Building and Browsing Tropos Models: The AVI Design

Tania Di Mascio[1], Anna Perini[2], Luca Sabatucci[2], and Angelo Susi[2]

[1] University of L'Aquila, Monteluco di Roio, L'Aquila, I-64100, Italy
tania.dimascio@univaq.it
[2] Fondazione Bruno Kessler –IRST, Trento - Povo, I-38050, Italy
{perini,sabatucci,susi}@fbk.eu

Abstract. This paper proposes the use of the HCI paradigm and techniques to support software system designers in building and browsing visual models during the development of complex distributed systems. In particular, we adopt Usability Evaluation Methods (UEMs) to analyse the first version of the interface of TAOM4E, the tool supporting the Tropos Agent-Oriented methodology. Using the results of this usability study, we collect different requirements to design an Advanced Visual Interface (AVI) of TAOM4E taking into account requirements of supporting software designers during Tropos models design process browsing.

1 Introduction

Visual modeling is a core activity in the so called model-driven approaches to software development. Tools played a key role in the diffusion of these practices in industrial settings (an example is the use of Object-Oriented modeling tools such as the IBM Rational Rose). In the latest years, Agent-Oriented (AO) modeling is becoming a reference paradigm for the design of complex distributed software systems. Several methodologies have been proposed so far, but only a few are supported by appropriate modeling tools. Current tools have been built giving considerable attention to practical issues, such as model interoperability, while their user interfaces seem to have been designed neglecting relevant principles of HCI. In our opinion, exploiting HCI principles and techniques will allow to build user interfaces which better fit the needs of the different users of these methodologies, resulting in a considerable contribution towards the diffusion of AO methodologies.

In this work, we focus on TAOM4E (http://sra.itc.it/tools/taom4e), the tool supporting the *Tropos* AO methodology, briefly described in Section 2. In order to build a user interface supporting users in *Tropos* models design process and browsing, the current GUI of TAOM4E has been evaluated. In order to do that, we used the usability evaluation methods (UEMs), which belong to the HCI research field; the different UEMs existing are described in Section 3. The TAOM4E GUI results evaluation, carried out using the Cognitive Walkthrough method [6], confirm that, mapping the actions prescribed by the Tropos methodology process to the TAOM4E AVI, can improve the users' interaction. The design guidelines of TAOM4E AVI, carried out taking into account requirements of supporting software designers, are described in Section 4. Conclusions and future works are described in Section 5.

M.J. Smith and G. Salvendy (Eds.): Human Interface, Part II, HCII 2009, LNCS 5618, pp. 269–276, 2009.

2 Tropos and TAOM4E

The *Tropos* AO methodology [8] adopts a model-driven software development process that guides users in building an initial model of domain stakeholders, with their own goals and social dependencies for goal achievement. This model is then incrementally refined and extended into intermediate design models.

The methodology provides a visual modeling language that offers primitive concepts such as actor, goal, plan, and resource, together with relationships, such as strategic dependencies between actors and goal and/or decompositions. Actor diagrams and goal diagrams give a view of the actors' strategic dependencies and of the way an actor's goals have been decomposed into subgoals and implemented via plans, respectively.

The modeling process has been defined in terms of the nondeterministic concurrent algorithm (*Generate Tropos Model*) recalled in Figure 1 [8]. This process guides a user in building *Tropos* models. During *Phase 1* - the initialization step - a set of domain *actors* and their associated *goals*, *plans* and *resources* are incrementally added to the model. During *Phase 2* - i.e. the modeling step - the procedure goal modeling is executed for each goal in the model in order to analyse it. The process goes on till all relevant entities have been elicited from the domain.

In the *goal modeling* procedure described in Figure 2, three possible decisions can be taken by users with respect to a given goal. The *delegate* choice can be performed in two ways, that is the goal can be delegated to an existing or to a new actor (adding a new *dependency* relationships). In the latter case, a new actor will be added to the model before delegating it the goal. For the choice *expand*, the goal is and/or decomposed into subgoals (adding a new decomposition relationship); the new subgoals are added to a list of goals to be processed. Finally, the goal can be solved (the step *solve*) associating it a plan p (and eventually a set of resources r_i) in the model via a means ends relationships that is added to the model.

