JOURNAL OF OBJECT TECHNOLOGY

Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2005**

Vol. 4, No. 2, March - April 2005

An Etymological and Metamodel-Based Evaluation of the Terms “Goals and Tasks” in Agent-Oriented Methodologies

Brian Henderson-Sellers, Quynh-Nhu Numi Tran and John Debenham,
University of Technology, Sydney, Australia

Abstract

Agent-oriented methodologies frequently make use of terms such as goal and task but do so in an inconsistent manner. We seek to rationalize the use of these terms by undertaking an etymological and metamodel-based analysis of a significant number of these AO methodologies and recommend that the word task be avoided; instead, the word action could be usefully employed to describe the work done to achieve a goal or subgoal. We also note that the notion of subgoal is ambiguous in either being an interim goal along the path of achievement of the main (final or overall) goal or, alternatively, a portion/part of the goal whose achievement contributes (at the same instant in time) to the achievement of the overall goal. If we accept subgoal for the former meaning, then we suggest “goal part” for the latter.

1 INTRODUCTION

Agent-oriented (AO) methodologies frequently make use of terms such as goal and task but do so in an inconsistent manner. We seek to rationalize the use of these terms by undertaking an etymological and metamodel-based analysis of a significant number of these AO methodologies. In Section 2 we outline the background to agent architectures in the context of how they are described in various AO methodologies (Section 3). In particular we evaluate how these various AO methodologies use the terms “goal” and “task” – some confound them while others clearly differentiate them. Based on this analysis and the use of etymological and metamodel analysis, in Section 4 we make some recommendations that try to both align with existing usage but at the same time avoid terms that have caused the original confusion.

Cite this column as follows: Author’s name(s): “Paper title”, in *Journal of Object Technology*, vol. 4, no. 2, March-April 2005, pp. 131-150 http://www.jot.fm/issues/issue_2005_03/article3

2 BACKGROUND

While there are many individual models of agent architecture, there is a general agreement that agents are able to act without the intervention of humans or other systems [36] [27, p35]: they have control both over their own internal state and over their behaviour. This may be achieved by some mechanism that determines which goals they should commit to achieving and then which decisions need to be taken in order to reach those goals [35].

While there are several internal architectural models for agents, including proactive and reactive reasoning models, there are commonalities regarding the notions of agency, including the notion that the agent is situated in an environmental context. In particular, we focus here on the important concept that agents that exhibit proactive reasoning have one (or more) internal and committed goals (future desired state) that they seek to achieve. Such a current commitment is continually being revised, possibly leading over time to the decommitment of goals that the agent no longer wishes to achieve as well as the establishment of commitments to new goals. When an agent decommits to a goal, it may be necessary to initiate a sequence of actions that "tidy things up" and undo some of the things that were done in the partial, incomplete attempt to achieve that now-decommitted goal.

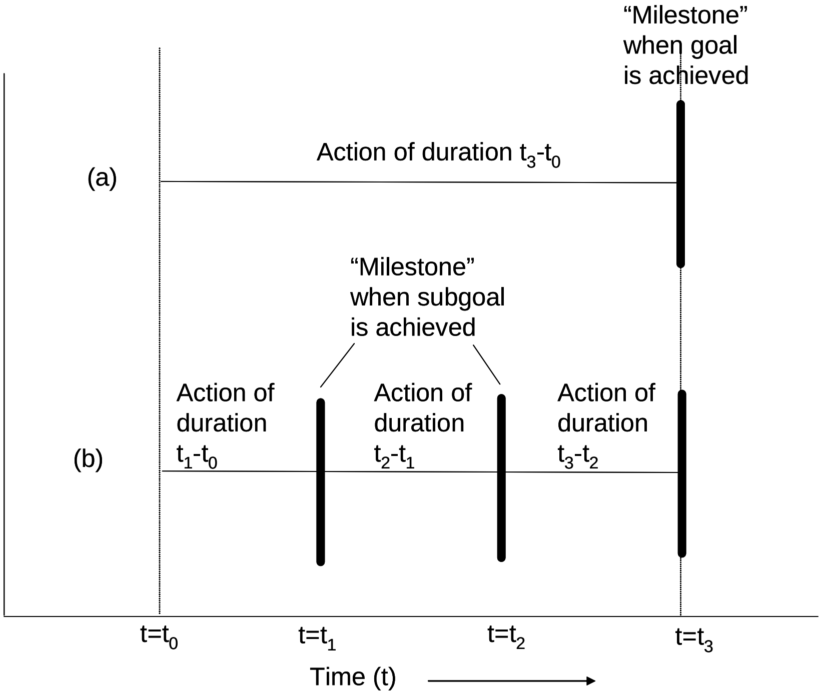
"Milestone" when goal is achieved
Action of duration t_3-t_0
(a)
"Milestone" when subgoal is achieved
Action of duration t_1-t_0
Action of duration t_2-t_1
Action of duration t_3-t_2
(b)
t=t_0
t=t_1
t=t_2
t=t_3
Time (t)

Figure 1 Milestones, subgoals and goals: (a) a single action attains the goal or (b) several actions are needed, each achieving a subgoal

In order to achieve a goal to which it is committed, an agent may need to do certain things. Thus there may be an action or, more commonly, a series of actions undertaken leading to the accomplishment of that goal[1] - assuming that that goal remains as one of the agent's commitments. We can think of this, intuitively, as a series of actions (or procedures or activities or tasks) each of which takes a finite amount of time. When an action is completed, and depending on the (sub)goal itself, the prior state of the agent and the prior state of the environment, the goal (or subgoal) may or may not have been achieved. The "milestone" that has been attained is associated with a single point in time (as compared to the action which acts over a specific temporal duration) and may or may not correspond to the intended subgoal. If not, an alternative atomic action is selected - this selection depending in general upon the states of the environment and of the agent at the time the selection is made. The case when the milestone corresponds to the achievement of a subgoal is illustrated in Figure 1a. For all successful actions other than the final one in the action series, we can map the milestone to an interim or sub-goal[2]. Each successful action thus links to the achievement of either a subgoal or the final goal. Figure 1b shows the situation in which two subgoals have been introduced, leading to a total of three actions that must be accomplished in order to fulfil the primary goal (at $t=t_3$).

