Enterprise Architecture Modeling using Agent Paradigm

Olfa Lamouchi, Amar Ramdane-cherif, Nicole Lévy

PRISM, Université de Versailles St.-Quentin,
45, Avenue des Etats-Unis, 78035 Versailles Cedex, France.
{olla, rca, nlevy}@prism.uvsq.fr

Abstract: New approaches to modeling and design of enterprise systems must enable enterprises to offer dramatically improved capabilities including more effective enterprise architectures, more efficient business processes, the ability to engage in new much more dynamic forms of global co-operation, and greatly improved interoperability. The focus of the work in this paper is the application of agent-oriented software development methodologies to architecture enterprise modeling as well as the management of the variability guided by a strategic cost/schedule/quality or benefit/value/risk goals. This study will be the first step of working towards enabling tool support for enterprise variability management, enterprise modeling, validating and deducting the consequences of constrains or feature interactions in enterprise organization.

1 Introduction

Planning and Managing an Enterprise Infrastructure correspond to a very active and steadily growing area. For instance, collaborative networked organizations (CNO), virtual enterprises/virtual organizations (VE/VO) are already supported by a large research and business practice community. These techniques just give one aspect of enterprise modeling which suggest new ways of work and put the emphasis on collaborative networks of human actors. Further to these main lines, other collaborative forms and patterns of collaborative behavior are emerging, not only in industry, but also in service sector, as well as governmental [GGBC01][OPR00]. One approach which accommodates such organizational modeling and analysis is founded on the premise that organizations are made up of strategic, intentional actors. The Strategic Dependency model [Y95] allows the modeling of how strategic actors relate to each other intentionally, while the Strategic Rationale model allows modeling of the means-ends reasoning employed by organizational actors as they explore alternative ways of relating to each other in fulfilling goals and accomplishing work. The Business Process model offers a declarative, logic-based notation for modeling, verifying and validating business processes.

Most existing process models mentioned above are deficient in that they cannot deal with descriptions of all the enterprise modeling concepts and variation management. These models do not provide standards semantics for validating and deducting the consequence of constrains of features interactions in the global enterprise organization.

The focus of the work in this paper is the application of agent-oriented software development methodologies [ONBCMB00] to architecture enterprise modeling as well as the management of the variability guided by business objectives.

2 UML for Agent-Oriented Analysis Models

A set of UML concepts can be used or extended easily for agent-oriented analysis. Agent-oriented analysis employees a rich set of concepts, which makes it difficult to understand all the aspects of the analysis model from a single viewpoint. It is convenient, therefore, to define a number of sub-models that emphasize different aspects of the full model. These are not disjoint models, but are rather different perspectives on a single complex model. We use the following sub-set of models given in [S99][E00] (for space limitation we will not detail more these models):

Organization Model: The Organization Model (OM) aims at defining the structure of organizations and the behavior of a group of agents belonging to organizations and working together to reach common goals. The Organization Model represents the responsibilities and authorities with respect to entities such as processes, information, and resources. It represents the structure of the organization in terms of sub-organization such as departments, divisions, sections, etc. (e.g.: Fig. 2.1)

Activity/Task Model: The Activity/Task Model (ATM) describes how high level goals (for example defining purposes of an organization) are decomposed into lower level goals (e.g. ones that can be assigned to constituent agents). Similarly, it shows how high level tasks (e.g. services provided by an organization) can be decomposed into sets of sub-tasks (e.g. services that are provided by the constituent agents).

