Abstract. In open MultiAgent Systems (MAS), interoperability is a key issue. The idea of open MAS is extremely important since it forms an adequate conceptual basis to model inherently distributed and heterogeneous scenarios. In this paper we argue that in open MAS built around the concept of organization (OC-MAS), beyond the communication and domain models, the organization model used by the agents can also be the source of interoperability problems. To deal with organizational interoperability issues we put forward an approach combining metamodels and transformations from Model Driven Engineering and ontologies from the Ontology Development.

Keywords: Multiagent System, Organization, Interoperability, Model Driven Engineering, Metamodel, Ontology.

1. Introduction

In the last few years, much of the research effort in software and system engineering has been applied to quest of new concepts and technologies suitable for the development of (large scale) open computational systems. Main features of such systems are [Hewitt and Inman 1991, Feiler and et al. 2006]: decentralization; integration of heterogeneous, inconsistent, and changing elements; development and use by a wide variety of stakeholders with unavoidably different, conflicting, complex, and changing needs; continuous evolutions and deployment. In the multiagent system (MAS) field, one of the main approaches deemed suitable to build open computational systems has been the Organization Centered MAS (OC-MAS) [Lemaître and Excelente 1998, Ferber et al. 2004]. An OC-MAS is an MAS conceived and implemented to have an explicit organization. Such organization is made explicit in the OC-MAS organization specification (Org-Spec), that is designed by using an organization modeling language (Org-ML). What characterizes the OC-MAS as open computational systems is the fact that they are able to accept external agents. External agents are agents whose design (at least their internal architecture) is not prescribed in the OC-MAS design. To enter the OC-MAS, the sole requirement imposed on the external agents is that they “understand” and abide by the specifications.
used inside the OC-MAS such as the communication, domain and organization specifications. Moreover, the OC-MAS as an approach to deal with openness is deemed suitable because much of the coordination and control necessary to achieve the global purpose of the OC-MAS is orchestrated by its explicit organization, which may be implemented to operate independently of any external agent.

In this paper we describe the problem of organization interoperability in the context of OC-MAS and sketch an approach based on metamodels and ontologies to deal with it. Moreover, we discuss in more details only part of the proposed approach. The paper is structured as follows: in the next section, we characterize the organizational interoperability problem, its importance and solution space. In section 3, we present the proposed approach to the organization interoperability problem. In section 4, we detail part of the work we have already made, according to the proposed approach. In section 5, we present our conclusions and future developments.

2. The Organizational Interoperability Problem

Interoperability is a key issue for open computational systems in general. Interoperability is the capacity of joining together two or more components to form a bigger system, despite differences in their interfaces, internal architectures or implementation technologies [Wegner 1996]. In particular, the acceptance of external agents by OC-MAS is only possible if interoperability mechanisms are in place.

Abstracting away the problems of how an external agent can access the services of the platform on which a given MAS is deployed, two major sources of interoperability problems may be considered. The first is the communication languages (communication primitives, their syntax and interpretations) used between agents that take part of an MAS. The second is the possibility of external agents hold different domain ontologies (the conceptualization of an application domain). To enter an MAS, an external agent has to be able to interpret the content of received messages in a way consistent with the sender’s intended interpretation.

Whenever we consider an OC-MAS, a third source of interoperability problems is identified: the Org-Spec and the Org-ML used to specify it. Similarly to the previous cases, to effectively operate together both the OC-MAS and the external agent, should agree in some way with respect to the used Org-Spec. In this context, we call such problem the organizational interoperability problem. Solutions to problems of organizational interoperability can be devised looking at the previous research on communication and domain interoperability. Roughly, this research can be divided into standards, mappings and integrations.

The definition of standard specifications, such as FIPA ACL [FIPA 2002] and OWL [W3C 2004], deals with the interoperability problem by smoothing away the heterogeneity that causes the interoperability problem. This is a solution for building new systems, but the problem may persist, for instance, with legacy systems. The definition of mappings between the specifications assumed by components that have to interoperate is a possible solution for these cases. The work on semantic interoperability in MAS has been driven by this mapping idea [Erdur et al. 2004, Park and Ram 2004, Kalfoglou and Schorlemmer 2003]. The mapping approach can lead to other problem: the high number of mappings to promote full interoperability between $n$ specifications. A
solution to this is to propose an integrated specification and $2 \times n$ mappings from and to this integrated specification.

