure d'organisation, les rôles, l'autorité. Les entités de base dans cette ontologie sont représentées comme des objets avec des propriétés et des relations spécifiques. Ces objets sont structurés dans des taxonomies et les définitions des objets, des attributs et des relations sont indiquées dans une logique de premier ordre. Les terminologies de cette ontologie sont réparties en deux groupes associés soit au comportement soit à l'organisation (structurelle). Les Terminologies de comportement permettent de représenter l'activité et ses états d'activation. Une activité est l'action primitive de transformation permettant de représenter les processus. Les activités ont un état dont la valeur est l'une des constantes suivantes : {dormant, exécution, suspendu, accomplie). D'autres états sont associés aux ressources et sont représentés par les attributs suivants : utilisation (s, a), consommation (s, a), libération(s, a), production(s, a).Ces attributs relient l'état avec la ressource exigée par l'activité. Une ressource est utilisée et libérée par une activité. Si l'activité est terminée, la ressource est libérée et aucune des propriétés de cette ressource n'est changée. Une ressource est consommée ou produite si une certaine propriété de la ressource est changée après avoir réalisé l'activité (disponibilité et la quantité de la ressource mais aussi d'autres propriétés liées à une action de production telle que la couleur…).

Dans la terminologie d'organisation, on défini une organisation comme un ensemble d'Agents d'Organisation (dits membres de l'organisation), d'un ensemble d'Unités d'Organisation (sous-composants récursifs ayant une structure semblable aux organismes) et un arbre de buts d'organisation qui indique le but (et sa décomposition dans des sous-objectifs. Un agent d'organisation exécute des activités afin de réaliser un ou plusieurs buts. Un agent peut être un être humain, un programme ou un groupe de personnes et/ou programmes. L'agent Individuel et le Groupe d'agents sont des sous-classes d'Agent d'Organisation.

2.2 Organisation du système de production

Cette partie sera consacrée à analyser la gestion de production afin d'avoir une bonne compréhension des ses différentes principes avant de nous intéresser à la philosophie de Lean manufacturing dans le but d'en identifier les concepts qui seront ensuite traités pour les transformer en règles d'urbanisation de système de production (en concordance avec les règles d'urbanisation de système d'information). Ce qui nous permettra de faire évoluer

simultanément le processus de production (prit en sens large) et le système d'information qui le supporte.

2.2.1 Définitions de base

La production c'est l'opération de transformation de matière premières ou de composants en produits qui ont une valeur sur le marché, conformément au processus de fabrication.

Le processus global de production consiste donc en un flux dans lequel des matières Premières sont transformé progressivement en produit.

Une entreprise de production collaborative doit être modélisé d'une façon globale, c'est-à-dire que la modélisation doit prendre en compte, au-delà des composants de l'entreprise elles-mêmes l'ensemble des acteurs intéressés, les interactions avec l'environnement, voire une partie de cet environnement.

Cette modélisation est à diviser en modélisation de « Pourquoi ? » et modélisation des « Quoi » et la modélisation de « Comment ». Autrement dit la modélisation globale des systèmes de production nécessite la modélisation cohérent de trois aspects: objectif, statique, dynamique.

La production comporte trois éléments en interaction (Fig. 5):

Le Client qui exprime un Besoin pour un Produit avec certaine fonctionnalité qui représente l'objet à réaliser par l'Entreprise en utilisant ces processus.

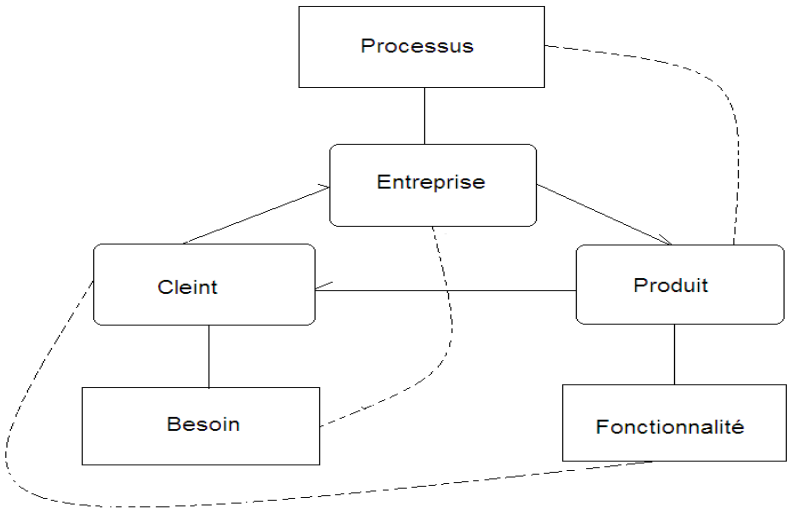

Figure 5 Les éléments de la production et leurs relations

Un produit est défini comme le résultat d'un processus [ISO 00]. C'est le « Quoi ». Un produit est matériel ou immatériel (service). Pour être conçu, un produit complexe est d'abord décrit et décomposé en constituants. On appelle Product Breakdown Structure (PBS) ce

découpage du produit. Le Processus représente le « comment » ou l'aspect dynamique de du système de production. Il s'agit d'un ensemble d'opérations finalisées. Un processus peut être divisé (selon le besoin et la complexité) en sous-processus par type de finalité, souvent un processus est représenté par une séquence d'activités liées entre elles par des relations de précédence temporelle et/ou d'objectif. Une activité est la description d'un travail à réaliser qui exige des spécifications organisationnelles de type : qui fait quoi et quand.

La gestion de la production [Fontanil 99] est l'ensemble des activités qui participent à optimiser les processus de production en améliorant de manière continue les flux allant des fournisseurs aux clients en tenant compte à la fois de la qualité de produits, et du délai et le prix acceptable par le client. La Gestion de Production est aussi définit comme la fonction qui permet de réaliser les opérations de production en respectant les conditions de qualité, délai, coûts qui résultent des objectifs de l'entreprise et dont le but est d'assurer l'équilibre entre :

- le taux d'emploi des ressources,
- le niveau des encours et des stocks,
- les délais.

Organiser des collaborations pour atteindre un but commun répondant aux besoins des clients impose donc de prendre en compte les caractéristiques du processus de production (ressources matérielles, matières, gammes et nomenclatures, procédé de fabrication…) afin d'avoir un niveau d'interopérabilité « d'affaire » et d'organisation permettant une cohérence de bout en bout. Ceci suppose de prendre en compte la typologie de système de production.

2.2.2 Typologie de la production

On peut classifier la production selon plusieurs critères comme modes de production, structure du produit, circulation des produits dans l'atelier, la relation avec le client, un système de production peut être issu d'une ou plusieurs de ces critère au même temps [Fontanil 99].

2.2.2.1 Typologies selon la structure du produit

Dans cette typologie on trouve :

1- Produits complexes au moyen d'une nomenclature arborescente à plusieurs niveaux :
   - Correspondant aux industries manufacturières de type automobile, électronique, électroménager, meubles, etc. Dans cette classe on distingue trois types de structure :

- Convergente « A » produits réalisés à partir de l'assemblage de composants. Cette structure est caractérisée par une arborescence présentant plusieurs niveaux qui correspondent à des sous-ensembles du produit final.
- Convergente « T »: cette structure cherche à concilier la production de masse et la personnalisation des produits. Il s'agit en fait d'un produit de conception modulaire destiné à permettre la réalisation de produits sur mesure en combinant de différentes façons des sous-ensembles dont certains sont standardisés.

2- Les structures divergentes et parallèles : correspondent à des usines fabriquant des produits peu variés, en très grandes séries, sur des périodes très longues avec un marché peu fluctuant. Dans cette classe on distingue deux structures :

- Divergente " V ": cette structure est celle des produits réalisés à partir de la transformation d'une matière première unique : pétrole, lait, acier, etc.
- Parallèle : les produits sont réalisés à partir de quelques matières premières faiblement transformées : industries de l'emballage.

2.2.2.2 Typologie selon la circulation des produits dans l'atelier

Cette typologie est basée sur le critère de circulation du produit dans l'atelier. On distingue deux grandes classes:

- Circulation des produits en Job Shop: Les produits circulent de machines en machines suivant un routage correspondant à leur gamme de fabrication. Ce mode est destiné à fabriquer une grande variété de pièces.
- Circulation des produits en Flow Shop *:* tous les articles suivent le même cheminement. C'est le cas des lignes transferts dédiées où les articles "visitent" systématiquement chaque poste de travail implanté sur la ligne, et toujours dans le même ordre.

2.2.2.3 Typologie selon la relation avec le client

Pour l'entreprise, deux situations peuvent se présenter, suivant que le produit correspond à :

- Un besoin immédiat : ce qui signifie que le délai de livraison acceptable par le client est nul; l'entreprise donc est obligé de produire sur Stock.Une production sur stock est déclenchée par anticipation d'une demande s'exerçant sur un produit dont les caractéristiques sont définies par le fabriquant. Ce type de production s'applique dans les cas suivants :
    - la gamme des produits finis est limité ;

- la demande des produits est prévisible;
- le délai de fabrication est supérieur au délai admissible par le client;
- la saisonnalité du produit est trop forte pour justifier le maintien de ressources en hommes et machines.

- Un besoin différé : ce qui signifie que le client accepte un délai de livraison non nul; l'entreprise donc est obligée de produire sur commande. Dans ce type de production le délai du cycle (achat + fabrication +assemblage + livraison) soit inférieur ou égal au délai acceptable par le client. Théoriquement, si la condition précédente est remplie, aucun stock n'est nécessaire. Si la condition précédente ne peut pas être remplie (le délai de fabrication est trop long) il est possible d'anticiper l'achat et la fabrication des composants, et de procéder à l'assemblage dès que l'on a une commande ferme. Ceci implique aussi d'avoir de bonnes prévisions de ventes afin de ne pas constituer de stocks excessifs de composants. En effet, dans ce type de production, le produit fini peut être personnalisé le plus en aval possible, tout en étant constitué de composants et sous-ensembles standards.

2.2.3 Organisation de la production :

C'est le mode regroupage des machines et des compétences dans les différents ateliers de production de l'entreprise. On peut distinguer essentiellement trois modes d'organisation de production [Courtois 03]:

- **Organisation fonctionnelle :** on regroupe les machines ayant les mêmes techniques ou les mêmes fonctions (atelier mécanique, électrique, ..), en règle générale le montage y est séparé de la fabrication et la réception des matières premières et les produits achetés est centralisé.

Avantage :

- Regroupement des métiers- les personnes qui travaillent dans un secteur sont des professionnels de ce type de machine. Ils peuvent facilement passer d'une machine à l'autre.
- Flexibilité : l'implantation est indépendante des gammes de fabrication il est donc possible de fabriquer tous les types de produit en utilisant les moyens de l'atelier sans perturber davantage le flux.

Inconvénients :

- flux complexe avec de nombreux points de rebroussement, d'accumulation.

- En-cours importants et donc des délais de production importants à cause de la complexité de flux.

- **Organisation en ligne de fabrication** : on trouve ce type d'implantation dans les processus continus. Les machines sont placées en ligne dans l'ordre de la fabrication de la gamme.
- **Organisation en cellules de fabrication (Ilots) :** une implantation en cellules de fabrication est constituée de petits ateliers de production spécialisés de façon à réaliser entièrement un ensemble de produits semblables. C'est un compromis entre la ligne et l'implantation fonctionnelle. Ce type d'implantation permet de diminuer les stockes et les délais dans le cas de processus discontinu.

Le tableau 1 montre une comparaison entre l'organisation linaire et l'organisation cellulaire [Hohmann 07]

Tableau 1 Comparaison entre l'organisation linaire et l'organisation cellulaire

Paramètre	**Ilots**	**Lignes**
Autonomie	Très importante	Très faible à inexistante
Taille des lots de production, des séries.	Petite à moyenne.	Moyenne à grande
Clés de succès	Répartition des charges et ressources. Formation. Motivation	Equilibrage des capacités et gestion des flux
Réserve de Progrès, Gisement de gains	Capitalisation et Mutualisation des bonnes pratiques développées dans certains îlots	Réduction des temps de changement de série.
Surface nécessaire	Faible	Importante
Stocks et encours globaux	Faibles	Importants
Mode d'approvisionnement	Multiple et répété à chaque ilot, en petites quantités	lots importants mis à disposition auprès des lignes
Gestion de production	Résultat cellule par cellule, gestion globale complexe.	Simple, visuelle, résultats prévisibles
Maîtrise des opérations	Apprentissage long, compétences et potentiels nécessaires	Intense répétition = robotisation.

Formation initiale	Bonne maîtrise du métier	Bonne maîtrise par l'extrême simplification Facilité de formation,

2.2.4 Vers le «Lean Manufacturing »

Les principaux flux de production qui traversent un système de production sont [Fontanil 99] :

- les flux physiques : composants achetés, fabriqués, pièces de rechange, sous-ensembles, produits finis, etc.
- les flux informationnels et d'information : commandes, ordres de fabrication, ordres d'approvisionnement, gammes, fiches opératoires, fiches de suivi, etc.

Les principaux facteurs d'amélioration des flux de production sont :

- Simplifier les flux physiques par l'amélioration de l'implantation des équipements; Plusieurs méthodes ont été développées pour répondre à cet objectif : Technologie de Groupe (TG), méthode des chaînons, algorithme de Kiusaku, King.
- accélérer les flux physiques en évitant les pannes, en diminuant les temps de changement de série, en améliorant la qualité, en développant la polyvalence des opérateurs, etc.
- rendre les flux informationnels plus cohérents et de faciliter la communication et l'échange de données. C'est ce dernier objectif qui présente le plus de difficultés pour les entreprises, compte tenu de la quantité, de la diversité et de la difficulté à fiabiliser des informations, mais aussi à cause de l'éclatement des données dans l'ensemble des services.

