

A Tropos Model-Driven Development Environment

Davide Bertolini, Loris Delpero, John Mylopoulos, Aliaksei Novikau, Alessandro Orler, Loris Penserini, Anna Perini, Angelo Susi, and Barbara Tomasi

ITC-IRST, Via Sommarive 18, I-38050, Trento, Italy

{bertolini,delpero,novikau,orler,penserini,perini,susi,tomasi}@itc.it,
jm@cs.toronto.edu

Abstract. This short paper illustrates a development environment supporting the use of the *Tropos* methodology for building complex Multi-Agent Systems, starting from the early requirements specification down to the concrete JADE agents implementation. A detailed description of the methods supported by the tool are described in a companion paper [7].

1 Introduction

Many Agent-Oriented Software Engineering methodologies have been proposed [5] to support the development of complex distributed systems including Multi-Agent Systems (MAS).

In our work we focus on the *Tropos* methodology [2], an Agent-Oriented software development methodology that proposes the use of knowledge level concepts, such as actor, goal, plan and dependency between actors, along the whole software development process. The process covers five software development phases: *early requirements analysis*, in which analysis focuses on the understanding of a problem domain by studying an *existing organizational setting* where the system-to-be will be introduced; *late requirements analysis*, in which analysis focuses on the system-to-be which is introduced as a new actor into the model; *architectural design*, that defines the system's global architecture in terms of subsystems, represented as actors; *detailed design*, that aims at specifying the agent micro-level, their capabilities and interactions, and finally *implementation*, that aims at producing an implementation skeleton according to the detailed design specification.

Tropos uses a visual modeling language based on the *i** framework [9]. So, a *Tropos* model can be represented as a set of diagrams like for instance *actor diagrams*, describing the network of dependency relationships among actors, *goal diagrams* that illustrates goal and plan analysis from the point of view of a specific actor. During the detailed design phase the UML/AUML interaction diagrams and activity diagrams are used in order to describe the processes that the agents can use for fulfilling their goals and the messages they exchange during the execution of these processes in the distributed environment. Finally, for the concrete system implementation, an agent platform, such as JADE or Jack, can be chosen.

In [7] we refined the methodology to address the problem of allowing the traceability of MAS capabilities to the stakeholders needs elicited at the early requirements phase, defining concepts and process guidelines for its application. In order to be used

in practice these refinements need the development of CASE tools at support of the different modelling and design tasks, such as the analysis of models or the automatic transformation among different model specification languages. The final objective is the improvement of the quality of software products and of the development process, allowing for the reuse of models and the mappings between models. These latter issues are considered in the Model-Driven Architecture (MDA) initiative of OMG [6] that proposes an approach to software development based on modeling and on the automated mapping of source models to target models by defining model interoperability standards.

In this short paper, we describe a development environment that supports the entire *Tropos* software development process, from the early requirements definition to the JADE agent implementation. In the design of the environment we considered the issues related to the MDA perspective.

2 The Tropos process support environment

The development environment at support of the *Tropos* methodology allows the analyst to specify a *Tropos* model, to automatically translate part of the model into an AUML model which should be editable by an AUML modeler and finally to support the implementation of the Multi-Agent models into an Agent-Oriented framework. The environment is based on the Eclipse Development Platform [1] that offers a flexible solution to the problem of component integration. New tools can be integrated into the platform through plug-ins, the smallest unit of function in Eclipse, that provide new functionalities to the environment.

The resulting development environment is composed by different tools that support the analyst in the different steps of the methodology, as shown in Fig. 1. In the following, we give a sketchy view of the use of the development environment along the five development phases, giving also a brief description of the tools currently integrated in the environment.

Early Requirements Phase. During this phase is specified the initial organizational setting of the domain before the introduction of the system-to-be, in terms of actors, their goals and their dependencies among actors.

The specification of this initial model is performed using the *TAOM4E* (labeled (1) in Fig. 1) modelling tool [8]. The tool allows to model entities of the domain following the *Tropos* language metamodel defined in [8] designed according MDA directives and to the Meta Object Facility (MOF) standard [4] which defines a set of modeling constructs that allow to manage meta-models interoperability via the XML Model Interchange (XMI). *TAOM4E* has been developed inside the Eclipse platform, on top of two existing plug-ins, the Graphical Editing Framework (GEF) plug-in that manage the editing functions and the EMF plug-in which offers a modeling framework and code generation for MOF compliant metamodels¹.

The *TAOM4E* consists of two plug-ins: (a) the *TAOM4E model*, which implements the *Tropos* meta-model exploiting the EMF plug-in; (b) the *TAOM4E platform*, which