Terminology across different agent models is, however, inconsistent. In some architectures, the word goal is used to describe some desired state, of either the agent or the system (environment plus agents). To reach that state some action or task must be undertaken (Figure 1). In other agent models, the terms goal and task are used interchangeably. Often a single term is used to mean *both* the end point and the means to achieve the end point (the milestone and the action as shown in Figure 1) - likely to lead to confusion. In this paper, we use an etymological approach together with a metamodel representation of these various models and attempt to standardize this portion of agent terminology.

Agent-Oriented Methodologies

In a multi-agent system (MAS), individual agents can exhibit two different forms of reasoning. They may be described as deliberative, proactive or goal-directed or as reactive or event-driven. Agents combining both forms of reasoning are called hybrid agents. The former mode of reasoning identifies an end-point - an objective that the agent wishes to achieve - and then plans are drawn up, dynamically revised and actioned to achieve that objective. Plans are often depicted using statechart notation that also assists in identifying, describing and structuring subgoals.

A deliberative agent continually reviews its commitments in the light of its state and of its observations of its environments. It may decide to decommit to a partially achieved goal. If it does, so then it may be necessary to perform a sequence of actions to return the environment to an acceptable state, since the goal has ceased to be one of the agent's

[1] Each action, however trivial it may be, is intended to achieve some goal.
[2] Initially we assume these terms to be synonyms.

commitments. In the second (reactive) mode, the agent has no predefined plan but reacts directly to changes in its environment. Reactive agents are easy to build, and so are preferable if the agent's role may be encapsulated in reactive logic. These two forms of reasoning (proactive and reactive) have their respective strengths and weaknesses [31,34]. In this paper, we will, however, focus primarily on goal-directed or proactive behaviour as opposed to reactive behaviour [36]. Since reactive agents do not have plans, goals and tasks inherent in their construction, we remove them from our further discussion.

Etymological and Metamodel Analysis

The Shorter Oxford English Dictionary (OED) [24] defines a goal as an “object of effort” or a “destination”; whereas task is defined as a “piece of work to be done”. Thus the OED is certain about the difference: a goal is an end state, something to be achieved. It is the destination itself and NOT a recipe for how to reach that destination, while acknowledging that effort and time need to be spent in its attainment. A task, on the other hand, is clearly seen as a work unit; it is the work itself. An associated term, that of “Action”, is also worth defining here: as “exertion of energy” [24]. Furthermore, Zhang and colleagues [39] note that a goal describes “*what is to be done*” and an activity or process “identifies *how things are to be done*”.

In this paper, we assess how a number of AO methodologies measure up against this etymological definition. In addition, we supplement the terminological discussion with a series of metamodels. We describe the concepts underpinning the agent architectural models by means of a UML [28] class diagram but expressed at the M2 level, a level at which the rules of the model (here the agent architecture) are defined. This permits us to analyze objectively how different concepts relate to each other, thus supporting an analysis of whether, or to what extent, different models, as used in various AO methodologies, correspond to each other. From this comparative analysis using both etymology and metamodels we can readily identify similarities and differences between contemporary agent-oriented methodologies.

3 HOW METHODOLOGIES VIEW TASKS AND GOALS

Agent-oriented (AO) methodologies place different emphasis on the key concepts of agency and how one might use those concepts in analyzing and designing an MAS (multi-agent system). Although there are many dimensions along which AO methodologies can be categorized, one identifies the importance that is placed on roles[3]; another on whether the methodology has an object-oriented (OO) or a knowledge engineerign (KE) ancestry. In all these, since an agent is autonomous and can strive to attain certain goals in its provision of services to other agents within the MAS, some notion of “goal” is utilized. In some methodologies a second concept, the achievement of

[3] in the sense of a set of, usually temporary, behaviours.

a goal, is identified as clearly distinct from the goal itself. This may be called variously action, activity or task. As noted above, in other methodologies, the same term is used for both the end point (the goal) and the process by which the goal is attained (the task), thus confounding two concepts as expressed in the intuitively-derived and dictionary definition-supported Figure 1.

In this section, prior to analyzing specific AO methodologies, we first discuss a commonly accepted architecture: the BDI (Beliefs, Desires and Intentions model [16]) description of deliberative agents (since a large number of AO methodologies use this or a similar model of agency). We then analyze the etymology and metamodels for a number of other commonly used/commonly cited AO methodologies.

The BDI Architectural Model and BDI Methodology (BDIM)

An important and influential deliberative agent architecture is BDI [16,25], which which describes the Beliefs, Desires and Intentions held by an agent. Winikoff and colleagues [31] offer a succinct summary of the BDI architecture proposed original by Rao, Georgeff and colleagues [16,25]. They distinguish between three “layers” or abstraction levels: philosophical, theoretical (called here “design”) and implementation (Table 1). Beliefs, Desires and Intentions (which give the model its acronymic name) are seen by these authors as high level, abstract, external characteristics, which can then be mapped to internal agent characteristics. Beliefs are mapped to a knowledge repository (for example, a link to a relational database (RDB)); desires are mapped to an agent's goals, ultimately implemented in terms of events; and intentions are mapped to plans implemented as actions intended to achieve the current subgoal. Each goal must have a link to at least one plan.

Table 1 Relationships between terminology (adapted from [31])

Viewpoints			
Philosophy	Belief	Desire	Intention
Design	Belief	Goal	Intention/Plan
Implementation	Knowledge base (e.g. RDB)	Event	Running Plan/Current action

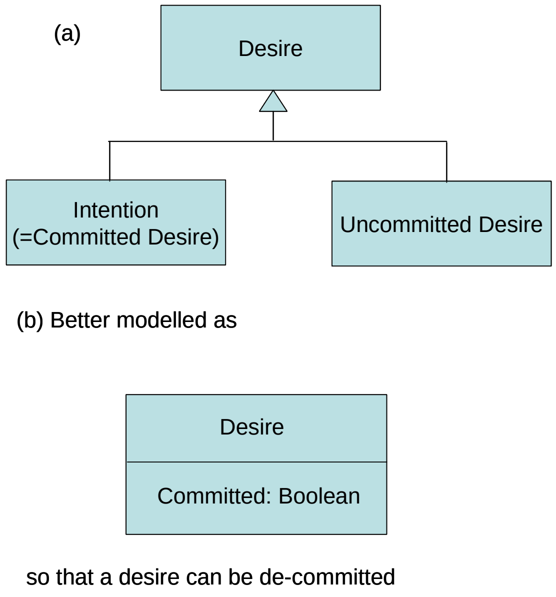

Figure 2 Modelling Desires and Intentions: (a) using an inheritance structure and (b) modelling with an attribute. The latter case permits a goal to be de-committed which is not possible with (a) since it is generally agred that objects cannot change their type dynamically.

