An Ontology of Interoperability in Inter-Enterprise Communities

Toni Ruokolainen[1], Yannick Naudet[2], and Thibaud Latour[2]

[1] Dept. of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 UNIVERSITY OF HELSINKI, FINLAND
[2] Innovation Center by Information Technologies (CITI)
Henri Tudor Public Research Center, LUXEMBOURG

Summary. The ability of a system to interoperate with another is a multi-dimensional concern which must be considered simultaneously from different perspectives and covering all the concerns relevant for different stakeholders. In this paper, we present a model of interoperability for service-based inter-enterprise computing environments which is based on the Ontology of Interoperability (OoI) developed under INTEROP-NoE and the Pilarcos framework for federated service communities. We propose a refined version of the OoI, generalizing the originally considered scope of interoperability and adding concepts required to address dynamic aspects in interoperability. The ontology is then specialized to the domain of federated service-based communities and we illustrate with some concrete examples how interoperability problems and solutions in that domain could be related to it.

1 Introduction

The ability of a system to interoperate with another is a multi-dimensional concern which must be considered simultaneously from technical, semantic and pragmatic perspectives and covering all the concerns relevant for different stakeholders. To grasp the problem of interoperability, one must have a clear definition of what interoperability comprises and what are the corresponding problems that can occur during operation of the system. Given this characterization of interoperation, mechanisms to identify interoperability problems can be constructed and corresponding solution methods provided.

The FIG-OOI (Focused Interest Group on Ontology of Interoperability) group of the INTEROP-NoE [3] has provided a general framework for managing interoperability problems. The Ontology of Interoperability (OoI) [10], resulting from previous work of Rosener *et al.* [12, 13], provides a formal, domain-independent definition of the interoperability problem with a pragmatic, problem-solving view.

To be usable as a decision aid, the OoI needs to be specialized to construct a domain-specific model of interoperability. In this paper, we present a model of interoperability for inter-enterprise computing environments in the Pilarcos framework [8, 16]. The later allows the establishment of federated collaborations between autonomically administered business services, by providing dedicated meta-models and middleware services. In this context, heterogeneity, autonomy, and dynamism are inherent properties of the operational environment which induce severe interoperability problems when collaborations involving several organizations should be established [14]. To address these kind of environments, the OoI has to be extended with additional concepts introduced in [15].

The structure of this paper is as follows. First, the Ontology of Interoperability is briefly presented and refined by generalizing the originally considered scope of interoperability and adding concepts required to address dynamic aspects [14]; the corresponding enhanced meta-model is discussed Section 2. Section 3 presents the different aspects defining interoperability in dynamic service-oriented enterprise environments. After that, we propose a specialization of the OoI to this specific domain in Section 4 and illustrate in Section 5 with some concrete examples how interoperability problems and solutions in that domain could be related to the ontology. In particular, interoperability intra- and inter-aspects is addressed. Finally, we draw some conclusions about the applicability of the interoperability model in Section 6.

2 The interoperability meta-model

Interoperability is a requirement inside a system for allowing interaction or composition of its components, but also for the system itself, when it needs to be sufficiently flexible to exchange information with another system, or if it needs to be open to new components. As soon as this ability is not achieved when systems or system's components need to operate together, interoperability becomes a problem that must be solved. Contrary to what can be found in most of the available definitions, interoperability is not only related to communication. Generally speaking, the components of a system do not necessary have to communicate, but might simply have to be composed together for a specific purpose. From a pure compositional point of view, this can be viewed as a *structural interoperability* need. When communication or other kinds of action define the relation between the system's components, this concerns the *behavioral aspect of interoperability*.