¹ <http://www.eclipse.org/gef/>, <http://www.eclipse.org/emf/>

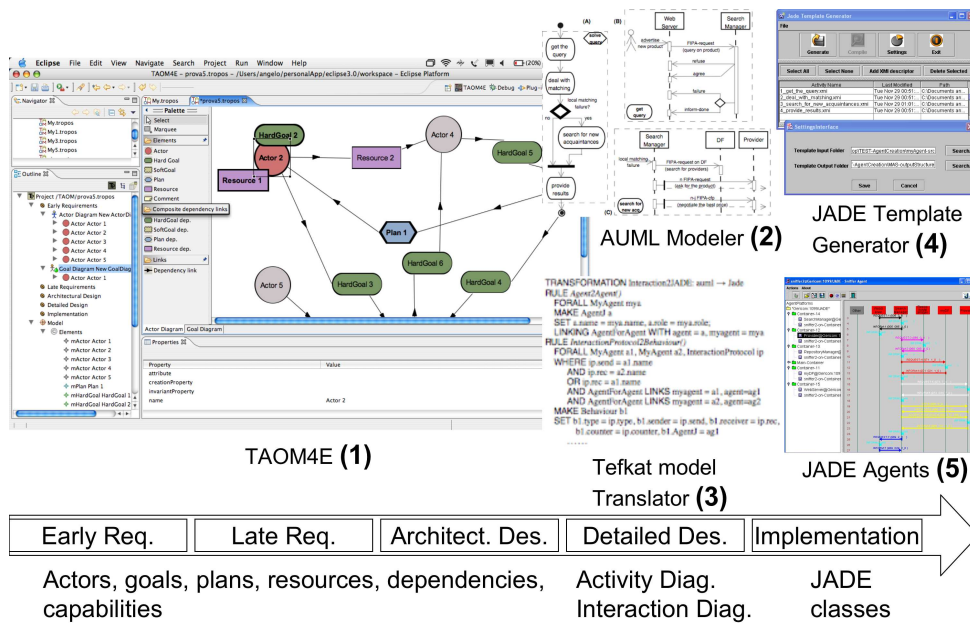


Fig. 1. The *Tropos* five phases process, and the Model-Driven development environment which support the methodology.

implements modeler functions needed for building and managing a *Tropos* model exploiting the GEF plug-in.

The modelling tool supports the specification of the *Tropos* model via the GUI that makes available the editors for the description of the actor and goals diagrams, for the navigation of the model and for the specification of the model entity properties. Moreover, the tool allow to save the *Tropos* models in XMI format.

Late Requirements phase. Also this phase is supported by the TAOM4E tool. Here, the System-to-be is introduced in the domain model as a new actor. The dependencies among the system and other actors in the domain are specified, and an internal system analysis is performed, via the goal analysis techniques. This way the system requirements are specified.

Architectural Design phase. In this phase TAOM4E allows to specify the subsystems of the system-to-be and their relationships with the other subsystems and with the external actors. Moreover, here, the set of capabilities of the system are specified in terms of both means-ends goal/plan relationship and plan/softgoal contributions [7]. The set of capabilities are then maintained in a capability table.

Detailed Design phase. At this stage, we use AUML activity and sequence diagrams to model part of the dynamic properties of agent capabilities specified in the previous phases.

In our framework, we propose the use of metamodel level transformation techniques related to the proposals on Query/View/Transformation (MOF QVT [4]). In particular the framework uses an Eclipse transformation plug-in as a model transformation engine, called *Tefkat*² (labeled (3) in Fig. 1), that implements a subset of requirements and of the transformation language described in the DSTC proposal for MOF QVT as in [3]. We use the QVT to transform the model of the capability of agents, expressed via the *Tropos* entities and structures (in particular the goal/plan means-ends association as described in [7]), into the definition of the process underlying the capability execution via AUML activity diagrams that can be edited via a *UML/AUML modeller* (labeled (2) in Fig. 1). In the same phase, a second transformation is performed in order to transform, at a metamodel level the activity and sequence diagrams that specify the behavior of the agent, into a set of classes representing the JADE metamodel [7].

Implementation phase. During this phase, a last transformation is performed in order to produce the JADE (Java) template classes of the set of agents specified so far, starting from the JADE classes defined during the last part of the Detailed Design. This transformation is performed via the *JADE Template Generator* tool (labeled (4) in Fig. 1) which starts from the agent capabilities specification (given in XMI files) automatically generated from the Tefkat engine and produces the *JADE Agents* code (labeled (5) in Fig. 1).

References

1. *ECLIPSE Platform Technical Overview*, object technology international edition, July 2001. <http://www.eclipse.org>.
2. P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, July 2004.
3. CBOP, DSTC, and IBM. MOF Query/Views/Transformations, 2nd Revised Submission. Technical report, 2004.
4. T. Gardner, C. Griffin, J. Koehler, and R. Hauser. A review of omg mof 2.0 query / views / transformations submissions and recommendations towards the final standard. In *MetaModelling for MDA Workshop*, York, England, 2003.
5. B. Handerson-Seller and P. Giorgini. *Agent-Oriented Metodologies*. Idea Group, 2005.
6. S. J. Mellor, K. Scott, A. Uhl, and D. Weise. *MDA Distilled*. Addison-Wesley, 2004.
7. L. Penserini, A. Perini, A. Susi, and J. Mylopoulos. From Stakeholder Intentions to Software Agent Implementations. In *Proceedings of the 18th Conference On Advanced Information Systems Engineering (CAiSE'06)*. LNCS, Springer-Verlag, 2006.
8. A. Susi, A. Perini, P. Giorgini, and J. Mylopoulos. The Tropos Metamodel and its Use. *Informatica*, 29(4):401–408, 2005.
9. E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, Department of Computer Science, University of Toronto, 1995.

² Tefkat is part of Pegamento project of the DSTC in the University of Queensland <http://www.dstc.edu.au/Research/Projects/Pegamento/tefkat/index.html>