Specification of Assessment Requirements for Business Collaborations

Bart Orriens

January 18, 2006

Abstract

Service-oriented computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing business collaborations. In order to realize this vision the ability to monitor collaborations is a critical issue that must be addressed. Current work in this area is low level in nature without any connection to high level assessment requirements. In this report we present a layered specification of assessment properties for business collaboration. The approach supports specification of high level assessment objectives, the mechanisms to achieve those, and the measures needed to implement these mechanisms. Moreover, dependencies among objectives, mechanisms and measures are made explicit as such creating a traceable path from objectives to measures.

Infolab Technical Report Series, no. 28

January 2006

1 Introduction

Recently there has been increasing focus on service-oriented computing (SOC), the new emerging paradigm for distributed computing and e-business processing, to deliver flexible and adaptable corporate business services by utilizing existing services across organizational boundaries. *Business collaboration* refers to a cooperation between multiple enterprizes working together to achieve a common business goal. In order to realize the vision of utilizing services as fundamental elements for developing applications [14] for business collaboration, assessment is a critical issue that must be addressed. Businesses will be averse to participating in cooperations which they can not monitor and subsequently manage, for example to avoid problems concerning not being notified of an important business occurrence, or not being able to prevent non-repudiation because of insufficient logging.

Therefore, for the successful adoption of SOC within the business collaboration domain, the paradigm must provide the means to make cooperation between enterprizes monitorable. At the moment, the most successful manifestation of SOC can be found in web services technology. A web service is a specific kind of service that can be unambiguously identified (generally by means of a URI) and whose service description and transport utilize open Internet standards, such as XML-based SOAP messages. Unfortunately, most work in the web service security arena concentrates solely on low-level monitoring and notification provisions; leaving the issue of relating these to higher level business requirements unaddressed.

In this report we briefly introduce our business collaboration context framework, which provides the context required for business collaboration development and management. We also show how this context can be described using collaboration models. Subsequently, we explain how the introduced collaboration models can be augmented to support the specification of assessment requirements from high level goals to low-level assessment measures such as provided by current web service assessment solutions.

The remainder of this report is structured as followed: we first introduce a running example based on a complex insurance claim handling scenario in section 2. Next, in section 3 we briefly discuss our framework for business collaboration context; after which we explain our model driven approach for the definition of this context in section 4. After that, in section 5 we analyze the role of assessment in business collaboration, and show how the specification of security requirements can be facilitated. Finally, we present conclusions in 6 and outline future work.

2 Example

To exemplify the ideas presented throughout this paper an example inspired by the case study in [9] is used. The example describes a complex multi-party scenario, which outlines the manner in which a car damage claim is handled by an insurance company (AGFIL). AGFIL cooperates with several contract parties to provide a service level that enables efficient claim settlement. The parties involved are Europ Assist, Lee Consulting Services, Garages and Assessors. Europ Assist offers a 24-hour emergency call answering service to policyholders. Lee C.S. coordinates and manages the operation of the emergency service on a day-to-day level on behalf of AGFIL. Garages are responsible for car repair. Assessors conduct the physical inspections of damaged vehicles and agree repair upon figures with the garages. The scenario outline is as followed (more details are introduced in the remainder of this paper where needed):

The policyholder (customer) phones Europ Assist using a free-phone number to notify a new claim. The claim is received by a call handler within Europ Assist's telephone assistance department. After verification of the customer' credentials to ensure that the provided policy details are valid and the occurred loss is covered, the call handler finds an approved repairer nearest to the customer's location. The customer is notified that this repairer will arrive at the scene shortly, if necessary with a replacement car and towing service. The call handler subsequently contacts the selected repairer to notify him of the incident. If the repairer is not available, another one will be selected and contacted. The customer is kept posted of such changes by phone. Once the repairer is on its way, the call handler contacts AGFIL to inform them of the made claim.