Considering the organizational interoperability problem, we can see a similar and yet unexplored solution space. Unlike communication languages and domain ontologies, there is no standards to Org-ML for MAS. In part, this can be explained because the field is relatively recent and still in expansion with many research groups trying to promote their models. Also, to the best of our knowledge, we do not know about works that define mappings or integrations between Org-ML or Org-Spec in order to provide interoperability in OC-MAS. Arguably, the very same solution space for communication and domain interoperability composed of standards, mappings and integrations, with their virtues and shortcomings, can also be explored as a feasible solution space in the case of organizational interoperability.

3. The General Approach

Our general approach to provide organizational interoperability in OC-MAS consists of using some Model Driven Engineering (MDE) techniques [Schmidt 2006, Bézivin 2005, Favre 2004] and Ontology Development [Gasevic et al. 2006] in a framework based on mappings and integration between Org-MLs. Notably, we borrow from MDE the techniques of language specification by means of metamodels and metamodel based model transformations. From Ontology Engineering, we borrow the techniques for ontology modeling and ontology relation. The main idea is to represent mappings between Org-MLs as model transformations. To this, we first have to specify the Org-MLs using metamodels. As long as, in the current MDE technological space, the mature metamodeling and model transformations techniques are syntactic-driven, we propose the use of ontologies to provide a more meaning-oriented specifications to Org-MLs and ontologies relation techniques to aid in the process of defining the syntactic-driven model transformations.

In Fig. 1, we show the big-picture of our general approach to organization interoperability. It contains two external agents that represent possible scenarios where organizational interoperability must happen. For instance, the external agent named (1) interprets the specification Org-Spec-1 of an OC-MAS that was written in an Org-ML, that we call Org-ML-1. Suppose that this agent could be reused in another OC-MAS, that was specified via Org-Spec-2, which was written using Org-ML-2 (Org-ML-2 ≠ Org-ML-1). The solution proposed suggests the use of an intermediate Org-Spec Transformation Layer. This layer translates from Org-ML-2 to Org-ML-1, and this could be generalized pairwise. The translations are operationalized by model transformations between pairs of Org-ML and the transformations take into account only the Abstract Syntax of the Org-ML. Such transformations could be benefited from the use of semantic information derived from the Org-MLs metamodels. Moreover, we propose the definition of ontologies based on the Org-MLs’ abstract and concrete syntax to describe the Org-MLs metamodels (Org-ML ontologies). Each Org-ML Ontology is a more detailed and semantic oriented account of the organization concepts and relations behind the modeling elements that were found in each Org-ML metamodel. After the definition of the Org-ML ontologies it is possible to build an articulated ontology through the alignment of the Org-ML ontologies. The articulated ontology serves as a common vocabulary to define an integrated ontology that unifies the concepts from the Org-ML ontologies.
Nevertheless, the model transformations are in accordance to the Integrated Ontology in the following way: they map modeling elements from an Org-ML metamodel to modeling elements of another Org-ML metamodel; this happens when the corresponding concepts and relations on the respective Org-ML Ontologies correspond one to the others; this correspondence is established by following ontology mappings from an Org-ML-i ontology to the Integrated Ontology (the $h_i$), then from the Integrated Ontology to the Articulation Ontology (the $g$), and finally from the Articulation Ontology to a Org-ML-j Ontology (the $f_j$). We note that this solution to the scenario (1) works only when the mismatches between Org-MLs are manageable. This is not always the case but Org-ML generally overlap in their elements and concepts because they have the same the domain of discourse (MAS organizations) and inspiration source (human organizations) [Coutinho et al. 2007, Coutinho et al. 2005]. Another possible scenario is sketched by the external agent named (2). Suppose that it is able of entering/leaving in/from different OC-MAS during its lifetime, no matter the Org-ML behind the OC-MAS. Based on Fig. 1, one way to realize this is to design the External Agent with the capacities of both: reasoning in terms of the Integrated Ontology and translating this reasoning to a target Org-ML-i. The translation could be made by composing the mappings $g$ and $f_i$.

For the moment we have successfully designed, implemented and initially tested some Org-ML metamodels and syntactic-driven Org-Spec transformations (not considering Org-ML ontologies). These rather promising results, which illustrates the feasibility of the MDE part of our proposal, will be presented in the next section. Regarding Org-ML ontologies and their integration, concrete experiments remain to be done. In the Conclusion section, we will discuss how we are planning to do them.
4. Metamodels and Transformations

Let us imagine that the number of different Org-ML to deal with were reduced to only two. In the next sections, we illustrate this minimalist scenario considering AGR [Ferber et al. 2004] and MOISE+ [Hübner et al. 2002] as the two Org-MLs. First, we show metamodels to AGR and MOISE+. Second, we propose Org-Spec transformations from AGR to MOISE+ and vice-versa. Finally, we conclude by pointing to the need of Org-ML ontologies and their alignments to semantically enrich the Org-ML syntactic transformations.

