Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2007 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2007

Assessing Business Process Security Awareness: A Service-Oriented Approach

Mariagrazia Fugini
Politecnico di Milano

Ernesto Damiani
Università di Milano

Karl Reed
La Trobe University

Assessing business process security: a service-oriented approach

Ernesto Damiani(+), Mariagrazia Fugini(++), Karl Reed(+++)
(+) Universita' degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
via Bramante 65 26013 Crema (CR), Italy
damiani@dti.unimi.it
(++) Politecnico di Milano
Dipartimento di Elettronica e Informazione
Piazza da Vinci, 32 20133 Milano
fugini@elet.polimi.it
(+++) La Trobe University
Dept. of Computer Science,
Bundoora, Melbourne, Australia
kreed@cs.latrobe.edu.au

Abstract The aim of this paper is to present some preliminary ideas about practical metrics and measurements useful for (i) assessing business process risk at design time and (ii) computing security and trust metrics at run time on business process orchestrations.
In particular, the study is focused on a priori metrics applied to behavioral specifications of business processes (e.g., business rules and UML 2.0 /UMM diagrams) and to run-time metrics applied to the high-level e-services composing them. Design-time metrics deal with the risk connected to information leaking (including privacy-related concerns) and to other disclosure threats, while run-time service-oriented metrics regard security as a Quality of Service, and therefore include factors like trustworthiness, completeness, and correctness of the services composing the business process when deployed on a Service Oriented Architecture (SOA).

1. Introduction

Today, the term ``extended enterprise" is used designate any collection of organizations sharing a common set of goals. In this broad sense, an enterprise can be a whole corporation, a government agency, or a network of geographically distributed entities. Extended enterprise applications support digitalization of traditional business processes, adding new processes enabled by e-business technology (e.g. large scale Customer Relationship Management).
Often, they span company boundaries, supporting a web of relationships between a company and its employees, managers, partners, customers, suppliers, and markets. Moreover, enterprises can collaborate in networks established on a *need-to-work* basis, hence creating a highly dynamic collaborative environment, where however, security requirements must be defined and met to preserve enterprises own business. In such a scenario, *Business Process Modeling* (BPM) techniques are becoming increasingly important. Less than a decade ago, BPM was known as `workflow design" and was aimed at describing human-based processes within corporate departments.
Today, BPM is used to design orchestration of complex system interactions, including communicating with the processes of other companies according to well-defined technical contracts. Also, it is used to check compatibility and consistence of the business processes of collaborating business entities. A number of methodologies, languages and software tools have been proposed to support digital BPM; nonetheless, a lot remains to be done for assessing a business process model validity with respect to an existing organizational structure or w.r.t. external constraints like the ones imposed by security compliance regulations. In particular, web-based business coalitions and other inter-organizational transactions pose a number of research problems. Besides, considering dynamic collaborative environments, models and mechanisms for access control and trust management are needed.

The Object Management Group's (OMG) *Model Driven Architecture* (MDA) provides a framework for representing processes at different levels of abstraction. Model-based development starts with the creation by application experts of models of the business process they want to represent. These business process models are built using a variety of formal notations but graphic notations are common. From these models, specifications are generated for the IT systems that they require. Available tools allow business process models to be edited and analyzed, and in some cases even executed for simulation and validation purposes. In this paper, we rely on a MDA-driven notion of business process model, composed of a *static domain model* including the domain entities and actors, plus a platform-independent *dynamic model* providing a specification of process activities.
Also, when designing an organization's business processes it is necessary to assess their criticality and to quantitatively determine the impact and consequences of loss of service or a reduction in normal customer service levels. We shall discuss how a quantitative risk assessment can be carried out on the basis of available information on the threats to normal service levels of each process. Such an assessment must be based on suitable metrics for measuring the impact on profitability and continued viability.
The paper is structured as follows. In Section 2 we introduce MDA-based business modeling. In Section 3, we discuss a priori and service oriented metrics, specifically, a priori, design time evaluation of knowledge-sharing risks on business process model descriptions. Section 4 specifies how security issues can be dynamically associated to collaborating e-services, also by means of suitable *Security Contracts* regarding trustworthiness of e-services, outlining how the reputation of a SME or site can increase or decrease according to a QoS approach. Finally, Section 5 draws the conclusion and outlines our future work.

