Proceedings of the 2000 IEEE International Workshop
on Robot and Human Interactive Communication
Osaka, Japan - September 27-29 2000

Human-Robot Interface Using Agents Communicating in an XML-Based Markup Language

Maxim Makatchev

Dept. of Manufacturing Engineering and
Engineering Management
City University of Hong Kong
Hong Kong, China

S. K. Tso

Centre for Intelligent Design,
Automation and Manufacturing
City University of Hong Kong
Hong Kong, China

Abstract

This paper concerns an agent-based human-robot interface via the Internet. A user client and an embedded software are viewed as agents with limited computational and communication resources. To facilitate the communication between the real-time embedded agents and the user interface agents via the communication channel of uncertain quality the proxy agent is proposed as a mediator. The functions assigned to the proxy agent target reduction of inter-agent communication load and minimization of computational resources taken by the embedded agents and user interface agents for communication-related tasks. An XML-based language, RoboML, is designed to serve as a common language for robot programming, agent communication and knowledge representation. The human-robot interface software prototype is developed for an autonomous guided vehicle to evaluate the proposed techniques.

1 Introduction

As applications of robots have been extended to such areas as service and entertainment leading to a gradually increasing number of varieties of robot designs and interfaces, the lack of open standards for robot hardware and software implies increasing costs of robot production and decreasing quality of end-user experience. Indeed, the variety and complexity of different hardware and software interfaces in modern robotics may account for some of the disappointments on the pace of adoption of robots into human environments.

An open architecture to provide a common interface at the application programming and hardware levels has been proposed (OpenR [5]). We recognize a need for similar initiatives in the area of human-robot interface (HRI). The wide accessibility and relative simplicity of many Internet-based interface technologies may provide the power that will bridge the gap between humans and robots. This paper is concerned with two issues of HRI via the Internet: an agent-based architecture and a common markup language for robot programming, agent communication and knowledge representation.

1.1 Human-robot Interfaces via the Internet

Starting from the early works on Internet-based HRI there is a number of successful implementations of teleoperation interfaces via the Internet (e. g. [21, 23]). Weak quality-of-service guarantees over the Internet however limit the applications of Internet-based telerobotics and suggest using more complex interface models, i. e. interface agents that control the robot at a higher level rather than through direct teleoperation, thus relaxing the time constraints imposed by the real-time robotic system. Despite the fact that there are agent applications on robots [11, 17] and agent-based interfaces [14, 18], few works combine agents embedded in robots and user interface agents in a single agent space [12, 22].

In this paper we propose the architecture that includes user interface agents (IAs) and embedded agents (EAs) communicating over the Internet. The proxy agent, with functions close to those of *communication facilitators* [7], is utilized to reduce inter-agent communication load and computational resources taken by the EAs for interface-related tasks. The proxy agent dynamically creates and removes links to IAs and EAs, builds and supports representations of the domain ontologies related to the respective IAs and EAs by means of XML Schemas [25], performs translation of inter-agent communication, provides data journaling to facilitate asynchronous bidirectional communication between IAs and EAs.

1.2 Robot and Agent Languages

A number of task-level programming languages has been developed for robots: TDL (Task Definition Language) [20]; the human-oriented robotic programming language CURL (Cambridge University Robot Language) [10]; reactive robot control languages RPL (Reactive Plan Language) [16], RAP (Reactive Action Packages) [4], ESL (Executive Support Language) [6]; the logic-based action language GOLOG [15]. The need for simplification of robot programming have led to a

0-7803-6273-X/00/$10.00©2000 IEEE

number of dedicated national projects, e. g. in Japan [1] and in Germany [13]. One way to simplify programming is through utilization of graphical interfaces, e. g. Roboglyph [10], Robot Programming Simplification Project [1], Onika [8]). An alternative (and complimentary) strategy is to simplify the underlying data representation so that its interpretation and editing with the help of relatively simple software or directly by a human is possible. In this paper we follow the latter approach.

There have appeared several proposed common languages for agent communication and knowledge representation. The agent communication languages include KQML (Knowledge Query Manipulation Language) [2], FIPA (Foundation for Intelligent Physical Agents) [3], AOP (Agent-Oriented Programming) [19] and Telescript [24]. The first order predicate calculus-based language for expressing the content of a knowledge-base is the KIF (Knowledge Interchange Format) [9].

