A KNOWLEDGE-BASED COMPONENT FOR HUMAN-ROBOT TEAMWORK

Pedro Santana, Luís Correia
LabMAg, Computer Science Department, University of Lisbon, Portugal
m35353@alunos.di.fc.ul.pt, Luis.Correia@di.fc.ul.pt

Mário Salgueiro, Vasco Santos
R&D Division, IntRoSys, S.A., Portugal
mjs@introsys.eu, vps@introsys.eu

José Barata
UNINOVA, New University of Lisbon, Portugal
jab@uninova.pt

Keywords: Human-robot teamwork, multi-agent systems, cooperative workflows, knowledge-based systems, demining.

Abstract: Teams of humans and robots pose a new challenge to teamwork. This stems from the fact that robots and humans have significantly different perceptual, reasoning, communication and actuation capabilities. This paper contributes to solving this problem by proposing a knowledge-based multi-agent system to support design and execution of stereotyped (i.e. recurring) human-robot teamwork. The cooperative workflow formalism has been selected to specify team plans, and adapted to allow activities to share structured data, in a frequent basis, while executing. This novel functionality enables tightly coupled interactions among team members. Rather than focusing on automatic teamwork planning, this paper proposes a complementary and intuitive knowledge-based solution for fast deployment and adaptation of small scale human-robot teams. In addition, the system has been designed in order to provide information about the mission status, contributing this way to the human overall mission awareness problem. A set of empirical results obtained from simulated and real missions demonstrates the capabilities of the system.

1 INTRODUCTION

As highly appealing the idea of humans and robots enrolling in teamwork might seem, their significantly different perceptual, reasoning, and actuation capabilities make the task a daunting one. Typically, human-robot teamwork (Tambe, 1997; Scerri et al., 2002; Sierhuis et al., 2005; Nourbakhsh et al., 2005; Sycara and Sukthankar, 2006) solutions grow from work on multi-robot and multi-agent systems adapted to include humans. This paper proposes to see the problem from the other end, i.e. to include robots as participants on human-centred operational procedures, supported by knowledge management concepts usual in human organisations. Both views are complementary rather than mutually exclusive.

Human teamwork operational procedures are typically knowledge intensive tasks (Schreiber et al., 2000), which can be approximately represented by a set of templates. Knowledge engineering methodologies can be used to grasp and formalise domain experts' knowledge into the form of templates. These templates specify stereotyped (i.e. recurring) team plans, which need to be adapted to the situation at hand. Here, much of the work on automatic teamwork (re)planning (Sycara and Sukthankar, 2006) can be employed. Nonetheless, visual interfaces through which humans can manually adapt the plan are paramount. See for instance that most of the clues to detect and solve exceptions to a plan in complex situations are not observable without complex tacit human knowledge, which is continuously evolving as the mission unfolds. Bearing this in mind, a formalism that directly maps the plan and its visual representation is essential. In addition to meet this requirement, workflows also have the advantage of being common for the representation of activities in human organisations, thus providing a natural integration of robots in human knowledge intensive tasks.

Another benefit of considering a knowledge-based component for human-robot teamwork is that actual human-robot teams encompass few elements, meaning that the cost of having one human taking care of major strategic decisions for the entire team is possibly less than having the system taking uninformed strategic decisions. In addition, providing humans with a visual description of the mission state is essential to improve their *overall mission awareness* (refer to (Drury et al., 2003) for a thorough study on the awareness topic in the human-robot interaction domain). The need for this improvement is suggested by the limitations of common map-centric and video-centric interfaces on fostering mission awareness (Drury et al., 2007).

The paper is organised as follows: Section 2 presents the knowledge-based concepts of the proposed approach, whereas Section 3 describes the multi-agent system for human-robot teamwork. In Section 4, a case study is described, and a set of empirical results, obtained from both simulated and real experiments, are discussed. Finally, conclusions and pointers to future work are given in Section 5.