In order to support the specific analysis techniques adopted in *Tropos* we developed a modelling environment, called TAOM4E (Tool for Agent-Oriented Modelling for Eclipse), which is based on an implementation of the metamodel described in [9].

Generate_Tropos_Model(*model*)
While other actors, goals, plans and resources are found in the domain
  Phase 1: initialization
  Collect the set of:
    domain actors A *model.add(A)*;
    domain goals G *model.add(G)*;
    domain plans P *model.add(P)*;
    domain resources R *model.add(R)*;
  For each goal g
      assign g, a;
  Phase 2: modeling
  Concurrently For each actor a
    Concurrently For each goal g
      goal_modeling(g);

Fig. 1. The modelling procedure of the Tropos methodology

Procedure $goal\_modeling(g)$
$decision = decision\_on\_goal(g)$;
case decision:
- $delegate$ g to existing actors or new actors $\{a_i\}$,
  $model.add(\{a_i\})$,
  $model.add(dependency(a_{depender}, g, a_{dependee}))$;
- $expand$ g in subgoals $\{g_i\}$,
  $model.add(\{g_i\})$,
  $model.add(decomposition_{and/or}(g, \{g_i\}))$;
- $solve$ g associating it a plan p and eventually some resources $r_1, ..., r_n$
  $model.add(means\_ends(g, < p, r_1, ..., r_n >))$

Fig. 2. The goal modelling procedure of the Tropos methodology

Among the main requirements we considered in developing TAOM4E tool are the following:

- *Visual Modelling*. As pointed out before, the visual aspect of the language is one of the fundamental characteristics of the *Tropos* methodology, so the modelling environment should support the user during the specification of an AO visual model (e.g., according to the *Tropos* visual notation). Moreover, the environment should allow us to represent new entities that will be included in the *Tropos* metamodel, language variants, as well as to restrict its use to a subset of entities of the modelling language.
- *Specification of model entities properties*. The modelling environment should allow us to easily annotate the visual model with model properties like invariants, creation or fulfillment conditions that are typically used in Formal *Tropos* specification.
- *Extensibility*. The modelling environment should be extensible and allow for different configurations by easily integrating other tools at will.

An effective solution to the requirement of a flexible architecture and to the component integration issue is offered by the Eclipse Platform.

New tools are integrated into the platform through plug-ins that provide the environment with new functionalities. A plug-in is the smallest unit of function in Eclipse and the Eclipse Platform itself is organized as a set of subsystems, implemented in one or more plug-ins, built on the top of a small runtime engine. The TAOM4E architecture is depicted in Figure 3. It follows the Model View Controller