Upon receipt of the claim a claim handler will be assigned within AGFIL. The claim handler will gather all related claim information like customer records, claim history, etc. to Lee C.S. After that the claim handler will fill out the claim details on a claim form, which is subsequently stored pending further developments. Lee C.S. in the meanwhile has one of its consultants working on the claim. The first thing this consultant does, is contact the garage to inquire about the status of the car. The garage has picked up the car while the previous was going on and has worked out an estimate of the car repair cost. If this cost was below $500 then the garage will have started repairs. But if the costs were higher, the consultant at Lee C.S. contacts an assessor to go to the garage and check out the car for him -or herself. This assessor makes an independent estimate of the repair costs and negotiates a final price with the garage.

The result of the assessment is next reported back to the consultant at

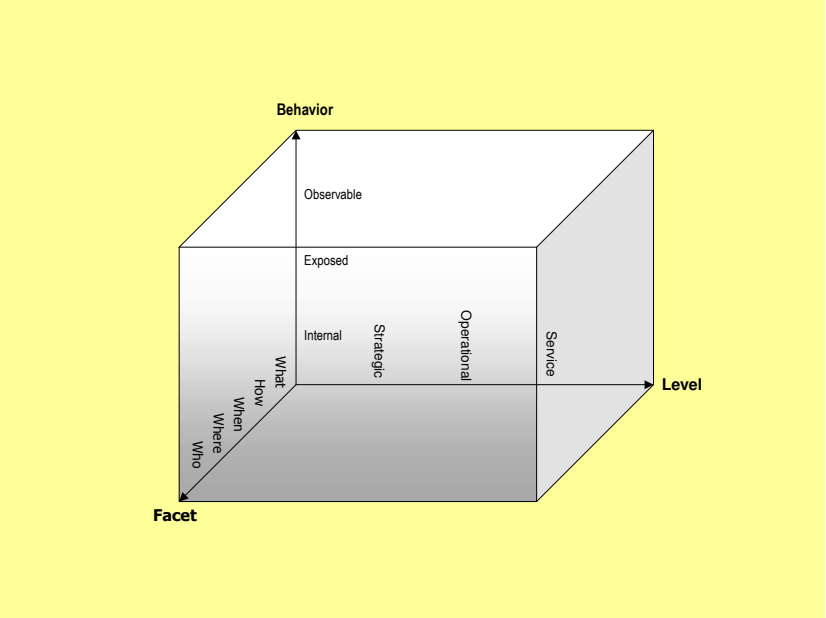

Fig. 1: Business Collaboration Context Framework (BCCF)

Lee C.S. The consultant reads the report and approves repair. An approval notification is sent to the garage, which consequently starts repairs on the car. Lee C.S' consultant also informs the claim handler at AGFIL of the final repair cost estimate upon which the claim handler incorporates the new information in the claim form. Once the garage has completed its repairs on the customer's car, an invoice is communicated to the consultant at Lee C.S. The consultant checks the invoice to see if it matches the earlier received cost estimate. Once the invoice is approved, the consultant sends the invoice onwards to AGFIL. The claim handler receives the invoice and adds it to the claim form. Payment for the claim is also issued.

3 Business Collaboration Context Framework

At the heart of our approach stands the Business Collaboration Context Framework (BCCF). The BCCF captures the context in which business collaboration development and management takes place by adopting a three dimensional view. Through this three dimensional view modularization of the definition and management of business collaborations is achieved. An overview of the framework is shown in Fig. 1.

As the figure illustrates we modularize the business collaboration context along three dimensions in the BCCF, being *behavior*, *level* and *facet*. We briefly discuss these in the following. For more information the reader is referred to [12, 11].

3.1 Behavior

The first dimension, **behavior**, places emphasis on the different behaviors that an enterprize exhibits in business collaboration; where consequently the purpose and target of development and management varies. The behavior dimension encompasses three types of behavior captured in three corresponding so-called collaboration aspects (inspired by among others [6, 15, 17]): observable, exposed and internal behavior expressed in the *conversation*, *participant public behavior* and *internal business process* aspect respectively.

