On Distributed Knowledge Bases for Small-Batch Assembly

Stenmark, Maj; Malec, Jacek; Nilsson, Klas; Robertsson, Anders

2013

On Distributed Knowledge Bases for Small-Batch Assembly

Maj Stenmark, Jacek Malec, Klas Nilsson
Dept. of Computer Science, Lund University, Sweden

Anders Robertsson
Dept. of Automatic Control, Lund University, Sweden

***Abstract*— This paper presents ongoing research involving design and evaluation of different architectures for providing knowledge-based solutions in industrial robotized automation systems. The conclusions are that distributed, cloud-based approaches offer many possibilities, in particular for knowledge exchange and reuse, and facilitate new business models for industrial solutions. However, there are many unresolved questions yet, e.g., those related to reliability, consistency, or legal responsibility. There is a definite need for further research and better infrastructure before this approach would become attractive for industrial actors.**

I. Introduction

Knowledge-based systems are large and are becoming even larger, outside the grasp of a single creator, programmer or maintainer. Usually they are built by large teams, maintained using semi-automatical computer-supported routines and updated fully automatically. The burden of manual care would be prohibitive. Their hardware configuration and physical location becomes to a large extent irrelevant; only availability and accessibility of their services play a major role. Therefore cloud-based solutions are highly relevant in this context.

Industrial robot systems are, to the contrary, usually a very localized endeavor, even though they are also highly complex and outside the grasp of a single person. However, the complexity of the manufacturing domain and of the automatized production solutions make the systems very often task- and location-specific. Even in the highly robotized automotive industry, each installation of a production line is different, even in case of the same car manufacturer and car model. At least this is the view of systems integrators, delivering each system to a specific end-user.

Cloud computing is normally understood as an infrastructure offering remote computing as a service to demanding clients. It may also involve distributed on-line resources, shared among interested clients. It becomes an opportunity in the robotized automation as it offers possibility of knowledge expansion and sharing among installations, knowledge transfer between different users and runs, better customer support from system integrators, simpler system installation and bootstrap, and new services based on creation and maintenance of specific knowledge-bases.

This paper is based on earlier experiences gathered during two finished EU-funded research projects; SIARAS and ROSETTA, and two on-going research projects; PRACE and SMErobotics. It is organized as follows. First, the SIARAS and ROSETTA solutions are presented, with comments about their relevance for the topic of the workshop. Then the PRACE and SMErobotics complementary ongoing efforts are presented, followed by a brief description of related work and conclusions. The pre-competitive and open nature of the research implies a focus on architectural aspects.

II. The SIARAS approach

From the AI perspective the main interest of the SIARAS (Skill-Based Inspection and Assembly for Reconfigurable Automation Systems) project[1] was knowledge-based automatic reconfiguration of automation systems. The results of this work have been presented in [1]. The outcome was an intelligent support system for reconfiguration and adaptation of robot-based manufacturing cells. Declarative knowledge was represented first of all in an ontology expressed in OWL, for a generic taxonomical reasoning, and in a number of special-purpose reasoning modules, specific for the application domain. The domain-dependent modules were organized in a blackboard-like architecture.

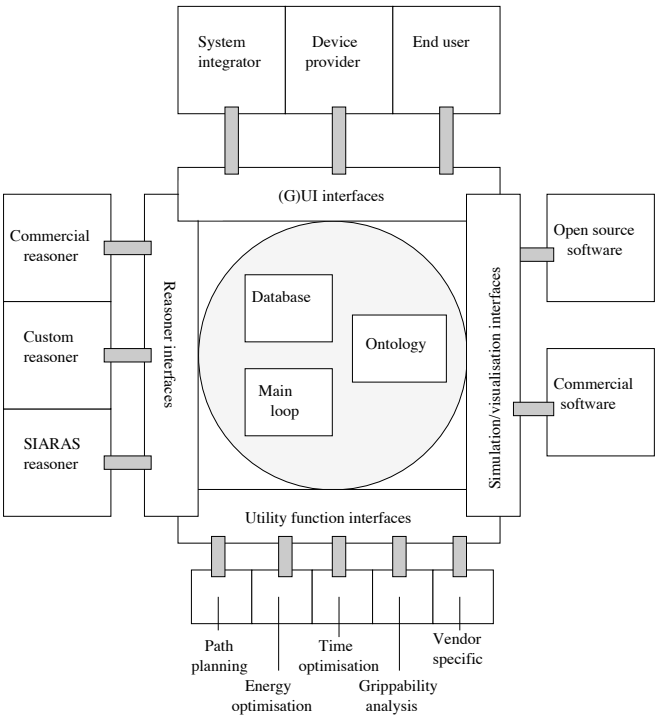

Fig. 1. The SIARAS blackboard architecture

[1] http://cordis.europa.eu/search/index.cfm?fuseaction=result.document&RS_RCN=12197834A