We propose here a revision of this overview table as follows:

a) although suggestive in the name BDI, these three characteristics are not in fact orthogonal. In particular, it is generally agreed (e.g. [34]) that intentions are a subset of desires – they are those desires that have been committed to (Figure 2). Similarly, at the design level, Goals plus a commitment leads to the notion of a committed goal or, often more simply, a Commitment (Table 2).

Table 2 Revised relationships between terminology

Viewpoint [1]	**[2]**	**[3]**	**[4] = [3] + commitment**	**[5]**
Psychology	Belief	Desire	Intention	Wherewithal (“how”)
Design/Model	World Model	Goal	Commitment	Plan
Implementation	Knowledge Base	-	-	Running (or instantiated) Plan

b) Plans are not simply design-level intentions nor implementations of intentions. Rather, Desires and Goals (and their commitment subsets of Intentions and Commitments) all address the issue of “what” needs to be done; whereas Plans clearly address the issues of “how” the goal/commitment is to be achieved. We thus introduce a new column describing these(column [5] in Table 2). Although excluded from the original BDI work, it is being increasingly recognized [16,p2],[31,30] that Plans must be an integral part of any BDI-based agent approach.

c) In column 2 we introduce a Philosophy/Design differential between Belief and World Model

d) Events are removed from Table 1 since it is normal in agent technology to associate the term Event with external environmental occurrences.

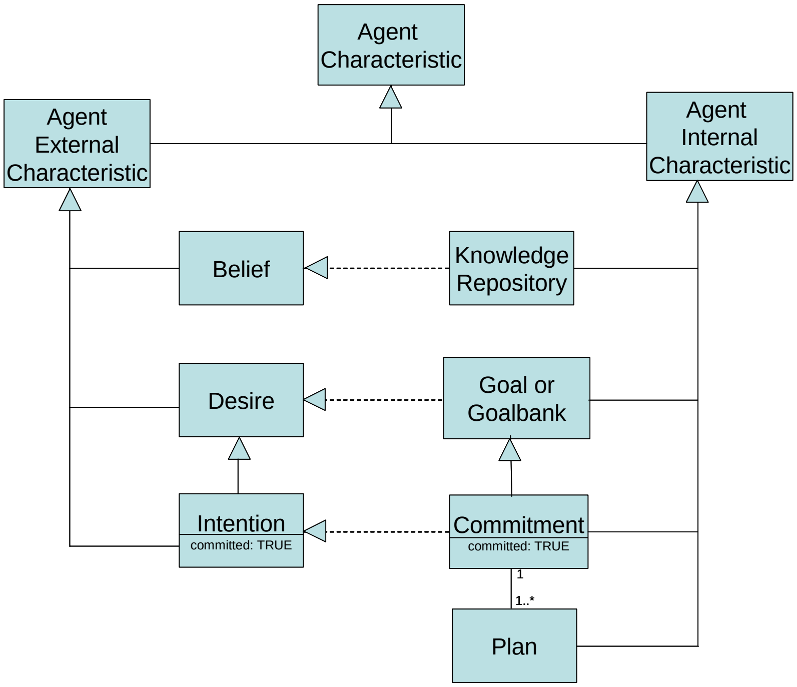

Figure 3 Metamodel of concepts used in BDI architecture

The terminology used in the three viewpoints of the BDI architecture, as summarized in Table 2 and modelled in Figure 3, is as follows:

- Beliefs are the agent's information about its environment and about the other agents.
- A Percept is information acquired by the agent from its environment. A change in the environment may cause an internal Event to occur.
- Desires represent heterogeneous objectives to be accomplished. They need not be consistent and may therefore contain implicit or explicit contradictions.
- A Goal, or perhaps more expressively a Goalbank, is often said to be a consistent set of desires[29,31], which, when committed to, becomes a Commitment. It is therefore also an objective to be accomplished or achieved, usually by the execution of Plan(s). Not all Goals can be held concurrently without contradiction so a subset is committed to. Once a commitment is made, the goal can be considered as encompassing the current intentions of the agent [18]. Thus each commitment is a (high level) goal and each committed goal is the subject of at

least one plan. While a goal remains as a commitment, the actions being executed may need to be revised (perhaps according to a plan or as a reaction to changes in the external environment).

- Intentions relate to a set of selected goals together with their state of processing [18], enacted by the currently chosen course of action [31].
- A Plan is a means by which a selected future state (as represented by a goal) can be achieved. Plans represent both the means and available options [31] and are often depicted using statecharts where the states are (sub)goals. Thus plans represent in some sense the structural decomposition of goals together with events causing transitions between subgoals. Deriving a plan, ideally containing atomic subgoals, for a specific goal involves means-end reasoning [34] - an important technique. [We note that, in [25], it is suggested that Plans are special forms of Beliefs. In the context of Tables 1 and 2, this is hard to understand.]
- The entry in Table 1 of RDB is a single example of how an agent may store its beliefs (knowledge of its world model) at the implementation stage.

Other terminology needed for a complete picture (but not shown in Figure 3) is:

- Events are linked to perceived changes in the environment, known as percepts (q.v.) or may be generated internally by the agent e.g. by an internal clock.
- Proactive agents focus on the achievement of goals; reactive agents react primarily to events.
- An Action represents something that is done. It either fails or else it succeeds if its (sub)goal is achieved. This is similar to the definition of Task in, for example, [7].

These various definitions relating to a BDI architecture, as summarized by [31], allow us to construct an underlying metamodel, which is shown in Figure 4. Note that in [25] and [31], the body of a plan is usually described by a statechart in which the states represent subgoals. In [16], BDIM as applied to internal modelling is said to have two steps. The first recommends the designer to "decompose each goal into activities, represented by subgoals, and actions". This model is shown in Figure 5 in which it is clearly seen that the milestone Goal consists of a static Subgoal (a.k.a. Activity) and an Action (which has duration). This is clearly untenable as discussed above (see also Figure 1). If we are lenient in our interpretation we could replace "goal" by "goal achievement" and "activity as represented by subgoal" by "subgoal achievement". The corresponding metamodel (Figure 6) could be more easily defended - although far from perfect. This introduces a confusion between goals and actions which we foreshadow here. Indeed, we will argue below that if we think of goals as being achieved by the execution of plans, then a simple revision of Figure 6 would show a plan as consisting of subgoals and actions (Figure 7) - indeed this is borne out by statements in [16] in the second step of BDIM.