The observable behavior constitutes the externally visible behavior between participants in a business collaboration; and is expressed in the *conversation aspect*. Captured in the *participant public behavior aspect* the exposed behavior describes how an individual participant can publicly behave in a business collaboration (i.e. its potential collaboration behavior). In contrast, the internal behavior (specified in the *internal business process aspect*) is also individual to each participant; however, it is only of interest to this particular participant, i.e. it can not be observed by other participants.

3.2 Level

The second dimension, **level**, recognizes the fact that the different business collaboration behaviors of an enterprize take place at several levels; where consequently the domain, degree of abstraction and the type of developers in development and management varies. In the BCCF three layers of abstraction are identified (inspired among others by [10, 18]): the *strategic*, *operational* and *service* level spanning from high level requirements to technical realization of collaboration behaviors.

At the strategic level the focus is on behavior that is abstract in nature, describing the purpose and high level requirements an enterprize has with the behavior. The operational conditions under which enterprizes exhibit their behavior are part of the operational level. This level establishes how high level strategic behavior (private, exposed and observable) will be operationalized. The technical realization of operational behavior is done at the service level, describing how the services provided by the IT-infrastructure support the operational activities.

3.3 Facet

The third dimension, **facet** captures the fact that the collaboration behaviors conducted by enterprizes affect many different parts. Facets represent these different parts of a business collaboration behavior that can be observed; and where consequently the focus and type of developer involved in collaboration development and management varies. Five facets are distinguished (inspired by among others [5, 16, 18]): *what*, *who*, *where*, *when* and *how* facet.

The *what* facet emphasizes the structural view of a collaboration behavior, focusing on what things are used to perform a collaboration behavior. The *how* facet takes a functional standpoint, and thus concentrates on how a collaboration behavior is conducted. The *who* facet concerns the participant(s) conducting the collaboration behavior. The location(s) at which the behavior is carried out are expressed in the *where* facet, whereas its temporal dimension is covered in the *when* facet.

4 Modeling the BCCF

To capture the three dimensions of collaborations aspects, levels and facets of BCCF we employ two types of model: meta models and models, both of which are defined for individual levels. Meta models provide design guidelines in terms of classes and their relationships, where depending on the collaboration aspect being modeled additional constraints are placed on the meta-model. Models represent a particular application design, and are derived by populating a meta model's *classes*.

Every meta model consists of six classes, where each class captures a particular facet; i.e. for *what*, *how*, *where*, *who*, *when* and *why* facet. Every class constitutes a set of logically related *attributes*. *Associations* connect the classes expressing dependencies among facets. *Mappings* define dependencies among levels by providing links between classes that describe the same facet at different perspectives (illustrated by the arrows between facets at different perspectives in Fig. 1).

Snippets of exemplary models for the AGFIL application are illustrated in Fig. 2, showing its strategic, operational and service model respectively; where the models are represented based on UML conventions. In order to distinguish different facets, we represent them in different shapes in their UML models (see also legend in Fig. 2): *what* facet is shown as folded corners, *how* facet as rounded rectangles, *who* facet as octagons, *where* facet as plaques, and *when* facet as hectagons. For more information the reader is referred to [11, 12]; where [13] contains the most recent details.

