Companion paper of "Specifying and Verifying Holonic Agents with GDT4MAS": the case study as a stand-alone paper

B. Mermet, G. Simon

GREYC-UMR 6072 & Université du Havre
Campus Côte de Nacre, Boulevard du Maréchal Juin
BP 5186, 14032 CAEN cedex, FRANCE
e-mail: Bruno.Mermet@univ-lehavre.fr

1 Case study

The example presented here is an extension of the problem presented in [Bordini et al. (2003)]. Although it may look very simple, it allows to give examples of proofs that are tractable by the reader. The initial problem, called *RoM* (Robots on Mars) in the sequel, is the following: Two robots have to clean Mars. Robot $R1$ walk through the surface of Mars (represented by a grid) looking for wastes. If it finds one, it tries to pick it up (this goal is achieved with at most three attempts), brings it to the other robot $R2$, goes back to the position where it had found the waste, and continues to explore Mars. Robots $R2$ cannot move but it can burn wastes that are given to it.

In their paper, Bordini *et al.* use model checking to verify their specification. So, they have chosen to verify the correctness of their specification on a 5×5 grid containing two wastes. In a previous work, it has been shown that the GDT model allows the verification for a grid of any size and with any number of wastes [Mermet et al. (2007)]. More recently, we have shown that with the GDT4MAS model, the system can easily be verified even if there are several robots of each type and if $R1$ must go to the nearest robot $R2$ [Mermet and Simon (2009)] (Extended RoM – ERoM – problem). The complexity of the proof process does not depend upon the number of robots.

In this article, we briefly present how both types of agents $R1$ and $R2$ can be implemented by super-agents. In order to do so, the specification of the *ERoM* problem is slightly modified as follows:

Problem 1.1 (Compound ERoM Problem (CERoM)) *Robots of type $R2$ are made of two parts: an arm and an incineration system. The arm picks up wastes on the cell in which the robot is situated. It then puts the waste it holds in a tank with a limited capacity of 10 wastes. In the incinerator system, a conveyor belt starts when the tank contains three wastes or more. Then, it brings all the wastes in the tank one by one to an incinerator. During this process, if the incinerator*

Copyright © 2009 Inderscience Enterprises Ltd.

supervisor detects that the temperature of the incinerator is abnormal (either too low or too high), it changes the status of the robot $R2$ to down. *Robots of type $R1$ bring pieces of garbage they find to the nearest* up *robot $R2$. While moving to a robot $R2$, if the target robot breaks down, they compute a new target robot.*

This new specification can be analysed in terms of design patterns given in the previous section:

- the $R2$ type of agent is made of two actuators that act concurrently. So, the design pattern *several actuators* is used to decompose the main goal of robot $R2$ with a parAND operator in two agents *arm* and *incineration system*;
- while the incineration system conveys wastes (its basic behaviour), it must detect abnormal beaviours of the incinerator. If this occurs, it must interrupt its basic behaviour. This typically corresponds to the design pattern *Task interruptible by the occurrence of an event.*
- while a robot $R1$ is bringing a waste to the nearest *up* robot $R2$, it must detect if this robot breaks down. If this occurs, it must react by modifying the target. This corresponds to the design pattern *Combining cognitive and reactive behaviours*, with two sub-agents: the *motor* and the *tracker*.

In the following specification, we use several variables:

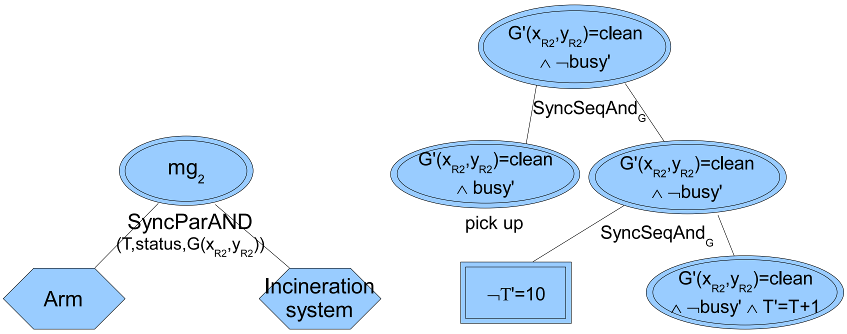
- G represents the grid. It is a variable of the environment of the system;
- x_{R2} and y_{R2} are constants of $R2$ that are initialized by parameters;
- *busy* is a variable of the arm of $R2$ specifying if it is carrying a waste;
- *status* is a variable of $R2$ specifying if it is *up* or *down*;
- T is a variable of $R2$ specifying the number of wastes in the tank;
- *temp* is a variable of the incinerator supervisor specifying its temperature;