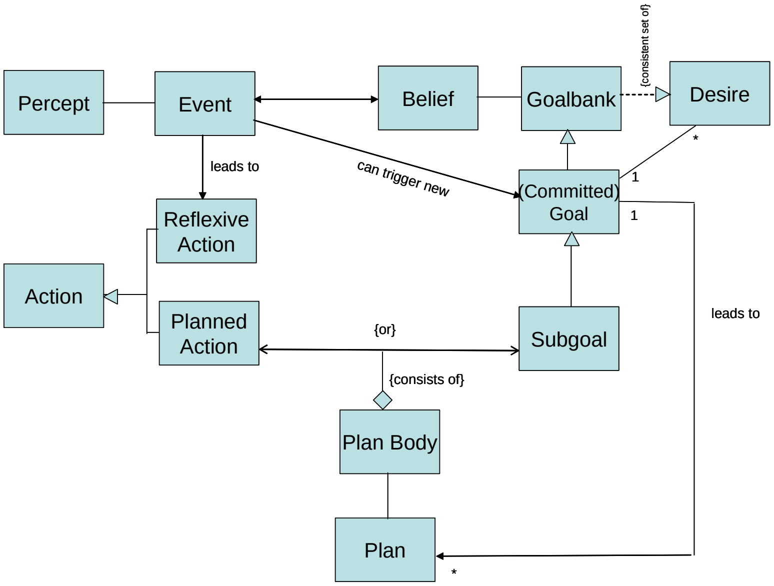

Figure 4 Metamodel of concepts used in BDIM

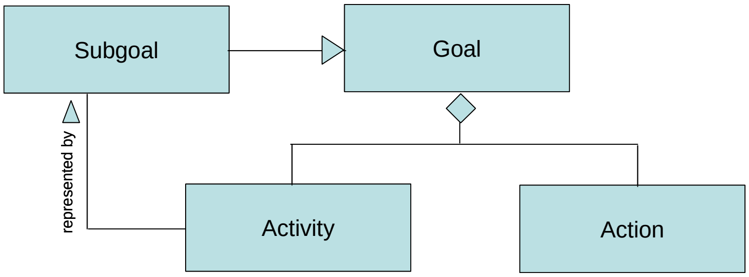

Figure 5 Goals, Activities and Actions metamodel

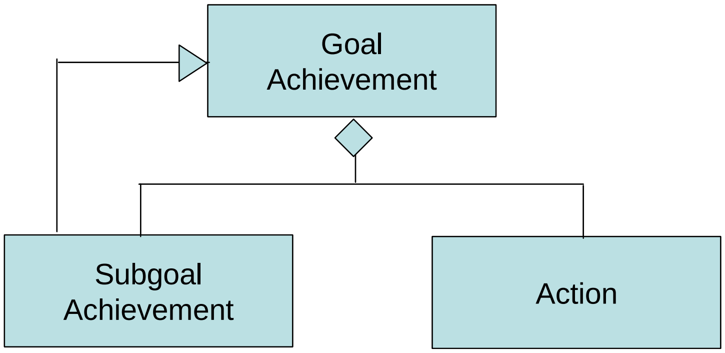

Figure 6 Revision of Figure 5

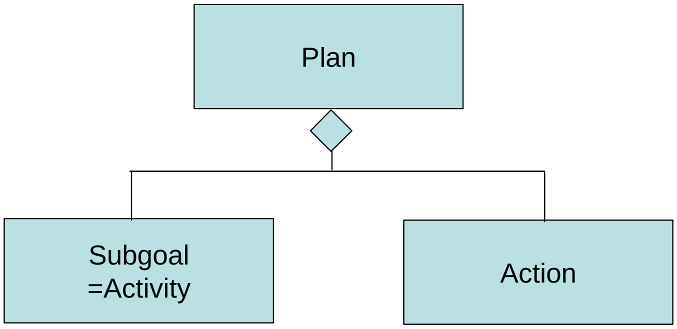

Figure 7 Metamodel for Plans, Actions and Activities

Furthermore, it is etymologically unclear why the word “activity” (a work unit usually possessing duration) is often equated with “subgoal” (a target state or milestone). One possible (mis)interpretation might be that some subgoals have plans associated with them and some have activities (atomic chunks of action) associated with them, and some may be associated with both. One might surmise here an influence from the Object Modeling Technique (OMT [26]) (said to have influenced BDIM), in which an activity was permitted to occur while residing for a finite duration in a given state. Regrettably, in usage, the activity name, intended to be secondary, was frequently elevated to become effectively the state name, thus leading to an easy confusion by which OMT statecharts were accidentally transformed into data flow diagrams (DFD). Figure 3 of [16] could easily be (mis)read as representing a substate called “activity formula”.

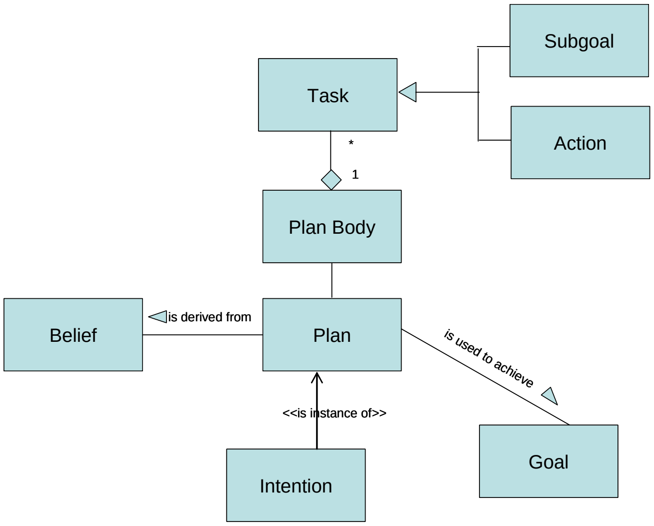

Figure 8 Relationship between Tasks, Plans, Beliefs, Intentions and Goals (in BDI)

There is further confusion: in [35, p70] goals are further confounded with intentions, which clearly disagrees with the well-accepted BDI architecture (Table 1). Further terminological confusion is exemplified in [15] in a discussion of strong (as opposed to weak) agents. Here the authors state that (strong) agents reason about *beliefs* in order to select a *plan* for achieving their *goals* (Figure 8). An instantiated plan is said to be an intention[4], whereas the body of the plan is a set of tasks, said to include, for example, actions and subgoals (Figure 9). Interestingly, Figure 9 is but a minor elaboration on Figure 7, as well as some parallels with Figure 4, but both are arrived at by different lines of argument (see above). However, it does introduce the polymorphic relationship between Tasks and Subgoals, which we argue above is etymologically incorrect; yet explains why some writers so readily exchange the words Task and Subgoal. We suggest that Figure 9 epitomizes the current misunderstandings and ambiguities in the literature, while Figure 7 offers an acceptable resolution, in which Task may be recommended as a synonym for Action if preferred.

