Using the Tropos Approach to Inform the UML Design: An Experiment Report

Andrea Capiluppi Cornelia Boldyreff
School of Computing, Information Technology and Engineering
University of East London
London, United Kingdom
{*a.capiluppi,c.boldyreff*}*@uel.ac.uk*

***Abstract*—Tropos is an agent-oriented software engineering (AOSE) methodology, based on the notion of actors, with goals and plans, and spanning all the phases of software development, from the very early phases of requirements analysis down to the actual implementation. The effectiveness of such methodology in the production of better design documents is evaluated in this study, by investigating the null hypothesis "using the Tropos Methodology before the analysis and design phases can produce a more accurate and complete set of UML diagrams than when no such technology is used".**

The evaluation of a real case scenario was given as part of a coursework in a BSc module at the University of East London, and the Tropos and UML diagrams were requested as part of the deliverables. The results of how students performed in such tasks, and how the Tropos approach helped in the drawing of the UML diagrams, are presented here.

The results show that generally, and confined to this experiment, the Tropos methodology has not helped in the design of the UML diagrams, and that students failed in understanding the link between the two methodologies.

***Keywords*-Software Quality; UML; Tropos Methodology**

I. Introduction

Among the core skills employed during the phase of requirements gathering and elicitation, is that of being able to identify and model the basic concepts of the application domain upon which the software system will be built. Such activity has been named *conceptual modelling*, and it serves the purpose of glueing together the requests by the customers, and the expertise of designers and developers, providing a platform to ease communication, meet users expectations and distribute knowledge [1]. Two techniques have recently been considered and compared for the modeling of such concepts, one based on scenarios of *how* the system is going to behave (or how the users will interact with it (e.g., the UML approach [2]); the other expressing *what* are the needs that the built system will fulfill, relating the business goals of the stakeholders with the functional and non-functional requirements of such system. The latter has been termed *goal-based* approach, and the agent-oriented software engineering (AOSE) methods have been one the more developed branches of such approach in the requirements elicitation.

Among the goal-based approaches, Tropos is an AOSE methodology based on two key ideas: agents and their interactions within the system environment. The main aim of Tropos is to produce a better understanding of the application domain where a system will operate, and of the kind of interactions that should occur between such system and the human agents. Within Tropos, the notion of agent, together with their goals and plans, are used since the early analysis of requirements elicitation: in the early phase of such analysis, the organizational setting is studied for the purpose of better understanding the scenario. In the late phase of requirements gathering and elicitation, the system is also inserted in the operational environment as one actor: the dependencies with the other actors represent the system's functional and non-functional requirements.

In both phases, the *actor* and *goal* diagrams are produced as outcomes, with the system being inserted in the diagrams in the late phase, but not in the early phase. The actor diagram represents the overall view of all the actors with their high-level dependencies to other actors, while the goal diagram is a refinement of the former with emphasis on the goals of a single actor (see Figure 1).

The focus of this work is on the early and late phases of requirements elicitation covered by the Tropos methodology, where the business entities are identified as actors, their goals assessed, and their inter-dependencies defined. In the UML notation instead, as summarised in Table I, these two phases correspond to the production of a *model in the problem space* (MOPS [3]). Such a model comprises of a set of use cases and business class diagrams (i.e., diagrams documenting business entities, their attributes and behaviors). When the business entities are converted into implementable entities, the UML notation produces the Model of Solution Space (MOSS) with the aim of feeding such model to the design phase.

The aim of this paper is to compare the UML outcomes from the MOPS phase (use cases and business class diagrams) as produced by undergraduate and postgraduate students, when combining (or not) the Tropos methodology as a "treatment". The rationale of such experiments is to determine through evaluation whether the joint use of

	Early Requirements (ER)	**Late Requirements (LR)**
Tropos	ER actors and goals	LR actors and goals (with system)
UML	MOPS	MOPS + MOSS

Table I
TROPOS AND UML DELIVERABLES IN THE EARLY AND LATE PHASES OF REQUIREMENTS GATHERING

goal-based (Tropos) and scenario-based (UML) approaches should be preferred to the use of only a scenario-based approach in the production of quality UML diagrams.