2. Business Modeling

In today's highly competitive business environments companies increasingly recognize the importance of business process engineering for achieving improvements in terms of the performance of their business organization, e.g. by increasing efficiency, reducing costs, and improving process effectiveness and innovation. Choosing the "right" high-level representation of business processes (i.e., one providing the desired level of rigorousness and formality) is of paramount importance for effective business process engineering. The Object Management Group's (OMG) Model Driven Architecture (MDA) [Uhl04] has developed an open, vendor-neutral approach to this challenge leveraging OMG's widespread modeling standards like the Unified Modeling Language (UML).
MDA models are organized into multiple views, covering different abstraction levels and a variety of aspects (e.g., workflow, domain modeling, deployment). In this paper we focus on MDA *core models*, also called *Platform Independent Models* (PIM), which are independent of any particular implementation platform. MDA modeling usually starts with the creation of a PIM representing business functionality and behavior, undistorted by technology details. Within MDA, variations in business logic are handled at the platform-independent level through variability patterns that produce generic implementations. In other words, PIM only capture the business logic, i.e. the problem space in business terms. A series of model transformations can then be applied to map the model for specific platforms. Specifically, PIM descriptions built using MDA are transformed into *Platform Specific Models* (PSM), specifying how the functionality represented in a PIM is implemented on a particular IT platform. PSM are expressed using UML (extended with platform specific profiles) and can be implemented using any major open or proprietary software platform, including CORBA, Java, .NET, and Web services component platforms.
At the PIM level, Business Process Modeling is composed of two main layers: *Business Rules* (BR), i.e., declarative statements describing business domains, easy enough to be understood by humans, and Business Process Models (BPM), i.e., stepwise specifications of individual business processes. BPM are machine-understandable and detailed enough to be translated into computer-supported workflows. Both the static and dynamic part of the PIM rely on a business vocabulary perspective: using shared, domain-wide business vocabularies for modeling business processes is expected to support integration. In the static part, business rules are often stated in controlled English, than translated in terms of a

standard *Semantics of Business Vocabulary and Business Rules* (SBVR) [Dam07] . For the purposes of this paper, however, we shall refer to UML class diagrams, increasingly used for domain models in business environments. For representing the dynamic part, several approaches exist, some based on highly expressive models like Event Calculus and Petri nets. Here we focus on UMM, a UN standard for designing inter-organizational e-business process models [Hof06]. UMM activity graphs are a simple yet effective BPM specification whose expressive power is comparable to the one of a Deterministic Finite Automaton (DFA) with input/output, where I/O messages represent information interchange among workflow nodes realizing the business process.

3 Assessing Design-time Risk in Inter-organizational Business Processes

An important component of a successful business strategy is related with the organization of process flows. In this case, a business process is viewed as the sequence of activities and decisions arranged with the purpose of delivering a service, assuring security and effectiveness, in accordance to the life cycle of the service. In this section we shall consider business process descriptions used in the framework of Business Process Management (BPM) and discuss how one can assess the level of risk associated to knowledge sharing in inter-organizational processes. Our ongoing Tekne Project is aimed at studying and developing a priori techniques for evaluating privacy and disclosure risks related to integration and replacement of business processes in dynamic coalition of companies.

To fix our ideas, let us consider the various tasks in the client subscription process of a supplier of *Fiber-to-the-Home* (FTTH) telecom services. These tasks (activating lines, creating or verifying customer account, billing the customer) correspond to business processes usually performed by different entities (such as, supplies, purchasing, invoicing, and the customer himself). With the help of BPM, the FTTH provider can graphically model the tasks, and check the execution of most of them. Graphical notations aimed at describing process flows are one of the most widespread tools for supporting business process modeling. Their popularity is due to the capability of supporting both immediate reading and rigorous formalization. Another important advantage is that graphical notations are understandable by all business users (e.g., business analysts designing a process, technical developers implementing it, business people monitoring the process, etc.). Due the the procedural nature of notations typically used for these descriptions, process flows are typically validated against process termination, verifying the absence of interferences and procedural inconsistencies. Still these characteristics do not exhaust the possible metrics that can be calculated on a flow.

A clear distinction between coarse-grained design languages such as BPM and finer grained workflow specification emerges from this description. First, BPM automatically executes the tasks in applications (by calling Web services or through transaction calls to appropriate middleware). Further, BPM offers functionalities for supervising the entire activity through business. In our approach, a state-transition diagram represents each activity of the BPM.