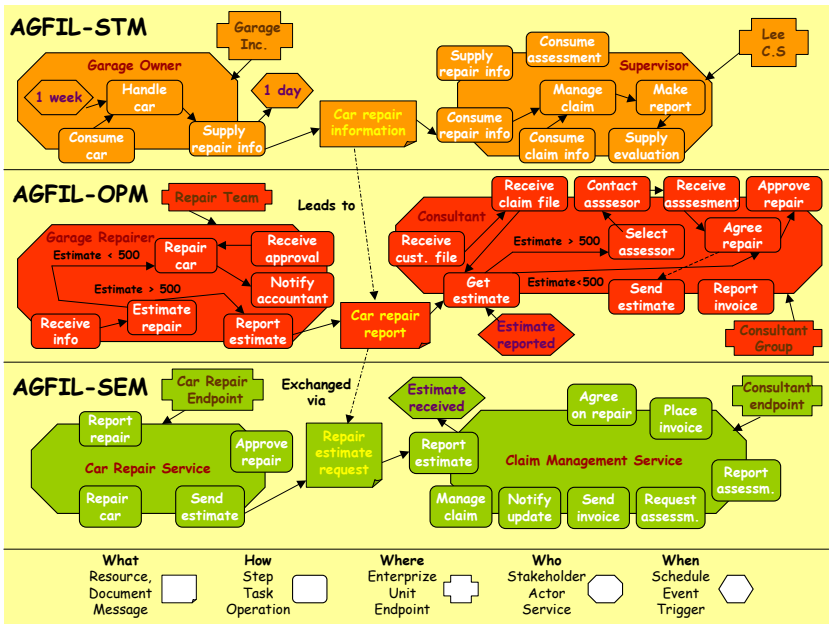

Fig. 2: AGFIL Collaboration Models

4.1 Strategic Models

At strategic level, strategic models like the AGFIL-STM in Fig. 2 capture purpose and high level requirements of business collaborations, akin to requirements analysis [2, 17]. Strategic models are expressed in terms of resources, steps, stake holders, enterprizes, and schedules. Resources such as `car repair information` provide abstractions for means such as financial, human and informational capital. Resources are used and produced by steps which represents high level functions.

Steps are of type 'internal' (like `handle car` presented inside the stakeholder boundary in Fig.2); or type 'communication' representing resource supply and consumption e.g. `consume repair information`. Stake holders like `garage owner` describe the participants involved who are responsible for carrying out defined steps. Stake holders belong to an enterprize, where enterprizes are manifestations record the information about the participating enterprize where behavior is carried out. Stake holders and their enterprizes are bound by schedules reflecting temporal constraints, like the deadline of `1 week` for `handle car`.

4.2 Operational Models

At operational level, operational models like the AGFIL-OPM in Fig. 2 depict how high level strategic behavior is realized in terms of operational activities. These are expressed in terms of documents, tasks, actors, units, and events. Documents (like `car repair report` represent the flow of information in a collaboration behavior. Documents are used and produced by tasks. Tasks represent specific business functions, and are of type 'internal' or 'communication' (represented inside or on the boundary of the actors respectively), e.g. `collect claim form` and `report invoice` respectively.

Actors such as `garage repairer` and `consultant` are responsible for carrying out tasks. Actors instantiate the **Actor** class and belong to units such as `repair team` unit, whose abstract definition is provided by the **Unit** class. In order to assess progress, keep logs to ensure non-repudiation, and etceteras, events are published and subscribed to by actors. Events describe business occurrences which have properties such as 'date', 'time', 'severity'.

4.3 Service Models

At service level, operational models are translated into service models that specify how the described operational behavior is realized using the services offered by the IT-infrastructure. Service models are defined in terms of messages, operations, services, endpoints, and triggers. Messages represents containers of information (e.g., `repair estimate request`), consisting of meta-data and actual data. Messages function as the inputs and outputs of operations such as `place invoice`.

Operations, just as steps and tasks at strategic and operational level respectively, can be dependent on one another. Additionally, they can be of type 'internal' or 'communication'. Operations are grouped in services (e.g. `car repair service`, which constitute collections of logically related operations. Services themselves are provided by endpoints (like `claim handling endpoint`) and have properties 'network location' and 'type'. To express technical occurrences triggers like `claim request acknowledged` can be defined on the basis of the **Trigger** class.

4.4 Mappings between Models

For the specification of dependencies between different collaboration behaviors at different levels, we employ vertical mappings. Vertical map-

pings are realized by providing links between the classes in different metamodels and instance models at different perspectives. The vertical mappings are based on the implicit links that exist between classes that describe the same facet at different levels in the same collaboration behavior. We define the following mappings:

Resources at strategic level are mapped to documents at operational level. Documents themselves are mapped to messages using *exchangedVia* relations. Steps are mapped via *decomposedIn* relations to tasks; whereas tasks are *realizedBy* operations. Stake holders *control* actors, where each actor is *representedBy* a service. Enterprizes are *organizedIn* units, where each unit itself *offers* one or more endpoints. Schedules are *splitInto* events, where each event *causes* multiple triggers.