The point of view adopted for building the OoI [10] considers interoperability from a problem-solving perspective, not restricted to communication matters. This is illustrated by the following definition, first proposed in [13]:

Definition 1. *An interoperability* problem *appears when two or more* heterogeneous *resources are* put together. *Interoperability* per se *is the paradigm where an interoperability problem occur.*

As can be seen in figure 1, the OoI is composed of three distinct meta-models, covering different aspects of definition 1:

- The *decisional model* defines the relation existing between problems and their solutions and new problems they potentially induce. It covers the problem-solving position adopted

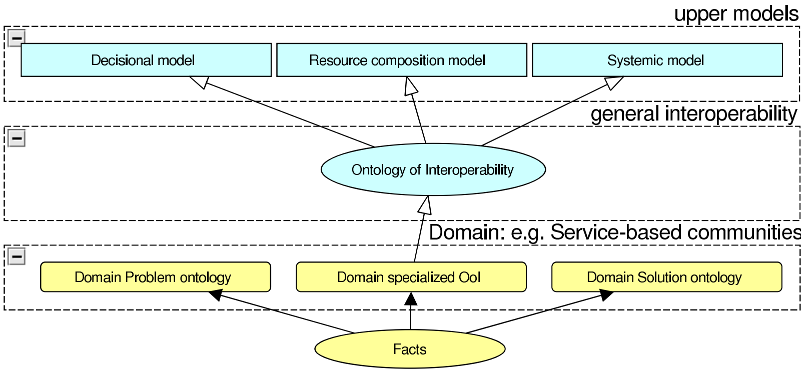

Fig. 1. Framework of the Ontology of Interoperability: origins, and instantiation for domain specific problem / solution description.

in definition 1 and the fact that interoperability is defined as a problem. This model aims at being the main support for a reasoner or a decision-aid system to propose solutions to identified problems.

- The *resource composition model* identifies the resources, together with their objectives, their interfaces, and their composition relations. Quite logically, this model is needed as interoperability problems are defined to appear when some heterogeneous resources are put in relation. In this model, resource heterogeneity concerns the resources' interfaces, which are their "material" realisations, actually in contact when there exist a relation between resources.
- The *systemic model* defines more precisely the notion of resource by identifying it to a system. Because a general system theory [18] point of view has been adopted, the considered systems are not limited to IT ones. This is especially important to have this non restrictive approach when considering interoperability, which has different facets, like the organisational or conceptual aspects [1], the technological aspect being as important as these two others. According to this view, a resource is a system. It can thus be real or abstract and can belong to any part of another system. A software component, as well as a hardware component, a person, a business rule, etc., can all be resources of a system. The systemic part consists thus mainly on the concepts of system; the model that can describe the system or define a system that will be build, this model being also describable by a meta-model; the representation of the model which constitutes a perceivable symbolic description of this model; and the environment, which represents what is outside of the system.

The interoperability meta-model, formalized now as an ontology [10], uses the three upper models presented before, by defining Interoperability as a problem, existing when there exist a relation between some heterogeneous resources and which can be solved using some specific solutions like bridging or homogenization. The OoI is not meant however to be used as-is in any application domain. For each domain, it will need to be specialized. Then it will be usable in a decision-aid system, using descriptions of problems and solutions known in that domain. Such descriptions will refer to the concepts of the specialized OoI and potentially to domain-specific problem and solution ontologies allowing a richer formalization.

2.1 An enhanced version

In figure 2, we propose a revisited version of the OoI, resulting of the work conducted for its use in the context of service-based inter-enterprise computing environments. The main enhancements concern: 1) a change in the considered scope of interoperability, leading to the introduction of the *Incompatibility* concept, which is wider than the previously considered *ResourceHeterogeneity*; 2) the introduction of dynamic aspects, revisited from [15].