An overview of the adopted architectural solution is shown in Fig. 1. The main focus has been put on generic interfaces, allowing independent service, data or knowledge providers to adapt to the system expectations. In particular, some experiments have been made using several simulation/visualisation tools, providing independent user interfaces suited for different needs and, last but not least, exploiting external "utility functions" (knowledge sources in blackboard architecture terms) provided by robot manufacturers, sensor manufacturers and system integrators.

The SIARAS system demonstrator used several machines communication using Ethernet (simulation software required different operating systems), but only locally within the engineered system. Some further experiments were done with distributing device database (shown in the central module in the Figure), allowing several device manufacturers to provide their data independently of each other, using their own computer systems connected to Internet [2]. However, the ontology used was single and centralized, available locally on the SIARAS system.

III. The ROSETTA system

The ROSETTA project[2] (RObot control for Skilled ExecuTion of Tasks in natural interaction with humans; based on Autonomy, cumulative knowledge and learning) focused on simplifying interaction between the user and the robotized automation system, and on ensuring human safety in all circumstances. The former goal in particular demanded knowledge-based solutions, although the latter one also exploited some explicit knowledge encoded in a specific injury ontology.

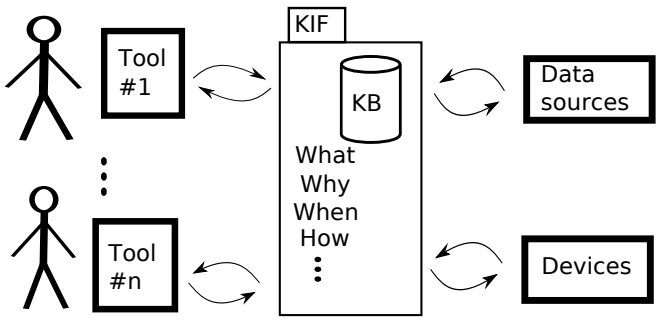

Fig. 2. The KIF idea

The main idea behind the ROSETTA solution is illustrated in Fig. 2 [3]. A central knowledge broker, called Knowledge Integration Framework (KIF[3]), is organizing access to knowledge and data sources, provides information about available devices and their capabilities, and serves a number of tools enabling human users of various kinds (factory floor operators, system integrators, device manufacturers, skill designers, system maintainers) to perform their tasks in the simplest possible way. The solution is generic in the sense that no particular data formats are mandated, only the tool APIs are specified in order to ensure interoperability.

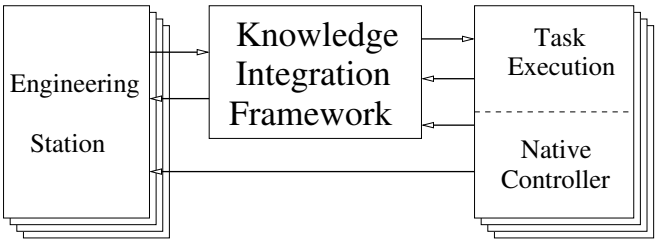

Fig. 3. The ROSETTA architecture

The concrete system built around this idea has been designed with assembly tasks as the main domain of application. This has limited the set of skills necessary to specify, kinds of sensors used as well as the end effectors that robots need for fulfilling their objectives and made creation of the test system possible. The architecture, depicted in Fig. 3, is an instantiation of the previous one, assuming a concrete simulation and visualization environment, here called Engineering Station, and concrete brands and models of robots for which the Task Execution system generates code, executable by the Native Controller. The architecture assumes a number, possibly geographically separated, engineering stations, and a number of independent robot system installations, connected to a common knowledge server.