5 Assessment in Business Collaboration

Assessment in general is defined as ”having the capability to to judge or decide the amount, value, quality or importance of something” [3]. Interpreted in the context of business collaboration assessment deals with providing assurance to enterprizes that they can assess the progress of their cooperation. In other words, assessment is concerned with providing peace of mind for the businesses involved, where they can rely on the fact that their collaboration is safe from risks like undetected deadline violations, unduly notification of relevant business occurrences, and etceteras.

When put into the business collaboration context as presented in section 3, it follows that for business collaboration assessment can be perceived at a strategic, operational and service levels, each of which represents a level of abstraction with its own content and meaning regarding security. In the remainder of this section we shall discuss the role of assessment and the specification of assessment requirements at the different levels. An overview of the requirements that can be specified is provided in Figure 3.

5.1 Strategic Level Assessment

As observed at an abstract strategic level a business collaboration constitutes a cooperation between enterprizes making use of each other's business services to exchange resources to further their business goals. At this level assessment specification deals with the definition of the objectives for assessing the progress of these resource exchanges. Objectives analysis here is aimed at 1) identification of the objective, 2) measuring the mag-

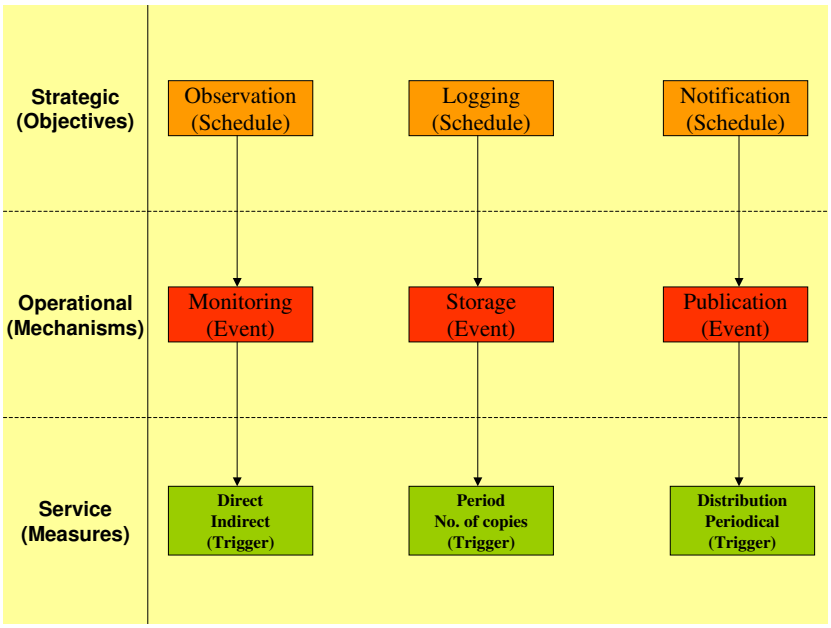

Fig. 3: Assessment in Business Collaboration

nitude of the potential loss if this objective is not dealt with, and 3) the probability that the loss will occur. We identify three main objectives:

- *Observation*

  A first, and basic, objective is that of *observation*, which expresses the desire to be able to observe progress. In the context of business collaboration this involves being able to monitor whether or not the schedules of the different steps are being met; e.g. the `1 week` schedule associated wit `handle car`.

- *Logging*

  Often simply monitoring the progress is not sufficient. Rather, enterprizes like to keep physically track of the progress made e.g. to be able to analyze it and make adjustments if necessary if there is a risk of not meeting the set deadline; or to be able to have proof of completion of activities in case things go awry.

- *Notification*

  In addition to logging progress it can occur that parties other than the party doing the monitoring, need to be notified. For example, if monitoring has been outsourced to a third party, then notification of progress has to take place to inform the stake holders involved about the current state of the collaboration.

The above three main assessment objectives have been extrapolated from the current work in the web service arena in this area, being WS-Events, WS-Eventing, WS-Notification; where the first two aim to address monitoring and logging at technical level, and the third one attempts to facilitate notification at this level.