This paper details one experiment where BSc students at the university of East London, UK, produced both Tropos and UML diagrams towards the assessment of a scenario where a software system has to be built. The UML and Tropos diagrams were assessed against the benchmark produced as a marking scheme, and it is questioned whether the presence of the Tropos methodology has helped in the completeness of the resulting UML diagrams. This paper is the first of two experiments, where the Tropos methodology is used to inform the UML design: we plan to replicate this experiment in the semester starting in February 2011, without the Tropos "treatment": students will be required to work on the same scenario, but no Tropos diagrams will be required (or taught), therefore allowing for the comparison of two different sets of UML diagrams. This will provide the basis for comparing the effectiveness (or not) of the two combined approaches.

II. BACKGROUND AND RELATED WORK

This paper builds upon the scenario-based and the goal-based approaches as two viable tools in the requirements elicitation phase and for validation purposes. As a practical exemplification of the scenario-based requirement engineering method, we have used the Jacobson's Use Case technique, which has been lately incorporated into the UML notation language [2]. Such a model is based on the notion of "scenario" which is a *sequence of interaction events between a system and its environment in the restricted context of achieving some implicit purposes* [4], [5].

On the other hand, this paper relies on the concepts of agents and the agent-oriented paradigm (AOSE), as one example of goal-oriented approach [6], [7], [8]. This second approach is based on agents interacting as a group within a system, not just reacting to stimuli, but also communicating, coordinating, and cooperating as an autonomous and social entity that can to achieve individual and organizational goals.

The main notations of UML (as a scenario-based methodology) and Tropos (as goal-based) are summarised in Figure 1 (taken from [5]). Specifically for the Tropos notation, every system can be thought of several actors, having goals to fulfill with the use of such system. Such goals could be "hard" or "soft", depending on whether it is clear what actions and plans (or resources) should be performed (or used) in order to achieve such goals. A Tropos "actor diagram" details the overall connections between all the actors in the scenario, where a dependee (e.g. actor_3 in Figure 1) fulfills the goal(s) of a depender (e.g., actor_1 in Figure 1). A Tropos "goal diagram" focuses more precisely on one actor, and tries and elaborates on what plans, actions and resources should be performed to achieve each goal, and which actors are needed to fulfill these goals.

In the literature, the effectiveness of goal-oriented and scenario-based approaches is analyzed in several works illustrating the application of different methods to case studies (e.g., [9], [10], [11] or comparing the strengths and limitations of the approaches according to different criteria (e.g., [12], [13]). However, to the best of our knowledge, experimental comparisons of these requirements modelling paradigms using different visualization methods are rare [5]. Such comparisons may raise insights and help decide which modelling paradigm to adopt for a given software development project. The "quality" of UML models, comprised guidelines for the aesthetic quality, have also been evaluated [14].

One important factor for comparison or evaluation is the immediacy in understanding the respective models by projects stakeholders, for instance by requirements analysts [15], who have to understand a given model during analysis and refinement tasks to accommodate new or changed requirements.