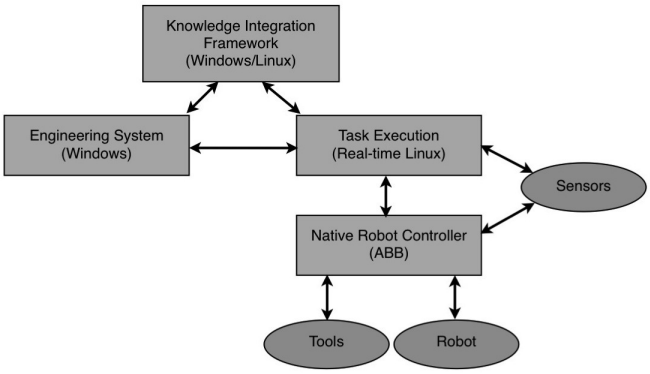

Fig. 4. A concrete ROSETTA system

During the ROSETTA project we have built KIF server and made it available for testing by all project partners. The server provides access to semantic storage with skill descriptions, task specifications and station (robot installation) definitions. It also contains a set of ontologies expressed in the OWL language. Besides, it provides a set of knowledge-based services like task consistency checks, rudimentary planning and scheduling, natural-language-based task definition, or process parameter learning. The engineering system has been realized as a plug-in to the ABB RobotStudio software, as we used ABB robots in our experiments.

The KIF concept can also be used in a hierarchical way with local servers. By setting up a local server in a factory or a lab, it is possible to address some of the problems of

[2] http://www.fp7rosetta.org

[3] We are aware of the acronym conflict with Knowledge Interchange Formalism, but chose to stay with this name anyway.

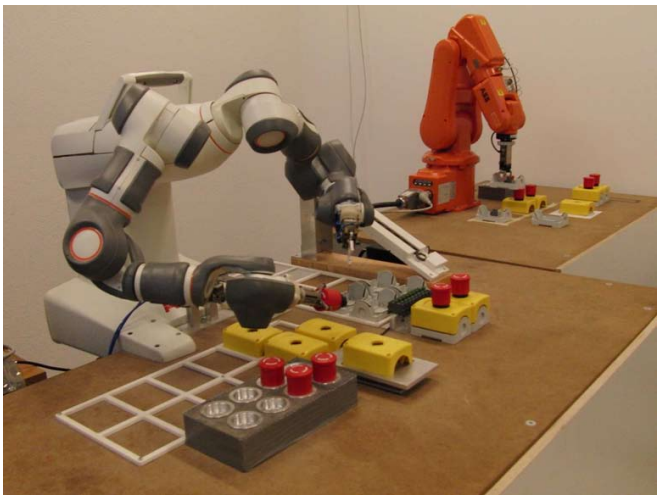

Fig. 5. A setup of two different robots, the two-armed concept robot Frida and the ABB IRB120, for individual or coordinated assembly of emergency buttons within the ROSETTA project.

a distributed system. The server can be complementary to the global KIF by storing additional, perhaps non-public, ontologies and storing shared cell and task information only relevant for the factory or lab. From these local servers, generic concepts can be uploaded to the global server.

An experimental setup for a robot assembly scenario within the ROSETTA project, where this was investigated, is shown in Figure 5. The workcell contains two different manipulators and different hardware configurations with respect to sensor information (e.g., force/torque measurements) are used. The main research platform in the ROSETTA project was the two-armed concept robot Frida from ABB Robotics, seen to the left in the picture and to the right we have a conventional industrial robot (ABB IRB120) extended with an open robot control interface [4].

Based on the assumptions of a shared workcell information scenario, the local KIF may "automatically propagate" e.g, the status of a local tool or fixture calibration also between task realizations for individual robots whereas 'generic functionality' such as descriptions for different standard operation such as a 'peg-in-the-hole' or a 'snapfit' functionality may be retrieved from a higher level in the hierarchy. The instantiation and realization of a peg-in-the-hole operation is of course strongly dependent on local configuration and access to sensor information. Although this may be considered as a reference implementation, it is worth to point out that a common high-level task description was evaluated not only between the different ABB robots, but also in two completely different laboratory setups: at the setup of RobotLab, Lund, shown in Fig. 5 and at the lab of our project partners at KU Leuven, where not only the robot manipulators were of very different nature (ABB Frida and the KUKA LWR, respectively), but also the robot system software from the very low-level control up to the high-level of state machines/SFCs and robot programming languages differed substantially.