2 KNOWLEDGE-BASED APPROACH

Under the assumption of knowledge-based human-robot teamwork, domain knowledge must be acquired, formalised, adapted, and employed for the coordination of team members performing a mission. Such knowledge is mainly composed of *mission templates* specified by a domain expert in terms of workflows. Mission templates are then adapted and instantiated by the mission coordinator (a human) to the actual team on field, in order to build an operational team plan. That is, *physical entities* (i.e. robots and humans) are assigned to *participants* in the mission template. The proposed approach is composed of four major steps, namely:

Mission Template Specification. Mission templates are knowledge intensive tasks specifications, i.e. domain knowledge, maintained in a knowledge base supported by a well specified ontology. Mission templates are non operational team plans, in the sense that: (1) on field adaptations to the template are expected; and (2) no knowledge of which physical entity (i.e. human or robot) will play the role of a given participant is known beforehand.

Mission Template Adaptation. On field, the mission coordinator selects and adapts templates according to the environment and work to be done by the team. An example of an adaptation is the addition of a new activity to deal with a specific exception. In the process, some of the adapted mission templates are added to the mission templates library for reuse.

Mission Template Instantiation. In this step the mission coordinator instantiates the adapted mission template towards an operational team plan, by recruiting physical entities to the team.

Team Plan Execution. Finally, the operational team plan is distributed to each team member and executed.

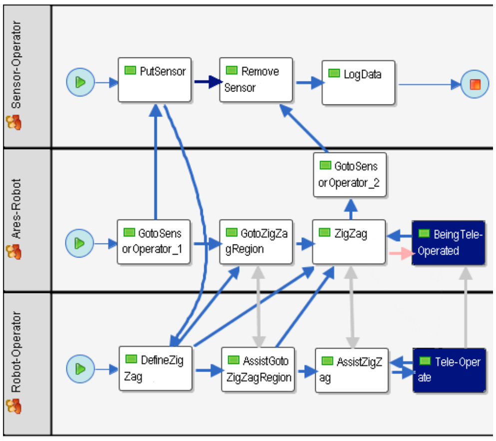


Figure 1: Partial view of the WFDM tool. Dark boxes represent active activities. Transitions, data-flow links, and exceptions are represented by blue, grey, and pink arrows, respectively. Each row corresponds to a team participant.

Mission specification, adaptation, instantiation, and monitoring are performed in the WorkFlow Design and Monitor (WFDM) tool (see Fig. 1). It was developed by the authors over the Together Workflow Editor (TWE) community edition[1] tool for workflow design applied to human organisations.

2.1 Data-Flow Links

Typically, workflows describe sequences of activities, which can exchange data at transition time. This limits their application in domains where activities must exchange data in a tightly coupled way, i.e. during their execution. To cope with this limitation, the concept of *cooperative workflows* has been proposed in (Godart et al., 2000). Although in cooperative workflows, activities can share data while executing, being mostly business management oriented, the exchanged data is performed sporadically and in the form of documents.

---

[1] TWE homepage: http://www.together.at/

The introduction of robots as team members adds new challenges to the execution of cooperative workflows. Robots require the use of structured data, i.e. with an explicit semantics. In addition, since many of the interactions have the purpose of allowing one participant (typically a human) to modulate the behaviour of another one (typically a robot), messages must be exchanged in a frequent basis. An example of a tightly coupled interaction is when a robotic team member is being teleoperated by a human team member.

Bearing this in mind, the cooperative workflow formalism is here extended with *data-flow links*, which allow activities to exchange structured messages, i.e. according to the ontology, in an asynchronous and frequent basis.

In addition to task-dependent interactions, subordination relationships also play a relevant role in human-robot teamwork. For this purpose, some activity parameters are defined at system level. For example, one team member (typically human) must be able to terminate another team member's (typically robotic) activity. This termination order is sent through a data-flow link. This feature can be seen in Fig. 1, in which the robot is teleoperated only while the human operator considers necessary.