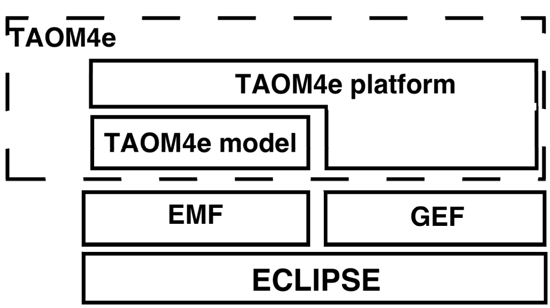

Fig. 3. The architecture of TAOM4E

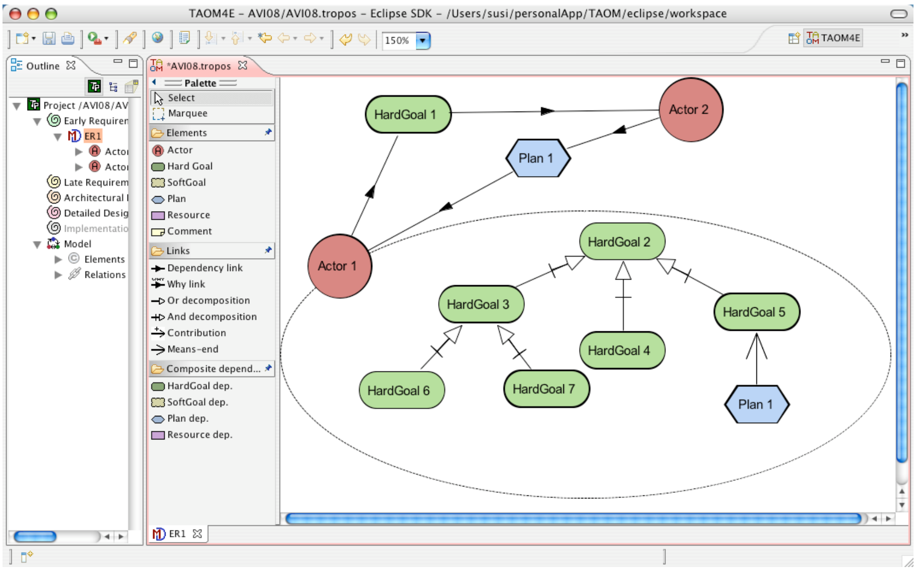

Fig. 4. Snapshot of the current TAOM4E GUI

pattern and has been devised as an extension of two existing plug-ins. First, the EMF plug-in (http://www.eclipse.org/emf/) offers a modelling framework and code generation facilities for building tools and other applications based on a structured data model. Given an XMI model specification, EMF provides functions and runtime support to produce a set of Java classes for the model. Most importantly, EMF provides the foundation for interoperability with other EMF-based tools and applications. The resulting plug-in, called TAOM4E model implements the *Tropos* metamodel. It represents the Model component of the MVC architecture. Second, the Graphical Editing Framework (GEF) plug-in (http://www.eclipse.org/gef/) allows developers to create a rich graphical editor around an existing metamodel. The functionality of the GEF plug-in helps to cover the essential requirement of the tool, that is supporting a visual development of *Tropos* models by providing some standard functions like drag & drop, undo-redo, copy & paste and others. The resulting plug-in, called TAOM4E platform represents both the Controller and the Viewer components of the tool. A snapshot of the tool's GUI is depicted in Figure 4. It consists of three main windows:

- the project/model browser on the left;
- the diagram editing window in the center;
- the *Tropos* entities/relationships palette at the left of the diagram editing window.

In particular, the project model window contains the set of model entities and artifacts of the *Tropos* project; the diagram editor allows the user to specify the *Tropos* artifacts, in particular actor and goal diagrams, using the *Tropos* concepts (actors, goals, tasks, resources and the relationships between those concepts) contained in the palette area. A simple example of such a *Tropos* diagram is shown in Figure 4. In this diagram the actors are represented via circles, goals via elliptical shapes and tasks via hexagons. Moreover, in the figure is shown the internal goal diagram of the *Actor 1* contained inside the dashed line ellipse (named balloon in *Tropos*). Focusing on the relationships specified in the diagram, two dependencies are shown between the two actors (*Actor 1* and *Actor 2*): a *goal dependency* whose label is the *HardGoal 1*, and a *task dependency* whose label is the *Plan 1*. Inside the balloon the and decompositions of the root goal *HardGoal 2* and of the subgoal *HardGoal 3* are described together with a *means_ends* relationship between the *Plan 1* and the *HardGoal 5*.

Currently, the model building process, executed by users, is not visually supported by the TAOM4E GUI. In fact, the user can add to the model entities without particular constraints or guidelines, so having also the possibility to specify *Tropos* models in a way that is not consistent with the process envisaged before. The designer can start building a new *Tropos* model directly creating the project in the project window and specifying the kind of diagram to start with (actor or goal diagram); at this stage the designer can start specifying model entities having also the possibility to specify goals or tasks without having specified the actors those goals/tasks are associated to; this way the user building task could fail in following the sequence of actions defined by the process, breaking the rational underlying the *Tropos* methodology.

We aim at designing an AVI that improves the user support in building and browsing tasks. In the next section we delineate some HCI guidelines to be followed in the new GUI supporting the design process.

3 Usability Evaluation Methods

The introduction of usability evaluation methods (UEMs) to assess and improve usability in which systems has led to a variety of alternative approaches and a general lack of understanding of the capabilities and limitations of each approach. In [4], authors present a practical discussion of factors, comparison criteria, and UEM performance measures that are interesting and useful while comparing different UEMs.