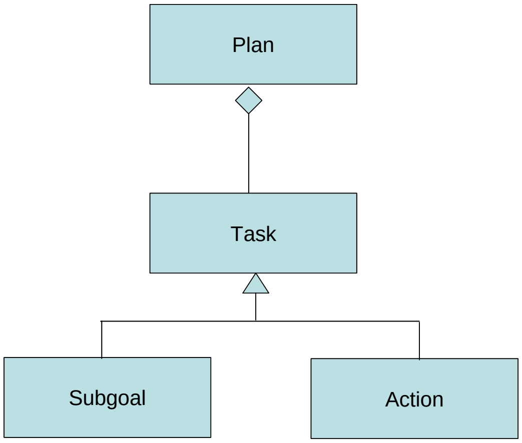

Figure 9 Illustration of current ambiguities in the literature

Tasks and Goals Differentiated

There are a group of AO methodologies that are fairly clear in their discrimination and thus uphold the etymological sources of the two words (task and goal). They may allow goals to be broken down into subgoals and tasks to be decomposed into subtasks. The metamodel is shown in Figure 10. The details of the extent to which this metamodel is used in each of the relevant AO methodologies is briefly discussed in the following subsections.

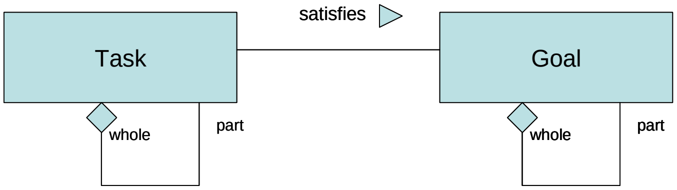

Figure 10 Relationship between Goals and Tasks when differentiated

[4] In the light of Table 1, this would appear to be an error, since a Plan instantiates the Intention.

ACR. While differentiating goals and tasks, ACR [10] supports decomposition of tasks but does not mention decomposition of goals (Figure 11). A goal is a state (employing the same concept of goal as the original BDI work of [16,25]) and a task as being performed by a role in order to fulfil the goal(s).

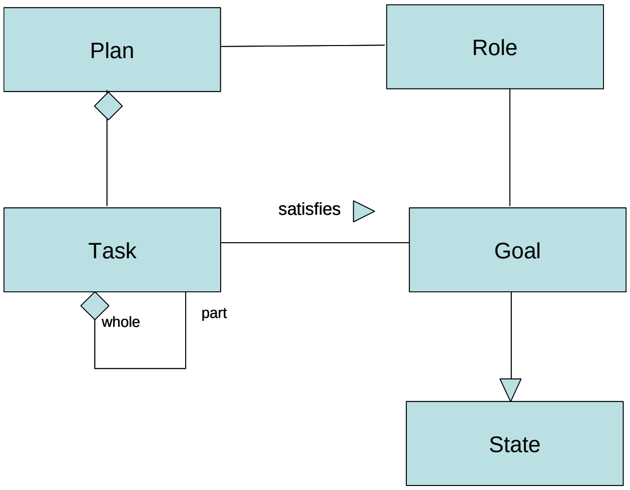

Figure 11 Goals and Tasks in ACR

Cassiopeia. While differentiating goals and tasks, Cassiopeia [6] appears not to permit any further decomposition. Goal is not used explicitly but rather the concept is replaced by the term “collective task” as a representation of the main functionality of the MAS. Goal attainment is described in terms of “elementary behaviours” which are required to achieve the collective task. The terminology is thus different from many other AO methodologies and etymologically misdirectional in using “collective task” to represent the overall goal.

HLIM. HLIM [8] differentiates goals and tasks, and permits further decomposition of both. The methodology states that “An agent may adopt goals to reach a desired state” whereas a Task is a means to fulfil goals. Both goals and tasks for an agent are identified from Use Case Maps (UCM). A stub in the UCM path segments represents a block of responsibilities or activities from which the tasks are directly mapped. If the stub is dynamic, it is mapped to a “subgoal” (as in Figure 1) and if static it is mapped to a complex task. The responsibilities inside each stub are then mapped to tasks in order to achieve the subgoal or to decompose the complex task.

MASE. While differentiating goals and tasks, MaSE [33] only permits decomposition of goals, initially mapped to roles, in a Goal Hierarchy Diagram and not tasks. A goal is an objective or declaration of system intent, which is clearly mappable to the notion of a state; a task is a structured set of communications and activities depicting how a role goes about fulfilling a goal; in other words, a means to achieve the goal. The goal of each role is then simply mapped to one or more tasks.

MESSAGE. While differentiating goals and tasks, MESSAGE [9] only permits decomposition of goals and not tasks. Each leaf of a Goal Decomposition Diagram is associated with a Workflow Diagram showing a partially ordered set of tasks to accomplish this goal. A goal is defined to "associate an agent with a state" and a task as "a knowledge-level unit of activity within a single prime performer" i.e. a means to achieve a goal.

Prometheus. While differentiating goals and tasks, Prometheus [21] appears not to permit any further decomposition. The task expresses functionality and is the means to achieve the goal. However, it is unclear whether it is intended that the goal should be a large task or a state. Here we assume the latter, since Prometheus is built on a BDI architecture. However, we note in passing that while Actions (Tasks), Events and Plans have their own notation, there is no notation in Prometheus for Goals. Detailed design focusses on capabilities of the agent and a progressive refinement thereof. Only at the bottom level are capabilities linked to plans.

MAS-CommonKADS [14] uses both goal and task but does not appear to have goal decomposition. The term task is used to represent the desired/required functionality of the MAS and it is permitted for these tasks to be decomposed into subtasks. Goals of tasks are assigned to agents and enhanced CRC cards are used for this purpose. Neither term is, however, well defined. Instead, in section 1 of the paper [14], goals are said to be a subtype of task [not upheld in the rest of the paper and therefore assumed to be in error].