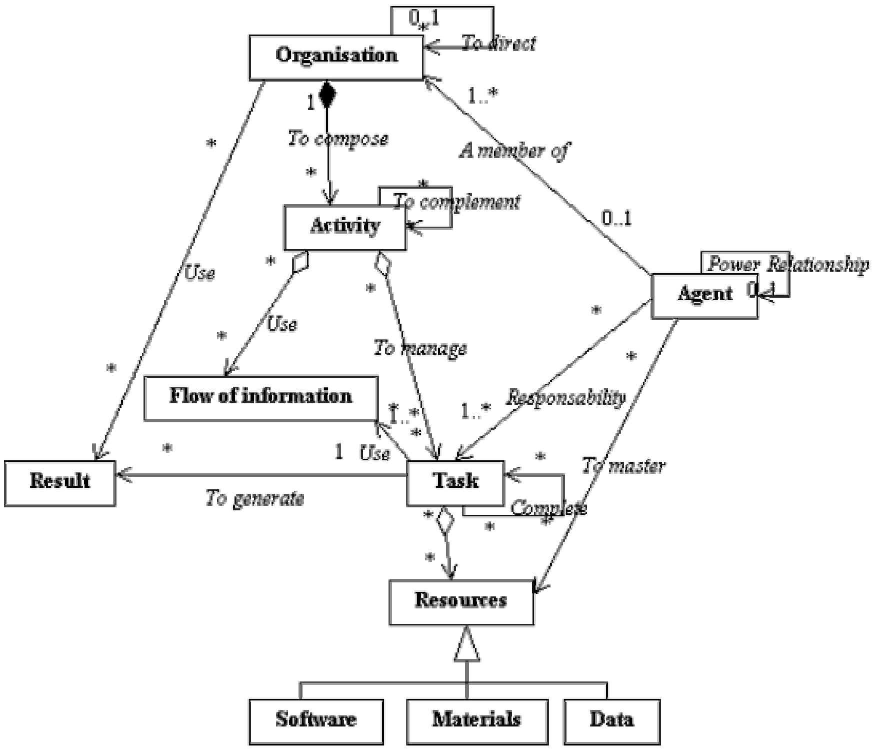

Figure 1: Structure of the Organization Model of Enterprise

Agent Model: The Agent Model (AM) consists of descriptions of the purpose, relationships, behavior, and other attributes of individual agents. Each description gathers together information on an agent from other models and adds internal detail.

Interaction Model (IM): The need for the agent communication language ACL is very important insofar as the agents must communicate efficiency for cooperate, coordinate and negotiate.
In deriving a model of Multi-Agent System to deal with the enterprise organization, there is a natural ordering to development of the models. First OM and ATM are developed and linked so that there is a correspondence between sub-goals and the purposes of agents and between sub-tasks and services the agents can provide. This process identifies the agents and their main features, the details of which are elaborated in the AM. However, to achieve the higher-level goal the agents must cooperate, and this drives the development of the IM.

3 Design Algebra formalism

The models mentioned above are deficient in that they cannot deal with descriptions of variation management and deducting the consequence of constrains of features interactions in the global enterprise organization. However , in order to provide:

- A variation model which consists on a set of variation points located into different levels of the enterprise organization and the relation between them.
- A range of alternatives for a given variation point (deducting the consequence of a variation on the global enterprise organization)
- A range of decisions for deriving a strategic cost/schedule/quality or benefit/value/risk goals to reduce the set of alternatives for a given variation point.

We integrate in our approach the design algebra formalism. This formalism is largely used in the variability management for product line families [KCNS90][TA03]. In this formalism, we consider that the feature model of enterprise organization is spanned by an independent set of dimensions. We define a dimension as a mandatory feature of a concept. As such the dimensions of *C* (e.g.: Fig.3.2) are the sub-features *C12, C13, C14, C15*. The set of dimensions of a concept is defined as its dimension set. In design algebra, we define the model of C (e.g.: Fig.3.2) as follows: C=(C12 ∧ C13 ∧ C14 ∧ C15), the symbol '∧' defines the composition relation in the feature diagram. The set of coordinates of a dimension are defined as the coordinates set of each dimension. The coordinates of a dimension may be a mandatory feature, alternative feature, optional feature or an or-feature [KCNS90][TA03]. These different feature properties are represented using the following symbols: '∧' mandatory, ';' alternative, '?' optional, '∨' or. Consider the example (e.g.: Fig.3.2) we can give the following expressions

Feature Model

C=(C12 ∧ C13 ∧ C14 ∧ C15)
C12=(C121 ∨ C122)
C14=((C141 ; C142) ∧ C143?)