Specifications of $R2$ and its subagents are presented in figures 1 and 2. The main goal of robot $R2$, named mg_2, consists, if $R2$ is up, in picking up the waste on its cell and in maintaining its tank not full. As a consequence, it has the following satisfaction condition:

$$SC_{mg_2} \equiv (status = up \rightarrow (G(x_{R2}, y_{R2}) = clean \wedge T < 10))$$

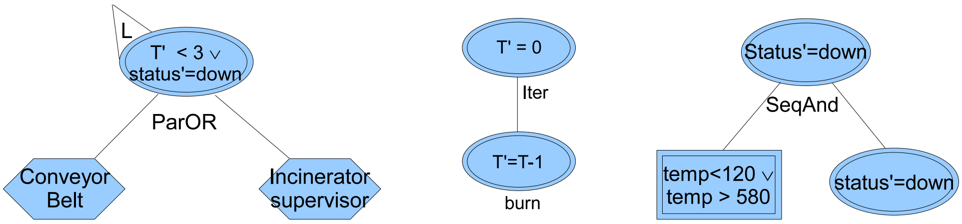
The pGDT associated to type $R1$ is summarized in figure 1. Some parts, summarized by a triangle, are not detailed here. The complete description can be found in [Mermet and Simon (2009)]. For the purpose of this article, we only precise that goal 13 is to bring the waste held to the nearest robot $R2$.

The behaviour of the motor is presented in figure 3(b). It is a part of the behaviour of $R1$ in the *ERoM* problem given in [Mermet and Simon (2009)] and is not detailed here. The behaviour of the tracker is presented in figure 3(c). It is a standard usage of the design pattern *Combining cognitive and reactive behaviours*. So, the satisfaction of its main goal is *false*, the satisfaction condition of goal $t2$



(a) pGDT of $R2$ (b) pGDT of the arm

Figure 1 pGDTs of $R2$ and of its arm



(a) pGDT of the incineration system (b) pGDT of the conveyor belt (c) pGDT of the incinerator supervisor

Figure 2 pGDTs of the incineration system and its subsystems

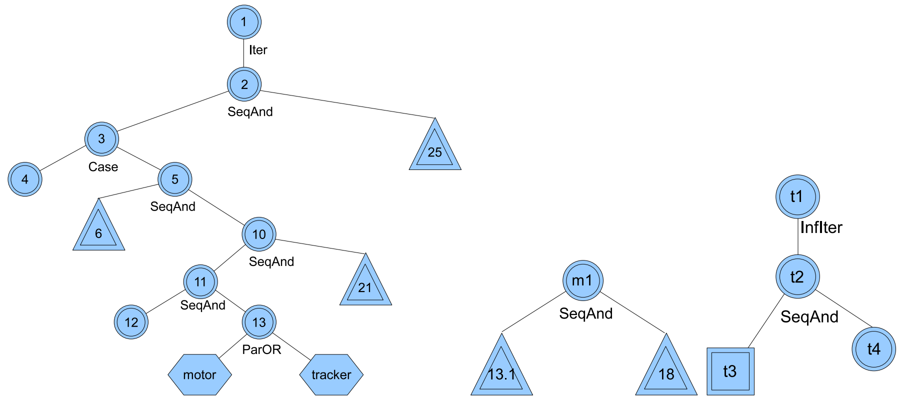
is the same as the one of goal $t4$ (this goal, that consists in finding the nearest robot of type $R2$ that is up, has the same SC as goal 12 of robot $R1$), and the satisfaction condition of external goal $t3$ expresses that the status of the target agent is *down*.