4.1. Org-ML metamodels

AGR and MOISE+ are examples of Org-MLs that can be mapped. As expected, they present mismatches but this do not completely hinder partial Org-Spec transformations from being defined. In previous works [Coutinho et al. 2005, Coutinho et al. 2007] we have described and compared Org-MLs regarding their conceptual coverage of some modeling dimensions. Among the dimensions, the structural, functional, dialogical and normative were identified as the more explored aspects of organization modeling in the literature. Moreover, among these four dimensions, the structural dimension was the one that appears in all analyzed works. For this reason we will focus the discussion of organization interoperability on the structural dimension of the organization modeling.

AGR (Agent, Group, Role) [Ferber et al. 2004] is the evolution of the AALAADIN model [Ferber and Gutknecht 1998]. In AGR agent, group and role are the primitive modeling elements. In Fig.2(a), we present a metamodel for AGR containing these three core elements plus complementary modeling elements and their valid relationships. The metamodel is written in Ecore\(^1\) and it is strongly inspired in the metamodel presented in [Ferber et al. 2004]. In fact, the basic structure of the metamodel is the same of [Ferber et al. 2004] with minor modifications and some additions. Regarding the minor modifications what is more visible are some changes in cardinalities. For example, in [Ferber et al. 2004, p. 222] the cardinality between the classes GroupStructure and Group is 1 meaning that only one Group is described by each GroupStructure. In our metamodel (Fig. 2(a)), we have changed this cardinality to \(\star\) based on [Ferber et al. 2004] With respect to the additions, are the new classes ConcreteOrganization and OrganizationalStructure. The concepts behind these classes are presented in the text related to AGR but not in the metamodel of [Ferber et al. 2004]. We have decided to include them in our metamodel (Fig. 2(a)) because we think they are important concepts originated from the composition of Group and GroupStructure, respectively.

MOISE+ (Model of Organization for multiI-agent SystEms) [Hübner et al. 2002] is an Org-ML that explicitly distinguishes three aspects in an organization: the structural, the functional and the deontic aspects. The structural aspect defines the agents’ relations through the notions of roles, links and groups. The functional aspect describes how an MAS usually achieves its global goals, i.e., how these goals are decomposed in plans.

\(^1\)The choice of Ecore as metamodeling language is a practical one. As Ecore is part of the Eclipse Modeling Framework (EMF), we can more easily implement and test the proposal in the Eclipse platform.
Figure 2. Org-ML metamodels.

(a) AGR Metamodel.

(b) MOISE+ Structural Specification metamodel.
and are distributed to the agents by missions. Together, global goals, plans and missions give rise to patterns called social schemes. Finally, the deontic aspect describes the roles’ permissions and obligations for missions.

As explained earlier, our focus is on the structural aspects of MOISE+. A metamodel detailing the notions of roles, links and groups and their interrelationships in MOISE+ is presented in Fig. 2(b). In the metamodel we can see that a StructuralSpecification is composed of Role and GroupSpecification. Role can have superRole (inheritance relation). Role can also be the source or target of RoleRelation. There are two kinds of RoleRelation: Link and Compatibility. A GroupSpecification is composed of Link, Compatibility, SubGroup and GroupRole. SubGroup relates GroupSpecification to other GroupSpecification (the possible sub-groups of a group). GroupRole relates GroupSpecification to Role (the possible roles in a group). Both GroupRole and SubGroup are associated with Cardinality.

4.2. Org-ML Transformations

From the metamodels of AGR and MOISE+ two Org-Spec transformations are possible: from AGR to MOISE+ and from MOISE+ to AGR.

Transformation: \( AGR \rightarrow MOISE+ \)

1. let \( mss:MOISE+!StructuralSpecification \);
2. assign to \( mss.role \)
   
   (a) \( soc:MOISE+!Role \) such that \( soc.id='soc' \);
   
   (b) one different \( mr:MOISE+!Role \) for each \( ar:AGR!Role \) such that \( mr.id=ar.name \) and \( mr.superRole={soc} \);
3. assign to \( mss.groupSpecification \)
   
   (a) one different \( mgs:MOISE+!GroupSpecification \) for each \( ags:AGR!GroupStructure \) such that \( mgs.id=ags.name \) and
      
      i. let \( mrAGS \) be the subset of \( mss.role \) corresponding to the roles in \( ags.role \) respecting 2(b);
      
      ii. for each \( r \) in \( mrAGS \) assign to \( mgs.groupRole \) one new \( mgr:MOISE+!GroupRole \) such that \( mgr.role=r \);
   