Many equivalent state-transition formalisms can be used to express BPMs; here, for the sake of simplicity, we shall refer to the standard UN/CEFACT Modeling Methodology (UMM).

In UMM, business transactions are defined via *activity graphs* which use two *swim lanes*, one for the transaction initiator and the other for the reacting partner. For our sample FTTH company, a typical transaction may start with a salesperson sending a *quote request envelope* containing its identity and location and requesting a quote for a voice line. The quote will heavily depend on the final customer's location, as extending the fiber-optic network has an high per-meter cost. The envelope is input to the responding business activity (e.g., request_confirm_activity, see Fig.1) which is triggered by its receipt. A suitable *class diagram* can be used to model the business information covered in the confirm activity envelope for this particular application.

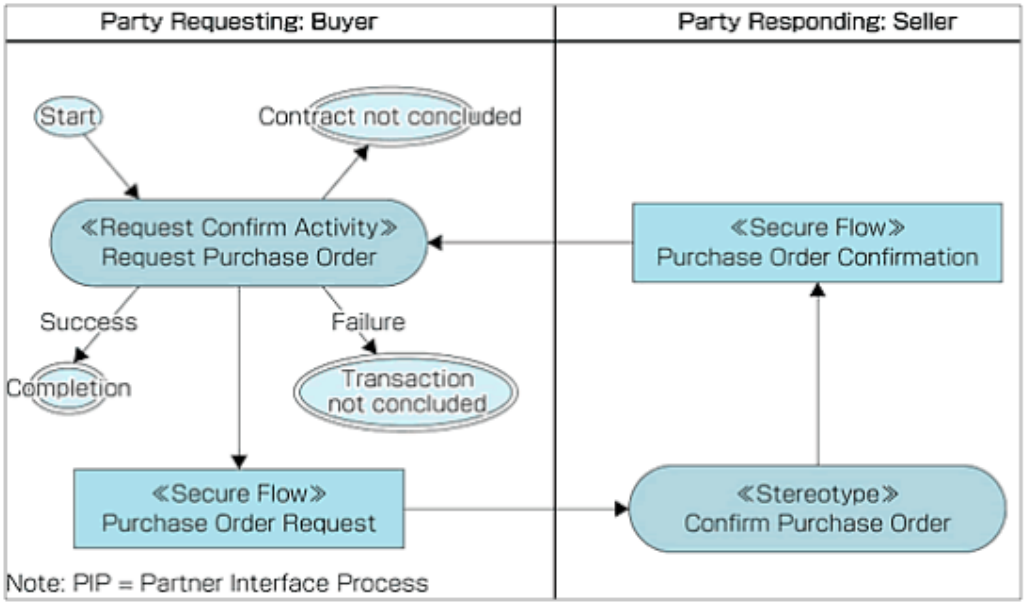

Fig. 1: A UMM activity graph showing two swim lanes

4 Assessing risk: design-time metrics

Assessing the risk linked to a specific business process design requires the definition of suitable *a priori* metrics, to be computed on the representation of the business process design itself. Of course, design-time a priori metrics are complementary to run-time metrics, based on data (e.g., process logs). linked to a specific implementation and execution of the business process. In our approach, each object instance included in the envelope is given an associated *risk value*, measuring the potential loss (expressed in a currency, e.g. in Euros) the sender will incur in should the receiver disclose it. In our example, the risk value associated to the envelope measures the potential loss the salesman would incur into if the precise location of the customers requesting the quote is disclosed by the FTTH company to some other telecommunication operator, e.g., in the framework of a reciprocity agreement with an ADSL provider.

Of course, the potential loss is not rigorously speaking a "risk value", since risk consists of impact times probability. Here, although the probability part cannot be measured on the side of the service consumer, it can however be estimated on the part of the service provider, e.g. based on frequency of attacks.

Although our example deals with privacy risk, this is by no means the only type of risk which can be considered at process design time. Doctored malformed messages, authentication attacks and the like and the other security threats mentioned in Section 4.1 can be modeled at design time in terms of risk/impact.

It is important to remark that (i) our risk value is a rough estimate on the part of the envelope sender, and may well be zero when data are publicly known (ii) the risk is associated to object instances in the activity diagram (rather than to classes in the class diagram) because the loss can be different depending of the specific interaction step.