Cela nous guide à la notion de *lean manufacturing* [Shah 03] qui est une philosophie générique de pilotage de processus industriel dérivée du système de production de Toyota mais également d'autres sources. La focalisation est sur la réduction de toutes sources de pertes afin d'améliorer la valeur globale délivrée au client final. Cela nécessite l'identification des activités qui sont source de valeur ajoutée pour le client, autrement dit le « Value Stream ».

Un "value stream" est l'ensemble des activités bout à bout « processus » nécessaires pour transformer les matières premières en produit fini au client final y compris les services après vente. Tracer le "value stream" pour chaque produit fournit une base pour exécuter une analyse détaillé de chacune des différentes actions dans le "value Stream". Chaque activité est

classifiée dans une des catégories suivantes : (1) elle crée la valeur, (2) il ne crée pas de la valeur mais est indispensable (3) il ne crée aucune valeur et peut être éliminée immédiatement (1) et (2) doivent être analysés plus loin dans le but de les améliorer le plus possible, en éliminant la consommation inutile de ressources. L'analyse de “value Stream” devrait être adopté pour toutes les activités nécessaires pour la production d’un produit ou un service. Ceci implique de penser au delà des frontières de l’entreprise et d'incorporer au processus tous les acteurs qui contribuent au produit final. Généralement dans le cas de la collaboration interentreprises, on peut identifier deux sources importantes de perte (en plus de des sources traditionnelles): les coûts d'opportunité perdue, et l’inefficacité structurelle. Le coût de la perte d’une opportunité dans le marché est dû à une stratégie de collaboration mal définies. La perte liée à la d'inefficacité structurale est produite, par les structures d'organisation inadéquates, ou les mauvaises structures des business modèles qui doivent intégrer des entités telles que des propriétaires, des collaborateurs, des fournisseurs, et des clients. Toute inefficacité structurale existe à n'importe quel niveau de l'entreprise créera la perte et gênera la création de la valeur pour certains ou tous les participants. Quand on parle de structures d'organisation inadéquates ceci inclut (sans s’y limiter) des activités mal intégrées au processus de l'entreprise, l'existence des interfaces inutiles entre les entités qui gênent la collaboration et la coordination…

Le paradigme lean nécessite donc la rationalisation de ces interfaces - sources de perte - et la création d'un environnement plus coopératif entre toutes les parties pour réaliser efficacement les buts du pour les étapes de création de valeur doivent couler facilement dans et entre les entreprises. Ceci signifie que les interfaces entre deux activités quelconques dans le “value stream” - soit elles sont internes ou externes à la entreprise-, doivent être réduites au minimum et amélioré de telle manière que le produit ne rencontre aucune résistance en se déplaçant à la prochaine étape dans le processus. Cela peut être réalisé par remplacer la structure fonctionnels traditionnels, par des équipes de produit intégrées organisées le long du « value Stream ».

Intégrer la démarche lean dans le « Collaborative Business » impose donc de rendre plus efficace l’interconnexion des sous-processus exécutés par les partenaires et de réduire les activités supports non rentables (comme celles engendrés par la manque d’interopérabilité technologique).

2.3 Urbanisation de Système d'information

Urbaniser le système d'information de l'entreprise, c'est organiser la transformation progressive et continue du système d'information visant à le simplifier, à optimiser sa valeur ajoutée et à le rendre plus réactif et flexible vis à vis des évolutions stratégiques de l'entreprise, tout en s'appuyant sur les opportunités technologiques du marché.

L'urbanisme définit des règles ainsi qu'un cadre cohérent, stable et modulaire, auquel les différentes parties prenantes se réfèrent pour toute décision d'investissement dans le système d'information [URBA-SI 03].

L'objectif de l'urbanisation est de concevoir une architecture applicative supportée par une architecture technique adaptée et cohérente avec l'architecture métier, elle-même alignée sur la stratégie de l'organisation.

Traditionnellement la démarche d'urbanisation du SI de l'entreprise consiste à étudier les différents secteurs fonctionnels d'une entreprise (production, administration, ventes, etc.), afin d'être en mesure d'en réaliser une cartographie puis d'étudier de la même manière son système d'information [Longépé 04]. Il s'agit donc de découper le SI en sous-ensembles (zone, quartiers et îlots). Une zone est un regroupement de quartiers qui sont des regroupements d'îlots qui sont des composants homogènes, remplaçables et développable séparément du système (compte tenu de la nature de l'information manipulé). Ils correspondent au niveau plus fin de décomposition du SI. L'ilot est une entité du système correspondant à une finalité fonctionnelle et comprend des traitements et des accès à des données pour cette finalité. Un îlot émet des résultats normalisés exploitable pour d'autres îlots [Longépé 04].

Plusieurs travaux sont effectués afin de traiter les différentes problématiques liées à ce thème. Par exemple [Longépé 04] présente une méthodologie d'urbanisation de SI où les cartographies sont au cœur de démarche à suivre pour un projet d'urbanisation du SI. Ces cartes sont utilisées pour décrire le SI existant et le SI cible. On distingue quatre cartographies (associées à quatre visions du SI): métier, fonctionnelle, applicative et technique.

La vision applicative est au cœur de l'opération d'urbanisation car le processus d'urbanisation se base sur l'architecture existante pour aboutir à une proposition d'architecture applicative future et propose des étapes permettant de passer de l'une à l'autre.

La démarche commence par placer le projet d'urbanisation du SI dans le cadre général de la stratégie d'entreprise de manière à définir une cible pour le SI, cible qui doit être en phase avec le processus et le métier, lui-même en phase avec la stratégie de l'entreprise. Dans cette phase il faut réaliser le diagramme d'entreprise pour dégager toutes les entités fonctionnelles

(vente et commerciale, administration et finance, production et logistique…) et détailler les activités et le flux d'information échangé entre ces entités, et faire apparaitre les clients, fournisseurs, partenaires….

Une deuxième étape consiste à identifier les processus métier ainsi que les contributions de chaque processus dans la réalisation des objectifs stratégiques de l'entreprise.

La troisième étape consiste à cartographier l'architecture fonctionnelle (réponse à la question « quoi ») sans tenir compte des acteurs de l'organisation. Cette cartographie est constituée en s'appuyant sur des règles d'urbanisme dédiées à l'architecture fonctionnelle. C'est la base sur laquelle le SI sera construit. Ceci fournit la structuration du SI en blocs fonctionnels communicants. L'architecture fonctionnelle est composée de zones, quartiers et îlots fonctionnels.

La quatrième et la dernière étape commence par construire la cartographie applicative existante afin de mettre en évidence les différents composants du SI actuel et les flux entre ces composants applicatifs. Il faut ensuite constituer l'architecture applicative cible en se basant sur l'architecture fonctionnelle cible.

Dans ce travail, la stratégie d'urbanisation est basée sur l'architecture fonctionnelle de l'entreprise sans tenir compte de la logique de processus qui implique une organisation transversale ou horizontale. Cela conduit à l'utilisation de logiciels spécialisés et augmente la rigidité au niveau opérationnel. Ceci rend donc difficile la collaboration. En outre, la vue externe qui pourrait impacter la conception du système a été négligée.

Pour pallier ces limites, dans son approche, l'auteur de [Le Roux 04] propose de voir l'entreprise selon quatre points de vue :

a. **Vue externe**: c'est la vue des partenaires et clients et qui met en évidence et formalise les missions de l'organisation. Cette vue est axée sur les objectifs qui se traduisent par des sollicitations (comme un événement métier). Elle fait le lien entre l'extérieur et une organisation assurant une mission en rendant (vendant) des services. Dans cette vue un événement métier est un élément qui déclenche un processus d'une organisation. On doit donc avoir une connaissance exhaustive des événements métier qu'une organisation doit traiter. Cette vue représente le « Pourquoi » et une partie de « Comment » (les aspects externes du « comment » sont les partenaires et les fournisseurs).
b. **Vue interne** (organisationnelle ou fonctionnelle): On définit les unités fonctionnelles et les compétences associées, les acteurs de l'organisation et leurs domaines de responsabilité, l'organisation des décisions et les habilitations d'accès

aux informations. Elle décrit les agents de l'organisation qui collaborent pour la réalisation des objectifs. Elle met en évidence la façon dont l'organisation prend en charge ses missions. Une organisation est structurée en sous structures définies sur la base des compétences, chaque sous-structure rassemble des acteurs ayant le pouvoir nécessaire pour réaliser des actes de gestion d'une certaine nature et d'une certaine portée. Cette vue considère l'entreprise comme une chaîne de valeur : pour satisfaire un client, l'entreprise produit une valeur ajoutée « objectif » à partir d'éléments obtenus auprès des fournisseurs et grâce à l'aide de partenaires. Pour cela elle réalise des activités successives coordonnées et imbriquées, qui transforment un « produit » d'entré en un produit de sortie à partir d'une sollicitation externe ou « un événement métier ». Cette vue est basé sur l'étude de processus métier qui conduit à une description du métier de l'organisation « qui fait quoi, quand et comment ? ».

c. **Vue informationnelle** : c'est la vue des informations manipulées par l'organisation pour satisfaire ses objectifs. Cette vue représente le « quoi » : quelles sont les informations manipulées et comment les manipuler. De manière générale les informations manipulées dans le contexte d'un système d'information peuvent être classées selon le niveau de formalisation :

   - *Informations structurées* : cette catégorie est associée aux objets métier stockés dans les référentiels de données (les bases de données), les modèles d'informations structurés ou modèles d'objets métier permettent de créer des vues correspondant aux différents types d'échange.

   - *Information non structurées* (l'ensemble du référentiel documentaire) : cette catégorie représente une partie des savoir et savoir faire de l'entreprise qui ne peuvent pas être structurés dans une base de données car il s'agit de questions et thèses à partir desquelles une activité peut être conduite et une information peut avoir un sens.

d. Vue applicative : elle décrit les outils informatiques mis à la disposition des utilisateurs, sensés supporter l'exécution opérationnelle du métier. Elle recouvre deux visions :

   - la vision des applications et des flux inter-applicatif.
   - l'architecture technique : (i.e. les choix techniques et la localisation physique).

Ces quatre points de vue sont introduits dans par un modèle selon quatre cadrans liés les un aux autres permettant de construire une vue complète du SI. Si ce travail présente une

approche claire et simple pour modéliser et décrire l'entreprise pour la problématique de l'urbanisation, une méthode pour exploiter un tel modèle afin de parvenir à une transformation progressive vers le système cible doit être introduite.

Afin d'éviter certaines limitations dans les approches présentées ci-dessus, [Chapron 04] définit une méthodologie de gestion de l'évolution du système d'information prenant en compte un existant tant au niveau des technologies que de l'environnement organisationnel. Ce travail vise à fournir non seulement une méthodologie et des outils formels pour gérer l'évolution du système d'information mais aussi une démarche pragmatique à mettre en œuvre dans l'entreprise pour gérer le changement organisationnel.

Cette méthodologie de gestion de l'évolution est basée sur une vision globale de l'entreprise en s'appuyant sur les processus métier. Il s'agit d'un cadre conceptuel permettant de gérer les évolutions du système d'information (appelé dans cet ouvrage l'urbanisme organisationnel du système d'information). Cette méthode est structurée selon trois étapes de conduite de changement. Elle débute par une représentation dédiée du système d'information en s'appuyant sur la définition d'un méta-modèle spécifique à la problématique d'urbanisation. Il s'agit d'un modèle formel permettant de représenter puis de conduire les processus de changement dans le domaine du système d'information basé sur une approche « orientée métier » permettant la formalisation des processus métier et de leurs contextes organisationnels. Pour cela, on utilise un méta modèle qui met en évidence les interfaces entre les processus et leur environnement. Cette approche de modélisation met en évidence la flexibilité et la souplesse de l'organisation à travers la notion de découplage et la décomposition. L'auteur propose également une approche pour obtenir une gestion affinée des différents processus de l'entreprise et pour comprendre les liens qui les unissent à travers l'identification des zones dans lesquelles les processus sont fortement liés. Ceci permet de les faire évoluer de manière conjointe et facilite la gestion globale du changement.

Dans ce travail, le groupement de processus métier est la méthode choisie pour délimiter les frontières de zones cohérentes associées à des processus fortement liés. Il y a donc un découplage fort entre les zones identifiées. L'auteur part du postulat que les organisations sont composées de multiples éléments interdépendants et que tout changement sur un élément a un impact sur les autres éléments dépendant de cet élément. Pour obtenir une bonne gestion des processus métiers, il est donc nécessaire d'identifier les processus « traversant » les différents domaines organisationnels de l'entreprise, de rationaliser les flux et gérer les dépendances entre eux. La dépendance représente un élément de stabilité de l'entreprise et se traduit par la capacité de maintenir la cohérence entre les éléments de l'entreprise après le changement ou

l'évolution. L'interdépendance représente donc une contrainte pour la gestion de l'évolution. Il faut donc bien maîtriser les interdépendances vis-à-vis de la gestion de l'évolution. la démarche consiste donc à identifier les objets utilisés dans un processus et analyser les correspondances dans d'autre processus pour quantifier les dépendances entre ces deux processus. Pour cela, l'auteur a choisi d'identifier les dépendances entre les processus en identifiant les trois composants majeurs de dépendance à savoir : les acteurs, les ressources matérielles et les informations.

En s'appuyant sur les différents types de dépendances l'auteur propose de les mesurer et de les agréger sous forme d'une matrice de dépendances élémentaires. Les matrices résultantes peuvent être traitées pour produire un graphe de dépendance (interrelations). Une fois ce graphe défini, on applique une méthode de classification hiérarchique afin de définir des regroupements entre ces processus. Le résultat obtenu est une carte de clusters de processus (regroupement de processus fortement dépendants). Cette carte de clusters est utilisée comme un outil par les managers pour le diagnostic du système d'information.