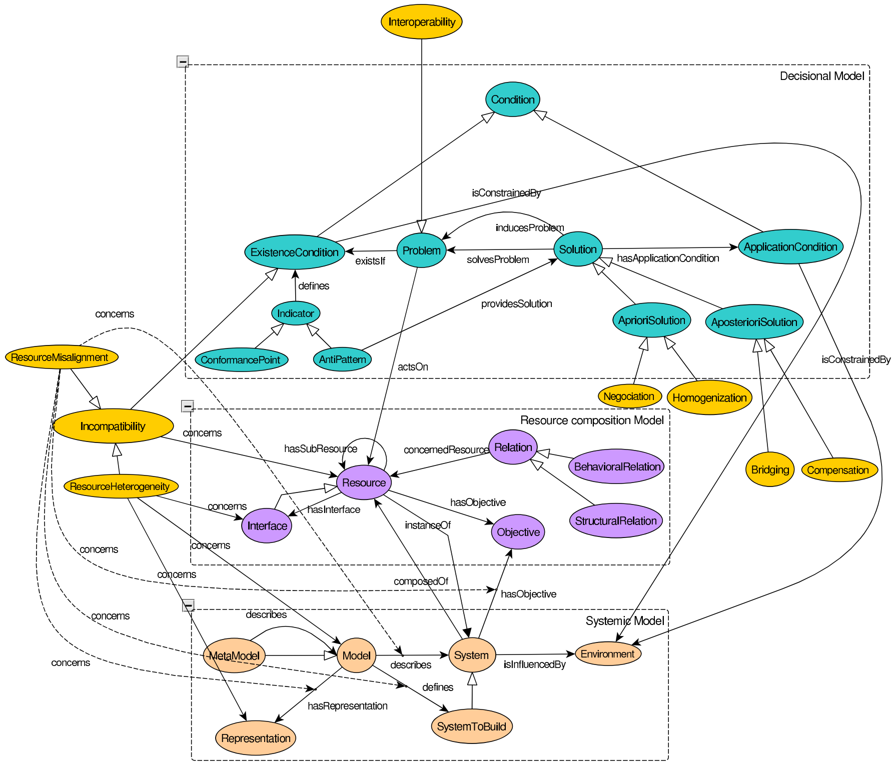

Fig. 2. Conceptual graph of the enhanced version of the OoI.

According to definition 1, an interoperability problem occurs when some connected resources are heterogeneous. The heterogeneity we consider here can only occur between resources of a same nature, at a same holistic level in the system. The application of the OoI to the specific domain addressed in this paper rises the issue of considering conflicting relations existing between resource of different nature as a problem pertaining to interoperability. To tackle this issue, we replace the concept of *ResourceHeterogeneity* considered in previous work [10], by the more general concept of *Incompatibility*, which is then partitioned into two kinds: *Heterogeneity* and *Misalignment*. Incompatibility concerns resources of any nature. More specifically, heterogeneity relates then to resources of the same nature and concerns either their *interfaces*, their *models* (describing the system a resource is), or the *representations* of the models which is the aggregation of symbols used to materialize the systems. Differently, misalignment can be observed when one resource constraints the way another one will be build, structured or will behave. The two resources are in this case of different nature. Misalignment does not occur between the resources themselves, but at the level of the relations: between a resource and its model (e.g. constraint on the structure); between a resource and its representation (e.g. constraint on the syntax of the model); between a resource and its objective (e.g. constraint on the behavior).

To tackle interoperability problems occurring in federated enterprise environments, concerned by autonomically administrated business services collaborations, additional concepts addressing dynamic aspects of interoperability have been proposed in [15]. These new concepts allow to more precisely address relational and environmental aspects in interoperation, introducing effects that cannot be dealt appropriately with or within static system models.

As dynamically detecting problems is an important issue, the concept of *Indicator* has been introduced in the decisional model. Indicators define concrete (formalized) *existence conditions* for a problem. They are declined into *anti-patterns* and *conformance points*, which both can be checked at runtime to detect the occurrence of potential conflicts. Anti-patterns and conformance points are dual indicators: the verification of the first one indicates the existence of a problem, while the last one indicates a problem if it is verified as being false. Conformance points provide a description of checking points (*e.g.* in the form of a set of rule) that must be verified to test the correspondence between the expected behavior and actual operation of the system; while anti-patterns describe specific knowledge or bad solutions or habits leading to problems. What is interesting with anti-patterns is that they also provide a solution to avoid the problem they relate.

Environmental conditions are also important as they will constrain not only the problem as was suggested in [15], but the whole system. According to this, the new OoI presented here possesses a concept *Environment*, linked to the system and representing all that is external to a system but can have an influence on it. This more general concept is coherent with the general system theory. As it influences both the existence of problems and the applicability of solutions, the concept is directly linked to *ExistenceCondition* and *ApplicationCondition*. Finally, two additional concepts related to solutions, have also been proposed in [15]: *Negotiation* and *Compensation*. They provide respectively *a priori* and *a posteriori* solutions to find common agreements at the organizational level.