More details about this system may be found in [5], [6] and the constraint-based task specifications and the combination of high-level action specification and low-level motion execution is described in [7].

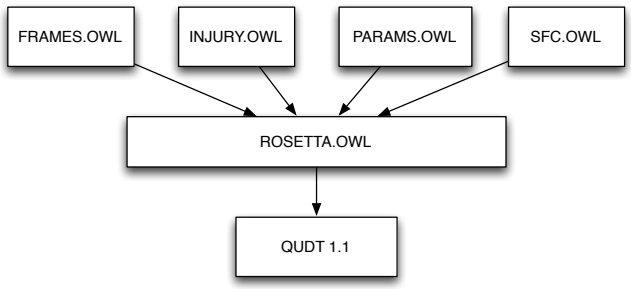

Fig. 6. The ROSETTA ontologies

One of the core insights of the project was that the robotic ontology used for supporting all connected subsystems cannot be monolithic, as it used to be in our previous work. There are too many agents with too many overlapping demands: e.g., engineering stations requesting data about physical objects in the station environment or demanding knowledge about skills available for a particular brand of robot equipped with a specific force/torque sensor; dialogue managers demanding a translation of text with constraints imposed by a concrete production environment; error management systems requesting specifics of a concrete skill; or, a safety controller interested in limit values for maximum robot speed given a human body part close to the end effector, etc. We have investigated the possibility of ontology modularisation and reached a preliminary and rather ad-hoc solution, presented in Fig. 6. We import the QUDT[4] ontology (quantities, units, dimensions and types) into the core robotic skill ontology centered around devices (`rosetta.owl`). This ontology in turn serves as a basis for defining feature frames substantially simplifying task definition (`frames.owl`), providing limit values for robot-human contact (`injury.owl`), specifying several methods for describing behaviour using graphical representation of transition systems (`sfc.owl`) or concretizing parameters of robot skills (`params.owl`).

IV. The PRACE apprentice

In the European projects SIARAS and ROSETTA, we have developed architectures for distributed robotics systems. Both have knowledge bases and ontologies for knowledge representation of robot cells and skills, as well as reasoning services. In the ROSETTA project we have created a system for high-level programming, where the user can combine preprogrammed skills into a new task and adapt the skill parameters to the new station. Executable code is then generated for the task. The skill representations and services are further developed in the ongoing PRACE project. The goal

[4] `http://www.qudt.org`

of the Productive Robot ApprentiCE project[5] is development of highly adaptable two-handed mobile robot systems for automation of small batch assembly operations.

The focus is on fast and intuitive training of the robot task by using programming-by-demonstration techniques to synthesize a task solution. The learnt task is to be stored in a central knowledge base. PRACE stores learnt tasks in terms of assembly operations. The knowledge base also contains knowledge about mapping operator demonstrations into assembly operations.

The architecture is built from knowledge-based web services interacting with a legacy ABB controller and a ROS system. At the moment, we are working on a demonstrator where a two-armed robot is mounted on a mobile base and the system is programmed using a tablet. The system uses ROS-based components for high-level computations, while the low-level sensor control uses realtime protocols. Both the tablet and the mobile robot have limited local computing power and battery time. Thus, we use a distributed system for code generation, planning, trajectory generation and control, where computationally heavy, non-realtime services are located on more powerful machines and accessed remotely.

The modular approach of the ROSETTA project ensures that the PRACE system can reuse the online services for planning and scheduling and natural language programming, see Fig. 7, thus extending the system capabilities with very little effort.