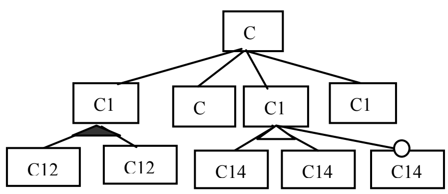

Figure 2: Feature model

Similar to the composition relations [KCNS90][TA03] in feature models we adopt constraints to express the constraints between various features in the model. We apply the *mutex-with* and *requires* composition rules. We can identify the following set of constraints, we called it 'Constraints model' ('.' is used to denote the bindings).
Mutex-with : defines a mutual exclusion relation between two concepts or features.
Requires : defines which features the selected feature requires (interdependent relations).

Constraints Model: *C.C12.C121 mutex-with C.C14.C142; C.C12.C122 requires C.C14.C141; C.C12.C122 requires C.C14.C142; C.C15 requires C.C14.C142.*

Besides of these constraints we can also define others constraints in a similar way. In the feature model the set of dimensions is related to the set of variation points and the set of constraints describe the relation between variation points. Once the feature model and the constraints model have been defined we need to derive for a given variation point a set of alternatives (the impact of one modification on the all enterprise organization).

Set of alternatives: *If (C142 is modified) then the following dimensions will be affected C142→C14→C→C12→C122; C142→C14→C→C15; If (C141 is modified) then the following dimensions will be affected; C141→C14→C→C12→C122.*

In design algebra the operation *unfold()* can be applied, which results in the total set of alternatives that can be derived from a given variation point in the feature model. All the alternatives will be analyzed and select them separately. In order to reduce the set of alternatives so that only the relevant alternatives are considered. The selected alternatives must satisfy some quality attributes following a strategic cost/schedule/quality or benefit/value/risk goals. A set of rules of the form *If 'condition' then select 'alternatives'* can be used to select the relevant alternatives. The condition can be made up of several logical expressions based on some metrics satisfying the flexibility, security, productivity, and/or other quality requirements. Therefore, the acceptable alternatives are driven by quality goals.

Decision making model : *If (the quality attribute 'productivity' related to the dimension C15 is higher) then C141→C14→C→C12→C122 will be selected and then the decision to modify C141 and not C142 will be taken.*

4 Application Example

In order to illustrate our approach, we apply the OM on a concrete example [TA03]. This example present some constraints and rules to manage an insurance company (g.: Fig. 4.3). Using our approach, we will derive the following models:

Feature model

Headquarters = (ParisAgtency ∧ LyonAgency ∧ NiceAgency); LyonAgency = (Insurance ∧ Accountingdepartement); Accountingdepartement = (Managementofthepay ∧ Out/InInsurance); Out/InInsurance = (Collection ∧ Payement); Insurance = (InsuredObject ∧ Converage ∧ Payment ∧ Payee); Payement = ((Service ; Amount) ∧ OwnRisk?); Converage = (Life ∨ Loss ∨ Dommage); Payee = (Person ∨ Corporation); InsuredObject = (Corporation ; Realty ; MoveableProperty ; Person).

Constraints model

InsuredObject.person mutex-with *Coverage.damage => If the ensured object is a person then the insurance product cannot include coverage of damage.*

InsuredObject.corporation requires *Payee.Corporation => If the insured object is a corporation then the claimer shoud also be a corporation.*

Set of alternatives for one scenario of variation

a) Modification Scenario 'ChangeAgent'

We suppose to replace the agent "agent1" by an other agent "agent3". The agent "agent3" don't perfect the commands of the software tool "AssurTT". We supose the folowing caracteristics: Agent1: Competence 80%; Agent3: Competence 10%; AssurTT: Usability 50%. We can define the following relations:

R1: (AGENTS →QUALITY-ATTRIBUTES) →VALUES. Where AGENTS is a set of agents of the organization QUALITY_ATTRIBUTES is a set of quality-attributes; VALUES is a set of percentage values.