3 Aspects of inter-enterprise interoperation

Modern networked enterprises require flexibility and openness of computing systems to tolerate changes in ever-changing technology and business domains, as well as to gain competitive edge. Dynamic and agile establishment of business collaborations comprised of loosely-coupled and autonomic services is currently considered as one of the best options to match these requirements. While in traditional software engineering approaches interoperability is typically enforced by requisition of aggressive integration or unification over technology and modeling concepts, this is hardly possible or acceptable in the modern networked business environments; autonomy of administration is too important for enterprises as means to maintain their agility.

Instead, a federated model of collaboration is needed for establishing interoperation in such a way that autonomy is preserved. In federated approaches, the model of collaboration is dynamically constructed within a controllable environment providing the necessary interoperability facilities. Federated collaboration is established via utilization of contractual relationships, an infrastructure providing a selection of interoperability service utilities, and shared meta-models providing the basis for negotiations toward shared understanding about the target concepts of the collaboration.

The Pilarcos framework [7,8] provides such an environment for establishing federated collaborations between autonomous enterprises. Interoperation is based on utilization of shared meta-models, such as Business Network Models [8], service offers, and service types [17], and a set of interoperability service utilities consisting of meta-information repositories, populators [7] and network management agents.

In the context of federated service-based communities and the Pilarcos framework, interoperability means effective capability of mutual communication of information and can be viewed from technical, semantic and pragmatic perspectives. *Technical interoperability* means that messages can be transported from one business service to another. *Semantic interoperability* asserts that actual content of collaboration is understood by the senders and the receivers; this means both correct interpretation of exchanged information elements and of business services' behavior. *Pragmatic interoperability* captures the willingness of partners for the actions necessary for the collaboration. The willingness to participate involves both capability of performing a requested action, and policies dictating whether it is preferable for the enterprise to be involved in.

The multi-dimensionality of interoperation must be managed by identifying and separating the different concerns into orthogonal aspects where each aspect grasps a different need or viewpoint of enterprise computing. Inter-enterprise interoperation is characterized in the Pilarcos framework as a layered collection of aspects [16] with four interoperation levels having increasing degree of abstraction: 1) technology, 2) service, 3) community, and 4) business level. Each abstraction level is orthogonal to the others and is further divided into orthogonal aspects. The layered view of interoperability concerns is illustrated in Figure 3. The three highest levels represent meta-information or knowledge that is needed to establish interoperability while in the lowest, technology level interoperability is achieved via integration of technological artifacts.

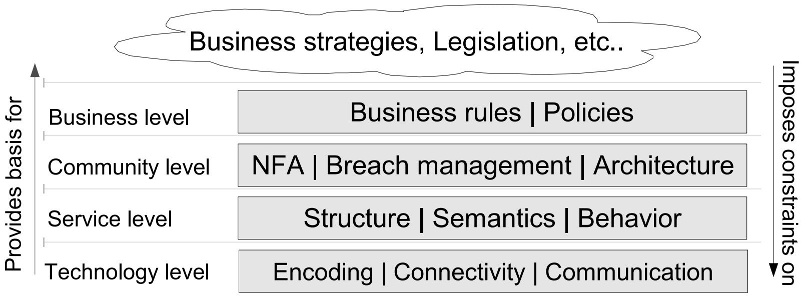
Business strategies, Legislation, etc..
Provides basis for
Imposes constraints on
Business level
Business rules | Policies
Community level
NFA | Breach management | Architecture
Service level
Structure | Semantics | Behavior
Technology level
Encoding | Connectivity | Communication

Fig. 3. Aspects of inter-enterprise interoperation.

4 The model of inter-enterprise interoperability

The different levels of inter-enterprise interoperation presented in Section 3 can be used for partitioning a system of service-based federated communities into sub-systems, where each one is concerned by one interoperation level. We start the specialization of the OoI from this hypothesis. Then, the different aspects of a level characterize its composing resources, by defining categories (e.g. Business rules) or properties (Behavior, as a property of services) of resources, or more generally domains (e.g. Connectivity, Architecture, Communication). As they can not be formally considered as sub-systems of levels, aspects will be linked in the specialized ontology to an *Aspect* concept, which is linked to a system and can be used to locate more precisely where a problem occur (*e.g.* in which aspect of a system). Figure 4 shows these main links between the domain-specific concepts and the OoI.