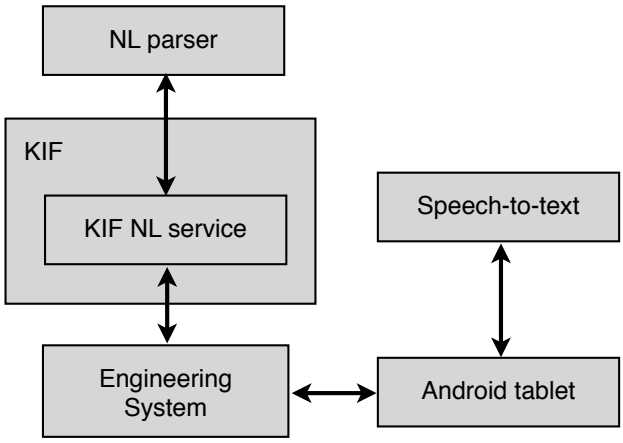

Fig. 7. The service architecture for the natural language programming interface. The NL service on KIF receives text, sends it to a general natural language parser (NL parser) and outputs a sequence of robot actions. These actions are generated into executable code using the architecture displayed in Fig. 4. The Android tablet can use its own speech-to-text cloud service and send the text to the Engineering System, thus extending the user interface to a portable platform.

V. SMERobotics for the small enterprise

In the SMErobotics[6] project one of the main foci is on applicable robot solutions for small and medium sized enterprises (SMEs). In a typical SME-scenario, short production series call for less use of expensive fixture-based production for economical reasons, but then require easily reconfigurable setups which need to cope with and compensate for large structural uncertainties. The higher level of uncertainty can be handled by advanced sensor-based systems, but today even the conventional industrial robot programming, without the above mentioned extensions, is still a bottleneck with respect to both time and expertise. Based on high-level task descriptions and intuitive interaction where the worker's process knowledge (not the knowledge of robotics) can be fully utilized, the goal is not to reach a fully automated system, but a system with high productivity due to the interaction of an operator with the robot system, giving flexibility in production changes, and short error recoveries.

Cognition is needed both on the robot side and on the human side, symbiotically, and must be integrated with learning. Although newly learnt functionality on the robot side may be immediately distributed, locally or globally, this does of course not count for the human operators where each individual will have a different level of experience and expertise and thereby different abilities and preferences on how to interact and instruct. A personalized interface and dialogue system that the individual operator can access remotely, may be beneficial in this human-robot-collaboration. Concepts and symbols used in dialogues need to have a grounding that is shared by the human and the machine, e.g., to support the user in error situations.

In this scenario the possibility to extend the available services with online reasoning and policy generation of error handling procedures opens also for efficient individual operator dialogues. In the SMErobotics project this topic is investigated in the context of modular knowledge and distributed reasoning systems.

Whereas both the PRACE and the SMErobotics efforts are based on compositional knowledge-bases, and with an emphasis on the notion of skills for reuse/portability of motions [8], our focus in SMErobotics is on wood-working including handling, which shares many of the issues with small-batch assembly.

VI. Related work

The rapid growth of network services and the latest development of cloud solutions in a very general form has definitely had its impact also in automation and robotics (in industrial as well as in service robotics).

Although the concept of network distributed control and functionality in not new, see e.g., [9], the term "cloud robotics" has spread tremendously since James Kuffner has introduced it in 2010; it has attained a large interest in the robotics community reflected in a number of publications, it is appearing in research calls and has led to important development in open-source and open-access projects[7] during the last couple of years. An important European project addressing the topic of knowledge sharing online, is RoboEarth[8] [10], that aims at creating a World Wide Web

[5] http://prace-fp7.eu
[6] http://www.smerobotics.org
[7] See e.g., http://code.google.com/p/rosjava/
[8] http://www.roboearth.org

for robots. Their knowledge base contains ontologies, tasks and environmental data [11], which are shared by robots. The RoboEarth infrastructure, named Rapyuta, has recently become available publicly [12] in its alpha version. However, RoboEarth project focuses mostly on the service robotics domain and does not discuss the needs of a industrial robotics domain.

Departing from network research, and in particular, networked robotics systems, Hu and coworkers analyze the opportunities offered by cloud robotics infrastructure [13]. Their applications focus though on the typical mobile robotics domains, like SLAM, navigation and grasping, leaving manufacturing outside the scope of interest.

Another popular cloud service, also in and for robotics, is natural language interpretation, such as speech-to-text techniques which we also see commonly appearing in e.g., smartphones. Thomas and Jenkins [14] describe a system for commanding a robot using natural language. Stenmark and Nugues [15] describe a more advanced solution to this problem.

An recent survey of work related to cloud robotics and automation was made by Goldberg and Kehoe [16]. Kehoe et al. [17] have also presented a concrete application of cloud computing for robot grasping using Google's object recognition infrastructure. However, it is not available publicly, making it less atractive for the cloud robotics community.