   (b) one different \( mgs:MOISE+!GroupSpecification \) for each \( aos:AGR!OrganizationalStructure \) making \( mgs.id=aos.name \) and
      
      i. let \( mgsAOS \) be the subset of \( mss.groupSpecification \) corresponding to the group structures in \( aos.groupStructure \) respecting 3(a);
      
      ii. for each \( gs \) in \( mgsAOS \), assign to \( mgs.subGroup \) one new \( msg:MOISE+!SubGroup \) such that \( msg.subGroup=gs \);

\(^2\)The notation \( r:MM!C \) means “\( r \) is a reference to an instance of the class \( C \) from the metamodel \( MM \)”. This is inspired in the Atlas Transformation Language (ATL) [Jouault and Kurtev 2006] that we have used to implement the proposed transformation. We have chosen ATL because it is expressive enough to code our transformations and because it is the transformation language integrated with Ecore and EMF on the Eclipse platform (used to write the Org-ML metamodels).
iii. assign to mgs.link the singleton \{l\} such that
l.scope=INTRA\_GROUP, l.type=COMMUNICATION and
l.source=1.dest=soc;
iv. for each gr in mgsAOS.groupRole, assign to mgs.compatibility two new cl,
c2:MOISE+!Compatibility such that: cl.scope= INTRA\_GROUP, cl.source=soc, cl.target=gr.role;
and, c2.scope=INTRA\_GROUP, c2.source=soc, c2.target=gr.role.

The general idea of the transformation is to make (instances of) the classes Role, GroupStructure and OrganizationalStructure from the AGR metamodel (Fig. 2(a)) correspond, respectively, to (instances of) the classes Role, GroupSpecification and GroupSpecification from the MOISE+ metamodel (Fig. 2(b)). These correspondences are detailed in the steps 2, 3(a) and 3(b), respectively.

To formally prove that this general idea works - in the sense of preserving the semantics of an AGR specification - is not an easy task. It implies of having some AGR and MOISE+ semantics' definition that is not the goal of the metamodels we have presented. So, we have to content ourselves with pure natural language considerations regarding the definitions of AGR [Ferber et al. 2004] and MOISE+ [Hübner et al. 2002]. Nonetheless, we acknowledge this as a crucial point in our work - how to justify Org-Spec transformations from/to Org-ML metamodels? - and that is why we have put forward in this paper the idea of using ontologies to ground transformations between organization metamodels.

Informally, our rationale for the previously mentioned correspondences is:

- The Role definition in AGR (an abstract representation of a functional position of an agent in a group) corresponds in general lines what Role is in MOISE+. However, two details have to be considered: in an Org-Spec MOISE+ there is always a soc:Role instance and this is the root of an inheritance relation.
- A similar argument applies in the correspondence AGR!GroupStructure and MOISE+!GroupSpecification with the exception that instances of Role are owned by AGR!GroupStructure and not by MOISE+!GroupSpecification. In MOISE+, Role and GroupSpecification are related through RoleGroup.
- AGR!OrganizationalStructure corresponds to MOISE+!GroupSpecification because there is the concept of some GroupSpecification being the root of a hierarchical group structuring in MOISE+. In AGR there is no hierarchical group structuring but we can suppose that OrganizationalStructure contains GroupStructure as a one level hierarchical structuring in the sense of MOISE+. Other details that have to be considered are: in AGR communication is not permitted between two different groups and there is no concept of role incompatibility in the same group or between two different groups.

It is important to note that the other classes in the AGR metamodel, with the exception of InteractionProtocol, are not contained in the specification of an OrganizationalStructure. In this way, these classes are not considered in the AGR to MOISE+ transformation. In the case of InteractionProtocol, it is not
considered because it is a dialogical modeling concept and we are focusing on structural
modeling concepts.

However, there are three classes, RoleConstraint, Correspondence and
Dependency, that are in the scope of structural modeling and that are not considered
in the AGR to MOISE+ transformation. This fact illustrates a general basic shortcoming
in the mapping approach to provide interoperability: some source elements can find no
similar replacement in the target of a mapping. The important question of how an external
agent can be reused in face of such ORG-ML mismatches is outside the scope of this paper
and will be postponed to future works.

Using a similar reasoning, a mapping from MOISE+ to AGR can be defined.