While these languages are powerful and convenient for expressing the information related to their respective applications, a simultaneous utilization of a robot programming language, agent communication language and knowledge representation language in a single application, as it would be necessary for an agent-based HRI, creates significant difficulty for a human operator.

Our goal is to propose a language for the purposes of Internet-based HRI applications that would unify the languages of the three categories considered above while being transparent for a human with minimal training. Namely, the language should:

- be powerful enough to express everything that can be expressed by the mentioned languages for robotic programming, agent communication and knowledge representation;
- allow its manipulation via relatively simple software or directly by a human; and
- be suitable for cross-platform applications.

In order to satisfy these requirements we utilize Extensible Markup Language (XML) [25] to develop the specification of RoboML, a markup language for robotic applications.

The rest of the paper is organized as follows. Section 2 describes the agent-based HRI architecture. RoboML is presented in section 3. The proposed HRI model and RoboML are evaluated on a HRI software prototype as shown in section 4. Section 5 gives our conclusion remarks.

2 Proxy-Mediated Human-Robot Interface Architecture

The idea of utilization of facilitator agents to mediate the communication between agents of different domains is a well known agent architecture, e. g. [7]. The applications of HRI, however, impose some specific constraints, i. e. the embedded agent runs in real time and has limited computational resources. To provide the true Internet-based HRI it is also useful to avoid extra assumptions about the computational resources available for the user interface agent. Thus not only the information should be preprocessed to reduce communication load, but as much of the communication-related processing as possible should be performed outside the EA/IA host computers. On the other hand, it is often the case in the Internet-based HRI that there is a dedicated computer to host internet services (*httpd* et al.) with weaker limitations on computational resources compared to EA/IA hosts. In this case it can be advantageous to use this computer for hosting of a facilitator agent which would provide some additional services to ease the computational and communication load of the other agents. In this paper we call such agent *proxy*.

The proxy agent in our model supports a knowledge-base to facilitate the asynchronous communication between IAs and EAs. To reduce the inter-agent communication load and to provide a set of data access levels it supports data schemas announcing the data available for subscription and the currently subscribed data for input and output data streams of each IA/EA. These schemas can be viewed as representation of subsets of the domain ontologies corresponding to each of the IAs/EAs. Thus four data schemas for each of the IAs/EAs are hosted by the proxy agent: available data schema (ADS) and subscribed data schema (SDS) for to-agent and from-agent data streams (see fig. 1).

The possible scenario of the communication using the proxy agent is as follows. The IAs and EAs send queries and data to the proxy agent in their own domain ontologies. The query can be expressed in the **get** or **subscribe** performatives of RoboML, and the generic data transmission is done under the **set** performative. Query of the type **subscribe** from agent A updates the to-agent SDS corresponding to agent A and from-agent SDS corresponding to agent B, recruited to provide the necessary information. Query of the type **get** from agent A may not cause permanent or temporary changes in its to-agent SDS but the request should be announced in the from-agent SDS of agent B, chosen to respond to the query. A message with the **set** performative from agent A may update its from-agent ADS and the content of the message can be stored in the proxy agent knowledge-base for future use. It may seem to be natural, at least for many applications, to have SDS as a subset of the corresponding ADS. However, we would not like to specify such restriction in the model, allowing a wider range of interaction scenarios. It may be useful, for example, for a user to subscribe to information in advance, thus modifying to-IA SDS even if the information requested is not yet available in the respective to-IA ADS.