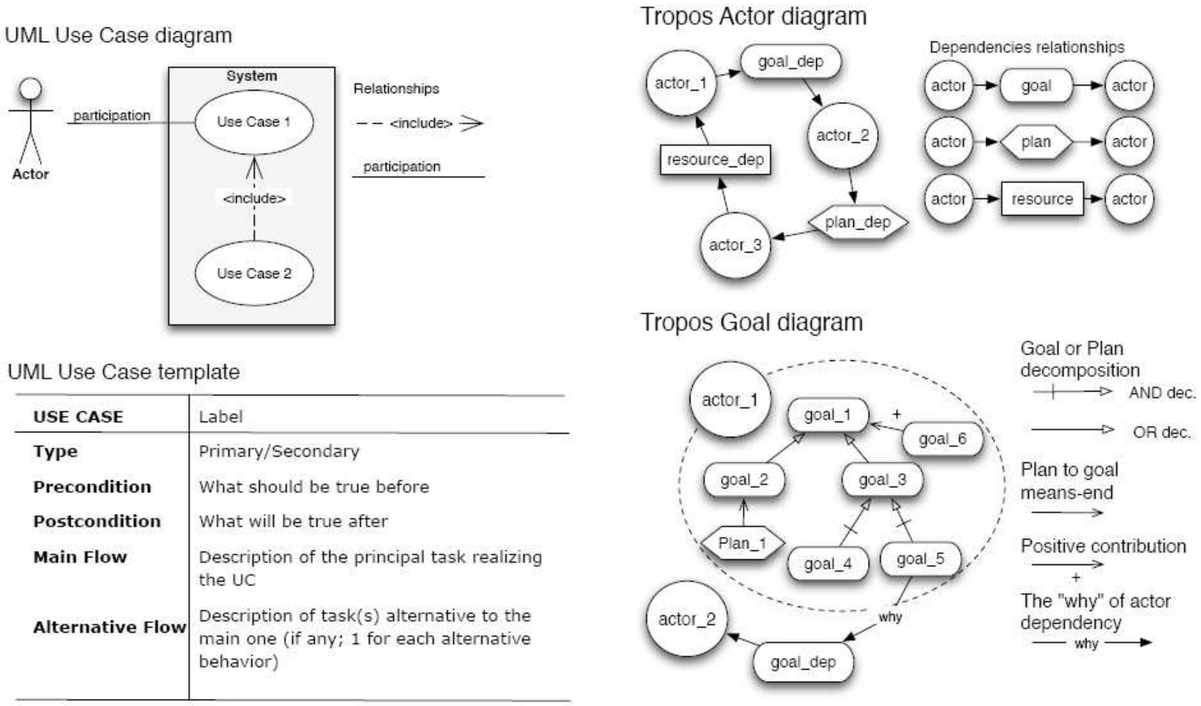
III. EMPIRICAL APPROACH

This section introduces the definitions used in the following empirical study and presents the general objective of this work, and it does that in the formal way proposed by the *Goal-Question-Metric* (GQM) framework [16]. The GQM approach evaluates whether a goal has been reached, by associating that goal with questions that explain it from an operational point of view, and providing the basis for applying metrics to answer these questions. This study follows this approach by developing, from the wider goal of this research, the necessary questions to address the goal and then determining the metrics necessary for answering the questions.

Goal: The long term goal of this research is to evaluate whether the Tropos methodology (as an experimental "treatment"), jointly with the UML MOSS notation, produce higher standards of conceptual modelling than the UML notation alone.

Question: In this paper, and considering a given scenario as a case study, the following research questions will be evaluated:

1) Are the models produced by the students with the Tropos notation "complete" against a given benchmark?



UML Use Case template

USE CASE	Label
Type	Primary/Secondary
Precondition	What should be true before
Postcondition	What will be true after
Main Flow	Description of the principal task realizing the UC
Alternative Flow	Description of task(s) alternative to the main one (if any; 1 for each alternative behavior)

Figure 1. Main UML and Tropos concepts and notations (from [5])

Rationale: the aim of this question is to check whether the diagrams produced with the Tropos notation are compliant with a minimum list of actors and goals directly derived from the scenario. Such list of actors and goals should be considered as the "absolutely mandatory" in a typical requirements elicitation and gathering phase.

2) Are the models produced with UML complete against a given benchmark?
Rationale: similarly to the question above, the aim of this question is to check whether the diagrams produced with a UML editor (Rational Rose, ArgoUML, etc) can be mapped to a minimum list of use case diagrams, necessary to describe the how the users of the system interact with its functionalities.
3) Can students map the Tropos actors and goals to UML use cases?
Rationale: the aim of this question is to evaluate whether the use of "goals" and "actors" can help in focusing on the main functionalities of the system, expressed as UML use cases. Given the set of Tropos diagrams produced by any group of students, and a benchmark mapping of "Goals-to-use-cases" (see last column of Table III), it will be evaluated how the Tropos diagrams have informed the specified group of students in the creation of use cases.

Metrics: The Tropos actor and goal diagrams for this scenario have been listed in their minimum form, i.e., the minimum number of functionalities that are expected for (and from) this system, corresponding to both functional and non-functional requirements (see Table II). Also, the minimum set of UML use cases has been developed and it served as a benchmark to evaluate how students assessed the scenario (see Table III). Each group coursework was evaluated against these two lists, and the number of correct diagrams produced by the students evaluated against these baselines.

IV. Experimental Design

The first part of the experiment was set up at the University of East London, during the Level 3 module "Advanced Information Systems Development". The experiment population comprised some 65 students, divided in 17 groups of 3 to 4 members[1]. Each group was in charge of producing two sets of diagrams: the Tropos goal and actor diagrams (for both the early and late phase of the requirements); and the UML use cases and class diagrams. All the students in the module had already studied the basic UML concepts in a previous module, while the Tropos concepts were introduced during several lectures, and their practical implementation was assisted in the lab sessions. The scenario was distributed to students on week 4 (out of 12 weeks in the module), and it represents the coursework needed to pass the module,

[1] Since the selection of students and groups was not random, the study should be referred to as a *quasi-experiment*. We will use the term "experiment" as a synonym throughout the study

together with the final exam. The students had 9 weeks to complete the task.