5.2 Operational Level Assessment

Based on the objectives analysis at the strategic level assessment measures are determined at the operational level to facilitate realization of these objectives. At this level a business collaboration constitutes the sending and receiving of appropriate documents by enterprizes to further the state of the business collaboration. As such, the assessment mechanisms that will be employed, are to enable evaluation of the progression of this state. In correspondence with the described objectives in subsection 4.1 we identify the following assessment mechanisms:

- *Observation* → *Monitoring*

  If the progress of a collaboration step with regard to its schedule is to be observed at strategic level, at operational level one or more of the events into which this schedule is split must be monitored. Which event is monitored and which is not, is defined in an event's *monitoring* property; where monitoring can be 'direct' or 'indirect'.

- *Logging* → *Storage*

  Realization of a logging objective takes the form at operational level in the *storage* characteristic of individual events. Storage can be done on a 'permanent' or 'temporary' basis, depending on the requirements (e.g. to meet certain legal obligations, internal protocols, and so on).

- *Notification* → *Publication*

  In case stake holders need to be notified concerning the progress of a schedule, at operational level for each event it is defined in what manner *publication* is performed. Currently, we distinguish between 'push' and 'pull' like mechanisms.

Together these three assessment mechanisms provide the building blocks required to achieve the strategic objectives present in the business collaboration environment. We are aware that they are rather basic in nature, and as such we expect them to be further enhanced in the future.

5.3 Service Level Assessment

The assessment mechanisms selected at the operational level must subsequently be realized at the service level via assessment measures. This level is the domain of the service oriented computing paradigm. In this paradigm a business collaboration is viewed as a set of interacting technical services, where these interactions are message based and facilitate the communication of information among services. In order to meet the business driven security demands at this level, the message based interactions, i.e. message exchanges, must be adequately monitored, stored and/or published.

For this purpose we have identified the following security measures; where these are discussed grouped in accordance with the security mechanism they realize:

- *Monitoring → Broker*

  When monitoring is done in a direct manner, no third party needs to be specified. If, however, monitoring is set to 'indirect' for an event, then the *broker* property must be defined for each related trigger at service level; where this property provides a reference to the third party like an URL.

- *Storage → Period, number of copies*

  If an event is to be stored, then at service level its corresponding triggers (together constituting this event) have the *period* and *number of copies* to be stored specified as part of their definition. In case storage is indefinite, then period is set to minus one; otherwise its value is greater than zero.

- *Publication → Distribution, periodical, access point*

  Publication can be done in a push or pull fashion. If an event is pushed, then the *distribution* characteristic captures how this is exactly done, e.g. via 'publish/subscribe' or 'broadcast'. The *periodical* property depicts how often a distribution is performed. If an event is pulled, then the *access point* where the information can be gathered from must be specified.

Observe that the above described measures at service level are not intended to be exhaustive in nature. The authors are aware that many other measures exist; the above is therefore intended to be of illustrative nature to show how operational assessment mechanisms might be realized at a technical service level.

6 Conclusions

In this technical report we addressed the issue of specification of assessment requirements for business collaborations. This work is motivated by the lack of support thereof in the current research with regard to relating high level assessment objectives to concrete assessment measures; as most work (like [1,4,7,8]) focuses on provision of low level assessment measures without taking higher level, business driven assessment requirements into consideration.

To remedy this situation we introduced our generic framework for capturing the business collaboration context; and explained how this context can be described via the usage of various meta models and models. After that we explained how these meta models and models can be augmented to facilitate assessment requirement specification at strategic, operational and service level. Furthermore, we established relations between the requirements at these different levels; as such enabling traceability of strategic assessment objectives to operational assessment mechanisms to service level assessment measures (and vice versa).

A caveat concerns the defined assessment properties: as also mentioned in the report already these are not intended to be exhaustive in nature nor do the authors expect them (and the relations between them) to be final. As the authors are not themselves experts in the field of assessment, more work to further develop the (currently basic) support for assessment requirement specification is required. However, we believe that the presented approach provides a first step on the road to comprehensive assessment requirement specification for business collaboration.