Linking of ADS and SDS is represented by the sign

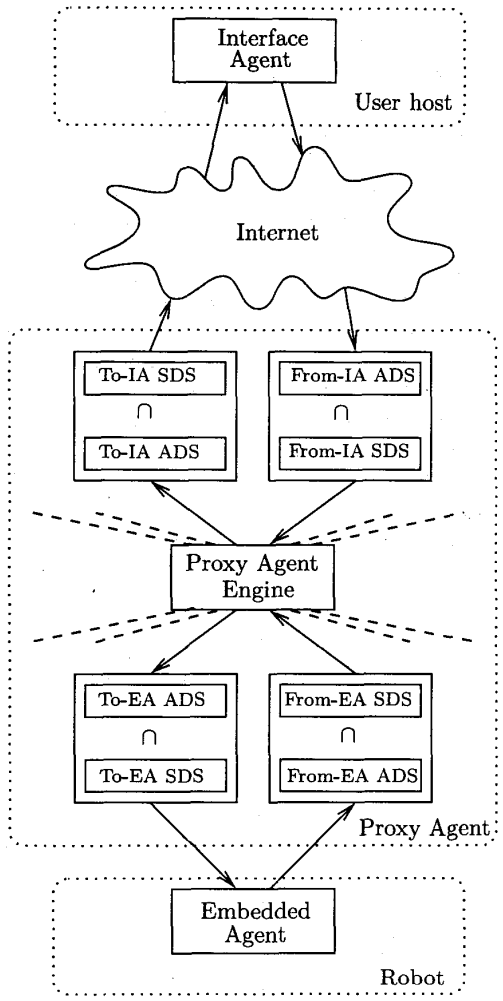

Figure 1: Architecture of the proxy-mediated human-robot interface.

of intersection of sets ∩ in the figure 1 for both to-agent and from-agent streams, assuming that in most cases the data stream is filtered to satisfy both SDS and ADS. However this may not always be true for the from-agent streams, as the proxy agent may choose not to ignore unsubscribed data but to store it for future reference. The mechanisms of advance subscription and of storage of unrequested data can effectively allow asynchronous interaction between IAs and EAs, accounting for the real-time constraints imposed on EAs (and, possibly, IAs).

The syntax of available data schemas and the subscribed data schemas is based upon XML Schemas specification [25] and RoboML. The sample schemas that are created by an experimental HRI application are described in section 4.

3 RoboML

Our goal is to develop a single Internet-oriented language to account for the major needs of agent-based HRI. We have chosen to use XML as a base for the specification of RoboML, the markup language for robotics. In view of the requirements outlined in section 1.2 the choice to build the language using XML is based on the following considerations:

- *Suitability to express what can be expressed by the known languages for robotic programming, agent communication and knowledge representation.* XML has proven to be convenient to describe various type of structured data, as demonstrated by a growing number of XML-based languages for a wide range of domains (see links in [25]). The availability of the specifications of XML-based languages for many engineering and scientific domains makes it possible to utilize their ontologies and syntactic models and to reduce the work on the design of a new language.

- *Convenience for manipulating by a human by means of simple software or directly.* Despite the fact that XML allows to specify languages with high complexity, understanding the concepts of domain-oriented implementations is usually within the capability of non-experts. For example, HTML (that have many commonalities with XML-based languages) or its editing software can be generally used by school students. This degree of human-friendliness of the language is what we aim to achieve with RoboML. XML-compliant code is easy to parse and generate by a software. The ability to manipulate the language code with the help of relatively simple software and its transparency for the user can be particularly important for service and entertainment robotics. A native support of XML is becoming a de facto standard for web-browsers.

- *Suitability for cross-platform applications.* As a consequence of the adoption of XML as a standard for the WWW and its support by browsers, it is effectively cross-platform. Parsers, translators, browser plug-ins and other types of software components for XML are available (often free of charge) for many applications and popular computer platforms.

In the rest of this section we shall outline the main concepts of RoboML. For the sake of clarity we omit the language constructs that are not directly related to the vocabulary of RoboML (e. g. the XML document type declarations).

3.1 Agent Communication

We utilize the KQML model to define the semantics of agent communication elements of RoboML. The

following performatives are introduced as elements of RoboML: **set** as analogous to *tell*, **get** as analogous to *ask*, **subscribe** as analogous to *subscribe* of KQML. Each of these elements can have optional attributes **sender** and **receiver** (corresponding to the communication layer of KQML model) and **ontology** (corresponding to the message layer of KQML model), e. g. a RoboML request can be wrapped into tags of the **get** element as:

```
<get sender="MyInterface" receiver="AGV1"
ontology="UserInterface">
 ...
</get>
```

3.2 Knowledge Representation

To express the content of a RoboML message the following alternatives are considered: (a) the content is already expressed in a language which has an XML-based analog, and (b) there is no known XML-based representation of the content. In the former case the known XML-based language can be readily adopted to express the content of the message. To account for the latter case and for the case when the utilization of the vocabulary of another XML-based language may be undesirable, we propose to use a few basic elements for description of the information of robot-related domains.