Tasks and Goals Not Differentiated

In a second group of AO methodologies, the terms task and goal are effectively used as synonyms in that they typically use one term and eschew the second. To be more precise, only one of these actually uses the word goal at all and the rest define task as the end point of achievement AND the means by which to achieve that endpoint. They too may allow goals to be broken down into subgoals (for any methodology that uses the term "goal") and tasks to be decomposed into subtasks (although two only permit use of the top level notion of "task"). The metamodel is shown in Figure 12. The details of the extent to which this metamodel is used in each of the relevant AO methodologies is briefly discussed in the following subsections.

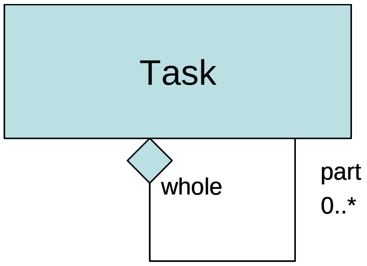

Figure 12 Goals and Tasks not differentiated

COMOMAS [12] does not use the term goal and therefore does not have goal decomposition. It does use the term task and permits these tasks to be decomposed into subtasks. "Tasks of MAS are those that help realize an organizational function".

MASSIVE [17] does not use the term goal and therefore does not have goal decomposition. It does use the term task and permits these tasks to be decomposed into subtasks. A task is defined as the "specification of what the system should do".

SODA [20] uses the terms task and goal but effectively as synonyms since it states that `The application goals are modelled in terms of the tasks to be achieved" and these tasks are made up of responsibilities. It also does not permit any further decomposition of tasks into subtasks.

Other Viewpoints

Gaia [37] focusses on roles rather than a BDI architecture. Roles are defined by four attributes: responsibilities, permissions, activities and protocols. They would thus appear to be a significantly enhanced OO model of a class (now an agent class), particularly one associated with Responsibility Driven Design as originally proposed by [32]. Agent functionality is expressed in terms of *services* associated with each role, as well as by its responsibilities, particularly its liveness responsibilities. Overall, Gaia is relatively weak on internal agent architecture stressing instead the societal aspects of agents in terms of its acquaintance model **.**

Tropos. In the Tropos methodology [3,5,22,23], we have a slightly more unusual situation, derived from the i* framework [38]. The focus here is on using AO concepts not for the internal architecture of an individual agent but rather for modelling the requirements and the requirements capture process. At the same time, the target internal architecture is recommended as BDI so it is largely BDI concepts that influence the Tropos RE Modelling Language.

Goals can be decomposed into subgoals in two ways such that the goal itself is achieved if (i) one of the subgoals is met (OR-decomposition) or (ii) all the subgoals are met (AND-decomposition). Plans are then used to achieve these goals/subgoals (which are also characterized as being hard goals or soft goals) [3,11]. However, in their discussion of means by which a goal is achieved, an ambiguity occurs - from both an etymological and metamodel viewpoint. It is said [3] that "a goal (the end), and a *Plan, Resource* or *Goal* (the means)" is a relevant model, based on Means-end Analysis which consists of "a discovery of goals, plans or resources that can provide a means for reaching a goal" (Figure 13). Thus the word "goal" is used (incorrectly in our view) to describe both the end-point and the means to achieve that end point [38]. This triad (of Plan, Resource and Goal[5]) is also used directly in the Tropos technique of Dependency analysis where one of these three provides the context for inter-Actor dependencies (Figure 14). This metalevel diagram stands in contrast to that of Figure 13 from which one could erroneously deduce that one means of achieving a goal is a goal - which is etymologically

[5] In other Tropos papers, e.g.[19], Plan is renamed Task and Softgoals and Goals are differentiated. Softgoals were also added in [2].

unsound. It is perhaps a failure to distinguish the semantic difference between an action of finite duration and an (instantaneous) milestone as depicted in Figure 1.

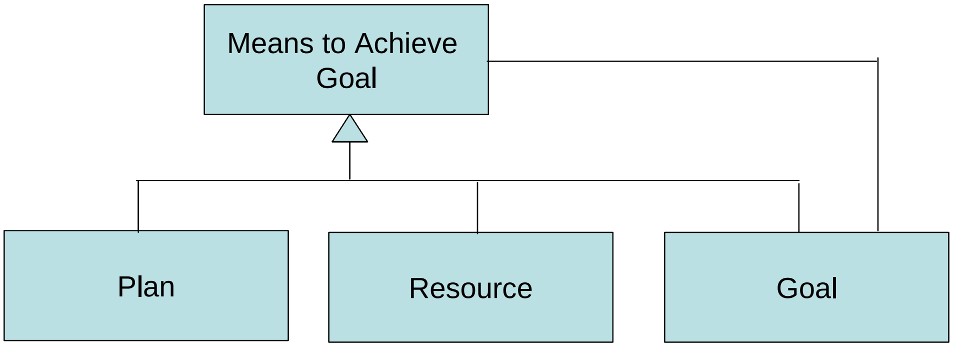

Figure 13 Metamodel of Means to Achieve Goal in Tropos

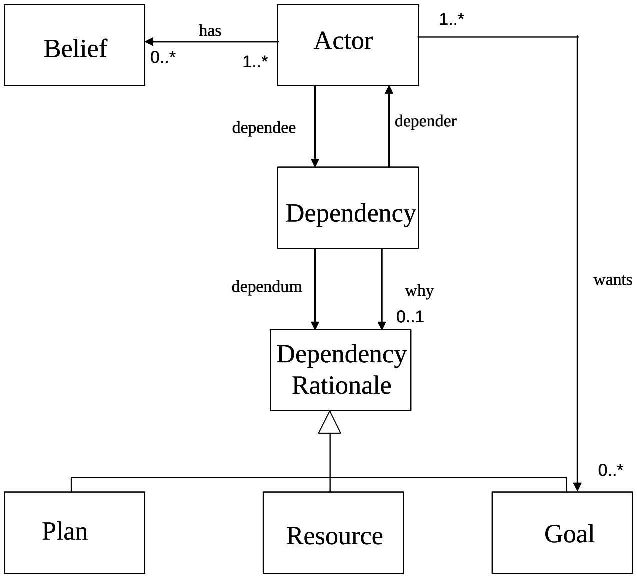

Figure 14 Tropos model of inter-Actor dependencies

Task is only used as a term in some of the Tropos papers where it is clearly described (e.g. [4,19] as a way of achieving the needs stated in goals.

