Tropos

Tropos is an Agent-Oriented software development methodology that uses knowledge level concepts, such as actor, goal, plan and dependency between actors, along the whole software development process. The process covers five development phases: early requirements analysis, late requirements analysis, architectural design, detailed design and implementation. Tropos exploits a visual modeling language based on the i* framework. A Tropos model can be represented as a set of diagrams. During the requirements phases actor diagrams and goal diagrams are used to describe respectively the network of dependency relationships among actors and the goal and plan analysis from the point of view of a specific actor. In the design phases UML/AUML interaction diagrams and activity diagrams are used to describe the processes that agents exploit for fulfilling their goals and 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 is exploited.

In our work, we aim at supporting the entire Tropos development process, providing guidelines and CASE tools. Traceability from stakeholder needs elicited at the early requirements phase, to the final MAS implementation, is favored.

The Tropos development framework

A set of CASE tools are integrated into the environment and support the analyst in the different steps of the methodology: when specifying a Tropos model, in automatically translating part of the model into an AUME/UML model which should be editable by an AUME/UML modeller, and finally in implementing the Multi-Agent models into an Agent-Oriented framework. The environment is based on the Eclipse Development Platform that offers a flexible solution to the problem of component integration through plug-in mechanisms.

The specification, of requirements and system architecture model, is performed using the TAOM4E modelling tool. The tool allows to model entities of the domain following the Tropos meta-model designed according to MDA directives and to the Meta Object Facility (MOF) standard. TAOM4E has been developed inside the Eclipse platform and consists of two plug-ins: (a) the TAOM4E model, which implements the Tropos meta-model exploiting the EMF plug-in, (b) the Tefkat engine, called Tefkat. The transformation mechanisms are here exploited to transform the model of each capability of agents, expressed via Tropos entities and structures, into AUME/UML activity diagrams. This first transformation allows the designer to deal with dynamic properties of the capability by means of activity and sequence diagrams. The second transformation deals with the behavior of the agent as a set of classes in the JADE metamodel.

Moreover, we propose the use of MDA metamodel level transformation techniques related to the proposals on Query/View/Transformation. In particular the framework uses the Eclipse transformation plug-in as a model transformation engine, called Tefkat. The transformation mechanisms are here exploited to transform the model of each capability of agents, expressed via Tropos entities and structures, into AUME/UML activity diagrams. This first transformation allows the designer to deal with dynamic properties of the capability by means of activity and sequence diagrams. The second transformation deals with the behavior of the agent as a set of classes in the JADE metamodel.

A fragment of the process: the case of an on-line service

Tropos is an Agent-Oriented software development methodology that uses knowledge level concepts, such as actor, goal, plan and dependency between actors, along the whole software development process. The process covers five development phases: early requirements analysis, late requirements analysis, architectural design, detailed design and implementation. Tropos exploits a visual modeling language based on the i* framework. A Tropos model can be represented as a set of diagrams. During the requirements phases actor diagrams and goal diagrams are used to describe respectively the network of dependency relationships among actors and the goal and plan analysis from the point of view of a specific actor. In the design phases UML/AUML interaction diagrams and activity diagrams are used to describe the processes that agents exploit for fulfilling their goals and 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 is exploited.

In our work, we aim at supporting the entire Tropos development process, providing guidelines and CASE tools. Traceability from stakeholder needs elicited at the early requirements phase, to the final MAS implementation, is favored.

The Tropos development framework

A set of CASE tools are integrated into the environment and support the analyst in the different steps of the methodology: when specifying a Tropos model, in automatically translating part of the model into an AUME/UML model which should be editable by an AUME/UML modeller, and finally in implementing the Multi-Agent models into an Agent-Oriented framework. The environment is based on the Eclipse Development Platform that offers a flexible solution to the problem of component integration through plug-in mechanisms.

The specification, of requirements and system architecture model, is performed using the TAOM4E modelling tool. The tool allows to model entities of the domain following the Tropos meta-model designed according to MDA directives and to the Meta Object Facility (MOF) standard. TAOM4E has been developed inside the Eclipse platform and consists of two plug-ins: (a) the TAOM4E model, which implements the Tropos meta-model exploiting the EMF plug-in, (b) the Tefkat engine, called Tefkat. The transformation mechanisms are here exploited to transform the model of each capability of agents, expressed via Tropos entities and structures, into AUME/UML activity diagrams. This first transformation allows the designer to deal with dynamic properties of the capability by means of activity and sequence diagrams. The second transformation deals with the behavior of the agent as a set of classes in the JADE metamodel.

Moreover, we propose the use of MDA metamodel level transformation techniques related to the proposals on Query/View/Transformation. In particular the framework uses the Eclipse transformation plug-in as a model transformation engine, called Tefkat. The transformation mechanisms are here exploited to transform the model of each capability of agents, expressed via Tropos entities and structures, into AUME/UML activity diagrams. This first transformation allows the designer to deal with dynamic properties of the capability by means of activity and sequence diagrams. The second transformation deals with the behavior of the agent as a set of classes in the JADE metamodel.

For further information

http://sra.itc.it/tools/taom4e/

Acknowledgements: This work has been partly supported by the STAMPS Project.

Center for Scientific and Technological Research ITC-IRST - Povo (Trento) – Italy

Department of Computer Science and Telecommunications - University of Trento, Italy

http://www.troposproject.org