VII. Conclusions

The research done so far allows us to make the following observations. However, we would like to stress that their scope is limited by the context in which our research has been done: industrial robotized manufacturing systems, mostly relevant for assembly.

Online knowledge bases: Their simplest possible advantage is to make deployment of a knowledge base and its associated services to every single system installation unnecessary. Instead, one central copy (or several mirrored ones) of the knowledge server needs to be created and maintained. In particular, the knowledge update and system upgrade can be made instantaneous, making fresh services immediately available to all users worldwide.

Depending on the model adopted for the knowledge base (monolithic or distributed, federated or governed by a single body) the chances of getting more knowledge provided to the system increase dramatically. However, this introduces also a whole set of issues that need careful attention, like e.g., reliability of knowledge coming from different, possibly unknown sources, guarantees of access to the (or a) knowledge base in all circumstances, depending on the business model adopted, consistency of knowledge provided by different actors, completeness of available resources with respect to a given set of tasks, knowledge overlap and possibility of choosing particular services based on experience, trust, or other criteria, just to name a few.

In particular, such federated model would allow many stakeholders like robot producers, system integrators, sensor providers, software deployers, to cooperate and contribute to a rich market of knowledge-based solutions, in a manner similar to what happens now in the ROS community regarding lower-level solutions for robotics.

Reuse of knowledge: As it has been shown in our ROSETTA investigations, such approach allows reuse of knowledge introduced by one stakeholder by others. This applies to skill definitions as well as concrete parameter modifications or adaptations, or fault detection routines inserted after a skill has been deployed in a concrete task. The experience gathered during deployment of a system may be made available to others without unnecessary delay or update burden.

On the other hand, the question of relevance of such experience is unclear and needs to be raised here. How can one judge whether parameters adapted at site A are useful for site B? How can one realize whether a user of a particular robot C possibly wants to deceive other users by uploading incorrect values? How to make the data/knowledge uploader responsible for their acts?

Probably reusability is the most significant advantage of such federated approach to knowledge services, however this possibility must be investigated further in order to provide concrete benefit.

Web services for robots: Web services are an attractive computational model, making it possible to request specific (possibly knowledge-based) services, without knowing any particular details about their implementation, residence, ownership, etc. With a thoroughly defined API it makes it possible to separate concerns between the installed system: specific, task-related, hardware-specific computations, and the service provider: generic, task-independent, hardware-independent, portable computations.

As in the cases above, the questions of reliability, responsibility of service providers, portability and generality need to be raised here. However, the responsibility question is clearer, as it seems rather straightforward how to make service providers accountable. In this manner, given sufficiently large market, robot system capabilities may increase substantially in a very short time. This is an opportunity for the robot manufacturers that seems to be very attractive.

It has to be made clear here that industrial players are rather conservative with that respect and that a lot of effort needs to be put into issues of security, reliability and dependability. Both on the side of manufacturers, be it robot, sensory equipment, or end effectors, as well as system integrators and end users, everyone is interested in keeping their know-how well protected as this is the main source of their profit. Sharing it freely, or just making it vulnerable to cyberattacks, is a risk that needs to be seriously considered and resolved. Without addressing those questions there is a risk that research activities will have only very modest influence on industrial practices.

On the other hand, there are solutions available, known from computer networking area and used in business-to-business communication, that address those issues. They are currently being extended to the context of cloud-based services (see e.g. [18]). What needs to be done is porting

them or adapting to a domain involving both software and heterogenous hardware subsystems, tightly interacting with each other. It is definitely a challenging task, but e.g., the automotive industry shows[9] that these questions may be answered in a manner satisfying to all market participants.

ACKNOWLEDGMENTS

The research leading to these results has received partial funding from the European Union's seventh framework program (FP7/2007-2013) under grant agreements No. 230902 (project ROSETTA), No. 285380 (project PRACE) and No. 287787 (project SMErobotics) and from the European Union's sixth framework program under grant agreement No. FP6-017146 (project SIARAS). The fourth author is member of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University and would like to acknowledge the Swedish Research Council grants VR 2010-5864 and C0590801 (Cloud Control).

The authors are grateful to Mathias Haage for many valuable comments that made this paper more legible.