3 MULTI-AGENT SYSTEM FOR TEAMWORK

This section describes the multi-agent system supporting the creation, adaptation, instantiation, and finally, execution of team plans. The system is built over the Java-based multi-agent platform JADE (Bellifemine et al., 1999), which provides two main facilities, namely: a yellow pages service for agent registration and lookup, plus an inter-agent messaging infrastructure.

Fig. 2 illustrates the major components of the multi-agent system. The coordinator represents human operators/experts responsible for formalising, adapting, instantiating, and monitoring a mission. The robotic and human participants represent robots and humans, respectively, involved in the mission's execution.

The explicit separation between human and robotic participant is essential at all levels. First, when defining domain knowledge, it is important to know which concepts must be followed by a human readable description. At execution time, humans are very good in understanding the situation at hand, even in the presence of incomplete information. With experience, humans are also very good in understanding when something is not working properly, like suspecting that messages are being lost when information appears in an intermittent way. Unlike robots, humans are not fully dependent on system level mechanisms to handle these situations. Consequently, watchdog and handshaking mechanisms, among others, are important in tasks to be performed by robots.

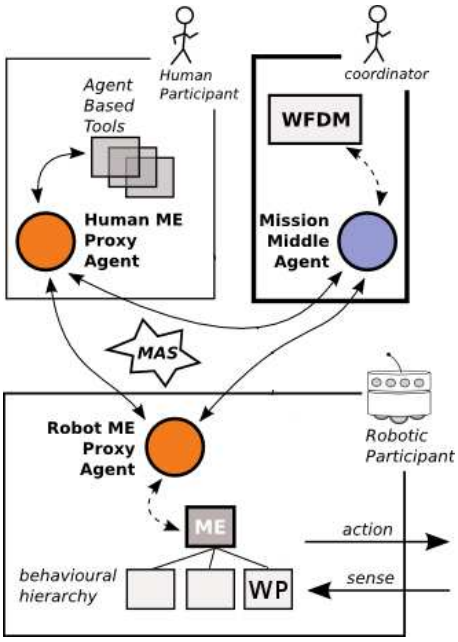
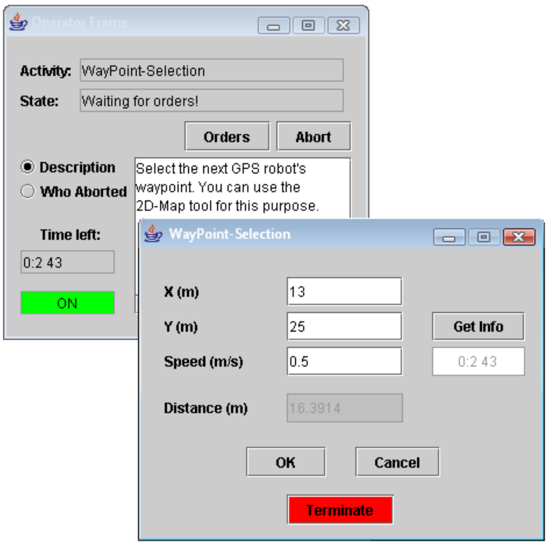


Figure 2: Multi-agent system for teamwork.

Understanding what activities are to be performed by humans and robots allows the system, for instance, to use the network in a parsimonious way. Handshaking mechanisms, which introduce network overhead, can be relaxed when performed in messages flowing from robots to humans. Considering humans and robots in such asymmetrical way, allows the system to exploit each one's specificities on its behalf.

3.1 Coordinator

By using the WFDM tool, the domain expert formalises knowledge, the coordinator adapts, instantiates, and monitors the execution of the mission. The WFDM tool interacts with the coordinator's proxy in the system, i.e. the *mission middle agent*, in order to provide the coordinator with a list of available physical entities able to play the role of each mission's participant. The coordinator is responsible for the final selection. Afterwards, the part of the plan corresponding to each participant is sent to its *Mission Execution (ME) proxy agent*. Finally the team initiation is done by the *mission middle agent*, by informing all *ME proxy agents* of the event.