Also, dealing with instances rather than classes allows us to ensure consistency of the estimated risk values at different levels of system specification, e.g. when passing from a process design iteration to the following one. A priori risks are measured by adding up impact values from instances of data entities. Increasing the level of detail of class diagrams can increase the number of instances, e.g. due to the explicit statement of the internal structure of classes via *part-of* relations. In order to prevent risk from increasing as well, risk pertaining to aggregated parts of an instance must be equal to the one previously estimated for the whole instance.

This way, it is easy to assess the global risk level associated to a UMM business process specification by adding up the risk values of all the individual interchanges. Disjunctive paths representing alternative flows can be dealt with by taking into account the maximum risk value, which corresponds

to the probabilistic interpretation of disjunction. Then, once an assessment of design-time risk, we can use the risk value (i) directly as a criterion for an outsourcing (or business process merging) decision (ii) as a warning on these processes which need to be monitored for run-time risk assessment.
Suppose for instance that an external service proposes to the FTTH operator to outsource the activity of providing quotes to salespersons. In this case, if the replacement is represented as a new UMM diagram allowing the direct communication of the salespersons to an external quote service, there will not be any increase in the risk value for the sending salesperson, but the replacement will introduce an additional risk value for the FTTH company, as the final customer data could be passed by the external service to a competitor.
Alternatively, suppose that the replacement is represented as a diagram where the FTTH provider can receive request for quotes itself, and by another diagram where the FTTH provider passes them to the external service, but without the final customer id. This alternative can be easily shown to decrease the disclosure risk, although most of the risk value in this case would be associated to the location rather than to the customer identity.
It is also important to observe that the evaluation of the two outsourcing alternatives can be done automatically, providing to the human in charge of a decision a list of UMM diagrams ranked according to their global risk level.

5. Security and dynamic adaptation on SOA business environments

After assessing risk at design time, practical countermeasures can be taken at the level of process implementation to decrease it to an acceptable level. Ensuring secure and authorized access to services and information resources in dynamic collaborative environments, such as the ones presented in the previous sections, is a challenging task [Bel04].
A success factor of service oriented business applications is the ability to dynamically handle openness and adapt to changes. The way dynamic business systems are used raises a number of trustworthiness issues, e.g., for privacy and security.
Security is often equated with access control, which consists of authentication and authorization, and is realized in such a way that unauthorized operations are identified in advance and subsequently forbidden. However, predefined access rights contradict with the openness and adaptation properties of dynamic business systems. All the existing security technologies are not effective in the dynamic handling of openness, which experiences continuous extensions.
Moreover, dynamic adaptation to business changes implies that the usage of technology is determined during utilization. In this section, we deal with security in such dynamic systems, in particular built on the web service paradigm. The Tekne (Towards Evolving Knowledge-based interNetworked Enterprise) project, funded by the Italian Ministry of Research (http://www.tekne-project.it) is aimed at defining an operational knowledge-based architecture for internetworked enterprises. In the framework of Tekne,.we are studying and developing an architecture and methods for creation and management of dynamic web services, able to detect anomalous situations, which may manifest as the inability to provide a service or to fulfill Quality of Service (QoS) requirements, among which security.
Two major security issues that need to be addressed in such a context of dynamic services are: specification of access control requirements and trust management. A dynamic system is privacy-aware if it ``enforces formalized and personalized privacy policies" [Sack06].
Specification of access control requirements for dynamic collaboration is limited by the lack of knowledge about remote services' and users' identities and affiliations. The access control policies and constraints defining privileges to invoke and to manage services need to be specified in terms of the attributes and properties, with related metrics, of both the users and the services. Moreover, the criteria for validating the attributes of the users and services should also be specified as part of access control requirements.
Trust management in the context of dynamic collaboration based on Service Oriented Computing involves validation of users' and services' attributes for secure interaction and prevention of unauthorized disclosure of policies and attributes. In this section, we outline models and mechanisms for access control and trust management, and specify how the model elements can be dynamically

associated to services by means of contracts regarding trustworthiness of services, using the metrics and business rules introduced in the previous section.