REFERENCES

[1] M. Haage, J. Malec, A. Nilsson, K. Nilsson, and S. Nowaczyk, "Declarative-knowledge-based reconfiguration of automation systems using a blackboard architecture," in *Proc. 11th Scandinavian Conference on Artificial Intelligence*, A. Kofod-Petersen, F. Heintz, and H. Langseth, Eds. IOS Press, 2011, pp. 163–172, doi: 10.3233/978-1-60750-754-3-163.

[2] P. Lidén, "Distributed knowledge sources in SIARAS," Master's thesis, Department of Computer Science, Lund University, 2009.

[3] A. Björkelund, L. Edström, M. Haage, J. Malec, K. Nilsson, P. Nugues, S. G. Robertz, D. Störkle, A. Blomdell, R. Johansson, M. Linderoth, A. Nilsson, A. Robertsson, A. Stolt, and H. Bruyninckx, "On the integration of skilled robot motions for productivity in manufacturing," in *Proc. IEEE International Symposium on Assembly and Manufacturing*, Tampere, Finland, 2011, doi: 10.1109/ISAM.2011.5942366.

[4] A. Blomdell, I. Dressler, K. Nilsson, and A. Robertsson, "Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers," in *Proc. ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, Anchorage, AK, May 2010.

[5] J. Malec, K. Nilsson, and H. Bruyninckx, "Describing assembly tasks in declarative way," in *Proc. IEEE ICRA 2013 Workshop on Semantics, Identification and Control of Robot-Human-Environment Interaction*, 2013.

[6] M. Stenmark and J. Malec, "Knowledge-based industrial robotics," in *Scandinavian Conference on Artificial Intelligence*, 2013, submitted.

[7] M. Stenmark and A. Stolt, "A system for high-level task specification using complex sensor-based skills," in *RSS 2013 workshop, Programming with constraints: Combining high-level action specification and low-level motion execution*, Berlin, Germany, 2013, poster presentation. [Online]. Available: http://robohow.eu/meetings/rss-2013-constraints-workshop

[8] K. Nilsson, E. A. Topp, J. Malec, and I.-H. Suh, "Enabling reuse of robot tasks and capabilities by business-related skills grounded in natural language," in *Proc of ICAS2013*, Lisbon, Portugal, March 2013.

[9] M. Inaba, S. Kagami, F. Kanehiro, Y. Hoshino, and H. Inoue, "A platform for robotics research based on the remote-brained robot approach," *Int. J. of Robotics Research*, vol. 19, pp. 933–954, 2000.

[10] M. Waibel, M. Beetz, J. Civera, and R. D. Andrea, "RoboEarth-A World Wide Web for Robots," *IEEE Robotics and Automation Magazine*, no. 6, pp. 69–82, 2011.

[11] M. Beetz and M. Tenorth, "Exchanging action-related information among autonomous robots," in *12th International Conference on Intelligent Autonomous Systems*, 2012.

[12] D. Hunziker, M. Gajamohan, M. Waibel, and R. D'Andrea, "Rapyuta: The RoboEarth cloud engine," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Karlsruhe, Germany*, 2013, pp. 438–444. [Online]. Available: http://roboearth.org/uploads/RCE2013.pdf

[13] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: architecture, challenges and applications," *Network, IEEE*, vol. 36, no. 3, pp. 21–28, May 2012, doi: 10.1109/MNET.2012.6201212.

[14] B. J. Thomas and O. C. Jenkins, "RoboFrameNet: Verb-centric semantics for actions in robot middleware," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 4750–4755.

[15] M. Stenmark and P. Nugues, "Natural language programming of industrial robots," in *Proc. Int. Symposium on Robotics, Seoul, Korea*, 2013, submitted.

[16] K. Goldberg and B. Kehoe, "Cloud robotics and automation: A survey of related work," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-5, Jan 2013. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-5.html

[17] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *Proc. IROS 2013*, Karlsruhe, Germany, May 2013.

[18] S. Yoshinori, I. Takahiro, M. Gen, W. Yoshikazu, and K. Shuichi, "Implementing secure communications for business-use smart devices by applying openflow," *NEC Technical Journal: Special Issue on Smart Device Solutions*, vol. 7, no. 3, 2013.

[9] www.autosar.org