A common difference among UEMs is based on the skills of the evaluators (in general, a person using a UEM to evaluate usability of an interaction design); in particular two main criteria exists, Expert-based criteria and User-based criteria. In the former, experts are requested to evaluate a prototype, comparing it with respect to existing rules and guidelines; in the latter evaluators assess usability through real users, having them using a prototype. In particular while the Expert-based Criteria UEMs include, among others, Heuristic Evaluation method [5], Cognitive Walkthrough method [3], and Expert-based method [6], the User-based Criteria UEMs [2] include, among others, Observational evaluation method [10], Survey evaluation method [10], and Controlled experiment method [3]. Among observational evaluation methods, we focus on Verbal Protocols, and Think Aloud Protocol [3]. In what follows, we briefly describe these methods:

- *Heuristic Evaluation*: This criterion foresees a small set of evaluators that evaluate the interface and judge its compliance with well known usability principles (e.g., the heuristics).
- *Cognitive Walkthrough*: This is typically performed by the interface designer and a group of his or her peers. Small-scale walkthroughs of parts of an interface can

also be done by individual designers as they consider alternative designs. In a group situation, one of the evaluators usually takes on the duties of *scribe*, recording the results of the evaluation as it proceeds, and another group member acts as facilitators, to keep the evaluation moving.

- *Expert evaluation*: This method exploits the knowledge of an HCI expert, so obtaining a prediction about the usability of the system. This method has a main disadvantage: it depends upon the quality of the expert(s).
- *Observational evaluation method* involves real users that are observed when performing tasks with the system (depending on the stage of the project, what the system is ranges from paper mock-ups to the real product). This method offers a broad evaluation of usability. Depending on the specific situation, we may either apply the Observational Evaluation by direct observation or record the interaction between the users and the system (using Usability Lab). Recording (done by video camera) is more valuable, since it allows for storing information, for example the critical points during the interaction (e.g., when the user has to consult the manual, when and where s/he is blocked), the time a user spends to perform a task, the mistakes a user makes, and so on. Obviously, recording, using camera, is very expensive (especially for the time required to analyze the recorded data).
  - *Think Aloud Protocol* provides the evaluator with information about cognitions and emotions of a user while the user performs a task or solves a problem. The user is instructed to articulate what s/he thinks and feels while working with a prototype. The utterances are recorded either using paper and pencil or using audio and/or video recording. By using the Think Aloud Protocol, the evaluator obtains information about the whole user interface. This protocol is oriented towards the investigation of the users problems and decisions while working with the system.
  - *Verbal protocols* aim at eliciting the users (subjective) opinions. Examples are interviews and questionnaires. The difference between oral interview techniques and questionnaire techniques lies mainly in the effort for setup, evaluating the data, and the standardization of the procedure.
- *Survey evaluation method*. In this case, structured questionnaires and/or interviews are used to get feedback from the users. This method offers a broad evaluation of usability since from the users viewpoint it is possible to identify the critical aspects in user-system.
- *Controlled experiment method*. This method is particularly valid to test how a change in the design project could affect the overall usability. It may be applied in any phase during the development of a system; it provides more advantages when it is possible to test separately the alternative designs, independently by the whole system. This method mainly aims at checking some specific cause-effect relations, and this is achieved checking as many variables as possible.

Since the TAOM4E tool design doesn't follow a particular HCI design methodology (as for example the User Centered Design Methodology [7], we use, to evaluate TAOM4E GUI, a method belonging to the Expert-based criteria: the Cognitive Walkthrough [3].

4 TAOM4E AVI Design

As mentioned the TAOM4E GUI has been analyzed from the usability point of view submitting it to an Expert-based evaluation [6], using the Cognitive Walkthrough [3]. The followed approach in the evaluation has been considering the main tasks of TAOM4E: building and browsing *Tropos* models. Consequently, the several remarks about visual issues are described considering building task and browsing task obtained, with respect to the possibility of improving user support in these tasks. In particular, mapping the phases and actions proposed by the process algorithm to structural elements of TAOM4E AVI we derived the different requirements. These requirements have to be applied to:

- the palette structure;
- the palette behavior;
- the diagrams dynamic layout.