The whole structure of the Robots on Mars system is summarized in figure 4 where only one instance of each type of agent is detailed.

Most generated proof obligations with this new version are the same than those generated for the previous versions and have already been verified, as it can be seen in [Mermet et al. (2007); Mermet and Simon (2009)]. So, the following example deals with POs associated to a PDO, namely the SyncParAND operator of $R2$.

We have to recall that:

- The triggering context of agent $R2$ is $G(x_{R2}, y_{R2}) = dirty$ and its precondition is *true*;
- the context of the main goal of an agent is the conjunction of the triggering context and the precondition of this agent. So here, $C_{mg_2} \equiv (G(x_{R2}, y_{R2}) = dirty)$.
- the triggering context of the arm is $G(x_{R2}, u_{R2}) = dirty$;



(a) pGDT of type $R1$ (b) pGDT of the motor (c) pGDT of the tracker

Figure 3 pGDTs of $R1$ and its sub-agents

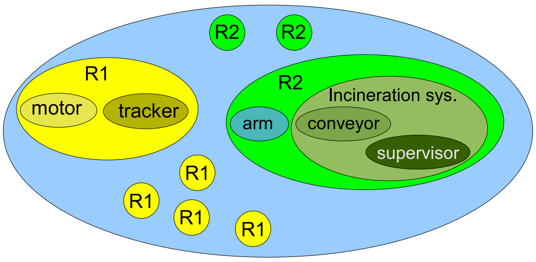


Figure 4 Stucture of the robots

- the triggering context of the conveyor belt is *true*.

The proof schema of the SyncParAND PDO leads to generate automatically the following proof obligations:

$$G(x_{R2}, y_{R2}) = dirty \rightarrow (G(x_{R2}, y_{R2}) = dirty \wedge true) \quad (1)$$

$$G_0(x_{R2}, y_{R2}) = dirty \wedge \left\{ \begin{array}{l} G_1(x_{R2}, y_{R2}) = clean \wedge \neg busy_1 \\ (T_1 < 3 \vee status_1 = down) \\ busy_2 = busy_1 \wedge G_2(x_{R2}, y_{R2}) = G_1(x_{R2}, y_{R2}) \\ status_2 = status_1 \wedge T_2 = T_1 \end{array} \right\} \quad (2)$$
$$\rightarrow \left(status_2 = up \rightarrow \left\{ \begin{array}{l} G_2(x_{R2}, y_{R2}) = clean \\ T_2 < 10 \end{array} \right\} \right)$$

Both formulae have been proven automatically by the theorem prover krt [Digilog (1997)]. The first one is obvious. The second one is true for the following reasons:

- it is obvious for $status_2 = down$. So it must just be verified for $status_2 = up$, which means that $status_1$ is up also (as $status_2 = status_1$).

- if $status_1 = up$, then $T_1 < 3$ and, as $T_2 = T_1$, $T_2 < 3$. So, $T_2 < 10$.
- $G_2(x_{R2}, y_{R2}) = G_1(x_{R2}, y_{R2}) = clean$ by hypotheses.

References

Bordini, R., Fisher, M., Pardavila, C., and Wooldridge, M. (2003). Model-checking AgentSpeak. In *AAMAS-03*, Melbourne, Australia.

Digilog (1997). Atelier B, Guide de l'utilisateur v3.0. Technical report, Digilog.

Mermet, B. and Simon, G. (2009). GDT4MAS: an extension of the GDT model to specify and to verify MultiAgent Systems. In *AAMAS*, pages 505–512.

Mermet, B., Simon, G., Saval, A., and Zanuttini, B. (2007). Specifying, verifying and implementing a MAS: A case study. In Dastani, M., Segrouchni, A. E. F., Ricci, A., and Winikoff, M., editors, *Post-Proc. 5th International Workshop on Programming Multi-Agent Systems (ProMAS'07)*, number 4908 in LNAI, pages 172–189. Springer.