4 DISCUSSION AND CONCLUSIONS

Agent-oriented methodologies frequently make use of terms such as goal and task but do so in an inconsistent manner. By using an etymological and metamodel-based analysis of a significant number of these AO methodologies, we recommend that the word task is to be avoided; instead, the word action could be usefully employed to describe the work done to achieve a goal or subgoal – as recently used also in TAO [30]. We also note that the notion of subgoal itself is ambiguous in either representing an interim goal along the path of achievement of the main (final or overall) goal as in Figure 1 or, alternatively, a portion/part of the goal whose achievement contributes (at the same instant in time) to the achievement of the overall goal. If we accept subgoal for the former meaning, then we suggest "goal part" for the latter.

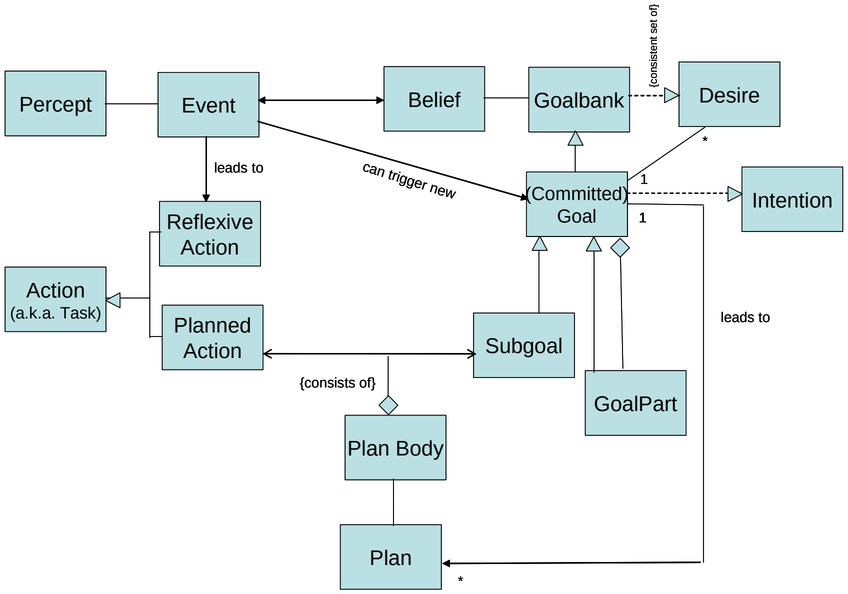

Figure 15 Final recommendation for etymologically sound metamodel for Goals and Tasks for agent-oriented methodologies

We thus conclude that an appropriate metamodel is that in Figure 7 with the addition of a whole-part relationship from Goal to Goal Part. Furthermore, we eschew the word Task in favour of Action and, finally, recommend that the Plan Body should consist of Actions and/or Subgoals (i.e. change the "or" to an "and/or"). This leads us to a final metamodel (Figure 15) to complement this etymologically recommended set of terminology.

ACKNOWLEDGEMENTS

We wish to acknowledge financial support for this project from the University of Technology, Sydney under their REGS (Research Excellence Grants Scheme). This is Contribution Number 04/14 of the Centre for Object Technology Applications and Research (COTAR).

REFERENCES

[1] Bratman, M.E., 1987, *Intentions, Plans, and Practical Reason*, Harvard University Press, Cambridge, MA, USA

[2] Bresciani, P. and Sannicolo, 2002, Applying Tropos early requirements analysis for defining a Tropos tool, **in** *Agent-Oriented Information Systems 2002* (eds. P. Giorgini, Y. Lespérance, G. Wagner and E. Yu), 135-138

[3] Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J. and Perini, A., 2004, Tropos: an agent-oriented software development methodology, *Autonomous Agents and Multi-Agent Systems*, **8(3),** 203-236

[4] Castro, J., Pinto, R., Castor, A. and Mylopoulos, J., 2003, Requirements traceability in agent oriented development, **in** *Software Engineering for Large-Scale Multi-Agent Systems* (eds. A. Garcia, C. Lucena, F. Zambonelli, A. Omicini and J. Castro), LNCS 2603, Springer-Verlag, 57-72

[5] Castro J., Kolp M. and Mylopoulos J., 2002, Towards Requirements-Driven Information Systems Engineering: The Tropos Project. *Information Systems*, **27(6)**, 365-389

[6] Collinot, A., Drogoul, A. and Benhamou, P., 1996, Agent-oriented design of a soccer robot team, *Procs. 2nd Int. Conf. on Multi-Agent Systems (ICMAS'96)*, 41-47

[7] Duncan, W.R., 1996, *A Guide to the Project Management Body of Knowledge*, Project Management Institute, PA, USA, 176pp

[8] Elammari, M. and LaLonde, W., 1999, An agent-oriented methodology: high-level and intermediate models, *Procs. 2st Bi-Conf. Workshop on Agent-Oriented Information Systems (AOIS'99)*

[9] Euroscom, 2001, *Methodology for Agent-Oriented Software Engineering*, http://www.eurescom.de/public/projectresults/P900-series/907ti1.asp

[10] Fan, X., 2000, Towards a building methodology for software agents, *TUCS Technical Report No. 351*, Turku Centre for Computer Science

[11] Giunchiglia, F., Mylopoulos, J. and Perini, A., 2001, The Tropos software development methodology: processes, models and diagrams, Technical Report #0111-20, Istituto Trentino di Cultura, 8pp

[12] Glaser, N., 1997, *The CoMoMAS Approach: From Conceptual Models to Executable Code*, http://citeseer.nj.nec.com/32190.html

[13] Henderson-Sellers, B., Giorgini, P. and Bresciani, P., 2003, Evaluating the potential for integrating the OPEN and Tropos metamodels, *Procs. SERP '03* (eds. B. Al-Ani, H.R. Arabnia and Y. Mun), CSREA Press, Las Vegas, USA, 992-995

[14] Iglesias, C.A., Garijo, M., Gonzalez, J.C. and Velasco, J.R., 1998, Analysis and design of multiagent systems using MAS-CommonKADS, *in Intelligent Agents IV (ATAL'97)* (eds. M.P. Singh, A. Rao and M.J. Wooldridge), Springer-Verlag, Berlin