Bien que l'approche proposée ici privilégie une méthodologie basée sur les processus (ce qui devrait améliorer la flexibilité au niveau opérationnel et rendre la collaboration plus facile), l'approche ne tient pas compte de la vue externe. En outre, cette approche utilise quatre éléments pour analyser la dépendance, à savoir: les applications, les acteurs, les événements et les informations. D'autres éléments importants dans le contexte de production tels que les points de stockage, les matières premières et les ressources matérielles (y compris les outils et les machines) n'ont pas été pris en considération.

2.4 Architecture Orienté Service (SOA) et gestion de la sécurité

Réorganiser l'entreprise pour favoriser la collaboration demande de prendre en compte principalement la dimension processus et de développer une infrastructure flexible. Or les systèmes supports du SI sont multiples et ne sont que peu flexibles et interopérables. L'introduction de technologies et architectures orientées services représente une base prometteuse pour mettre en œuvre une infrastructure technique et fonctionnelle de la collaboration entre entreprises car la notion de composition de services et le découplage entre l'implémentation et l'interface d'un composant favorisent la construction d'un espace de travail collaboratif.

Les services sont des applications offertes par des différentes entreprises qui peuvent être encapsulées et présentées comme des services indépendants. Une architecture orientée service repose sur les éléments suivants :

- Le Support d'interfaces des services est basé sur les standards du marché (SOAP, WSDL, UDDI…..) ce qui garanti une relative interopérabilité technologique
- Il est possible d'avoir une découverte dynamique du producteur par le consommateur via un tiers intermédiaire.
- La mise en place d'une approche SOA implique la séparation de l'interface et de l'implémentation. L'interface doit contenir tout ce qu'on doit connaître du service, sa mission, ses interfaces d'entrée et de sortie et les protocoles utilisables pour y accéder : c'est la description du « quoi ». Il est donc possible de localement changer l'implémentation (décrivant le « comment ») sans affecter le reste du système si la nouvelle mise en œuvre respecte les mêmes spécifications d'interface.
- La neutralité technologique est apportée par le découplage de la description de l'interface d'entrée/sortie (fonctionnelle) et de la description des protocoles d'accès (technique).

Les technologies sous-jacentes de ce type d'architecture sont :

- SOAP Simple Object Access Protocol: Langage XML standardisé de requête/réponse avec lequel on peut invoquer les services. Il est utilisé pour encapsuler les données qui permettent au service de s'exécuter ainsi que les données retournées par le service.
- WSDL Web Services Description Language: Langage XML standardisé de description des interfaces de service. Il est utilisé pour connaître les modalités techniques d'accès à un service, le nom des fonctions qu'il expose, les données qu'il attend pour s'exécuter et les données qu'il renvoie en retour.
- UDDI Universal Description, Discovery & Integration: Technologie de structuration d'annuaire dans le but de fournir une description standardisée des services référencés par celui-ci et de permettre la découverte dynamique de ces services.

Au niveau « business », les services doivent être fournis à un niveau d'abstraction qui correspond à l'activité réelle et à une fonction identifiable pour les processus métier. De cette façon des services issus de plusieurs partenaires peuvent être combinés pour former des services composés offrant une plus grande valeur ajoutée. La composition de services métier et l'organisation de l'exécution reposent sur la définition d'un tiers intermédiaire, dit orchestrateur, qui a comme vocation d'assurer la coordination et séquencer les appels de services dans des processus métier soit en réponse à des évènements soit d'une façon

programmée. Toutefois, la construction d'un processus collaboratif sur le paradigme SOA impose de dépasser plusieurs obstacles :

- Dans le contexte de la collaboration les services utilisés résultent d'une combinaison de services issus de plusieurs entreprises, dans ce cas, et bien que les services individuels soient simples, un besoin important concerne la transformation et la médiation des formats de données ainsi que la médiation sémantique entre des services hétérogènes provenant de plusieurs sources.
- L'exécution et la coordination de processus dynamiques et complexes demande la diffusion d'information et des événements vers des destinations multiples. Il faut donc garantir et gérer la sécurité des données et des échanges notamment dans un contexte d'échange avec des partenaires peu connus (dans le cas de collaboration dynamique). Pour cela il convient non seulement de sécuriser l'infrastructure mais aussi de décrire les services pour permettre de présenter la fonctionnalité du service tout en limitant la visibilité (sur ce service) lors de l'exécution au niveau nécessaire pour la coordination. Ceci impose de fournir des services à un niveau d'abstraction qui ne rend visible qu'une interface «métier» et d'enrichir la description du service par les contraintes de sécurité de façon à refléter les contraintes et les règles issues de l'analyse des risques aux niveaux métier et organisationnel. Autrement dit, cela impose l'intégration de la sécurité dans la conception des processus métiers et des systèmes d'information.

L'organisation d'une stratégie de sécurité est souvent réduite à la mise en œuvre d'une infrastructure de sécurisé. Pour cela plusieurs méthodes et normes ont été définies, par exemple, dans DoD Rainbow series [USDOD 85] La politique de sécurité est définie par l'étiquetage des informations selon les niveaux de confidentialité ainsi que par des descriptions des droits d'accès à l'information. Le standard Common Criteria [COC 04] définit un modèle de sécurité et une méthode d'évaluation des risques pour protéger les composants du système. Ces normes fournissent des solutions techniques à des menaces et des vulnérabilités mais ne prennent pas en compte les aspects organisationnels.

En revanche, le standard ISO 17799 [ISO 00], propose des normes de sécurité incluant à la fois des éléments organisationnels et techniques. Ce standard introduit une méthode pour définir, mettre en œuvre et gérer une politique de sécurité cohérente et une méthodologie intégrant les points de vues, organisationnels et techniques, dans une politique de sécurité globale. Malheureusement, cette norme ne fournit pas un cadre pour intégrer les exigences de sécurité dans les modèles de processus métier et ne définit pas une méthode systématique

d'analyse des risques. En outre, toutes les normes de sécurité introduites ci-dessus reposent sur des modèles de connectivité centralisés, néanmoins, ce qui ne correspond pas au paradigme SOA qui a un modèle de connectivité décentralisé. De plus, ces standards visent à définir et à mettre en œuvre une stratégie de sécurité à l'intérieur de l'entreprise et ne fournissent pas de cadre pour définir une stratégie de sécurité pour la collaboration inter-entreprises où les processus métier sont construits dynamiquement.

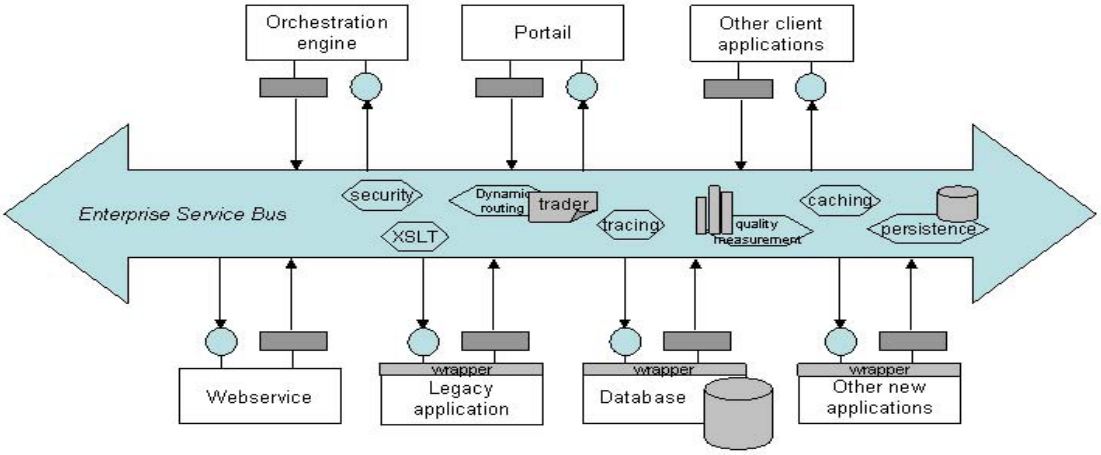

Figure 6 Enterprise Service Bus

Mettre en œuvre une architecture SOA exige une infrastructure de système d'information respectant les principes de SOA, à savoir, fournir les capacités permettant d'invoquer des services indépendamment de la localisation de services et du protocole de communication et de remplacer les implémentations des services sans affecter le client de ces services. Ces capacités sont parmi les nombreuses fonctionnalités fournies par l'ESB (Enterprise Service Bus) (Fig. 6). En plus des fonctionnalités requises pour assurer l'interopérabilité comme la communication asynchrone à base de message et le routage dynamique, la découverte des services par les consommateurs, l'ESB peut assurer d'autres activités non fonctionnelles liées à la gestion de qualité des services, la traçabilité, la gestion des transactions composées et la sécurité [Hérault 05]. Vis à vis des objectifs d'interopérabilité notons que la médiation reste le rôle le plus important de l'ESB. La médiation comprend des tâches telles que la transformation et la synthèse des données provenant de sources multiples et ayant des formats hétérogènes, voire même une analyse plus sophistiquée en utilisant les approches de l'ingénierie de connaissances.

Au niveau opérationnel, l'architecture orientée services rassemble les grandes applications de l'entreprise (dites composites) en services interopérables et réutilisables. Toutefois, pour permettre à ESB d'assurer la composition de services appartenant à des partenaires

sélectionnés « à la volée » dans des scenarios de collaboration à court terme, nous devons étendre ces fonctionnalités avec de nouvelles fonctionnalités pour permettre de sélectionner des service dont les spécification fonctionnelles et non-fonctionnelles correspondent « matchent » aux besoins en termes de fonctionnalité et de qualité de service. Cela nécessite d'introduire un cadre de compositions basé sur la gestion d'ontologies permettant de faire la confrontation sémantique entre des concepts exprimés en utilisant des termes spécialisés appartenant à des ontologies « métiers » différentes. En outre, l'ESB doit être enrichi par une architecture flexible de sécurité capable de confronter les différentes politiques de sécurités des les parties impliqués afin d'assurer la mise en ouvre d'une politique de sécurité globale de façon dynamique.

2.5 Conclusion

Les différents travaux présentés dans cet état de l'art montrent que les approches d'urbanisation du SI couplées à une approche orientée services permettent de doter le SI d'une relative réactivité et d'assurer une interopérabilité technologique. Toutefois, ces approches conduisent à une segmentation « fonctionnelle » du SI de l'entreprise alors que la mise en œuvre de la collaboration impacte la totalité de l'entreprise et son SI.

Pour palier les limites induites par les stratégies traditionnelles d'urbanisation dans le contexte de la production collaborative, nous avons besoin d'adopter une nouvelle stratégie d'urbanisation qui couple une organisation transversale du système de production de l'entreprise à une orientation « service industriel » afin de permettre d'une construction incrémentale des processus en fonction des objectifs industriels à atteindre.

Une telle approche doit prendre en considération la particularité du contexte de production. Cette nouvelle stratégie d'urbanisation doit envisager une organisation « lean » à la fois pour le système d'information et le système de production pour améliorer l'efficacité (en réduisant les gaspillages) et assurer l'interopérabilité non seulement au niveau de système d'information, mais aussi au niveau du système de production. Cela devrait permettre la reconfiguration « à la volée » des processus de production en intégrant les contraintes de production dans la stratégie d'urbanisation, c'est-à-dire en tenant compte de la façon dont l'entreprise transforme les matières premières en produits selon sa propre stratégie de production.

Chapitre 3

3 Nouvelle Stratégie d'Urbanisation

Réorganiser l'entreprise pour favoriser la collaboration demande principalement de prendre en compte la dimension des processus de production et de développer une infrastructure flexible. Pour cela nous proposons de construire des services industrielles qui pourront être composés de manière cohérente (en respectant les logique de production, d'approvisionnent…). Cette stratégie permet de construire les processus collaboratif de manière incrémentale par composition de services et donc d'éviter les organisations trop statiques. En outre, il importe que cette stratégie d'urbanisation soit compatible non seulement avec l'organisation fonctionnelle mais aussi avec l'organisation « horizontale » décrivant les liens entre les unités fonctionnelles mises en œuvre pour fournir le service.

3.1 Urbanisation de l'Entreprise, Vers une Gestion de Service Industriels

Notre objectif est d'identifier des services « industriels » qui seront publiés pour permettre via la composition d'organiser des collaborations cohérentes. Pour cela nous proposons d'organiser des groupes logiques de ressources de production (îlots de production), de gérer les flux d'échange entre ces îlots et d'intégrer l'ensemble des services nécessaires pour réaliser un produit final « à la demande ». Ceci conduit à définir une organisation du système de production qui permet de construire d'une manière dynamique et incrémentale des processus dans une logique « bottom-up ».

Pour organiser ces services il faut réaliser un regroupement « logique » des ressources de production. Pour cela, nous nous sommes basés sur les principes de l'urbanisation de SI pour définir une démarche d'urbanisation du système de production. En effet, l'urbanisation du système d'information consiste à découper le SI en modules autonomes et à définir des zones d'échange d'informations entre modules. Ceci permet de découpler les différents modules pour qu'ils puissent évoluer séparément tout en conservant leur capacité à interagir. On peut ainsi garantir la capacité à construire et à intégrer des sous-systèmes d'origines diverses et donc renforcer la capacité SI d'interagir avec d'autres SI.

Pour ce qui concerne les contraintes de production, notre démarche est construite pour permettre de composer des services nécessaires pour réaliser un produit finis en réponse aux demandes des clients. De manière similaire aux approches présentées dans le cadre de l'urbanisation du SI, notre démarche se base sur les étapes suivantes:

- Cartographier les processus de productions
- Identifier les interdépendances entre les processus de production.
- Identifier l'implantation cible "idéale" des unités de productions en basant sur les deux principes suivants :
    - Autonomie de production : le sous-système de production doit être aussi indépendant que possible des autres systèmes
    - Regroupement selon les dépendances entre les produits :
        a. Identifier les matières premières partagées (nécessaires pour différentes unités de production).
        b. Identifier les machines et les outils partagés.
        c. Identifier les compétences partagées.
        d. Identifier les unités de production dont les produits semi-finis ou finis sont nécessaires pour d'autres unités de production.