In order to produce the Tropos diagrams, the OME tool, implementing the i* notation[2], was taught and demonstrated during the lab sessions. In order to produce the UML class and use case diagrams, students could select the UML editor of their choice (e.g., the IBM Rational Rose toolkit, or the Open Source ArgoUML tool[3], etc.).

A. Scenario

The following problem statement was provided to the students, with the request for modeling such scenario via a scenario-based approach and a goal-based approach. This is based on a previous job placement where a student effectively designed and developed the system outlined below.

> *A company has supplied and supported its clients in the area of Tax and Returns Automation for more than 10 years. This involves an employee going to the client sites and inspecting the revenues that each of the client companies claim in a given year and giving advices and filling the necessary forms for Tax Return purposes. Once the employee has filled the relevant forms (on a per-client basis), these forms need to be saved to a couple of paper copies, one to be kept by the client, one to be archived within the company.*
>
> *The company is seeking to streamline and automatise its systems for record keeping, therefore enabling the business to offer their clients a better service. The aim of this project would be to develop a system allowing data collection during site visits to be entered onto an online application, that sits on the web: the employee visiting the site's premises would input the data to a specified form (which can be extended by a System Administrator to contain more fields and input data, it could be reused from existing form, and new forms can be created ad-hoc). The data once collected would be synchronized with the companys database, but during the initial trial period, the paper-based system, and the on-line system, would need to run together, and be synchronised.*
>
> *The data collected would be used to keep the clients informed of the results of the employee's visits and the next visit's date. This upgrade project would be expected to cover the following areas: data acquisition using online, secure systems; synchronizing of data; a database to store the data of clients; and a PC based management tool for the data-captured database.*

B. Expected Outcomes – Tropos Marking Scheme

In order to assess the courseworks produced by the students, a list of "model solutions" was produced, and checked against the delivered set of diagrams. In particular, a minimum list of the Tropos actors present in the scenario was produced and their main goals were identified: the following Table II was therefore used as the baseline for marking the assignment. These goals and actors were prepared by one of the authors (running the module) and the assistant, a PhD student whose focus is on the secure aspects of Tropos.

Three main actors (Client, Company and Employee) were identified as expressing goals within the interaction with the system, while other two (the System, and the HM Revenue and Customs agency – HMRC) are also present, acting as dependees in one or more of those goals by the three main actors.

The marks available for the completion of such task were 25 out of 50.

Goal-based approach – TROPOS			
Actor	Goals – (H)ard or (S)oft		Dependee
Company	GCo1	Schedule periodic meetings (H)	Client
	GCo2	Get data to fill forms (H)	Client
	GCo3	Get up-to-date Returns rules (H)	HMRC
	GCo4	Secure data based on client or employee (H)	System
	GCo5	Provide a better service to clients (S)	Self
	GCo6	Rationalise forms (S)	Self
Employee	GE1	Get training on up-to-date procedures (H)	Company
	GE2	Get online access during visits (H)	Client
	GE3	Access clients details on system (H)	System
	GE4	Log activity or duration (H)	System
Client	GCl1	Obtain copies of job performed (H)	Employee
	GCl2	Get Tax Return advices (H)	Employee
	GCl3	Browse activity logs (H)	System
	GCl4	Get secure service (S)	Company
System			
HMRC			

Table II
MARKING SCHEME – TROPOS ACTORS AND GOALS

C. Expected Outcomes – UML Marking Scheme

The following Table III summarises instead the list of UML use cases that were set up as a baseline for marking the scenario-based part of the assignments: three main UML actors were expected to be interacting with the system, with increasing amount and type of provileges: the clients of

[2] *OME3*, available online at http://www.cs.toronto.edu/km/ome/
[3] ArgoUML, freely available at http://argoUML.tigris.org/

the Tax Revenue company (c_i in Table III, $i = 1..5$), its employees (e_i in the same Table, with $i = 1..7$) and the system administrator (s_i in the same Table, with $i = 1..6$).

