Knowledge Representation for Reconfigurable Automation Systems*

Ola Angelsmark[1], Jacek Malec[2], Klas Nilsson[2], Sławomir Nowaczyk[2] and Leonardo Prosperi

***Abstract*— We describe the work in progress on knowledge representation formalisms chosen for use in the European project *Skill-Based Inspection and Assembly for Reconfigurable Automation Systems*. The goal of SIARAS is to make agile manufacturing easier by creating an intelligent support system for reconfiguration and adaptation of assembly systems.**

I. SIARAS

SIARAS is an acronym of an EU-funded (FP6 - 017146) STREP-project with the general aim to support end users and engineers of manufacturing systems, including robotic ones, and to make production engineering easier (and thus cheaper) in most circumstances.

Its main goal is to build an intelligent system, named provisionally the *Skill Server*, capable of supporting automatic and semi-automatic reconfiguration of a manufacturing processes in response to changing requirements. The main issue during the design phase was to merge two, somewhat opposed, views on the reconfiguration process: the top-down, AI-based view and the bottom-up, engineering one.

A. Top-down AI approach

The top-down approach describes the problem of reconfiguration as a (re)planning problem: given a new task (usually expressed as a goal condition), possibly being a modification of the previous one, and given a set of skills available in the system, understood as a description of the operations that might be performed by the devices available to the user, find such a sequence of operations that would ensure that the task is correctly executed (find a plan that achieves the goal). It is assumed that the domain is modelled sufficiently well.

B. Bottom-up Reparametrisation Approach

In this approach the skill server is used only for reconfiguration of an existing, correct, properly modelled production system. The system is not expected to propose novel solutions, nor to search for alternative ways of implementing the process. In particular, one should expect a detailed description of the task: what is produced and how (i.e. what are the steps of the process). Moreover, for each step it should be clear how does it contribute to the goal. On the other side, available devices must be described in terms of operations they are able to perform (skills) and conditions under which they can operate. Skill Server needs to map task into skills and parametrise them appropriately.

C. Finding the Golden Middle

It seems that the top-down AI approach is both computationally infeasible and impossible to model sufficiently well, while the bottom-up reparametrisation approach lacks generality and risks ending up as a database of *previously used* parameter settings for a number of devices in a number of scenarios. The main issue with this approach is guaranteeing scalability and extendibility to new domains or to new kinds of devices. There is a risk of limiting the approach to the previously considered cases and very similar ones only, thus precluding a more open-ended solution.

In order to make sure that we do not loose the larger perspective while we aim at restricting ourselves to a feasible problem, we imagine a layered approach, with reconfiguration level at the bottom and (re)planning level on top of it.

II. Ontology

We have decided to center knowledge representation around the concepts of devices (physical objects provided by their manufacturers) and skills (operations that can be performed). Task descriptions exist only during problem solving sessions, as dynamic structures, specific to a particular case. They can be seen as (arguably, quite complex) combinations of skills and parameters and therefore there is no need to have them explicit in the vocabulary.

The static part of the knowledge is represented in an ontology: a data structure storing all the necessary relations between the terms used. While ontologies are often used for classification purposes, in our case the classification is done when objects (skills or devices) are introduced in the structure. The main use of the ontology is to allow reasoning about skills matching particular tasks and about devices being suitable for particular operations, as well as to standardise the nomenclature used and the relationships of different concepts.

A carefully chosen set of representation primitives, together with a relatively rich ontology and a set of reasoning algorithms (available for complex formal systems such as description logics) allow us to keep the extension possibility open while focusing, in the beginning, on a concrete demonstrator case. As the project is done in cooperation with a German/Greek company INOS which provides system integration for automotive industry, we concentrate on a number of test cases from that domain.

We have chosen the open source tool Protégé for ontology creation and manipulation (together with reasoners such as Racer, Fact++ and Pellet), JGrafchart for task representation using Sequential Function Charts, and Python programming language for "gluing" the two together in the prototype.

* This work has been supported by the EU-project SIARAS (FP6 - 017146) http://www.siaras.org/.

[1] ABB, ola.angelsmark@se.abb.com

[2] Department of Computer Science, Lund University, Lund, Sweden, {jacek|klas|slawek}@cs.lth.se.