5 Formalizing federated interoperability management

Identification of the orthogonal aspects of interoperation is a crucial step toward managing interoperability in federated environments. However, it is not sufficient by itself and requires as a companion a characterization of how interoperation can be maintained within and between the distinct domains of concern. For this purpose, we introduce the concepts of *intra-aspect* and *inter-aspect* interoperability and discuss for each one solutions for achieving correct interoperation.

By intra-aspect (or horizontal) interoperability we mean aspect-specific characterizations for relationships guaranteeing interoperation between resources in the

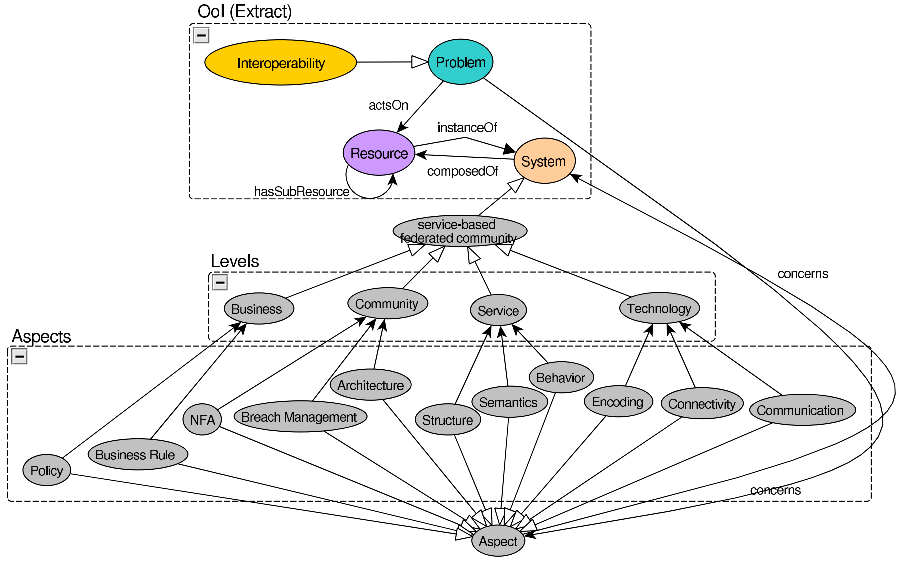

Fig. 4. Specialized OoI for service-based inter-enterprises communities, using the aspects of inter-enterprise interoperation.

same domain of concern. As intra-aspect interoperability problems concern resources of the same kind, belonging to a same domain of concern in one inter-enterprise interoperation aspect, they are clearly related to *heterogeneity* as defined now in the OoI.

At the technology level, technical interoperability must be achieved between communication and computation platforms by addressing issues related with connectivity, communication and encoding. Connectivity is a problem linked with *heterogeneity of interfaces*, while encoding issues are rather related to the model *representation*. Communication is indeed a specialization of *behavioral relation*, and thus related issues can occur at any level in the OoI. Interoperability is established *a priori* at the technology level by integration and unification of technologies and provides the foundation for interoperation at the higher levels. When a technology level interoperability problem is encountered during the operation of a community, a *compensation* action is needed to be performed outside the conflicting technology domain to fulfill the objectives of the community. This is an *a posteriori* solution. In the context of federated service communities, *bridging* is not applicable as a generic *a posteriori* solution at the technology level. If a technological incompatibility is found during operation, it might be too late (considering the commitments and contractual relationships) or impossible due to the autonomy of enterprises to try to homogenize the platforms and retry the failed activity.

At the service level, both technical (compatibility between business service and document structures) and semantic interoperability (semantics of information and behavior of business services) between service end-points must be established. Related issues of both aspects can be located at the level of *models*, in the OoI. Considering the information exchanged between business services, technical and semantic interoperability can be established *a priori* with homogenization of the vocabulary; for this purpose frameworks such as XML [20], ebXML [5] and OWL [19] can be applied. The behavioral aspects of business services can be formalized using methods such as pi-calculus [9]; accompanying analysis methods and tools can be applied for *a priori* interoperability validation.