The UML use cases listed, and intended as a "model solution", are a subset of what was documented during a business consultancy,where the described system was actually implemented by a student in a job placement. The listed UML use cases should be inferred by reading the problem statement of the scenario, and they should also become clearer after working on the Tropos goals and actors. Albeit more specified UML actors could be identified (e.g., the ISP administrator, the project manager in charge of delivering the requested system, the Tax Revenue company owner, etc.), the above three provide the minimum set of scenarios that fulfill most of the functional and non-functional requirements of the scenario. In some of these, one UML use case could be the extension, or being included in some other use case (for instance, the "log-in" use case is typically included in any interaction with the system, independently from the privileges).

The marks available for the completion of the UML task were also 25 out of 50. This was decided to balance the relative importance of both Tropos and UML tasks.

D. *Results*

As said above, the results obtained from the marking of the presented coursework represent the first part of this study: the second part will be based on assessing the diagrams produced by the students when the "treatment" Tropos is not taught or requested.

Table IV shows how each group coursework (G1 to G17) was evaluated against the list of Tropos goals and UML use cases, gathered around the main actors expressing their requirements, either in a goal-based approach, or a scenario-based approach.

At a first glance, the results found in the table show that the students found easier to assess the Tropos actors and goals, rather than producing the relative UML diagrams to describe how the actors are interacting with the system. Even when breaking down the aggregated results in the main components and actors, it is visible that some actors were assessed better than others: the Tropos models for the Company providing the Tax Revenue service are more complete than other actors (as visible in Table V where on average 70% of the groups assessed the benchmark goals from the Company actor).

The striking difference with such a finding is visible by observing the results of the UML cases, summarised in Table VI, where on average, only 38% of the groups assessed the "Employee" use cases, and only 43% delivered the "administrator" cases. As a grand average, some 64% of groups successfully assessed the set of Tropos goals proposed as a baseline, while only 38% of students assessed the set of UML cases of the benchmark.

Tropos Goal	Groups delivering	Perc	Average
GCo1	12	70.59%	70.59%
GCo2	14	82.35%	
GCo3	8	47.06%	
GCo4	11	64.71%	
GCo5	17	100.00%	
GCo6	10	58.82%	
GE1	7	41.18%	58.82%
GE2	15	88.24%	
GE3	13	76.47%	
GE4	5	29.41%	
GCl1	12	70.59%	58.82%
GCl2	14	82.35%	
GCl3	8	47.06%	
GCl4	6	35.29%	
Grand Average			63.87%

Table V
RESULTS – BY GROUP

UML use case	Groups delivering	Perc	Average
c1	9	52.94%	27.06%
c2	3	17.65%	
c3	4	23.53%	
c4	3	17.65%	
c5	4	23.53%	
e1	7	41.18%	38.66%
e2	11	64.71%	
e3	5	29.41%	
e4	15	88.24%	
e5	2	11.76%	
e6	2	11.76%	
e7	4	23.53%	
a1	12	70.59%	43.53%
a2	3	17.65%	
a3	13	76.47%	
a4	5	29.41%	
a5	1	5.88%	
a6	15	88.24%	
Grand Average			38.56%

Table VI
RESULTS – BY GROUP

These discrepancies are also visible when considering single students groups:

- among the Tropos goals, 4 goals out of 6 were on average correctly identified, with regards to the Company goals (average 4.23 goals); among the Employee goals, 2 goals out of 4 were on average assessed (average 2.35 goals); finally, among the Client goals, 2 out of 4 goals were identified (average 2.35 goals);
- with respect to the UML cases, 1 out of 5 cases were identified for the client (average 1.35 cases); 2 out of 7

Scenario-based approach – UML			
UML actor	UML use cases		Via Tropos goal(s)
Client	c1	Can log-in	GCl3, GCl4, GCo4, GCo5
	c2	Can update their details	GCl4
	c3	Can browse the log of activity	GCL3, GE4
	c4	Can browse relevant documentation	GCo3, GCl4
	c5	Has sole access to private area	GCl4
Employee	e1	Can schedule visit	GCo1
	e2	Can log-in	GE2
	e3	Can select appropriate forms based on client	GE1, GCo6
	e4	Can fill in forms	GCo2, GE4
	e5	Can fill in the log of activity	GE4, GCl3
	e6	Can upload relevant documentation	GCo3
	e7	Has privileged access to all clients private area	GCo4, GCo5, GE3
Administrator	a1	Can log-in	GCo5
	a2	Can create/update/remove employees	GCo4, GE1
	a3	Can create/update/remove forms	GCo6
	a4	Can create/update/remove clients	GCo4
	a5	Can monitor the activity of employees	GE1, GE4
	a6	Can synchronise the database	GE1, GE4, GCo6