References

[1] D. Box, L. Cabrera, C. Critchley, F. Curbera, D. Ferguson, A. Geller (Editor), S. Graham, D. Hull, G. Kakivaya, A. Lewis, B. Lovering, M. Mihic, P. Niblett, D. Orchard, J. Saiyed, S. Samdarshi, J. Schlimmer, I. Sedukhin, J. Shewchuk, B. Smith, S. Weerawarana, D. Wortendyke, Web Services Eventing (WS-Eventing), *http://www.ibm.com/developerworks/webservices/library/specification/ws-eventing/, 2004*

[2] P. Bresciani et al, Tropos: An Agent-Oriented Software Development Methodology, *Autonomous Agents and Multi-Agent Sytems, Vol. 8, No. 3, pp. 203-236, 2004*

[3] Cambridge Learner's Dictionary, *http://dictionary.cambridge.org*

[4] N. Catania (editor), P. Kumar, B. Murray, H. Pourhedari, W. Vambenepe, K. Wurster, Web Services Events (WS-Events), *devresource.hp.com/drc/specifications/wsmf/WS-Events.pdf, 2003*

[5] B. Curtis et al, Process Modeling, *Communications of the ACM, Vol. 35, No. 9, pp. 75-90, 1992*

[6] R. Dijkman et al, Service-oriented Design: A Multi-viewpoint Approach, *International Journal of Cooperative Information Systems, Vol. 13, No. 4, pp. 337-368, 2004*

[7] S. Graham (editor), P. Niblett (editor), D. Chappell, A. Lewis, N. Nagaratnam, J. Parikh, S. Patil, S. Samdarshi, I. Sedukhin, D. Snelling, S. Tuecke, W. Vambenepe, B. Weihl, Web Services Base Notification, *http://www-128.ibm.com/developerworks/webservices/library/specification/ws-notification/, 2004*

[8] S. Graham (editor), P. Niblett (editor), D. Chappell, A. Lewis, N. Nagaratnam, J. Parikh, S. Patil, S. Samdarshi, I. Sedukhin, D. Snelling, S. Tuecke, W. Vambenepe, B. Weihl, Web Services Brokered Notification, *http://www-128.ibm.com/developerworks/webservices/library/specification/ws-notification/, 2004*

[9] P. Grefen et al, CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises, *International Journal of Computer Systems Science & Engineering, Vol. 15, No. 5, pp. 277-290, 2000*

[10] Object Management Group, Model Driven Architecture, *http://www.omg.org/docs/ormsc/01-07-01.pdf, July 2001*

[11] B. Orriens et al, Bridging the Gap between Business and IT in Service Oriented Business Collaboration, *Proceedings of the IEEE International Conference on Services Computing, Orlando, Florida, USA, July 2005*

[12] B. Orriens et al, Establishing and Maintaining Compatibility in Service Oriented Business Collaboration, *Proceedings of the 7th International Conference on Electronic Commerce, Xi'an, China, August 2005*

[13] B. Orriens, Modeling The Business Collaboration Context, *INFOLAB Technical Report Series, No. 28, Tilburg, The Netherlands, January 2006*

[14] M. Papazoglou, G. Georgakapoulos, Introduction to the Special Issue about Service-Oriented Computing, *Communications of the ACM, Vol. 46, No. 10, pp. 24-29*

[15] C. Peltz, Web services orchestration: a review of emerging technologies, tools, and standards, *Hewlett Packard White Paper, January 2003*

[16] A. Scheer, Architecture for Integrated Information Systems - Foundations of Enterprise Modeling, *Springer-Verlag New York, Secaucus, NJ, USA, 1992*

[17] P. Traverso et al, Supporting the Negotiation between Global and Local Business Requirements in Service Oriented Development, *Proceedings of the 2d International Conference on Service Oriented Computing, New York, USA, 2004*

[18] J.A. Zachman, A framework for information systems architecture, *IBM Systems Journal, Vol. 26, no. 3, pp. 276-292, 1987*