Then we give the following elements of the relation R1:

Agent1 Є AGENTS, competence Є QUALITY_ATTRIBUTES, 80% Є VALUES,

((Agent1 ⟼ competence) ⟼ 80%) Є R1, ((Agnet3 ⟼ competence) ⟼10%) Є R1

R2: (RESSOURCES → QUALITY_ATTRIBUTES)→VALUES. Where RESSOURCES is a set of resources of the organization. QUALITY_ATTRIBUTES is a set of quality-attributes; VALUES is a set of percentage values.
Then we give the following elements of the relation R2:
AssurTT Є RESSOURCES, usability Є QUALITY_ATTRIBUTES, 50% Є VALUES
((AssurTT ⟼ usability) ⟼ 50%) Є R2

b) Set of alternatives

If (agent1 is modified) then the following dimensions will be affected; Agent1→Corporation; Agent1→Realty; Agent1→MovableProperty; Agent1→Corporation → InsuredObject→ Converge→Damage; Agent1→Corporation →InsuredObject→Converge→Loss Agent1→Realty →InsuredObject→Converge→Damage; Agent1→Realty → InsuredObject→Converge→Loss; Agent1→ MovableProperty → InsuredObject → Converge→Damage; Agent1→ MovableProperty → InsuredObject→Converge→Loss

The result of the agent change can be presented either in the form of report or in the form of graph, it gives the parts of organization structure which will be affected by the change. In order to evaluate the impact of this modification on the productivity attribute quality, we will use the following equation (Eq.1: Equation evaluation of productivity quality):

$$P_{rof} = nbrAct^{-1} \sum_{i=1}^{nbrAct} (nbrTask^{-1} \sum_{i=1}^{nbrTask} (2^{-1} ((nbrAgt^{-1} \sum_{i=1}^{nbrAgt} competenceAgt\ i) + (nbrRes^{-1} \sum_{i=1}^{nbrRes} usabilityRes\ i)))) \quad (1)$$

Where :

Word	Signification	Word	Signification
:P_{rof}	*% value of the productivity*	*:nbrAgt*	*Number of agents*
:nbrTask	*Number of tasks*	*:competenceAgt*	*% competence of the agent*
:nbrAct	*Number of activities*		
:nbrRes	*A number of resources used by the agent i1, in order to realize the task i1*		
:usabilityRes	*% value of the usability of a resource*		

In order to measure the impact of the replacement of "agent1" by "agent3" on the productivity of the enterprise, we will apply the equation (1) for each agent.

Productivity of agent1 :
$P_{rof} = 2^{-1}$ (80%+ 50%) = 65%

Productivity of agent3:
$P_{rof} = 2^{-1}$ (10%+ 50%) = 30%

"Agent1", who has a work experience within company, using the software "AssurTT", has a productivity rate of 65%. If we replace "Agent1" by 'Agent3' (which perfects less the software tool "AssurTT":10%), we decrease the global productivity to the rate of 30%. During the formation of "Agent3" and for a certain duration, the company will lose 35% (65%-30%) of its total productivity. In this example we don't introduced the time parameter.

5 Conclusion

Our approach for organization modeling is based partly on a UML models (OM, ATM, AM, IM) that involve social actors who depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. For describing the network of relationships among actors (and among all the elements of the OM), we use the design algebra formalism. Our approach provides a method for variability management of the enterprise organization driven by the improvements of quality goals. Our future work includes the implementation of all the above concepts (the conceptual modeling and their semantics) in a tool. This tool will help companies in the way they do and manage changes in their organization and help to capture and analyze the strategic relationships among business work units and external players.