Pour cela, un ensemble de règles doit être identifié pour permettre la décomposition et la recomposition du système de production de manière à satisfaire les contraintes suivantes :

a. Délimiter la responsabilité de chaque unité de production :
    - Une unité de production peut réaliser un produit du début à la fin, sans avoir besoin d'effectuer de traitements dans d'autres unités de production.
    - L'unité de production qui offre un produit est responsable de la qualité de ce produit.
b. Assurer l'autonomie des unités de production :
    - Les produits réalisés par chaque unité de production doivent satisfaire les standards et les normes reconnues.
c. Assurer l'interdépendance entre les unités de production :
    - Chaque unité de production a une seule unité de gestion des flux.
d. Simplifier le contrôle de flux et de gestion de la production :
    - Chaque unité de production a une seule unité de contrôle.
    - Chaque unité de production a une seule unité de gestion des ressources.

- Chaque objet dans le système a une référence unique définie selon sa nature.

e. Maîtriser la transformation des produits :

- Toute transformation d'un produit est confiée à une unité de production.
- Chaque opération réalisée sur un produit dans une unité de production est associée à une date.

Dans notre approche, nous distinguons deux types d'unités de production : l'îlot de production et le point de consolidation. Un îlot de production est un « atelier » dans lequel un ou plusieurs produits sont réalisés. Il contient des éléments liés à ces produits. Un point de consolidation est un regroupement de compétences, des points de stockage, des outils et des machines qui sont utilisés par de multiples îlots de production.

En nous basant sur l'identification et l'analyse des interdépendances au niveau des produits, nous identifions et isolons des groupes de ressources ayant un niveau important d'interdépendance. L'identification de ces groupes permet de délimiter les responsabilités de chaque unité de production. Afin d'apporter un niveau de visibilité suffisant, nous introduisons la notion d'objet métier défini comme un ensemble de ressources qui sont impliquées dans la production d'un produit concret.

Pour identifier les services industriels qui pourront servir de base à l'organisation d'une structure de collaboration, nous proposons deux étapes : tout d'abord une analyse des similarités (phase descendante) puis un processus de regroupement (phase ascendante) selon la complémentarité entre les unités de production.

La première étape consiste à identifier les produits principaux selon lesquels nous allons urbaniser l'entreprise. Nous proposons d'utiliser la décomposition selon le diagramme de causes et effets dit « diagramme d'Ishikawa » (Fig. 7) utilisé pour la décomposition et la représentation graphique des modèles d'objectifs. Adapté à notre contexte, il permet la décomposition et la représentation graphique des modèles des produits à partir des processus de production des produits finaux jusqu'aux processus de production des produits élémentaires, qui sont des produits réalisés par un seul processus élémentaire. L'utilisation du diagramme d'Ishikawa est justifiée par le fait qu'il permet une cartographie de processus de la production sur laquelle nous pouvons nous baser pour identifier les interdépendances entre les processus de production.

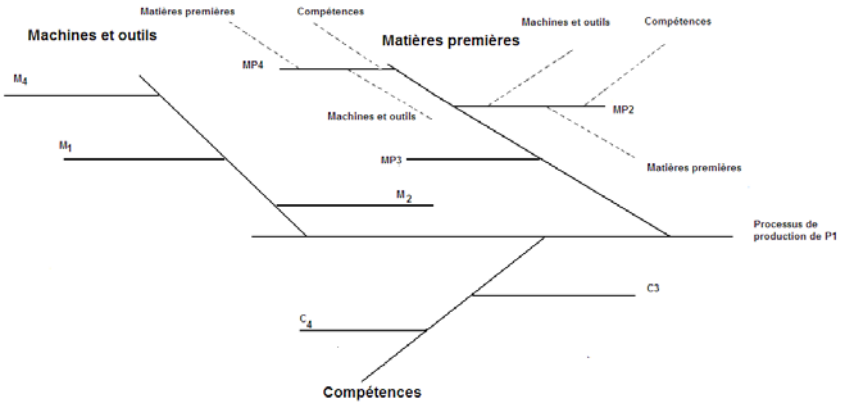

Figure 7 Diagramme Ishikawa

Après avoir réalisé cette étape, nous devons classifier les produits élémentaires, pour identifier les groupes de produits les plus proches puis former les îlots selon un critère de ressemblance.

Une deuxième étape porte sur le regroupement des îlots dans des îlots plus grands selon le critère de complémentarité et de partage jusqu'à arriver à couvrir la totalité du système de production.

Dans notre approche inspirée de la méthode de McCormick [Javel 04], nous regroupons les données de départ dans une matrice (tableau 2) dans laquelle les lignes représentent les produits et les colonnes représentent les ressources. Tous les types de ressources sont pris en considération : les machines (M), les ressources humaines (C) et les matières premières impliquées dans les processus de fabrication (MP). Notons que la prise en considération des matières premières permet de préciser la répartition des points de stockages dans les différents ateliers.

Dans chaque cellule du tableau, on note « + » si l'objet concerné (associé à cette colonne) est utilisé pour produire le produit correspondant à cette ligne. Nous indiquons « - », sinon.

La deuxième étape consiste à réorganiser les lignes pour rapprocher les lignes semblables (qui contiennent dans les mêmes champs un maximum de (+)) en mettant la ligne la plus dense au milieu du tableau. Puis nous déplaçons les colonnes, en mettant les colonnes les plus denses en premier, sans prendre en considération la nature de chaque colonne (c.à.d. sans prendre en compte la nature des objets associés à cette colonne) et nous reclassons les colonnes selon les nombres de « + » qu'elles contiennent dans un ordre décroisant. Les groupes de « + » permettent de repérer les objets partagés. Le tableau 1 montre les sorties de ces deux étapes. Ce tableau représente le « plan d'occupation des sols (POS) » cible, issu de la démarche d'urbanisation. Nous constatons que dans un premier temps les outils et les machines et les

points de stockage liés à un seul produit sont regroupés dans un ilot consacré à ce produit, alors que les ressources partagées sont réparties de façon à souligner l'implication de ces ressources dans les processus de production des différents produits.

Tableau 2 Regroupement d'Objets Métiers

Groupement Initial	M1	M3	M2	M4	MP1	MP2	MP3	MP4	C1	C2	C3	C4	C5
P1	+	-	+	+	-	+	+	+	-	-	+	+	-
P2	+	+	+	+	+	+	+	-	-	+	+	+	-
P3	+	-	-	+	+	+	-	-	+	+	-	+	+

Groupement Final	M1	M4	C4	MP2	M2	MP3	C3	MP1	C2	C5	M3	MP4	C1
P2	+	+	+	+	+	+	+	+	+	+	+	-	-
P1	+	+	+	+	+	+	+	-	-	-	-	+	-
P3	+	+	+	+	-	-	-	+	+	-	-	-	+

Une fois "urbanisée", l'entreprise doit ensuite exposer ses services dans des annuaires afin de pouvoir participer à différentes collaborations.

3.2 Organisation incrémentale de processus

D'après [Crowston 98], les business process sont habituellement définis comme des "sequences of goal-oriented actions undertaken by work units or business firms that repeat over time and which are measured in performance terms, such as time, resources or costs". La norme ISO 9000:2000 donne la définition suivante de la notion de processus : « Ensemble d'activités corrélées ou interactives qui transforme les éléments d'entrée en éléments de sortie. Ces éléments sont des objets matériels ou/et des informations ». Enfin, [Harrison 93] définit les processus comme: «any activity or group of activities that takes an input, adds value to it, and provides an output to an internal or external customer».

Notons que toutes ces définitions définissent le processus selon une logique pilotée par les tâches qui est une logique purement algorithmique et assez rigide où les modèles de processus sont explicites. D'autre part, toutes ces définition sont basées sur le fait que les activités, les ressources sont des notions séparées. Or en réalité, ils ne sont pas séparables car dans le contexte d'une entreprise individuelle, ces éléments font partie des ressources d'entreprise et des processus de l'entreprise (tout au moins pendant le temps d'exécution).

Dans le scénario de collaboration dynamique, les processus sont construits d'une façon dynamique. En outre, les modèles de processus ne fournissent que peu d'informations. Ces modèles de processus « pilotés par les tâches » dans une logique plutôt algorithmique sont donc difficilement applicables. Dans une logique de processus collaboratifs définis de

manière « ad-hoc », il serait souhaitable de redéfinir les processus en fonction des buts à atteindre plutôt que par la définition du « comment atteindre ces buts ». Ceci conduit à privilégier une logique « pilotée par les ressources » au lieu du traditionnel pilotage par les tâches. Pour répondre à ces objectifs, nous proposons tout d'abord d'intégrer la notion de ressources et de rôles dans la définition des processus avant de nous intéresser au mécanisme d'évolution d'une tâche générique.

Ceci nous conduit à envisager l'exécution d'une tâche, élément de base d'un processus, dans une logique pilotée par les ressources et non plus seulement dans la logique « algorithmique » (pilotée par les tâches). Dans notre approche, la tâche (ou plutôt son état) est alors considérée comme liée à l'état de ressources. Les évolutions sont contrôlées par des conditions d'activation de la tâche (pré-conditions) pour atteindre des effets prédéfinis (les buts) qui conduisent également au changement d'état de ressources. Ceci permettra d'activer d'autres tâches. Nous appellerons post-condition associée à une tâche, l'état final des ressources impactées par cette tâche. Compte tenu de cette définition, nous considérons qu'un processus est représenté par l'évolution (objective) de l'état de ressources selon le contexte. Cette approche facilite grandement la gestion simultanée du processus et des ressources en même temps.

3.3 Intégration des contraintes d'organisation

3.3.1 Ressource

Les ressources permettent de décrire les objets constituant l'entreprise et nécessaires pour exécuter une tâche. On notera que nous incluons les acteurs humains dans la notion de ressource tout comme les ressources matérielles (marchandises, matières, machines…). Seuls les objets de l'entreprise qui ne sont pas utilisés (d'une manière ou d'une autre) par une tâche ne sont pas considérés. Ainsi, une entité « ressource » est un objet associé à un type qui reflète ses caractéristiques statiques (taille, capacité, couleur, marque…) et des attributs dynamiques qui représentent les interfaces par lesquelles on peut contrôler et paramétrer le comportement et les fonctionnalités supportées par cet objet. Ainsi, nous proposons de classifier les ressources en groupes. Chaque groupe contient des ressources ayant des caractéristiques et/ou comportements semblables.

Par exemple des ressources peuvent être groupées pour faciliter le contrôle de leur coordination, leur partage et la gestion de leurs interdépendances. Pour cela nous proposons la décomposition suivante (Fig. 8) :

- Ressources matérielles : ce groupe est décomposé à son tour en ressources réutilisables (ressources consommables avec garantie d'approvisionnement ou ressource de production) et non réutilisables (que ces ressources soient partageables ou non). Par exemple, l'essence représente une ressource matérielle consommable alors qu'un véhicule peut être considéré comme une ressource de production.
- Ressources non-matérielles: ce groupe est composé de deux sous-groupes, à savoir les ressources passives (données, temps nécessaire pour l'exécution…) et les ressources actives, capables de produire des efforts ou d'avoir des effets sur l'environnement (acteurs, e-services…).

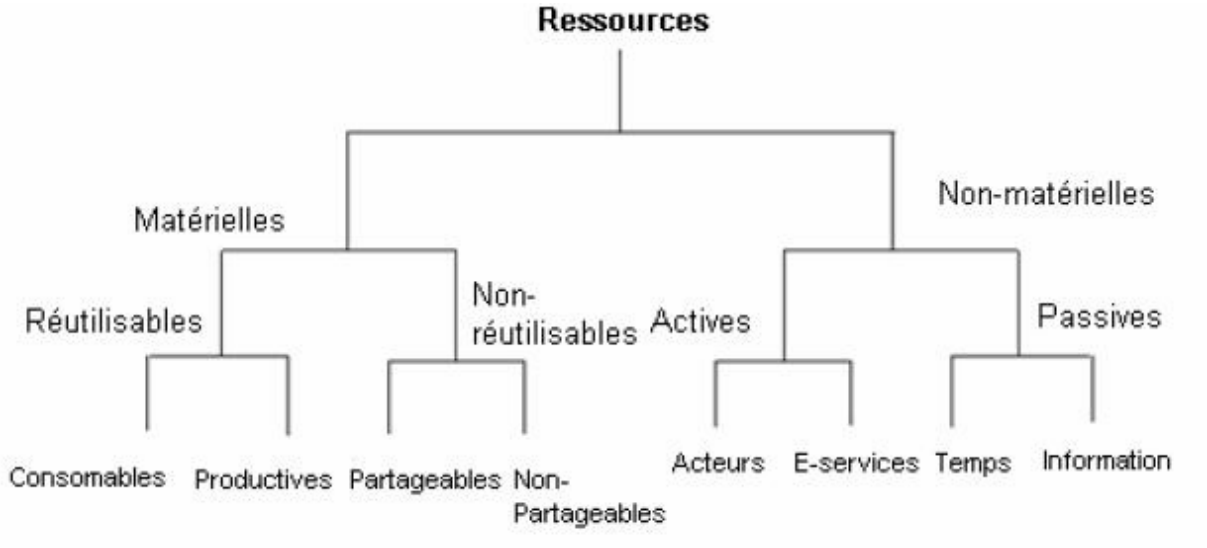

Figure 8 Décomposition des ressources

3.3.2 Organisation de l'orchestration

La contrainte d'autonomie des tâches impose que les ressources soient assignées, contrôlées et gérées par la tâche qui effectue les opérations sur ces ressources conformément aux objectifs qui lui sont donnés. Toutefois autonomie ne signifie pas « isolation » : la tâche doit aussi avoir une activité de coordination (objectifs et actions) avec les autres tâches pour atteindre l'objectif commun mais aussi pour pouvoir satisfaire ses propres objectifs dans des conditions négociées. Ce cahier des charges nous amène à définir la tâche et les concepts liés. La tâche (processus) dans notre vision représente un concept générique, elle peut être associée au processus complet, ou être décomposée en sous-tâches. Une tâche est associée à une transition contrôlant l'évolution de l'état des ressources vers un objectif. Une tâche est donc caractérisée par :

- L'état initial (ressources requises) pour atteindre un but et un événement déclencheur. Ces éléments sont regroupés dans les pré-conditions de la tâche.
- L'état en cours d'évolution est caractérisé par les effets produits ('mesurables goal').
- L'état final (post-condition) décrit l'état prévu des ressources après l'exécution de la tâche. On notera que cet état final peut représenter une partie des pré-conditions d'autres tâches.