Table III
MARKING SCHEME – LIST OF UML USE CASES

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
GCo1	√	√	√		√	√	√				√	√	√		√	√	√
GCo2	√	√	√	√	√	√	√	√	√		√	√		√		√	√
GCo3		√	√	√			√				√	√	√		√		
GCo4		√	√	√	√		√	√			√	√	√			√	√
GCo5	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
GCo6		√		√	√		√	√				√	√		√	√	√
GE1		√		√			√				√	√			√	√	
GE2	√		√	√	√	√	√	√	√		√	√	√	√	√	√	√
GE3	√	√		√	√	√	√	√	√		√		√		√	√	√
GE4		√				√	√	√								√	
GCl1		√	√	√	√	√	√				√	√	√		√	√	√
GCl2	√	√		√	√	√	√		√	√	√	√	√		√	√	√
GCl3	√		√		√		√	√			√					√	√
GCl4			√		√			√		√	√				√		
c1	√	√						√		√	√	√			√	√	√
c2								√			√				√		
c3	√				√											√	√
c4					√											√	√
c5		√								√		√				√	
e1		√		√	√	√				√		√	√				
e2	√	√	√		√		√		√	√	√	√			√	√	
e3	√		√							√			√	√			
e4	√	√	√	√	√	√	√	√	√	√	√	√	√	√		√	
e5									√	√							
e6		√							√								
e7	√							√		√	√						
a1	√	√	√	√		√	√				√	√		√	√	√	√
a2								√	√			√					
a3	√	√			√	√	√	√			√	√	√	√	√	√	√
a4		√		√			√		√			√					
a5		√															
a6	√	√	√	√	√	√	√	√	√		√	√		√	√	√	√

Table IV
RESULTS – BY GROUP

cases were identified for the employee (average 2.70 cases); and 2 out of 6 cases were assessed for the administrator of the system (average 2.88 cases)

Relatively to the experiment performed with the students of the University of East London, we can conclude that the use of the Tropos approach was not effective to inform the UML conceptual model.

V. Threats to Validity

Like any other empirical study, the validity of ours is subject to several threats. In the following, threats to *internal* (whether confounding factors can influence your findings), *external* (whether results can be generalized), and *construct validity* (relationship between theory and observation) are illustrated.

- **Internal validity** – the terminology "quality of UML models" was used to define whether "better" models could be obtained with the use of the additional Tropos analysis. Obviously the quality of UML diagrams is a multi-faceted dimension of several possible: aesthetic aspects could be considered, but also others based on design metrics of UML diagrams, as coupling, complexity, etc).
- **External validity** – the following threats to external validity have been identified:
  1) these findings cannot be generalised by one scenario, distributed to some 70 students, and based on one observation only. Replications are needed not only regarding the presence or absence of the Tropos "treatment", but also with more students involved.
  2) Despite the initial results, a stronger statistical formalism cannot be used for investigating the research questions: this is because since there is no comparison with a null hypothesis, such analysis cannot be properly performed. The results will become much more reliable when the second part of the experiment will be carried out.
- **Construct validity**: the minimum set of actor and goal diagrams, and the minimum set of UML use cases derived for the construction of the benchmark could play an important part in the outcomes of this experiment. The reason of choosing these use cases, and the relative Tropos actors and goals, are of a practical nature: the proposed one is a real scenario of a past job placement, where a student designed and implemented the system to be delivered: the "model answers" are a subset of the diagrams implemented for the deployment of such system.

VI. Acknowledgements

The authors would like to thank Dr H. Mouratidis and Michalis Pavlidis for the extensive comments, and the help in formulating the Tropos goals and actors. The authors would also like to thank the anonymous reviewers, since many improvements were added to the text, based on their suggestions.

VII. Conclusion and Further Work

The usage of visual modelling tools has become a common support for the design of a software system's capabilities; the use of such tools has become more valuable in the early phase of requirement gathering, where the interaction with non-technical stake-holders requires jargon-free and easily usable approaches. Among these techniques, this paper has considered the goal-based (Tropos) and the scenario-based (UML) methodologies, trying to assess whether the use of the first could be useful to inform the definition of more complete use cases.