As noted in [5], the robot's hardware usually can be represented as a semantic tree according to the physical and logical interconnection of the units. Software components and user interfaces can be locally represented by tree structures as well. This representation is particularly suitable for the nested markup structure of XML-based languages. This fact is exploited through utilization of the *container* elements.

RoboML uses container elements to represent basic compound objects within a domain. For example, the following container elements are defined for the *Hardware* ontology: **robot**, **wheel**, **motor**, **sensor**, **controller**, and so forth. These container elements can be further specified by the the **name** and the **ontology** attributes. The **name** is provided to uniquely identify the element within the respective layer of the nesting. Note that a container element may have its own ontology, which makes it possible to describe data of various domains in a single RoboML document. A container element can have a number of children elements, in accordance with the domain ontology.

The actual data should appear within the *token* elements, or, following the XML Schema terminology, elements that have *simple types*, i. e. those that cannot contain any subelements or attributes. The tokens specified for *Hardware* ontology of RoboML are: **position**, **velocity**, **acceleration**, the generic token **value**, and so forth. Other tokens may be adopted from the respective XML-based languages when other domain ontologies are utilized. A RoboML message from the *AGV1* embedded agent to the *MyInterface* user interface agent specifying the positions of steering motors of wheels 1 and 2 may look like:

```
<set sender="AGV1" receiver="MyInterface"
ontology="Hardware">
  <robot name="AGV1">

    <wheel name="1">
      <motor name="steering motor">
        <position>2577</position>
      </motor>
    </wheel>

    <wheel name="2">
      <motor name="steering motor">
        <position>-754</position>
      </motor>
    </wheel>

  </robot>
</set>
```

3.3 Robot Programming

Well developed formalism of programming languages, including those for robot programming, makes their translation into an XML-based markup language straightforward. Although this is unlikely to produce any advantage for direct manipulation of the code by a human, it allows reuse of parsing, and possibly of other components of the respective agent software by the language interpreter/editor. Simple programming constructs can then be expressed in RoboML between the `<program>` and `</program>` tags in the way analogous to using the *progn* element in KIF [9].

4 Human-Robot Interface Application

The proposed HRI architecture and RoboML have been evaluated with the software HRI prototype consisting of (a) the user interface agent IA, (b) the proxy agent (in the current version both hosted by an off-board Macintosh computer), and (c) the embedded agent EA running on Harmony RTOS (hosted by a Motorola MVME controller that is embedded in an autonomous guided vehicle). At the expense of some extra communication traffic between the EA and the proxy agent, the EA is relieved of any communication processing tasks (i. e. the EA ignores its to-agent ADS and from-agent SDS), except for periodical report/query generation and RoboML parsing.

The communication using the *Hardware* and *UserInterface* ontologies has been implemented. To demonstrate the use of XML Schemas for implementation of ADS and SDS we show a sample from-EA ADS generated during the operation of the HRI. The from-EA ADS represents the information available in the from-stream of the EA agent. In this case, it refers to the

positions of the steering motors and velocities of the driving motors corresponding to the four wheels of the *AGV1* robot (as before, we omit the XML document type declaration, the standard XML Schema wrapping and annotations):

```
<xsd:element name="set" type="SetPerformative"/>

<xsd:complexType name="SetPerformative">
  <xsd:element name="robot" type="Robot"/>
  <xsd:attribute name="sender" type="xsd:string"/>
  <xsd:attribute name="receiver" type="xsd:string"/>
  <xsd:attribute name="ontology" type="xsd:string"
  fixed="Hardware"/>
</xsd:complexType>

<xsd:complexType name="Robot">
  <xsd:element name="wheel" type="Wheel"
  minOccurs="4" maxOccurs="4"/>
  <xsd:attribute name="name" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="Wheel">
  <xsd:element name="motor" type="SteeringMotor"
  minOccurs="1" maxOccurs="1"/>
  <xsd:element name="motor" type="DrivingMotor"
  minOccurs="1" maxOccurs="1"/>
  <xsd:attribute name="name" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="SteeringMotor">
  <xsd:element name="position" type="xsd:decimal"
  minOccurs="1" maxOccurs="1"/>
</xsd:complexType>

<xsd:complexType name="DrivingMotor">
  <xsd:element name="velocity" type="xsd:decimal"
  minOccurs="1" maxOccurs="1"/>
</xsd:complexType>
```