Nous ne faisons pas de différence de traitement entre les tâches et sous-tâches. En revanche, le but possède une définition plus riche permettant d'améliorer la synchronisation. Le but (Goal) décrit les effets et objectifs à atteindre. Il peut être décomposé en sous-buts (sous-objectifs) pouvant à leur tour être décomposés en buts primitifs (insécables).

Ces buts, sous-buts et buts primitifs sont associés aux finalités respectivement des tâches, sous-tâches et tâches primitives (non décomposables). Nous introduisons également la notion de but intermédiaire (mesurable goal) associé à des états intermédiaires en cours d'exécution d'une tâche primitive. Ceci nous permet de renseigner l'état d'une tâche, de pouvoir surveiller le comportement réel du processus et de contrôler dynamiquement les états des processus par la production d'événements en respectant le plan d'exécution de la tâche.

Enfin, le plan de tâche gère le comportement des ressources qui interagissent au sein de la tâche mais aussi avec d'autres tâches. Ce plan de tâche est un document comprenant la structure générique de la tâche. Il comporte une description abstraite des ressources demandées ou traitées par la tâche (pré-conditions) et les événements à générer.

Dans notre vision nous considérons que les acteurs (humains et automatisés) sont des ressources de l'entreprise au même titre que d'autres objets. Toutefois, ils ont une importance capitale pour la structure de l'organisation, notamment parce qu'ils donnent à l'entreprise sa capacité fonctionnelle. Autrement dit ce sont les ressources par lesquelles l'entreprise exécute ses processus y compris sur d'autre ressources. Pour cela nous proposons d'encapsuler les notions liées à la structure d'organisation de l'entreprise dans une entité représentable dans le SI, de façon à ce que la modification dans l'organisation soit reflétée dynamiquement dans le SI et que nous puissions modifier la structure des processus à la demande.

Pour atteindre cet objectif nous proposons d'utiliser la notion du rôle. Nous pensons que cette notion qui est utilisée dans le domaine de la gestion du contrôle d'accès (modèle de RBAC) [Ferraiolo 95] peut être employée avec profit dans notre contexte moyennant quelques

restrictions et modifications. Ceci nous conduit donc à formaliser les définitions suivantes (Fig. 9):

- Acteur : peut être une personne ou un service automatisé, il peut avoir différents rôles.
- Rôle : concept lié au travail qui définit l'engagement, le devoir, les permissions, les aptitudes. Des rôles sont assignés aux acteurs actifs (les acteurs humains et les services automatisés). Les rôles représentent les métiers de l'entreprise.
- Aptitude : permet de définir les activités d'entreprise qui peuvent être assignées aux rôles. Chaque aptitude est définie par une activité identifiée par un nom (par exemple: traitement d'une commande d'un client). Il faut définir très précisément les frontières de chaque activité de telle sorte que ces activités évitent les « intersections » entre les aptitudes. Ceci permet donc d'éviter que ces activités ne se retrouvent sur des domaines fonctionnels différents (respect des contraintes d'urbanisation du SI).
- Les règles métier: définissent des contraintes sur la réalisation des activités.
- Opération: désigne un ensemble atomique d'actions conçu indépendamment de l'organisation.
- Processus: se compose d'activités partageant un objectif commun.
- Session: intègre acteur, les rôles et aptitudes, et considérées comme ensemble d'interactions entre les utilisateurs et les ressources.
- Evénement: une occurrence qui peut provoquer un changement dans l'état du rôle.
- Services: ils sont constitués d'un ensemble atomique d'activité qui offrent des prestations bien définies. Un service peut être composé d'autres services.

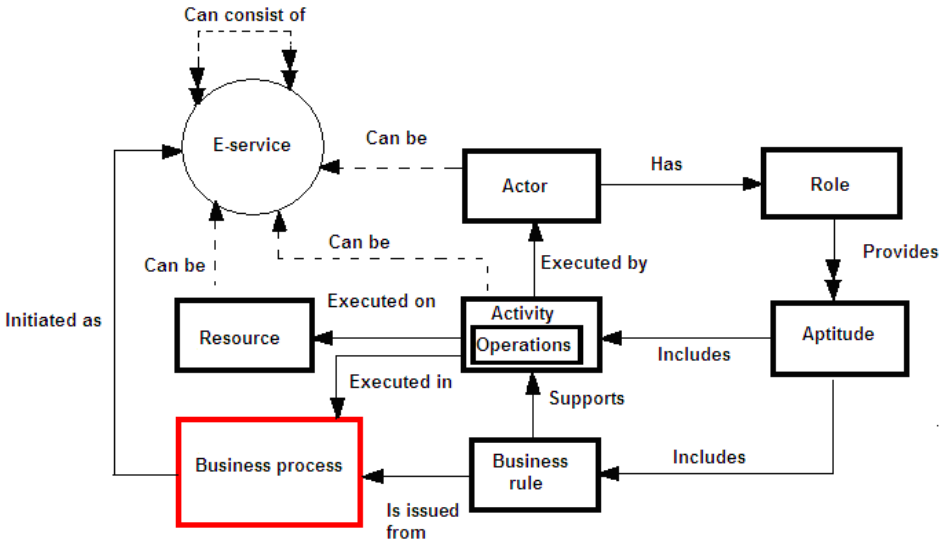

Figure 9 Relation entre les concepts de processus et les concepts de rôle/acteur

3.3.3 Le mécanisme d'évolution d'une tâche générique

Pour prendre en compte ce qui est effectivement fait, nous proposons un mécanisme générique d'évolution d'une tâche (Fig. 10): un événement déclencheur, issu d'un autre processus ou d'une entité externe permet d'activer des tâches (par satisfaction de pré-conditions).

Ceci permet de déclencher le traitement associé qui va faire évoluer l'état de ces ressources (post-condition). On notera qu'une tâche peut produire, consommer, modifier ou emprunter des ressources (cas de ressources réutilisables). Tout au long de l'exécution de la tâche, des buts intermédiaires (mesurable goal) sont produits ce qui permet de contrôler l'avancement globale. On notera que cette exécution « idéale » peut être perturbée par des événements externes conduisant à désactiver une ressource, provoquer une suspension de tâche ou au contraire lancer des reprises.

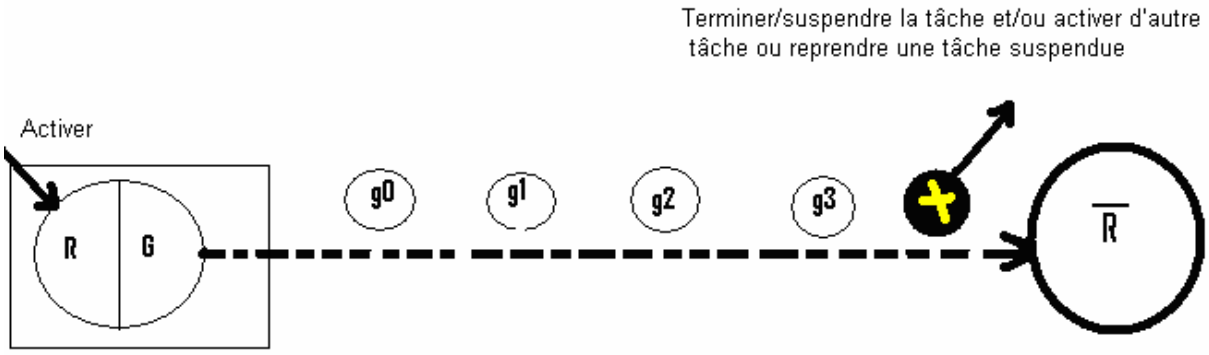

Figure 10 Mécanisme d'évolution d'une tâche générique.

3.3.4 Mécanisme d'Orchestration Intégrant les Concepts Décrits

Le mécanisme de construction incrémentale du processus en se basant sur la demande du client repose sur la sélection des services adaptés aux besoins décrits via des propriétés fonctionnelle et non-fonctionnelle. Pour cela, deux entités sont introduites: le gestionnaire de ressources et l'initiateur du processus. Le gestionnaire de ressources a pour vocation d'organiser et d'exposer des combinaisons de ressources disponibles qui assurent les fonctionnalités demandées et de les fournir à un client dans le cas d'un accord. Son rôle peut être détaillé à travers les activités suivantes:

- Obtenir les descriptions des objets gérés et indexer ces descriptions.
- Surveillance des objets gérés, afin d'obtenir leur état.
- La prise de décision concernant l'affectation des objets gérés en s'appuyant sur les informations collectées en temps réel et sur la politique de gestion des ressources.

- Prendre toutes les mesures nécessaires pour la construction d'une combinaison de ressources en réponse à une demande
- Redéfinir le comportement générique des ressources gérées.

L'initiateur du processus prend en charge la négociation avec les autres entités afin d'obtenir les ressources nécessaires pour l'exécution du processus tout en générant des événements liés à des objectifs mesurables mentionnés dans le plan d'exécution des différentes tâches.

Pour permettre de sélectionner les combinaisons de ressources adéquates de manière pertinente, il importe de les définir précisément et publier leurs descriptions dans un annuaire. Cette description doit reposer sur une ontologie « métier » permettant de décrire à la fois, les aspects fonctionnels et non-fonctionnels des objets du système de production. Les contraintes relatives aux propriétés fonctionnelles (contraintes matérielles, règles de gestion…etc.) et non-fonctionnelles (contraintes sur la qualité, sécurité, maturité des processus) sont, ainsi, intégrées dans les descriptions de l'interface de services pour permettre à la sélection de services.

Nous avons proposé une stratégie d'urbanisation d'entreprise et du SI. Basé sur l'approche SOA et sur la dimension « processus », et qui permet de satisfaire la contrainte d'agilité et l'interopérabilité à la fois pour le système d'information et pour l'organisation structurelle de l'entreprise et son système de production ce qui permet d'identifier et d'exposer des services « industriels » composable. Il devient alors possible de reconstruire les processus de production manière incrémentale pour s'adapter au contexte de la collaboration. Toutefois, les services industriels doivent être analysés afin de construire un modèle donnant une visibilité globale sur le niveau de sécurité à atteindre qui régit l'accès aux différentes interfaces dans un service composé de manière à assurer la sécurité pour l'ensemble des services. Pour cela, nous devons intégrer les besoins et contraintes de sécurité dans les définitions des services.

Chapitre 4

4 Sécurité pour Une Architecture Collaborative Orienté Service Industriels

4.1 Introduction

Du fait de l'ouverture induite par le business collaborative et du caractère dynamique de l'organisation de collaboration « à la volé » une gestion contractuelle et distribuée de la sécurité doit être mise en œuvre. Ces contraintes conduisent à la mise en échec de l'approche traditionnelle « des îlots de sécurité » puisque les mécanismes et politiques sécurité et de vérification des identités et des autorisations peuvent varier selon les organisations et les systèmes impliqués. Le problème est double. D'une part l'approche service est limitée par les couches d'orchestration et composition des services qui fonctionnent au niveau technique et cachent les détails de l'implantation technologique des utilisateurs. D'autre part, puisque chaque service abstrait l'identité de l'utilisateur et son contexte vis-à-vis des applications sous-jacentes, les utilisateurs peuvent accéder à des services situés dans des systèmes différents et à des moments différents. Les applications sous-jacentes ne connaissent alors plus le contexte global de l'utilisateur, contexte dont elles ont pourtant besoin pour autoriser les actions demandées. Par conséquent, les entreprises utilisant le paradigme de SOA ont besoin d'une gestion unifiée d'identité et d'une infrastructure de sécurité qui régit l'accès aux différentes interfaces dans un service composé de façon à assurer la sécurité pour tous les systèmes impliqués. Cela impose l'intégration des contraintes de sécurité dans les descriptions des services et l'utilisation d'une infrastructure adaptée mettant en œuvre la sécurisation demandée.

Au niveau de l'organisation des processus, la sélection de services doit aussi prendre en compte les préférences et besoins contextuels en termes de sécurité pour offrir le niveau de protection demandé.

4.2 Annotation Sémantique et Sélection à base de Qualité de Protection

Dans le paradigme SOA, la sélection des services et l'interaction entre les consommateurs et fournisseurs des services sont régis par les descriptions des interfaces des services. La mise en œuvre de la découverte et de la sélection de services appropriés sont déléguées à l'orchestrateur à travers un processus de confrontation entre la description des services et les exigences du consommateur. En plus des exigences fonctionnelles, un consommateur de service spécifie (ou peut spécifier) des exigences et des préférences non fonctionnelles. Ceci

conduit donc le fournisseur de service à annoter les descriptions de ses services par des propriétés non fonctionnelles décrivant la QoS ou la sécurité.
Cette extension des critères de sélection avec de critère non-fonctionnels impose de pouvoir traiter des règles d'autorisation qui prennent en compte l'identification des « utilisateurs » et leurs attributs (rôles, identités… etc.) nécessaires pour prendre la décision d'autorisation en fonction des besoins de confidentialité exigés par le service. En outre, la participation dans un scénario de collaboration impose de « signer » des accords contractuels non seulement sur les conditions de qualité de service (QoS) ou qualité de protection (QoP) mais aussi sur les conditions effectives de « réalisation » de cette collaboration. Cela est généralement assuré en termes de Service Level Agreement (SLA) afin de permettre une confrontation sémantique entre les offres et les exigences des consommateurs tant lors de la sélection que lors de la négociation.