Transformation : MOISE+ → AGR

1. let aos:AGR!OrganizationalStructure;
2. for each mgs:MOISE+!GroupSpecification
   (a) assign to aos.groupStructure one different
      ags:AGR!GroupStructure such that
      i. ags.name=msg.id;
      ii. for each mgr in mgs.groupRole assign to ags.role one differ-
      ent ar:AGR!Role such that ar.name=mgr.role.id;
   (b) for each mgr in mgs.groupRole where mgr.subGroupRole=true
      i. let mgsTC be the set of all MOISE+!GroupSpecification
         reachable from mgs by transitively following the
         mgs.subGroup.subGroup relation (transitive closure);
      ii. let mgrTC be the set of all gs.groupRole where gs is in mgsTC
      iii. for all gr in mgrTC such that mgr.role=gr.role then define a
          new ac:AGR!Correspondence such that
          A. ac.source=ar1 where ar1:AGR!Role corresponds to
             mgr.role according to 2(a)ii. ;
          B. ac.target=ar2 where ar2:AGR!Role corresponds to
             gr.role according to 2(a)ii. .

In this mapping, we have the following correspondences: MOISE+!Role
and AGR!Role; MOISE+!GroupSpecification and AGR!GroupStructure.
Some comments:

• Only concrete MOISE+!Role, i.e., roles associated with some
  MOISE+!GroupSpecification by means of a MOISE+!GroupRole
  instance are mapped to an AGR!Role. The reason for this is that in MOISE+,
  abstract roles (i.e., roles not concrete) are used only to define general restrictions
  shared by more specific concrete roles.
• The hierarchical structure of MOISE+!GroupSpecification is mapped to a
  flat AGR!GroupStructure set.
• Every MOISE+!GroupRole pointing to a role that can be played only in
  sub groups of a given group is mapped to an AGR!Correspondence.
  This is justified by saying that we have the same role referenced by two
MOISE+!GroupRole. This will give rise to two different AGR!Role by following rule 2(a)ii above. However, these two different AGR!Role are not meant to have different instance sets because in the MOISE+ specification the agents play the role in only one group.

Similarly to the previous mapping, it is important to note that some classes in the MOISE+ metamodel are not considered in the MOISE+ to AGR to transformation. Again, this fact illustrates shortcomings in this mapping approach: source elements have no similar replacement in the target of the mapping. Specifically, in the MOISE+ to AGR mapping all the information regarding role relations like permitted communications, authority and acquaintance are lost by the fact that AGR does not support these concepts.

4.3. E-market Org-Spec

In [Ferber et al. 2004], we found a simple Org-Spec of an e-market MAS. The Org-Spec is reproduced in the top of Fig. 3. The e-market is to be populated by external coming buyers and sellers agents intermediated by brokers agents. The Org-Spec is decomposed into three group specifications: a client group (ClientGS), a provider group (ProviderGS) and a contract group (ContractGS). Initially, an agent that wants to buy something (Client) enters the e-market in the client group and agents that have something to sell (Provider) enter in the provider group. The broker agent (Broker) is an internal agent (designed by the e-market designer) that can be at the same time in the client and provider groups. After having decided to buy something from a given Provider, both the Client and Provider go to the contract group (assuming the roles of Buyer and Seller, respectively) to formalize the transaction. In order to illustrate the transformations described in the previous section, the bottom part of Fig. 3 show how the e-market Org-Spec is translated from AGR to MOISE+. The encircled numbers denotes the steps taken from the AGR to MOISE+ transformation. For space reasons we will not present the inverse transformation.

5. Conclusions

We have presented a work in progress towards organization interoperability. An explicit characterization of organizational interoperability problems was presented, a general proposal to deal with them, and a first minimal implementation based on Org-ML metamodels and Org-Spec transformations.

The main characteristic of our approach is to address the process of defining mappings between metamodels by using well established results from MDE and Ontology Development. The main idea is to reuse concepts and technologies already in place in these fields to address the engineering of larger scale OC-MAS. Our short-term explicit objective has been to provide organization interoperability but, by using concepts and techniques from MDE and Ontology Development, we believe that our work is also contributing to: systematize the specification of Org-MLs; provide a detailed comparison of the available Org-MLs; provide a first ontological integration of existing Org-MLs.

Much work remains to be done. Our very next steps are to extend the Org-ML to all dimensions of organization modeling (here we presented only structural aspects of Org-ML) and to define Org-ML ontologies from the Org-ML. Regarding the Org-ML ontologies, we are planning to use an approach similar to the one presented in [Kappel and et al. 2006]. In this work, which proposes a general approach for semantic
integration of modeling languages comparable to our specific approach for Org-ML, the authors propose patterns to systematically promote metamodels to ontologies. After, we plan to begin the alignment and integration of the resulting ontologies. For this, we plan to use a semi-automatic process based on some tools like COMA++ [Aumueller et al. 2005]. Finally, we will address the problems of deriving Org-Spec transformations from the ontologies integration and validating the entire approach.

References