Assume that the IA requests the information related to the top view of the vehicle. The subscription could be expressed in the *UserInterface* domain ontology by the respective to-IA SDS. The from-EA SDS, corresponding to this request, would then be a translation of the to-IA SDS into *Hardware* domain ontology, e. g., it would subscribe for the steering motor positions and would ignore the driving motor data. In this case, the from-EA SDS would look essentially like the from-EA ADS above, except for the definition of the *Wheel* complex type lacking the declaration of the **motor** element of the *DrivingMotor* type (or, equivalently, having in this declaration `minOccurs="0" maxOccurs="0"`).

5 Conclusions

In order to use Internet as a communication medium for the human-robot interface, it is required to address a number of specific problems: communication between real-time systems via the channel with low quality-of-service guarantees, limited or uncertain computational resources available for the interface client and the embedded software, compliance with the Internet standards, and compatibility with available software.

This paper describes the approach to solve these problems by the utilization of the agent-based HRI architecture and the common XML-based language for HRI applications.

The proxy-mediated HRI model aims at reducing the communication and computational load of embedded and user interface agents and provides means for asynchronous data exchange, accounting for the real-time nature of the embedded agent and, possibly, of the user interface agent. Available data schema and subscribed data schema, expressed in XML Schema definition language, are supported by the proxy agent for each of the data streams to facilitate cross-agent translation and to reduce inter-agent traffic.

The common Internet-oriented language for agent communication, knowledge representation and robot programming is proposed. The XML-based markup language for robotic applications, RoboML, is intended to be used in conjunction with other Internet standards, including other XML-based languages for specific domains.

The proposed interface architecture and the language are implemented in the HRI for the autonomous guided vehicle prototype. The *Hardware* and *UserInterface* ontologies are developed for the experimental HRI configuration.

While the proposed methodologies are being successfully tested, we recognize that much more work needs to be done to refine the formal specifications of the proxy-mediated HRI architecture and RoboML. This work is closely related to the development of the parent Internet standards of XML-based languages and protocols, as well as to the progress in HRI, agent communication, knowledge representation, and robot programming. Possible further directions of work on RoboML include (a) specification of ontologies for robot control architectures, multimodal user interfaces, and robot programming, and (b) development of the RoboML application support via applets and browser plug-ins.

Acknowledgments

Special thanks to Lorenza Yeung for data on the proficiency of Hong Kong school students in HTML and the HTML-editing software.

This work is supported by an RGC grant #9040224 in Hong Kong.

References

[1] T. Arai, T. Itoko, H. Yago, "A Graphical Robot Language Developed in Japan," *Robotica,* vol. 15, pp. 99–103, 1997.

[2] T. Finin, Y. Labrou, J. Mayfield, "KQML as an Agent Communication Language," in *Software Agents*, ed. J. M. Bradshaw, AAAI Press/The MIT Press, pp. 291–316, 1997.

[3] http://www.fipa.org.

[4] J. Firby, "An Investigation into Reactive Planning in Complex Domains," *Proc. of AAAI-87,* Seattle, WA, 1987, pp. 202–206.

[5] M. Fujita, K. Kageyama, "An Open Architecture for Robot Entertainment," *Proc. of Int. Conf. on Autonomous Agents*, 1997, California, USA, pp. 435–442.

[6] E. Gat, "ESL: A Language for Supporting Robust Plan Execution in Embedded Autonomous Agents," *AAAI Fall Symposium: Issues in Plan Execution,* Cambridge, MA, 1996.