Typiquement, le consommateur de service exprime ses besoins et exigences liés à la sécurité en définissant des objectifs à atteindre, alors que les fournisseurs expriment leurs offre et garanties selon les termes de leurs politiques de sécurité. Afin d'être en mesure de confronter les deux descriptions, le consommateur et le fournisseur doivent partager un vocabulaire commun précisés dans des ontologies sécurité.

Nous avons défini une ontologie de sécurité organisée en deux sous-ontologies : une ontologie décrivant les concepts de sécurité (Concepts ontology) (Fig. 11) et une ontologie décrivant les objectifs de protection (Objective ontology) (Fig 12). Ces deux sous-ontologies sont liées par des relations, montrant comment les concepts de sécurité peuvent remplir un ou plusieurs objectifs de sécurité (Fig. 13).

Dans un souci de simplicité, nous définissons cinq objectifs pour exprimer les exigences de sécurité liées aux fonctionnalités de base de la sécurité :

- Confidentialité.
- Non-répudiation.
- Intégrité.
- Anonymat.
- Ouverture.

Ces objectifs permettent de répondre aux services de base de sécurité. Notons que le service de disponibilité est pris en compte au niveau de l'infrastructure en intégrant des mécanismes de protection alors qu'au niveau « business », il est assuré à travers la gestion de la QoS.

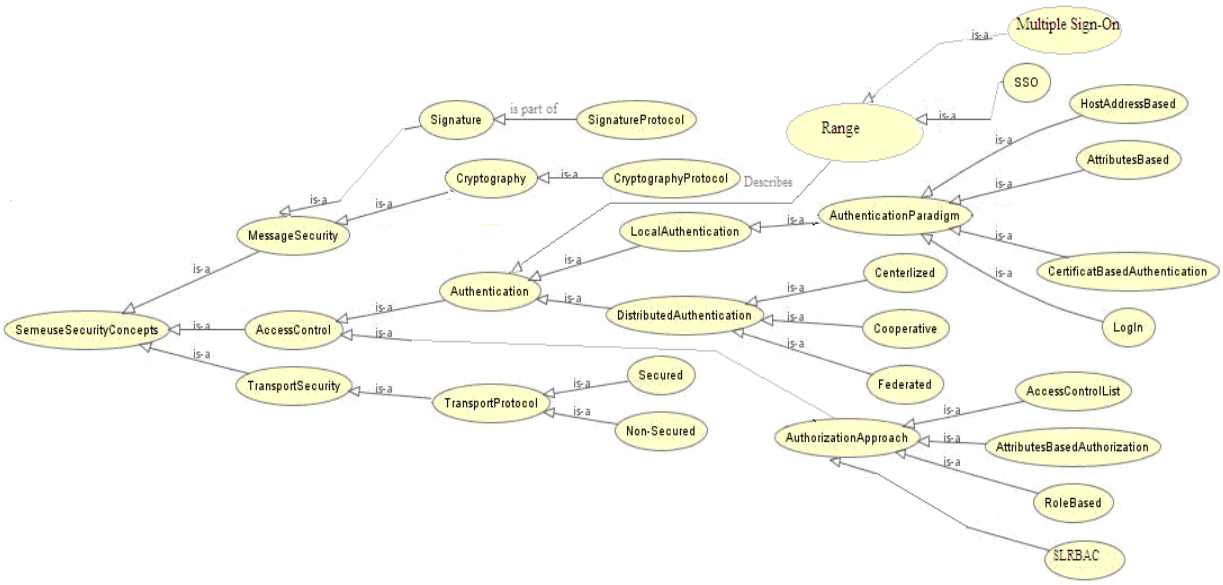

Figure 11 Ontologie de Concepts de Sécurité

Dans notre proposition, les besoins et les préférences liés à la sécurité coté consommateurs sont exprimés en utilisant la sous-ontologie des objectifs de sécurité et les fournisseurs de services expriment leurs offres de sécurité en utilisant la sous-ontologie des concepts de sécurité. Les relations entre les concepts de sécurité et les objectifs de sécurité sont utilisées pour choisir les services qui peuvent satisfaire les besoins de sécurité du consommateur (parmi les services dont les offres fonctionnelles correspondent à la requête du consommateur). Après avoir sélectionné les services appropriés, les termes de QoP convenus sont sémantiquement exprimées à l'aide de l'ontologie de sécurité dans les un accord en utilisant les spécifications de WS-agreement, notamment « wsag:BusinessLevelObjectif » et <wsag:GuaranteeTerm>.

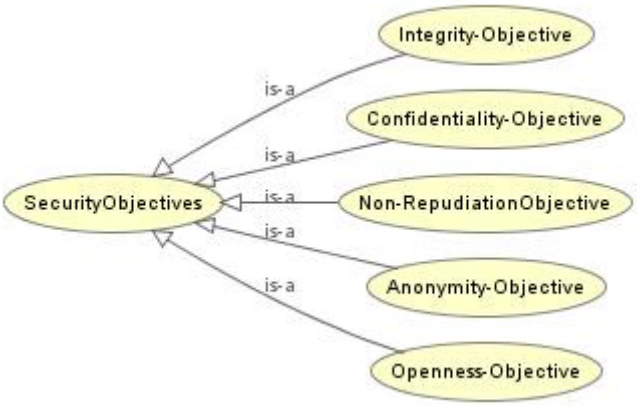

Figure 12 Ontologie des Objectifs de Sécurité

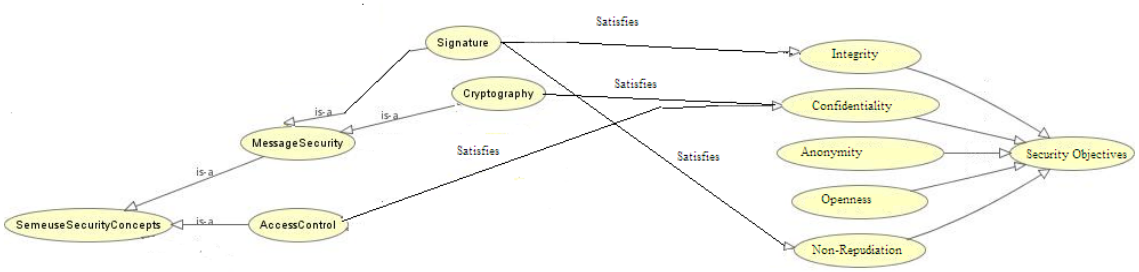

Figure 13 Relations entre les Sous-ontologies

Chapitre 5

5 Mise en œuvre

5.1 Introduction

Pour mettre en ouvre notre organisation basée sur les services industriels, il importe de disposer d'un middleware enrichi d'un niveau sémantique pour traiter les propriétés fonctionnelles et non-fonctionnelles. A partir de la plateforme SEMEUSE construite autour de l'ESB distribué PEtALS supervisé par Dragon (une solution de gouvernance de SOA), nous avons proposé de prendre en considération les préférences de sécurité comme des propriétés non fonctionnelles des services et de les intégrer dans les descriptions des services comme des annotations sémantiques (ce qui permettra de les intégrer directement dans le processus de sélection sémantique des services) ainsi que dans le Service Level Agreement (SLA), ce qui permet de construire un ensemble de contrats définissant le cadre global de la collaboration.

5.2 Architecture de Sécurité

L'architecture de sécurité est conçue comme un service externe « Pluggable », ce qui permet de le connecter à la fois à l'ESB et au système de gouvernance. Le service de sécurité est divisé en deux couches, la couche sémantiques et la couche services. La couche sémantique est chargée de prendre en compte les préférences de sécurité ainsi que les stratégies d'autorisation des parties impliquées. Le niveau sémantique comprend trois composantes (Fig. 14):

- Le gestionnaire de QoP: il est chargé d'orchestrer les différents composants de sécurité: elle doit d'abord activer le comparateur QoP pour sélectionner des services répondant aux objectifs de sécurité, avant d'extraire les rôles des différents acteurs pour activer le système d'autorisation. Enfin, il active le médiateur de sécurité situé dans la couche de service chargé de composer les composants techniques de sécurité.
- Le gestionnaire d'autorisation fédéré a pour vocation d'extraire la stratégie d'autorisation (à partir des exigences stockées dans la description du service) et de lancer le processus d'autorisation sur les différents nœuds hébergeant les services impliqués dans la chaîne de services.
- L'intermédiaire de sécurité : il est utilisé pour intégrer les exigences de sécurité lors de la composition de la chaîne de service. Il gère les paramètres de sécurité afin

d'instancier les composants de sécurité technologique appropriés dans la couche de service.

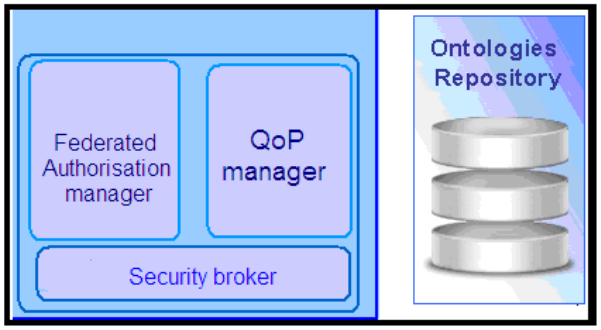

Figure 14 Couche Sémantique

Le niveau service est décomposé en deux sous-niveaux (Fig. 15):

- « Platform-Independent Component » implémente la vue globale sur l'authentification (mis en œuvre grâce à une gestion d'identité fédérée (Single Sign On), la gestion des transports sécurisés et la signature des messages.
- « Plate-forme-dependent Component » : a comme vocation de lier l'interface avec le protocole de communication approprié. Il gère également les cadres d'authentification mis en œuvre dans le bus « local ». Il est coordonné grâce au médiateur de la sécurité (couche sémantique).

A partir de l'étiquetage des messages, un module de routage dédié permet de sélectionner les protocoles de communication approprié (SSL, Dream, etc…).

Pour ce qui concerne la gestion des authentifications et des autorisations, nous avons retenu une approche « fédérative » organisé en royaumes. Un royaume est attaché à chaque noeud (PEtALS ou Dragon). Dans chaque royaume, le référentiel SSO comprend les informations d'authentification de ce royaume. Le lien entre royaumes est régi grâce au composant « gestionnaire de fédération ». Ce composant est utilisé pour vérifier si l'utilisateur est déjà enregistré ou pas dans un domaine référencé. Les relations de confiance entre les domaines sont utilisées pour identifier et échanger des informations sur les acteurs enregistrés. Une fois qu'un acteur est enregistré dans le répertoire SSO, un jeton est renvoyé et stocké. Grâce à un processus de « SignOff » la déconnexion d'un utilisateur est prise en compte sur l'ensemble des nœuds. Grâce à ce mécanisme, une seule authentification suffit jusqu'à ce que l'utilisateur se déconnecte.

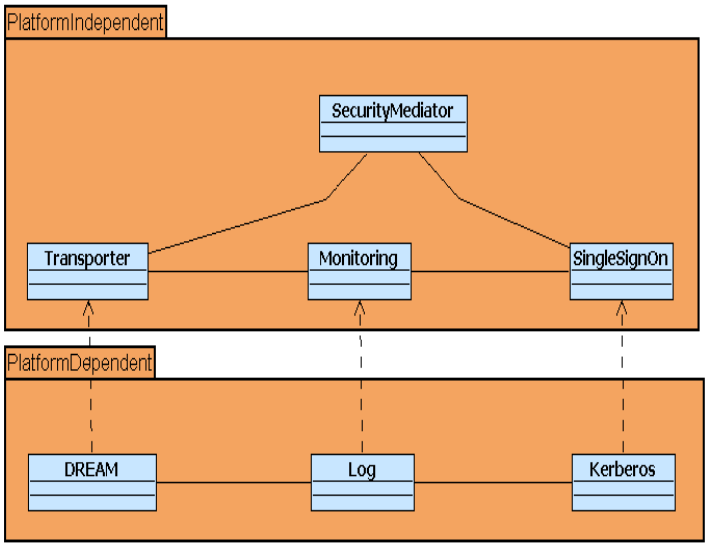

Figure 15 Couche de Service

Chapitre 6

6 Conclusion

Pour faire face aux contraintes du Collaborative Business, nous avons privilégié une démarche pragmatique permettant de répondre aux problèmes d'interopérabilité aux niveaux business organisationnel et technologique.

L'architecture retenue dans notre démarche d'urbanisation d'entreprise repose sur un modèle orienté services. Nous avons spécifié un modèle de service industriel construit par regroupement de toutes les fonctions nécessaires autour de la fabrication du produit ce qui permet de composer des services répartis sur différents sites pour construire un processus collaboratif. Or cette visibilité fonctionnelle ne suffit pas pour assurer la sélection de partenaires une attention particulière doit être accordée aux problèmes de sécurisation. Dans ce cadre, nous avons construit un modèle donnant une visibilité globale sur le niveau de sécurité à atteindre qui régit l'accès aux différentes interfaces dans un service composé de manière à assurer la sécurité pour l'ensemble des services. Cela est assuré par l'intégration des besoins et des contraintes de sécurité dans les définitions des processus métier, ainsi que dans les composants de la structure organisationnelle et technique de manière à proposer des accords globale de qualité de protection entre les partenaires.