[15] Kendall, E.A., Malkoun, M.T. and Jiang, C., 1995, A methodology for developing agent based systems for enterprise integration, EI'95, IFIP TC5 Working Conference on Modeling and Methodologies for Enterprise Integration, Heron Island, Queensland, Australia

[16] Kinny, D., Georgeff, M. and Rao, A., 1996, A methodology and modelling techniques for systems of BDI agents, Technical Note 58, Australian Artificial Intelligence Institute, also published *in Agents Breaking Away: Procs. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*, 56-71

[17] Lind, J., 1999, *MASSIVE: Software Engineering for Multiagent Systems*, PhD Thesis, University of Saarbrucken, Germany

[18] Müller, J.P., 1996, *The Design of Intelligent Agents. A Layered Approach*, LNCS 1177, Springer-Verlag, 227pp

[19] Mylopoulos, J., Kolp, M. and Castro, J., 2001, UML for agent-oriented software development: the Tropos proposal, **in** *UML2001* (eds. M. Gogolla and C. Kobryn), LNCS 2185, Springer-Verlag, Berlin, 422-441

[20] Omicini, A., 2000, SODA: Societies and Infrastructure in the Analysis and Design of Agent-Based Systems, *Procs. 1st Int. Workshop on Agent-Oriented Software Engineering (AOSE-2000)*, 185-194

[21] Padgham, L., and Winikoff, M., 2002, Prometheus: A pragmatic methodology for engineering intelligent agents, *Procs. Workshop on Agent-Oriented Methodologies at OOPSLA 2002*, COTAR, Sydney, 97-108

[22] Perini A., Bresciani P., Giorgini P., Giunchiglia G. and Mylopoulos J., 2001, A Knowledge Level Software Engineering Methodology for Agent Oriented Programming, In J.~P. Müller, E. Andre, S. Sen, and C. Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, May 2001, Montreal, Canada, 648-655

[23] Perini A., Bresciani P., Giorgini P., Giunchiglia F. and Mylopoulos J., 2001, Towards an Agent Oriented approach to Software Engineering, In A. Omicini and M. Viroli, editors, *WOA 2001 - Dagli oggetti agli agenti: tendenze evolutive dei sistemi software*, 4-5 September 2001, Modena, Italy, Pitagora Editrice Bologna

[24] OUP, 1960, *The Pocket Oxford Dictionary of Current English*, fourth edition, Clarendon Press, Oxford

[25] Rao, A.S. and Georgeff, M.P. 1995, BDI agents: from theory to practice, Technical Note 56, Australian Artificial Intelligence Institute

[26] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W., 1991, *Object-oriented Modelling and Design*, Prentice Hall, New Jersey, 500pp

[27] Russell, S. and Norvig, P., 1995, *Artificial Intelligence. A Modern Approach*, Prentice-Hall, Inc.

[28] OMG, 2001, *OMG Unified Modeling Language Specification, Version 1.4*, September 2001, OMG document formal/01-09-68 through 80 (13 documents) [Online]. Available http://www.omg.org

[29] Schlieder, C., Timm, I. and Hermes, T., 2002, Autonomous behavior in multiagent systems, Universität Bremen (available at http://www.informatik.uni-bremen.de/~hermes/lectures/ik2002/lecture4%20cs.pdf)

[30] Torres da Silva, V. and Lucena, C.P., 2004, From a conceptual framework for agents and objects to a multi-agent system modeling language, Autonomous Agents and Multi-Agent Systems, 8, 1-45

[31] Winikoff, M., Padgham, L. and Harland, J., 2001, Simplifying the development of intelligent agents, *Procs. 14th Australian Joint Conference on Artificial Intelligence (AI'01)*, Adelaide, 10-14 December 2001

[32] Wirfs-Brock, R., Wilkerson, B. and Wiener, L., 1990, *Designing Object-Oriented Software*, Prentice Hall, Englewood Cliffs, NJ, USA, 341pp

[33] Wood, M. and DeLoach, S., 2000, An overview of the Multiagent Systems Engineering methodology, *Procs. 1st Int. Workshop on Agent-Oriented Software Engineering (AOSE-2000)*, 207-222

[34] Wooldridge, M., 1999, Intelligent agents, **in** *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence* (ed. G. Weiss), MIT Press, 27-77

[35] Wooldridge, M., 2002, *An Introduction to MultiAgent Systems*, John Wiley & Sons, Ltd.

[36] Wooldridge, M. and Ciancarini, P., 2001, Agent-oriented software engineering: the state of the art***, in*** *Agent-Oriented Software Engineering* (eds. P. Ciancarini and M.J. Wooldridge), LNCS 1957, Springer-Verlag, Berlin, 1-28

[37] Wooldridge, M., Jennings, N.R. and Kinny, D., 2000, The Gaia methodology for agent-oriented analysis and design, *Autonomous Agents and Multi-Agent Systems*, **3(3),** 285-312

[38] Yu, E.S.-K., 1995, Modelling strategic relationships for process reengineering, unpubl. PhD thesis, University of Toronto, 124pp

[39] Zhang, T.I., Kendall, E. and Jiang, H., 2002, An agent-oriented software engineering methodology with applications of information gathering systems for LCC, ***in*** *Agent-Oriented Information Systems 2002. Procs. of AOIS-2002* (eds. P .Giorgini, Y. Lespérance, G. Wagner and E. Yu), 27-28 May 2002, Toronto, Canada, 32-46

About the authors

Brian Henderson-Sellers is Director of the Centre for Object Technology Applications and Research and Professor of Information Systems at University of Technology, Sydney (UTS). He is author of eleven books on object technology and is well-known for his work in OO methodologies (MOSES, COMMA, OPEN, OOSPICE), in OO metrics and, more recently, in agent-oriented methodologies. He can be reached at brian@it.uts.edu.au.

Quynh-Nhu Numi Tran is a PhD candidate in Information Systems at the University of New South Wales. Her dissertation is on the development of an agent-oriented methodology for ontology-driven multi-agent systems. She can be reached at numitran@yahoo.com.

John Debenham is Professor of Computer Science at the University of Technology, Sydney. He is author of two books on the design of intelligent systems. John is Chair of the Australian Computer Society's National Committee for Artificial Intelligence, and is Secretary of IFIP's TC12 Artificial Intelligence. His recent research has focussed on multiagent systems with business process management, e-Negotiation and argumentation systems as the focus. He can be reached at debenham@it.uts.edu.au.