4.1 Scenario and security obligations

We shall consider an instantiation of the scenario of e-commerce of broadband introduced in the previous Sections, involving the service provider, a wholesale-to-retail distribution channel and a network of SMEs. All these actors interoperate in a Service Oriented Architecture, each preserving its internal orchestration and choreography. In this setting, a sample security threat can occur as follows: suppose two services S1 (e.g., an on-line Retail web service) and S2 (e.g., a Wholesale web service) are cooperating, and a third service S3 (e.g., the Provider web service) is invoked by S2. If the communication between Retail and Wholesale requires to be protected, and therefore messages can not be broadcast to any kind of Provider (as it would happen in a standard web service style), a secure communication is established between Retail and Wholesale, e.g., through strong authentication. In such way, these services mutually authenticate, and Wholesale is enabled to select only trusted providers, depending on S1 identity and security requirements. Moreover, if the Provider S3 should fail and interrupt its cooperation with S2, it should be ensured that there is no information leakage on the interrupted channel, and that S1 and S2 can continue cooperating by having S2 invoking a substitute trusted provider. To guarantee that these security requirements are met, authentication of identities and of messages is needed, typically using passwords, and encryption keys, so that only authorized services are able to read the message and its contents or to perform operations on services.
Moreover, a set of trusted services must exist in the business coalition to be selected for invocation depending on the invoker's required security level. We consider *security properties*, according to the rules and metrics defined in the previous sections, as a *Security Class* (SC), composed as a *n*-uple of elements each with a Security Level (SL). We consider that an SC has 5 elements (n=5), namely,

<confidentiality, integrity, non-repudiation, authenticity, reputation>

properties of a service (n depends on the selected application domain and is usually fixed by the designers). The level of a security property gives the measure of an obligation in service provisioning with respect to its business operations. The level of a security property is expressed as a value in the range [0..8], where ideally value 0 corresponds to the “unclassified” value and 8 for “top secret” in mandatory policies for multilevel systems.
Obligations impose constraints on operations on service data, which can relate to time (e.g., data may have to be stored at least for 30 days), or technical or governance restrictions (e.g., encrypted storage or adherence to privacy standards). When a service is invoked, its SC, that is, its set of *SL*s, is checked using business rules in order to satisfy the invoker's expected *SL*s for the properties. This leads to addressing access control requirements in a dynamic environment.
For trust management, which is the second aspect of our run-time approach, a policy-based decision is taken when the service satisfies a sufficient set of security provisions and meets the required security obligations. A non-trivial task involves monitoring accepted obligations and taking appropriate actions upon security fulfillment and defaulting, respectively. This monitoring is approached here using *security contracts*. For example, in general, if the user agrees to pay a monthly fee for services as an obligation, the system should monitor this obligation fulfillment and, in case of failure, take necessary compensating actions. Analogously, for security, if a service provider agrees to make available some services with a given level for confidentiality, the security components of the system should monitor this security obligation fulfillment and, in case of failure, take compensating actions [Pre06]. Such compensating actions could range from decreasing the trustworthiness of the service provider, replacing unfulfilled obligations with (perhaps more costly) alternatives, and/or taking punitive actions such as informing relevant (certification) authorities of the default or terminating the policy in force. To replace obligations with more stringent ones, the services need to be tagged with contractual obligations documents.
Similarly, for obligations fulfilled as promised, it may be appropriate that a (positive) reward action

should be taken, such as acknowledging fulfillment of promised confidentiality levels and perhaps rewarding the Web service provider by upgrading its trustworthiness level.
A simple example of the general form of security contract between two interacting parties Ws1 and Ws2 is reported in Table 1, stating that the overall work flow of the Process is at SC= 3.The business interaction to be protected regards Ws1 and Ws2, which exchange customer data with the following *SL*s prescribed for the interaction *<4,4,4,4,4>*.
The Trust Area of the Process is fixed at level TA1. The business operation protected by the contract is BO1 consisting in "Customer id exchange" between Ws1 and Ws2. The security requirements for BO1 state that, if Ws2 rises a security alarm, e.g., because it is under attack, it can be executed by a set of services {Ws7,Ws9,Ws11} in the same trust area TA1. However, the substitution can be performed only on Ws2 in same states, which are specified in the contract, in order to maintain both BO1 consistency and security. For example, Ws2 can be substituted only if the customer has not paid yet (Ws2.State1= "Before Payment") *and* if the credit card data were not acquired by the system (Ws2.State4= "Customer-Credit-Card= not-acquired"). Instead, a recovery action on Ws2 (e.g., trying to repair it from the attack) can be attempted only at early stages of the cooperation, e.g., when Ws1 and Ws2 have not reached the point of establishing a cost for the goods being purchased by the customer in this piece of transaction (Ws2.State2= "ItemCost =NotShown").
Finally, the grant option permitted denotes an auto-substitution of Ws2, that is, it is up to Ws2 to select the best service that can substitute itself, within a pool of *trusted* Web services "known" by Ws2. This issue goes in the direction of providing services able to self-react to problems, e.g., security attacks, or in general, faults [Fug06]. Note also that the reputation of a service or of a whole process can be downgraded in the cooperation network due to unfulfilled security obligations.