La mise en œuvre de cette architecture repose sur la construction d'un middleware de services industriels capable de prendre en compte et assurer la sécurité des services composés et issus d'entreprises différentes. Ce bus ajoute aux fonctionnalités d'un ESB classique les services principaux nécessaires pour une composition orienté services métiers au niveau sémantique et non plus seulement technique tout en prenant compte du contexte global ainsi que les préférences de la sécurité des différentes parties impliquées. Les modules de notre bus de services sont développés au dessus de l'ESB open source « PEtALS » au sein du framework SEMEUSE.

References

[Abele 09] Abele, E. et al., Apache Online Documentation, Authentication, Authorization, and Access Control, available at: http://httpd.apache.org/docs-project/, last visit 1/11/2009.

[Abowd 99] Abowd, G., Dey, A., Brown, P., Davies, N., Smith, M., Steggles, P., Towards a Better Understanding of Context and Context-awareness, proc. HUC'99, Lecture Notes in Computer Science, Vol. 1707, (Springer Berlin) p.p. 304-307, 1999.

[AFGI 91] AFGI, Dictionnaire Anglais-Français, Français-Anglais des termes de gestion industrielle, et leurs définitions, AFGI, 1991.

[AFNOR 90] AFNOR, Norme française NF-X 50-150, Gestion de la production : Analyse de la chaîne de valeur, 1990, available at www.afnor.org, last visit 1/11/2009.

[AFNOR 91] AFNOR X50-310- Norme Française NF X50-310, Organisation et Gestion de la Production Industrielle, Concepts Fondamentaux de Gestion de Production, 1991.

[Agrawal 05] Agrawal, R., Kiernan, J., Srikant, R. & Xu,Y., Xpref :A Preference Language for P3P, Computer Networks Vol. 48, Elsevier B.V, p.p. 809-827, 2005.

[Alam 06] Alam, M., Breu, R. & Hafner, M., Modelling Permissions in a (U/X)ML World, proc. ARES 2006, IEEE Computer Society, Washington, p.p. 685 – 692, 2006.

[Alberts 01] Alberts, C. & . Dorofee, A., Octave Threats Profile. CERT White paper, 2001 available at: www.cert.org/archive/pdf/OCTAVEthreatProfiles.pdf, last visit 1/11/2009.

[Anderson 05] Anderson, A., Multiple Resource Profile of eXtensible Access Control Markup Language (XACML) v2.0, OASIS Standard, 2005.

[Andrieux 07] Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S. & Xu. M., Web Services Agreement Specification. Open Grid Forum (OGF), Grid Resource Allocation Agreement Protocol (GRAAP) WG, March 2007, available at: http://forge.gridforum.org/sf/projects/graap-wg, last visit 1/11/2009.

[Antonelli 99] Antonelli, M., Building Information Systems to Integrate the Manufacturing Supply Chain, Master Report, Massachusetts Institute of Technology, USA, 1999.

[ASQ] ASQ, Glossary of American Society for Quality, available at: www.asq.org/abtquality/glossary.cgi, last visit 1/11/2009.

[Aura 99] Aura, T., Distributed Access-Rights Management with Delegation Certificates Lecture Notes in Computer Science Springer Berlin / Heidelberg Volume 1603/199, , p.p. 211-23, 1999.

[Blackwell 98] Blackwell Encyclopedic Dictionary of Operations Management, , 272 pages, Blackwell Publishing, 1997.

[Biennier 05] Biennier, F. & Mathieu, H., Security Management: Technical Solutions v.s Global BPR Investment, Schedae informatica, vol. 14, p.p.13-34, Krakow, Poland, 2005.

[Blondel 97] Blondel, F., Gestion de la Production, 486 P, Paris, Dunod, , 1997.

[Brende 05] Brende, C., Buche, N., Delayre, S., Mocaër, A. & Nevers, J., SOA et urbanisme, Le rôle des Architectures Orientées Services dans l'alignement métier des Systèmes d'Information, Unilog Management White paper, 2005.

[Cahill 08] Cahill, C., Canales, C., Le Van Gong, H. A., Madsen, P., Maler, E., Whitehead, G., Liberty Alliance Web Services Framework: A Technical Overview, Version 1.0, Liberty Alliance Project, 2008.

[CCO 04] CCO, Common Criteria Organisation, Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and General Model Version 2.2, 2004, available at www.hds.com/fr/solutions/storage.../data-security/cc.html, last visit 1/11/2009.

[Chappell 04] Chappell, D. A., Enterprise Service Bus., 288 p, O'reilly Media, 2004

[Chapron 04] Chapron, J : .L'urbanisme Organisationnelle, Méthodes et Aide à la Décision pour Piloter l'Evolution de Système d'Information, Rapport de Thèse, Ecole Nationale Supérieure des Mines de Saint-Etienne, France, 2004.

[Chen 07] Chen, T.Y., Chen, Y.M, Wang, C.B., Chu, H.C. & Yang, H., Secure Resource Sharing on Cross-organization Using a Novel Trust Method, Robotics and Computer-Integrated Manufacturing Vol. 23, p.p. 421-435 Pergamon Press, USA, 2007.

[Choppy 04] Choppy, C. & Heisel, M., Une Approche à Base de Patrons pour la Spécification et le Développement de Systèmes d'Information, Report, Laboratoire d'Informatique de Paris-Nord, LIPN, 2004.

[Christensen 01] Christensen, E., Curbera, F., Meredith, G.& Weerawarana, S.,Web Services Description Language (WSDL) 1.1, W3C Note, 2001, available at: www.w3.org/TR/wsdl, last visit: 1/11/2009.

[Chu] Chu, C., 5S Home page, School of Information Sciences and Technology, The Pennsylvania State University, USA, http://net1.ist.psu.edu/chu/wcm/5s/5s.html, last visit: 1/11/2009.

[Clement 04] Clement, L., Hately, A., von Riegen, C., Rogers, T., et al. OASIS, Organization for the Advancement of Structured Information Standards, UDDI Spec Technical Committee Draft, 2004.

[Covington 01] Covington, J. M., Moyer, J. M. & Mustaque, A., Generalized Role-Based Access Control for Securing Future Applications, Proc. ICDCS'01, p.p. 391-401, IEEE computer Society, 2001.

[Cortada 95] Cortada, J.W. & Woods, J. A., Encyclopedia of Science & Technology – 8th Edition, New York: McGraw-Hill, 1995.

[Courtois 03] Courtois, A., Martin-Bonnefous, C., Pillet, M., Gestion de production, 454 p, Paris 2003.

[Cranor 03] Cranor, L., P3P: Making Privacy Policies More Useful, IEEE Security and Privacy journal, Vol. 1, IEEE press Security and Privacy, p.p. 50-55, 2003.

[Crowston 98] Crowston, K. & Osborn, C., A Coordination Theory Approach to Process Description and Redesign, IBM Systems Journal, 37(2), p.p. 227-245, 1998.

[Delen 03] Delen, D., Benjamin, P., Towards a Truly Integrated Enterprise Modelling and Analysis Environment, Computers in Industry, volume 51 , Issue 3, P.P. 257 - 268 , 2003.

[Dierks 08] Dierks, T.& Rescorla, E., Internet Engineering Task Force, Network Working Group, The Transport Layer Security (TLS) Protocol Version 1.2, 2008.

[Duncan 95] Duncan, W. L., Total Quality: Key Terms and Concepts, 192 P., AMACOM, New York, 1995.

[Eom 07] Eom, J., Park,S., Han, Y. & Chung, T., Risk Assessment Method Based on Business Process-Oriented Asset Evaluation for Information System Security, proc. ICCS 2007, Lecture Notes in Computer Science, Vol. 4489, p.p 1024-1031, Springer, Berlin, 2007.

[Eveaere 99] Eveaere, C., Management de la flexibilité, Economica, 203pages, Dound, Paris, 1999.

[Farlex 09] Farlex, The Free Dictionary, available at: http://www.thefreedictionary.com, last visit: 111/2009.

[Ferraiolo 95] Ferraiolo, D., Cugini,J., Kuhn, R., Role Based Access Control: Features and Motivations, Proc. Annual Computer Security Applications Conference, p.p.554-563, IEEE Computer Society Press, Washington, 1995.

[Fontanil 99] Fontanil, F., Intégration d'Outils de Simulation et d'Optimisation pour le Pilotage d'une Ligne d'Assemblage Multi-produit à Transfert Asynchrone, Ph. D. Dissertation, University of Paris 13, 1999.

[Forder 07] Forder, R. & Duffy, M., TOGAF to MODAF Mapping, Crown Copyright, Document Number C370-EP-01, Issue 2.0, 2007.

[Fox 95] Fox, M., Barbuceanu, M. & Gruninger, M.: An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour, Fourth Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises, p.p. 11-15, West Virginia

University, , 1995.

[Gadomski 98] Gadomski, A.M., Balducelli, C., Bologna S. & DiCostanzo, G., Integrated Parellel Bottom-up and Top-down Approach. Proc. of The International Emergency Management Society's Fifth Annual Conference (TIEMS 98), p.p.19-22, USA 1998.

[Gamma 95] Gamma, E., Helm, R., Johnson, R., Vissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, 416 p., Addison-Wesley, 1995.

[Gou 03] Gou, H., Huang, B., Liu, W., Li, X., A Framework for Virtual Enterprise Operation Management, Elsevier, Computers in Industry, Volume 50 , Issue 3, P.P: 333 – 352, 2003.

[Gunn 07] Gunn, R. A., Matrix Management Success: Method Not Magic, 161 p., Infinity Publishing, 2007.

[Graebsch 05] Graebsch, M., Information and Communication in Lean Product Development, Ph.D. Dissertation, Technical University of Munich, 2005.

[Gross 03] Gross, T., Security Analysis of the SAML Single Sign-on, Browser/Artifact Profile, proc. ACSAC2003, p.p. 298- 307, IEEE Computer Society, Washington, 2003.

[Haas 07] Haas, H., Hurley, O., Karmarkar, A., Mischkinsky, J., Jones, M., Thompson, L. & Martin, R., SOAP Version 1.2 Specification Assertions and Test Collection, Second Edition, W3C Recommendation, 2007, available at www.w3.org/TR/.../REC-soap12-testcollection-20070427/, last visit: 1/11/2009.

[Han 06] Han, J. & Khan, K., Security-oriented Negotiation for Service Composition and Evolution, Proc. APSEC06, p.p. 71-78, IEEE Computer Society, Washington, 2006.

[Harrison 93] Harrison, D. B. & Pratt, M. D., A Methodology for Reengineering Business, Planning Review, England, 21(2): p.p. 6-11, 1993.

[Hay88] Hay, E. J, The Just-In-Time Break through – Implementing The New Manufacturing Basics, 257 p., John Wiley & Sons, New York, 1988.

[Herault 05] Herault, C., Thomas, G., & Lalanda, P.: Mediation and Enterprise Service Bus, a Position Paper, First International Workshop on Mediation in Semantic Web Services p.p. 67-80, Amsterdame, 2005.

[Hohmann 07] Hohmann, C., La flexibilité des entreprises : Réflexions autour d'une mutation, Comparaison entre l'ilot et la ligne de production. Available at : chohmann.free.fr/flexibilite4.htm, last visit 06/2007.

[Hopp 95] Hopp, W. J. & Spearman, M. L., Factory Physics, Foundations of Manufacturing Management, 600 pages, Richard D Irwin, 1995.

[HPM] HPM, Kaizen Blitz - Two Words that Spell Success, High Performance Manufacturing Consulting Inc, www.hpmconsulting.com/, last visit 1/11/2009.

[Hutter 06] Hutter, D., Volkamer, M., Information Flow Control to Secure Dynamic Web Service Composition, proc. SPC 2006, Lecture Notes in Computer Science, Vol. 3934, p.p.196-210, Springer, Berlin, 2006.

[IBM 02] IBM & Microsoft, Security in a Web Services World: A Proposed Architecture and Roadmap, A Joint Security Whitepaper, Version 1.0, 2002 available at: download.boulder.ibm.com/ibmdl/pub/software/dw /library/ws-secmap.pdf, last visit 1/11/2009.

[IFIP 99] IFIP–IFAC Task Force on Architectures for Enterprise Integration Generalised Enterprise Reference Architecture and Methodology, Version 1.6.3, 1999. http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html, last visit 1/11/2009.

[Imai 86] Imai, M., KAIZEN: The Key to Japan's Competitive Success, 260 P., McGraw-Hill; New York, 1986.

[ISO 00] ISO, International Standardisation Organisation, Code of Practice for Information Security Management Information Technology: ISO/IEC 17799:2000 Standard, available at: www.iso.org/iso/catalogue_detail?csnumber=33441, last visit 1/11/2009.

[ISO9000] ISO9000/2000, International Standardisation Organisation, www.iso.org/iso/fr/

[Jackson 00] Jackson, M., Problem Frames: Analyzing and Structuring Software Development problems, 416 p., Addison-Wesley, 2000.

[Javel 04] Javel, G., organisation et gestion de production, 520 p., Dunod, 2004.

[Joiner 94] Joiner, B., Fourth Generation Management, , 289 p., McGraw-Hill, New York, 1994.

[Kearney 07] Kearney, P., Brügger, L., A Risk-driven Security Analysis Method and Modelling Language, British Telecom Technology Journal Vol. 25, p.p.141-153, Springer, 2007.

[Kim 02] Kim, S. & Jang, K.J., Designing Performance Analysis and IDEF0 for Enterprise Modelling in BPR, Elsevier, International Journal of Production Economics, Volume 76, Number 2, 21, pp. 121-133, 2002.

[Kanneganti 07] Kanneganti, R. & Chodavarapu, P. A., SOA Security, 512 p., Manning Publications, 2007.