Activity: WayPoint-Selection
State: Waiting for orders!
● Description
○ Who Aborted
Select the next GPS robot's waypoint. You can use the 2D-Map tool for this purpose.
Time left: 0:2 43
ON

X (m) 13
Y (m) 25
Speed (m/s) 0.5
0:2 43
Distance (m) 16.3914

Figure 3: Human participant graphical interface. The frame in the rear illustrates the main front-end containing the activity's description. The frame in the front is dynamically adapted according to the current activity.

During mission execution, each *ME proxy agent* informs the *mission middle agent* of its execution state (e.g. which activity is currently being executed). This information is then presented through the WFDM tool to the coordinator, reflecting the status of the mission. This symbolic information augments the coordinator's *mission awareness*.

3.2 Participants

Each *ME proxy agent* is composed of two main components: (1) the *Multi-Agent System Interaction Mechanism* (MAS-IM), and the (2) *physical entity interface*. The MAS-IM is the component that enables the agent to interact with other agents in the multi-agent community, as well as to exploit its middleware services (e.g. yellow pages service). In this work, this module is built over the JADE platform. The *mission middle agent* also aggregates a MAS-IM module for the same purposes. The *physical entity interface* is the module abstracting the physical entity, i.e. its control system in the robot case (e.g. the *behavioural hierarchy* in Fig. 2), and a set of graphical interfaces (see Fig. 3) in the human case.

3.3 Plan Execution

Let us start explaining the plan execution with a motivating example. At a given moment, the *human ME proxy agent* knows that its current activity is `waypoint-selection`. In this situation, its *physical entity interface* adapts the graphical interface as in Fig. 3, so that the human can fill in the next waypoint for the robot. Each time the operator updates this field, its *operator ME proxy agent* sends an inter-agent message to the *robot ME proxy agent*, currently executing the activity `goto-waypoint`. This message exchange has been specified in the team plan by means of a data-flow link. Then, through its *physical entity interface*, the *robot ME proxy agent* updates the robot's control system according to the incoming message, consequently modulating the *goto WayPoint (WP) behaviour*. The *WP behaviour* is implemented by a set of perception-action rules able to drive the robot towards the given waypoint. In addition, the human is also provided with a message suggesting the use of a 2-D visualisation (Santos et al., 2007) so as to enhance its *situation awareness*, before selecting the waypoint. This example highlights the main role of the *human ME proxy agent*: to provide *awareness*.

In detail, the plan execution proceeds as follows. As mentioned, each *ME proxy agent* receives from the *mission middle agent* the part of the team plan corresponding to the participant it is representing. Then, it executes its part of the plan according to the following algorithm:

1. Obtain participant's `start-activity`.
2. While the current activity is not terminated, update its input parameters with the contents of incoming, from other *ME proxy agents*, data-flow messages. In addition, *ME proxy agents* send data-flow messages to others alike, whose contents are the current activity's output parameters values.

   The aforementioned process of updating the activity's input parameters is done by sending a message to the participant's *physical entity interface*, which is able to interface directly with the entity's execution layer (e.g. robot's control system). In turn, the execution layer provides the *ME proxy agent* with the current values of the activity's output parameters, through the *physical entity interface*.

   The activity's termination event, along with its code (e.g. `not-ok-aborted`, `not-ok-time-out`, or `ok`), is provided to the *ME proxy agent* through the *physical entity interface*.
3. Being the current activity, *C*, terminated, the last obtained values of its output parameters are sent to the *ME proxy agents* of those participants that have active incoming transitions from *C*. These transitions become active if their associated conditions on the termination code of *C* are met. These messages are buffered in the receiving *ME*

proxy agents allowing subsequent asynchronous consumption. For further reference, these messages will be called `transition` messages.

4. Wait until one of the subsequent participant's activities becomes active. This activation occurs if all necessary, or one sufficient of its transitions, is active too. This is assessed by verifying if any of the received `transition` messages refers to the necessary and sufficient transitions.
5. The parameters encompassed in received `transition` messages are used to update the activity's input parameters. If more than one message (e.g. sent by activities in different participants) feeds the same input parameter, only one is selected according to a pre-specified – in the plan – priority. The actual update of the activity's input parameters is carried out as in step 2, i.e. through the *physical entity interface*.
6. Return to step 2 until the `end-activity` is reached.