[7] M. R. Genesereth, S. P. Ketchpel, "Software Agents," *Communications of the ACM,* vol. 37, no. 7, pp. 48–53 and 147, 1994.

[8] M. W. Gertz, D. B. Stewart, P. K. Khosla, "A Software Architecture-Based Human-Machine Interface for Reconfigurable Sensor-Based Control Systems," *Proc. of the IEEE Int. Symposium on Intelligent Control,* Chicago, Illinois, USA, 1993, pp. 75–80.

[9] M. Ginsberg, "Knowledge Interchange Format: The KIF of Death," *AI Magazine,* vol. 12, no. 3, pp. 57–63, 1991.

[10] W. S. Harwin, R. G. Gosine, Z. Kazi, D. S. Lees, J. L. Dallaway, "A Comparison of Rehabilitation Robotics Languages and Software," *Robotica,* vol. 15, pp. 133–151, 1997.

[11] H. Hiraishi, H. Ohwada, F. Mizoguchi, "Web-Based Communication and Control for Multiagent Robots," *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS'98),* Victoria, B. C., Canada, 1998, pp. 120–125.

[12] Y. Iwakura, Y. Shiraishi, Y. Nakauchi, Y. Anzai, "Real-world Oriented Distributed Human-Robot Interface System," *Proc. of IEEE Int. Workshop on Robot and Human Communication,* 1997, pp. 188–193.

[13] M. Kampker, "New Ways of User-Oriented Robot Programming," *Proc. of the 24th Annual Conference of the IEEE, IECON'98,* 1998, vol. 4, pp. 1936–1940.

[14] B. Lenzmann, I. Wachsmuth, "Contract-Net-Based Learning in a User-Adaptive Interface Agency," in *Distributed Artificial Intelligence Meets Machine Learning,* ed. G. Weiß, Lecture Notes in Artificial Intelligence, vol. 1221, pp. 202–222.

[15] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, R. Scherl, "GOLOG: A Logic Programming Language for Dynamic Domains," *Journal of Logic Programming,* vol. 31, pp. 59–84, 1997.

[16] D. McDermott, "A Reactive Plan Language," Research Report YALEU/DCS/RR-864, Yale University, 1991.

[17] T. Ohko, K. Hiraki, Y. Anzai, "Addressee Learning and Message Interception for Communication Load Reduction in Multiple Robot Environments," in *Distributed Artificial Intelligence Meets Machine Learning,* ed. G. Weiß, Lecture Notes in Artificial Intelligence, vol. 1221, pp. 242–258.

[18] T. Sawaragi, O. Katai, "Resource-Bounded Reasoning for Interface Agent for Realizing Flexible Human-Machine Collaboration," *Proc. of IEEE Int. Workshop on Robot and Human Communication,* 1997, pp. 484–489.

[19] Y. Shoham, "Agent-Oriented Programming," *Artificial Intelligence,* vol. 60, no. 1, pp. 51–92, 1993.

[20] R. Simmons, D. Apfelbaum, "A Task Description Language for Robot Control," *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS'98),* Victoria, B. C., Canada, 1998, vol. 3, pp. 1931–1937.

[21] T. Suzuki, T. Fujii, K. Yokota, H. Asama, H. Kaetsu, I. Endo, "Teleoperation of Multiple Robots through the Internet," *Proc. of IEEE Int. Workshop on Robot and Human Communication,* 1996, pp. 84–89.

[22] T. Suzuki, K. Yokota, H. Asama, H. Kaetsu, I. Endo, "Cooperation between the Human Operator and the Multi-Agent Robotic System: Evaluation of Agent Monitoring Methods for the Human Interface System," *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS'95),* 1995, pp. 206–211.

[23] K. Taylor, B. Dalton, J. Trevelyan, "Web-Based Telerobotics," *Robotica,* vol. 17, pp. 49–57, 1999.

[24] J. E. White, "Mobile Agents," in *Software Agents,* ed. J. M. Bradshaw, AAAI Press/The MIT Press, pp. 437–472, 1997.

[25] Extensible Markup Language (XML). The World Wide Web Consortium, http://www.w3.org/XML.