[Koubarakis 02] Koubarakis, M., Plexousakis, D., A Formal Framework for Business Process Modelling and Design, Journal of Information Systems, Volume 27 , Issue 5 , P.P: 299 – 319, Elsevier,2002.

[Lawrence 06] Lawrence, K., Kaler, C., Nadalin, A., Goodner, M., Gudgin, M., Barbir, A. & Granqvist, N. H., WS-SecurityPolicy 1.2 Committee Draft 01, Organization for the Advancement of Structured Information Standards, OISIS, 2006, available at docs.oasis-open.org/ws.../ws-securitypolicy/.../ws-securitypolicy-1.2-spec-cd-01.pdf, last visit 1/11/2009.

[L'encyclopédie] L'encyclopédie e-Business, le Journal du Net: available at www.journaldunet.com/encyclopedie/definition/325/51/20/business_process_management_system.shtml, last visit 1/11/2009.

[Le Roux 04] Le Roux, B., Desbertrand, L., Guerif, P., Tang, X.: Urbanisation et Modernisation de SI, 214 p., Hermes Science Publications, Paris, 2004.

[Li 02] Li, D., Hu,S., Bai,S., A uniform model for authorization and access control in enterprise information platform. Proc. EDCIS2002, Lecture Notes in Computer Science, vol. 2480, p.p.180-192, Springer Berlin 2002.

[Li 94] Li, H. & Williams, T. J., A Formalisation and Extension of the Purdure Enterprise Reference Architecture and Purdure Methodology, Report Number 158, Published as Part of Engineering Experiment Station Bulletin 143 Series, Purdue Laboratory for Applied Industrial Control, School of Engineering, Purdue University, USA, l994.

[Longépé 04] Longépé, C.: Le Projet d'Urbanisation du S.I, Second Edition,284 p., Dunod, Paris, 2004.

[Lossent, L.] Lossent, L. & Gzara, L., Système de Gestion des Données Technique et Wokflow associé, Université de Cuces, Nancy, France, available at www.cyber.uhp-nancy.fr/.../cours_2_1_1.html, last visit : 1/11/2009.

[Madsen 05] Madsen. P., Maler, E., Wisniewski, T., Nadalin E.T. , Cantor, S., Hodges, J. & Mishra, P., Security Assertion Markup Language (SAML) Executive Overview, version 2.0, Organization for the Advancement of Structured Information Standards, 2005, available at www.oasis-open.org/...php/.../sstc-saml-tech-overview-2%200-draft-10.pdf, last visit 1/11/2009.

[Madsen 08] Madsen, P., Itoh, H., Ishikawa, K. & Arisa, F., Liberty ID-WSF Multi-Device SSO Deployment Guide, Version: 1.0-02, Liberty Alliance Project, 2008, available at: www.projectliberty.org/liberty/.../draft-liberty-idwsf-mdsso-deployguide-v1.0-02.pdf, last visit 1/11/2009.

[McCabe 08] McCabe, F. G., Estefan, J. A., Laskey, K., McCabe, F.G., Thornton, D., Reference Architecture for Service Oriented Architecture Version 1.0, Public Review Draft 1, OISIS, Organization for the Advancement of Structured Information Standards, OISIS, 2008, available

at: docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf, last visit 1/11/2009.

[Mahoué 01] Mahoué, F., The E-World as an Enabler to Lean, Master Dissertation, Massachusetts Institute of Technology, 2001.

[Mayer 95] Mayer, R. J., Menzel, C. P., Painter, M. K., deWitte P. S., Blinn, T. & Perakath, B., Information Integration for Concurrent Engineering IDEF3 Process Description Capture Method Report, University Drive East College, Texas, 1995, available at: www.idef.com/pdf/Idef3_fn.pdf, last visit 1/11/2009.

[Mertins 05] Mertins, K., Jochem, R., Architectures, Methods and Tools for Enterprise Engineering, International Journal of Production Economics, Volume: 98, p.p. 179-188, Elsevier, 2005.

[MIT 2009] MIT, Massachusetts Institute of Technology, Kerberos: The Network Authentication Protocol, 2009, available at http://web.mit.edu/Kerberos/, last visit 1/11/2009.

[Monden 97] Monden, Y., Toyoto Production System – An Integrated Approach to Just-In-Time, 480 P, Georgia, Engineering &Management Press, Norcross, 1997.

[Monden 93] Monden, Y., Toyota Management System - Linking the Seven Key Functional Areas, Portland 222 p., Productivity Press, Oregon, 1993.

[Mondon 05] Mondon, C., Supply Chain Management en PMI : Le chaînon manquant, 378 p., AFNOR, 2005.

[NATO 07] NATO Consultation Command and Control Board, NATO Architecture Framework Version 3, 2007, available at: www.mod.uk/DefenceInternet/.../MODAF/, last visit 1/11/2009.

[Nickulland 05] Nickulland, D., Connor, M., MacKenzie, C., Watson, B., Cowan, M. & Chase, E.: Service Oriented Architecture Whitepaper, Adobe Systems Inc. 2005, available at: www.adobe.com/.../pdfs/Services_Oriented_Architecture_from_Adobe.pdf, last visit 1/11/2009.

[Nightingale 98] Nightingale,D. S., Lean Aerospace Initiative, Massachusetts Institute of Technology, USA, 1998, available at: lean.mit.edu, last visit 1/11/2009.

[NIST 93] NIST, National Institute of Standards and Technology, IDEFØ a Standard for Function Modeling, 183 p., FIPS Publication, 1993.

[OMB 07] OMB, FEA Consolidated Reference Model Document, Version 2.3, The White House Office of Management and Budget, 2007, Available at: http://www.whitehouse.gov/omb/e-gov/fea/, last visit 1/11/2009.

[Preibusch 06] Preibusch, S., Privacy Negotiations with P3P, Proc. W3C Privacy Workshop on Languages for

Privacy Policy Negotiation and Semantics-Driven Enforcement,Ispra, Italy, 2006, available at: www.w3.org/2006/07/privacy-ws/papers/24-preibusch-negotiation-p3p/, last visit 1/11/2009.

[Rafiy 07] Rafiy, S., Dawn, J. & Peter, B, Management of Users Privacy Preferences in Context, proc. IRI 2007, , p.p. 91-97, IEEE Computer Society, Washington2007.

[Ray 06] Ray, I., Kumar, M. & Yu, L., LRBAC: A Location-Aware Role-Based Access Control, Proc.ICISS2006, Lecture Notes in Computer Science Vol. 4332, p.p. 147- 161, Springer, Berlin, 2006.

[Razmerita 05] Razmerita, L., Services Contextualisés pour Utilisateurs et la Modélisation des Utilisateurs à base d'Ontologie: Défis et Perspectives, proc. EGC'2005 Workshop, Paris, 2005, available at: www-sop.inria.fr/acacia/.../Liana.../RazmeritaAtelierMUf.pdf, last visit 1/11/2009.

[Reithofe 97] Reithofe, W. & Naeger, G., Bottom-up Planning Approaches in Enterprise Modelling – the Need and the State of The Art, Computers in Industry, Volume 33, Issue 2-3, P.P: 223 -235, Elsevier,1997.

[Rembold 98] Rembold, U. & Janusz, B., The Role of Models in Future Entreprise, Annual Reviews in Control, issue 22, P.P: 73-83, Elsevier, 1998.

[Robiette 05] Robiette, A. & Morrow, T., Innovation in the use of ICT in Education and Research, Blueprint for a JISC Production Federation, JISC Development Group; Version 1.1, 2005, available at: www.jisc.ac.uk, last visit 1/11/2009.

[Rother 99] Rother, M., Shook, J.: Learning to See Value Stream Mapping to Add Value and Eliminate Muda, 102 p., The Lean Enterprise Institute, Brookline MA, 1999.

[Sassoon 98] Sassoon, J., Urbanisation des Systèmes d'Information, 122p, Hermes Science, Paris, 1998.

[Scavo 05] Scavo, T. & Cantor, S., Shibboleth Architecture, Technical Overview, Working Draft 02, 2005, available at: shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf last visit 1/11/2009.

[Seitz 03] Seitz, T.A., Lean Enterprise Integration: A New Framework for Small Businesses, Master Report, Massachusetts Institute of Technology, USA, 2003.

[Shah 03] Shah, R. & Ward, P.T, Lean manufacturing: Context, Practice Bundles, and Performance, Elsevier, Journal of Operations Management, Volume 21, pp. 129-149, 2003.

[Simon 95] Simon, K. A., From Structure to Process: A Vision of a Process-Based Organization, Proc. ENTER95 Conference, Springer Verlag, Innsbruck, Austria.Innsbruck, 1995.

[South95] South, D.W., Encyclopedic Dictionary of Industrial Automation and Computer Control, 352 p

Prentice Hall, New Jersey: 1995,.

[Sun 08] SunMicrosystems, Sun Java System Directory Server Enterprise Edition 6.3 Reference, 2008, available at: docs.sun.com/coll/1224.4, last visit 1/11/2009.

[Suresh 98] Suresh, N. C. & Kay, J. M., Group Technology and Cellular Manufacturing: State of the Art Synthesis of Research and Practice, 568 p., Springer, 1998,.

[Syukur 04] Syukur, E., Loke, S. & Stanski, P., A Policy Based Framework for Context Aware Ubiquitous Services, Proc. EUC 2004, Lecture Notes in Computer Science, Vol. 3207, p.p.346-355, Springer, Berlin, 2004.

[Szyperski 97] Szyperski, C., Component Software - Beyond Object Oriented Programming, 411 p., Addison Wesley, 1997.

[Tajiri 92] Tajiri, M.& Gotoh, F., TPM Implementation, 220 pages, McGraw-Hill, New York, 1992.

[Trabelsi 07] Trabelsi, S., Gomez, L., Roudie, Y., Context Aware Security Policy for the Service Discovery, Proc. AINA07, p.p. 477-482, IEEE Computer Society, Washington, 2007.

[URBA-SI 02] URBA-SI, Pratiques de l'Urbanisme des Systèmes d'Information en Entreprises, 184 p., Publibook, 2002.

[USDoD 04] USDoD, Department of Defence of USA, NRL Security Ontology, 2004, available at: chacs.nrl.navy.mil/projects/4SEA/ontology.html, last visit:1/11/2009.

[USDoD 07] USDoD, DoD Architecture Framework Version 1.5, Volume I: Definitions and Guidelines, 2007, available at www.defenselink.mil/cio-nii/docs/DoDAF Volume I.pdf, last visit 1/11/2009.

[USDoD 85] USDoD, United States Department of Defence, Trusted Computer Security Evaluation Criteria Orange Book, Report19855200.28-STD, 1999, available at csrc.nist.gov/publications/history/dod85.pdf, last visit 1/11/2009.

[Vallespir 92] Vallespir, B., Zanettin, M. & Chen, D., GIM, GRAI: A Methodology for Designing CIM Systems, Version 1.0, Unnumbered Report, LAP/GRAI, University Bordeaux 1, Bordeaux, France, 1992.

[Vanhaver. 99] Vanhaverbeke, W. & Torremans, H., Organizational Structure in Process-Based Organization, Knowledge and Process Management Review, Volume 6, p.p. 41-52, Wiley Interscience, 1999.

[W3C 03] P3P, Platform for Privacy Preferences, Project of W3C, Enabling Smarter Privacy Tools for the Web, available at http://www.w3.org/P3P/, last visit: 1/11/2009.

[W3C 02] W3C Working Group, A P3P Preference Exchange Language, Working Draft , version1.0,

2002, available at:http://www.w3.org/TR/2002/WD-P3P-preferences- 20020415/, last visit: 1/11/2009.

[W3C 04] W3C Working Group, P3P Using the Semantic Web (OWL Ontology, RDF Policy and RDQL Rules), 2004, available at: www.w3.org/P3P/2004/040920_p3p-sw.html, last visit: 1/11/2009.

[Wainer 07] Wainer, J., Kumar, A. & Barthelmess, P., DWRBAC: A Formal Security Model of Delegation and Revocation in Workflow Systems, Information Systems, Vol. 32, p.p.365-384, Elsevier Science, Oxford, 2007.

[Wason 05] Wason, T., Cantor ,S., Hodges, J., Kemp, J. & Thompson, P., Liberty ID-FF Architecture Overview, Version: 1.2-errata-v1.0, Liberty Alliance Project, 2005.

[Weber 05] Weber, B., Reichert, M., Wild, W. & Rinderle, S., Balancing Flexibility and Security in Adaptive Process Management, proc. CoopIS, DOA, and ODBASE 2005, Lecture Notes in Computer Science, Vol. 3760 p.p. 59-76, Springer, 2005.

[Williams 95] Williams, T. J.& Rathwell, G. A., Use of the Purdue Enterprise Reference Architecture and Methodology in industry (the Fluor Daniel Example), 1995, available at: http://www.pera.net/, available at: 1/11/2009.

[Womack 03] Womack, P. & Jones, T., Lean Thinking – Banish Waste and Create Wealth in Your Corporation, 384 p, Free Press, USA, 2003,.

[Yee 05] Yee, G., Korba, L., Negotiated Security Policies for E-Services and Web Services, Proc. ICWS 2005, IEEE Computer Society, Washington, 612-619, 2005.

[Yialelis 96] Yialelis, N., Lupu, E. & Sloman M., Role-BasedSecurity for Distributed Object Systems, Proc. ICE'96, IEEE Computer Society, Washington, p.p. 80–85, 1996.

[Zhang 06] Zhang G. & Parashar, M., Dynamic context aware access control for grid application, Proc. Of GRID'XY, Computers & Security, Volume 25, p.p. 507-521, SienceDirect, 2006.

[Zhou 06] Zhou, X., Tsai, W., Wei, X., Chen, Y. & Xiao, B., Policy Infrastructure for Verification and Control of Service Collaboration, proc. ICEBE2006, p.p. 307-314, IEEE Computer Society, Washington, 2006.