Bridging which utilizes adapters, translation protocols and model transformations can be used as an *a posteriori* solution [10] for resolving service level incompatibilities during community establishment. Bridging is possible, since the semantics and structure of business services are explicitly available at this level. Using this information, appropriate bridging methods can be applied before the actual operation, and in this context, bridging at the service level provides also an *a priori* homogenization solution with respect to the operation of the community.

At the community level, deciding about the interoperability management methods has to be a joint process, since the properties of all the participants must be taken into consideration. *Negotiation* mechanisms are used for this purpose and the mutual agreements about the properties concerning the aspects of interoperation are formalized into a collaboration contract. Negotiation is used in this scenario as an *a priori* solution for homogenizing the view on the collaboration among the participants; the resulting contract provides an *a priori* solution as a homogenized model of collaboration with respect to the community operation. Prescribed *compensation* procedures are used as *a posteriori* mechanisms for resolving incompatibilities encountered during the operation of the community.

At the business level, pragmatic interoperability which considers business rules and organizational policies, is addressed. These rules comprise the autonomic intention and character of enterprises. Business rules are declarative statements that define or constraint some aspect of a business. Typical examples of business rules are different kinds of service pricing policies based on customer classifications. Organizational policies regulate the use of business functionality and knowledge provided by an enterprise. For example rules addressing accessibility, authorization, trust and privacy with respect to the provided business services and information are typical examples of organizational policies.

As an *a priori* solution, constraint satisfaction algorithms [6] can be used for finding a compatible intersection of policy values between enterprises. Horizontal interoperability problems at the business level are instances of more general *policy conflicts*. They are related to heterogeneity at the level of resource *models* in the OoI. Policy conflicts can be resolved *a posteriori* if *meta-policies* which state modality precedences (ordering between negative and positive predicates) or priorities among business rules [2, 4] have been defined. This prioritization approach can be applied to both business rules and organizational policies.

The *AntiPattern* indicators (see Figure 2) identify intra-aspect problems by utilizing knowledge about best practices and practical experiences of the corresponding domain of concern and provide solution methods for well-known problems. An example anti-pattern considers non-determinism in the externally observed behavior of business processes exported by organizations towards their clientele. For protecting their business assets, enterprises do not want to publish their internal business processes at a detailed level: only a restricted view on the internal processes for establishing interoperation is available. Now, the behavior observed by a client may seem non-deterministic since all the criteria and information for making the choices in business process branching points are not available; non-deterministic behaviour leads to interoperability problems between service behaviors. As an *a priori* solution, the internal business processes can be refactored in such way that different branches encountered in the external view of the process can be distinguished based on the communicated information content.

Inter-aspect (or vertical) interoperability concerns resources of different nature, when one imposes constraints that others need to respect. From this regard, inter-aspect interoperability is related to the *misalignment* concept introduced in section 2.1. Characterisation of inter-aspect interoperability is provided by a set of relationships, consistency rules, and conformance criterion between domains of concerns residing at different abstraction levels.

The *ConformancePoint* indicators becomes very important when wanting to secure inter-aspect interoperability. Concretely, they could be formalized as rules that a monitoring system will use at runtime to check a whole system consistency. As soon as each requirement expressed within an aspect is correctly identified and formalized, conformance points allow to check that there are no predictable *misalignments* among different aspects, leading to interoperability problems. For example, when organizational policies are represented using deontic logic, conformance between a policy and service behavior can be validated using model checking [11].

6 Conclusions

This paper introduced an extended version of the FIG-OOI Ontology of Interoperability and specialized it to the context of federated service communities. In the context of federated service communities, interoperability is characterized as a layered collection of interoperation aspects and corresponding intra- and inter-aspect relationships. A preliminary identification of the methods for analyzing these relationships was conducted as part of this paper. These topics will be further utilized in the domain of Model-Driven Engineering (MDE). When formalized as models and transformations, the interoperability aspects and relationships become very valuable as facilities for managing and establishing interoperability during model-driven development processes. Joined with decision aid tools based on the OoI ontology, a platform for interoperable service development with automatic or semi-automatic identification and validation of interoperability problems could possibly be attained.