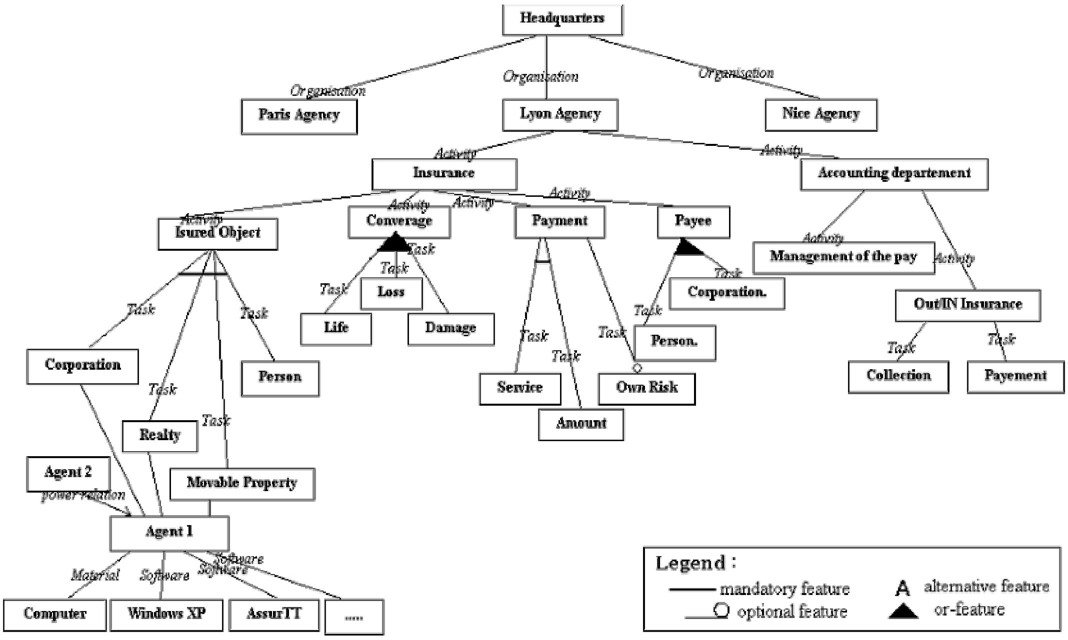

Figure 3 : Organization of company of insurance system

References

[E00] EURASCOM shareholder companies. Project P907-GI. Methodology for Engineering Systems or Software Agents, 2000.

[S99] Schreiber & al, Knowledge Engineering and Management: The CommonKADS Methodology, Bradford Books, 1999.

[ONBCMB00] O'Sullivan, D.; Núñez-Suárez, J.; Brouchoud, H.; Cros, P.; Moore, C.; Byrne, C.: Experiences in the use of FIPA agent technologies for the development of a Personal Travel Application, , Proceedings Agents 2000, Barcelona, June 2000.

[KCNS90] Kang, K.C.; Cohen, G.; Hess,J.A.; Novak, W.E.; Spencer Peterson, A.S.: Fearure-oriented Domain Analysis (FODA) Feasibility! Study. Technical Report, CMLUSET-90-TR-21 ESD-90-TR-222, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, November 1990.

[TA03] Tekinerdogan, B.; Aksit, M.: Managing Variability in Product Line Scoping using Design Space Models, in: J. van Gurp, J.; Bosch, J. (eds.), Proc. Workshop Software Variability Management, Groningen, IWI 2003-7-01, The Netherlands, February 13/14, 2003; pp. 5-12.

[GGBC01] Galland, S.; Grimaud, F.; Beaune, P.; Campagne, J.P.: A study case with a methodological approach for complex and distributed industrial system simulation, 15ème Conférence Internationale Industrial Engeneering and Production Management, Québec, Canada, août 2001.

[OPR00] Oquendo, F.; Papamichail, K.N.; Robertson, I.: Addressing Decision Making Issues in Enterprise, Process Models Proceedings International Conference of Enterprise Information Systems, Stafford, UK. July 2000.

[Y95] Yu, E.: Modelling Strategic Relationships for Process Reengineering, Ph.D. thesis, Department of Computer Science, University of Toronto,1995.