A set of "model solutions" was prepared for a given scenario, that was handed out as part of a coursework at the University of East London, UK. A baseline set of actors was prepared for the Tropos approach, and one for the UML use cases. Each coursework was assessed against these two baselines. Contrarily to what was expected, a larger number of students correctly assessed a larger amount of Tropos goals, whereas the UML cases were delivered less often, and more erroneously. Although the correct UML cases were assessed where the relevant Tropos actors were identified, this was not always the case: students found it difficult to connect the two approaches, and synchronise the actors and goals with how the system was supposed to perform.

These results are interesting, but we need to produce a similar set of observations when removing the Tropos approach from the experiment: we plan to replicate this experiment in a course starting in February 2011, where the same scenario will be provided, and where only the UML use cases will be requested. This will help in assessing whether the use of the Tropos approach can be considered to play a difference in the requirements gathering phase, when coupled to the UML notation.

References

[1] A. van Lamsweerde, "Requirements engineering in the year 00: a research perspective," in *Proceedings of the 22^{nd} International Conference on Software engineering (ICSE 00)*. New York, NY, USA: ACM, 2000, pp. 5–19.

[2] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide, 2^{nd} Edition (Addison-Wesley Object Technology Series)*. Addison-Wesley Professional, 2005.

[3] B. Unhelkar, *Verification and Validation for Quality of UML 2.0 Models*. Wiley-Interscience, 2005.

[4] A. Sutcliffe, "Scenario-based requirements engineering," in *Proceedings of the 11^{th} IEEE International Conference on Requirements Engineering*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 320–.

[5] I. Hadar, T. Kuflik, A. Perini, I. Reinhartz-Berger, F. Ricca, and A. Susi, “An empirical study of requirements model understanding: Use Case vs. Tropos models,” in *Proceedings of the* 25^{th} *ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2010, pp. 2324–2329.

[6] J. Mylopoulos, “Information modeling in the time of the revolution,” *Information Systems*, vol. 23, pp. 127–155, May 1998.

[7] E. S. Yu, “Modelling strategic relationships for process reengineering,” Ph.D. dissertation, Toronto, Ont., Canada, Canada, 1996.

[8] A. Dardenne, A. van Lamsweerde, and S. Fickas, “Goal-directed requirements acquisition,” *Science of Computer Programming*, vol. 20, pp. 3–50, April 1993.

[9] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, “TROPOS: An agent-oriented software development methodology,” *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, May 2004.

[10] J. Castro, M. Kolp, and J. Mylopoulos, “Towards requirements-driven information systems engineering: the tropos project,” *Information Systems*, vol. 27, pp. 365–389, September 2002.

[11] M. Kim, S. Park, V. Sugumaran, and H. Yang, “Managing requirements conflicts in software product lines: A goal and scenario based approach,” *Data & Knowledge Engineering*, vol. 61, pp. 417–432, June 2007.

[12] J. L.M.C. Filho, V.Werneck and E. Yu, “Agentgoal orientation vs object orientation for requirements engineering: A practical evaluation using an exemplar,” in *Proceedings of the* 8^{th} *Workshop on Requirements Engineering*, 2005, pp. 123–134.

[13] C. Rolland, G. Grosz, and R. Kla, “Experience with goal-scenario coupling in requirements engineering,” in *Proceedings of the* 4^{th} *IEEE International Symposium on Requirements Engineering*. Washington, DC, USA: IEEE Computer Society, 1999, pp. 74–.

[14] H. Eichelberger and K. Schmid, “Guidelines on the aesthetic quality of UML class diagrams,” *Information and Software Technology*, vol. 51, no. 12, pp. 1686 – 1698, 2009.

[15] C. F. J. Lange, “Improving the quality of UML models in practice,” in *Proceedings of the* 28^{th} *International Conference on Software engineering*. New York, NY, USA: ACM, 2006, pp. 993–996.

[16] V. R. Basili, G. Caldiera, and D. H. Rombach, “The goal question metric approach,” in *Encyclopedia of Software Engineering*. John Wiley & Sons, 1994, pp. 528–532, see also http://sdqweb.ipd.uka.de/wiki/GQM.