Building task. This task is the most important task of TAOM4E users. It consists in choosing *Tropos* model elements and managing them in order to create a consistent model, following the modelling process described in the Figures 1 and 2. In order to improve the user support during the model building task, the AVI should accomplish the following requirements:

- The palette structure has been divided into three areas. At the top we grouped general actions on the model diagram: simple objects selection; objects group selection; and *Tropos* balloon opening. In the middle the set of actor internal elements (goal, task, resource) and relationships are presented. At the bottom are positioned the set of social relationships, namely goal-, task-, resource- dependency.
- The palette behavior reflects the *Tropos* process algorithm steps. The system enables and disables object choices following the building process. When creating a new model all the palette icons are disabled except the actor icon. Once an actor is dragged & dropped from the palette in the diagram (so that an actor is added to the model), the icons related to internal elements, in the middle area of the palette, are enabled, together with the icons corresponding to the general actions. Internal and social relationships are not created in this step, so, their icons are maintained disabled. Once a second actor has been specified, the set of social relationships are enabled to allow users to start modeling actors' social relationships (see the delegate action in the algorithm).
- For each actor added to the model, users can choose to open the balloon, dragging it from the general actions palette to a specific actor in the diagram, in order to start doing actor internal goal modeling (the expand action in the algorithm).

Browsing task. The TAOM4E AVI should support users in diagram browsing by enriching, through visualization techniques, predefined views. In order to improve the user support in this task, we aim at introducing the layers paradigm to visualize selected objects, using a tool visualization palette containing a list of combo boxes associated to modeling objects; in this way, when users select one or more combo boxes, the system shows/hides the associated objects in the diagram. In order to better fit user needs in diagram visualization, the tool should propose different shape layouts (e.g. star, circle, rectangles) upon user request. Moreover, we rely on a Focus+Context technique [1] to support diagram browsing: when the user selects a particular actor, the diagram's layout changes: the actor is positioned in the center of the diagram (focus) while its proximity and the context appear in the background. It is worth noting that this operation is not an optimization since we are only temporary modifying the model view.

5 Conclusions

We used the HCI paradigm and techniques to evaluate the current TAOM4E GUI and to derive the new TAOM4E AVI design. This work has been motivated by requests and observations from users, that downloaded TAOM4E and exploited it as supporting tool in a novel software design approach. The evaluation of the current interface is carried out using the Cognitive Walkthrough method: TAOM4E AVI maps the phases and actions proposed by the *Tropos* process algorithm to palette structure, palette behavior, and diagram dynamic layout, integrating Focus+Context technique to support diagram browsing. Further investigation will be devoted to: (1) evaluate TAOM4E AVI through user-based methodologies; (2) support the definition of different user profiles based on different skill levels in specifying *Tropos* models.

References

1. Card, S.K., Mackinlay, J.D., Shneiderman, B.: Information Visualization: Using Vision to Think. M. Kaufmann, San Francisco (1999)
2. UEMs Consortium. Heuristic evaluation (2006)
3. UEMs Consortium. Performing a cognitive walkthrough (2006), `http://www.cc.gatech.edu/computing/classes/cs3302/documents/cog.walk.html`
4. Hartson, H., Andre, T., Williges, R.: Criteria for evaluating usability evaluation methods. International Journal of Human Computer Interaction 13(4), 373–410 (2001)
5. Hartson, H.R., Andre, T.S., Williges, R.C.: Criteria for evaluating usability evaluation methods. International Journal of Human Computer Interaction 13(4), 373–410 (2001)
6. Hix, D., Hartson, H.R.: Formative evaluation: Ensuring usability in user interfaces. Trends in Software 1, 1–30 (2000)
7. Norman, D., Draper, S.: User Centered System Design. LEA Hillsdale, N.J (1986)
8. Penserini, L., Perini, A., Susi, A., Mylopoulos, J.: High Variability Design for Software Agents: Extending Tropos. ACM TAAS 2(4) (2007)
9. Susi, A., Perini, A., Giorgini, P., Mylopoulos, J.: The Tropos Metamodel and its Use. Informatica 29(4), 401–408 (2005)
10. Yagita, Y., Aikawa, Y., Inaba, A.: A proposal of the quantitative evaluation method for social acceptability of products and services. Technical report, Ochan-omizu University, Otsuka Bunkyo-ku, Tokyo (2001)