The joint work presented in this paper has lead to some modifications of the original OoI of [10]. In particular, the scope of interoperability as been extended to cope also with misalignments, which are essential to consider in a multi-level organization where resources in one level can potentially influence directly resources in another level. Moreover, the different categorization of problems concerning technical, semantic and pragmatic interoperability considered in federated service communities and the Pilarcos framework, which we did not address here will be the subject of further investigations relatively to those already started concerning the enterprise interoperability framework proposed in [1].

References

[1] D. Chen. Enterprise Interoperability Framework. In *Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP)*, Jan. 2006.

[2] B. N. Grosof, Y. Labrou, and H. Y. Chan. A declarative approach to business rules in contracts: courteous logic programs in XML. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 68–77, New York, NY, USA, 1999. ACM Press.

[3] INTEROP. European Network of Excellence, 2004. `http://www.interop-noe.org`.

[4] L. Kagal and T. Finin. Modeling Communicative Behavior Using Permissions and Obligations. In *International Workshop on Agent Communication, AC 2004*, number 3396 in LNAI, pages 120–133. Springer-Verlag, 2005.

[5] A. Kotok and D. R. R. Webber. *ebXML: The New Global Standard for Doing Business Over the Internet*. New Riders, Boston, 2001.

[6] V. Kumar. Algorithms for constraint-satisfaction problems: a survey. *AI Mag.*, 13(1):32–44, 1992.

[7] L. Kutvonen, J. Metso, and S. Ruohomaa. From trading to eCommunity population: Responding to social and contractual challenges. In *EDOC 2006*. IEEE, Oct. 2006. To appear.

[8] L. Kutvonen, J. Metso, and T. Ruokolainen. Inter-enterprise collaboration management in dynamic business networks. In *OTM Confederated International Conferences, CoopIS, DOA, and ODBASE*, volume 3760 of *Lecture Notes in Computer Science*, Agia Napa, Cyprus, Nov. 2005.

[9] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, parts I and II. *Information and Computation*, 100:1–40 & 41–77, Sept. 1992.

[10] Y. Naudet, T. Latour, K. Haussmann, S. Abels, A. Hahn, and P. Johannesonn. Describing Interoperability: the OoI Ontology. In *Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP)*, Jan. 2006.

[11] N. Osman, D. Robertson, and C. Walton. Run-time model checking of interaction and deontic models for multi-agent systems. In *5th International joint conference on Autonomous Agents and Multiagent Systems (AAMAS'06)*, pages 238–240, New York, NY, USA, 2006. ACM Press.

[12] V. Rosener, T. Latour, and E. Dubois. A Model-based Ontology of the Software Interoperability Problem: Preliminary Results. In M. Missikoff, editor, *Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP)*. CEUR, 2004.

[13] V. Rosener, Y. Naudet, and T. Latour. A Model Proposal of the Interoperability Problem. In M. Missikoff and A. D. Nicola, editors, *Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP)*. CEUR, 2005.

[14] T. Ruokolainen and L. Kutvonen. Addressing Autonomy and Interoperability in Breeding Environments. In L. Camarinho-Matos, H. Afsarmanesh, and M. Ollus, editors,

Network-Centric Collaboration and Supporting Frameworks, volume 224 of *IFIP International Federation for Information Processing*, pages 481–488, Boston, Sept. 2006. Springer.

[15] T. Ruokolainen and L. Kutvonen. Extending the Model of Interoperability. In *Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP)*. CEUR, Jan. 2006.

[16] T. Ruokolainen and L. Kutvonen. Interoperability in Service-Based Communities. In C. Bussler and A. Haller, editors, *Business Process Management Workshops: BPM 2005 International Workshops*, volume 3812 of *Lecture Notes in Computer Science*, pages 317–328. Springer-Verlag, 2006.

[17] T. Ruokolainen and L. Kutvonen. Service Typing in Collaborative Systems. In *2nd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2006)*. Springer-Verlag, Mar. 2006. To appear.

[18] L. Von Bertalanffy. *General System Theory: Foundations, Development, Applications*. Georges Braziller, Inc., New York, USA, 1968.

[19] W3C. *OWL Web Ontology Language Guide*, Feb. 2004. W3C Recommendation 10 February 2004.

[20] W3C Working Groups. *Extensible Markup Language (XML)*. W3C, 2005. `http://www.w3.org/XML/`, valid 5th October 2005.