Table 1 Sample Security Contract between *Ws1* and *Ws2*

Security class of a Process P	Process SC= 3
Services involved in P: level of security *Ws1* and *Ws: <4,4,4,4,4>*	
Trust Area of P	TA=TA1
Business Operation BO1	Description "Customer id exchange" between *Ws1* and *Ws2*
Security *of BO1*	If Ws2 raised security alarm, permit **substitution** of Ws2 with {Ws7,Ws9,Ws11} in same Trust Area=TA1
	substitution permitted in Ws2.State1= "Before Payment"
	substitution permitted in Ws2.State4= "Customer-Credit-Card= not-acquired"
	recovery permitted in Ws2.State2= "ItemCost =NotShown"

grant option = ok
Authentication Mode/Key Strong/ 40sj)7)(“/£=£=?£?=0/”%&^?
Contract Lifetime Until receipt delivered to customer
Obligations SC not respected fee, and SL of Reputation property= -1

5. Conclusions

In this paper we presented some preliminary ideas about techniques for (i) assessing business process risk at design time and (ii) computing security metrics on the business process orchestrations at runtime. In our vision, design-time metrics deal with risks such as information sharing (including privacy-related concerns) while service-oriented metrics deal with business process instances security as a Quality of Service concept, and therefore include factors like trustworthiness, completeness, and correctness of the services composing the business process when deployed on a Service Oriented Architecture (SOA). Moreover, we outlined a scenario for creating security contracts and obligations regarding security in dynamic environments; the contract negotiation in SOA frameworks will be further investigated.

References

[Bel04] Bellettini, C., and Fugini, M. (Eds.) (2004). Security in Distributed Information Systems: Trends in Methods, Tools, and Social Engineering, IDEA Book Publishing.

[Cer07] Ceravolo, P., Fugazza, C. and Leida, M. (2007). Modeling Semantics of Business Rules, Proceedings of the IEEE Conference on Digital Ecosystems and Technologies (IEEE-DEST), Cairns, Australia, Feb. 2007.

[Dam07] Ceravolo, P., Damiani, E. Fugazza, C., Reed K., Wombacher A., Bodenstaff L., Representing and Validating Digital Business Processes, in T. Dillon, R. Rajugan, eds. "A State of the Art Semantic Web", LNCS-IFIP AdWebS series, Springer, to appear.

[Fug06] Fugini M., Mussi E., "Recovery of Faulty Web Applications through Service Discovery", SMR-VLDB WorkRetail, Matchmaking and Approximate Semantic-based Retrieval: Issues and Perspectives, 32nd International Conference on Very Large Databases Seoul, Korea, September 12-15, 2006, pp. 67-80

[Hof06] Hofreiter B., Huemer C., Liegl P., Schuster S., Zapletal M. . UN/CEFACT'S Modeling Methodology (UMM): A UML Profile for B2B e- Commerce, in Proc. of ER (WorkRetails) 2006: 19-31

[Pas06] Paschke A. (2006) Rule based Service Level Management, in Proc. of WorkRetail Quantitative Methods (SWQM) - Munich, Germany, 2006.

[Pre06] Pretschner, A., Hilty, M., Basin, D., (2006). Distributed Usage Control, in Comm. of the ACM, Special Issue on Highly Dynamic Systems, Vol. 49, n. 9.

[Sac06] Sackmann S., Strueker J, and Accorsi R. (2006). Personalization in privacy-aware-highly dynamic systems, in Comm. of the ACM, Special Issue on Highly Dynamic Systems, Vol. 49, n. 9.

[Uhl04] Mellor, SJ., Weise, D., Scott, K., Uhl A. (2004). MDA Distilled: Principles of Model-Driven Architecture Addison Wesley, 2004
-