To allow the coordinator to follow the mission unfolding, messages stating activities and transitions activation/deactivation events are sent to the *mission middle agent*, which in turn updates the WFDM tool.

4 CASE STUDY

In order to illustrate the proposed architecture, one case study has been selected: scanning a terrain with a scent sensor to detect minefields. The case study is defined as a high-level task involving one robot, and two humans, viz. one robot operator plus one sensor operator. The goal is to determine if a given terrain is a minefield. When the mission starts, the robot is equipped with a sensor able to determine the probability of the terrain to be a minefield. After analysing the terrain, the sensor is returned to the sensor operator, which is located in a safe location away of the potential minefield. The robot operator, also remote to the operations site, helps the robot whenever needed. Fig. 1 depicts how the team plan looks like in the graphical interface of the WFDM tool.

In this case study, the robot first moves towards the operator handling the scent sensor so as to get the sensor (`GotoSensorOperator_1`). After reaching the operator, `GotoSensorOperator_1` terminates activating `PutSensor`, in which the sensor operator equips the robot with the sensor. Then, the robot operator parameterises a zig-zag behaviour (i.e. a set of parallel lanes to be followed in a sequential manner) using the graphical interface of activity `DefineZigZag`. Afterwards, the robot moves in the direction of the defined zig-zag region (`GotoZigZagRegion`), while being modulated by the robot operator whenever necessary (`AssistGotoZigZagRegion`). The zig-zag specification is passed to the robot as a transition input parameter, whereas the `GotoZigZagRegion` modulating signal is passed through a data-flow link. As soon as the robot reaches the zig-zag region, the zig-zag behaviour is activated (`ZigZag`), which is also assisted by the robot operator (`AssistZigZag`). An example of assistance is "change to the next lane". If the robot departs too much from the lane being followed, caused for instance by the presence of a large obstacle, then `ZigZag` terminates with an exception. In response, the robot passes to teleoperation mode (`BeingTeleOperated`) and the current robot operator's activity is terminated.

Then, the robot operator is called to teleoperate the robot (`TeleOperate`). In this case, the `TeleOperate` provides `BeingTeleOperated` with teleoperation commands as data-flow messages. This corresponds to the mission state illustrated in Fig. 1. As soon as the operator considers the robot is again in a convenient position to resume its autonomous zig-zag behaviour, `TeleOperate` terminates, which in turn requests `BeingTeleOperated` to terminate as well. This is an example of a human activity terminating a robot activity by means of data-flow. Being again in autonomous zig-zag behaviour, the robot eventually reaches the end point of the zig-zag region and `ZigZag` terminates. Then, the robot moves towards the sensor operator (`GotoSensorOperator_2`), leaving the sensor there (`RemoveSensor`), whose data is logged (`LogData`). All `Goto*` activities are of the same type `GotoXY`.

4.1 Empirical Results

A set of simulated and field missions with the physical robot Ares for off-road environments (Santana et al., 2007; Santana et al., 2008), demonstrated the feasibility of the system. The information provided to the operator was enough for a proper awareness at each moment of the mission. The design of demining and surveillance missions, as complex as the one presented as case study, posed no major challenges to the user. However, for more complex tasks it was clear the need for workflow nesting capabilities. In terms of network load, the system showed to be sustainable, even for teleoperation cycles of 10 Hz. However, wireless communication temporary failures resulted in the loss of messages, resulting in system's performance degradation, and sporadic crashes.

5 CONCLUSIONS AND FUTURE WORK

To our knowledge, this paper contributes with a pioneering step towards exploitation of knowledge based techniques in human-robot teamwork. The goal was to enable cooperative execution of stereotyped tasks, essential in demanding scenarios, where timely decision making is required. The cooperative workflow formalism, usually employed for business oriented human organisations, was selected.

Clear distinctions on the way humans and robots interact required the workflow formalism to be adapted. Some adaptations were suggested, with particular focus on *data-flow links*. These links enable the implementation of tightly coupled coordination. This ability is usually disregarded in works of both theoretical teamwork and cooperative workflow fields, which typically focus on high-level tasks with sporadic interactions. Although multi-robots literature is more concerned with tightly coupled coordination, it lacks a structural approach to cope with the human factor. This paper presented a multi-agent system that explicitly considers the human. First, the workflow formalism is usually employed by humans and consequently natural to them. Second, by considering different message exchanging protocols and system level activity parameters, both human and robot asymmetries are explicitly taken into account. Third, human readable information is formally attached to the ontology concepts used by the human participant. As future work we expect to make use of nested workflows. In addition, the abstraction of human-robot sub-teams as work-flow participants will also be subject of analysis. Dynamic invocation of team sub-plans will be pursued as a way of applying well known stereotyped problem solvers (i.e. mission templates) to the situation at hand. Robustness against communication channels degradation must be further studied. Handshaking and message aging policies must be analysed, separately, for the human-robot and robot-robot interaction cases. A thorough analysis of non-expert user friendliness is still missing.

ACKNOWLEDGEMENTS

We thank Paulo Santos and Carlos Cândido for proofreading. The work was partially supported by FCT/MCTES grant No. SFRH/BD/27305/2006.

REFERENCES

Bellifemine, F., Poggi, A., and Rimassa, G. (1999). JADE – a FIPA-compliant agent framework. In *Proc. of PAAM'99*, pages 97–108, London.

Drury, J., Keyes, B., and Yanco, H. (2007). LASSOing HRI: Analyzing Situation Awareness in Map-Centric and Video-Centric Interfaces. In *Proc. of the HRI' 07*, pages 279–286, Arlington, Virginia, USA.

Drury, J., Scholtz, J., and Yanco, H. (2003). Awareness in human-robot interactions. In *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics*.

Godart, C., Charoy, F., Perrin, O., and Skaf-Molli, H. (2000). Cooperative workflows to coordinate asynchronous cooperative applications in a simple way. In *Proc. of the 7th Int. Conf. on Parallel and Distributed Systems*, pages 409–416.

Nourbakhsh, I., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burion, S. (2005). Human-Robot Teaming for Search and Rescue. *Pervasive Computing, IEEE*, 4(1):72–78.

Santana, P., Barata, J., and Correia, L. (2007). Sustainable robots for humanitarian demining. *Int. Journal of Advanced Robotics Systems (special issue on Robotics and Sensors for Humanitarian Demining)*, 4(2):207–218.

Santana, P., Cândido, C., Santos, P., Almeida, L., Correia, L., and Barata, J. (2008). The ares robot: case study of an affordable service robot. In *Proc. of the European Robotics Symposium 2008 (EUROS'08)*, Prague.

Santos, V., Cândido, C., Santana, P., Correia, L., and Barata, J. (2007). Developments on a system for human-robot teams. In *Proceedings of the 7th Conf. on Mobile Robots and Competitions*, Paderne, Portugal.

Scerri, P., Pynadath, D., and Tambe, M. (2002). Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17:171–228.

Schreiber, G., Akkermans, H., Anjewierden, A., deHoog, R., Shadbolt, N., VandeVelde, W., and Wielinga, B. (2000). *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press.

Sierhuis, M., Clancey, W., Alena, R., Berrios, D., Shum, S., Dowding, J., Graham, J., Hoof, R., Kaskiris, C., Rupert, S., and Tyree, K. (2005). NASA's Mobile Agents Architecture: A Multi-Agent Workflow and Communication System for Planetary Exploration, i-SAIRAS 2005. *München, Germany*.

Sycara, K. and Sukthankar, G. (2006). Literature review of teamwork models. Technical Report CMU-RI-TR-06-50, Robotics Institute, Carnegie Mellon University.

Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence*, 7:83–124.