Configuration

Luca Anselma
Università di Torino, Italy

Diego Magro
Università di Torino, Italy

INTRODUCTION

Configuring means selecting and bringing together a set of given components to produce an aggregate (or a set of aggregates) satisfying some requirements. All the component types are predefined and no new component type may be created during the configuration process.

The result of the configuration can be physical objects (such as cars or elevators), non-physical entities (such as compound services or processes) or heterogeneous wholes made of both physical and non-physical parts (such as computer systems with their hardware and software components).

The configuration process has to take into consideration both endogenous and exogenous constraints: the former pertain to the type of the assembled object(s) (therefore they hold for all the individuals of that type) and mainly come from the interactions among components, whereas the latter usually represent requirements that the final aggregate(s) should satisfy. All these constraints can be very complex and make the manual solution of configuration problems a very hard task in many cases.

The complexity of configuration and its relevance in several application domains have stimulated the interest in its automation. Since the beginning, Artificial Intelligence has provided various effective techniques to achieve this goal. One of the first configurators was also one of the first commercially successful expert systems: a production rule-based system called *R1* (McDermott, 1982, 1993). *R1* was developed in the early Eighties to configure VAX computer systems, and it has been used for several years by Digital Equipment Corporation.

Since then, configuration has gained importance both in industry and in marketing, also due to both the support that it offers to the mass customization business strategy and the new commercial opportunities provided by the Web. Configuration is currently an important application field for many Artificial Intelligence techniques and it is still posing many interesting problems to scientific research.

BACKGROUND

The increasing complexity and size of configurable products made it clear that production-rule-based configurators such as R1 are not effective, particularly in the phase of maintenance of knowledge bases. In fact, changing a rule may require, as a side effect, changing several other rules and so on, and, actually, for some products, the component library may change frequently.

To partly address this problem, in current configurator systems, domain knowledge and control knowledge for problem solving are separate. The domain knowledge is represented in a declarative language, and the control knowledge (i.e., inferential mechanisms) is general (i.e., not depending on the particular problem to be solved). This is a common approach in modern knowledge-based systems. A configurator is based on an explicit representation of the general model of the configurable entities, which implicitly represents all the valid product individuals. The reasoning mechanisms implement the control knowledge and they use the domain knowledge to draw inferences and to compute configurations.

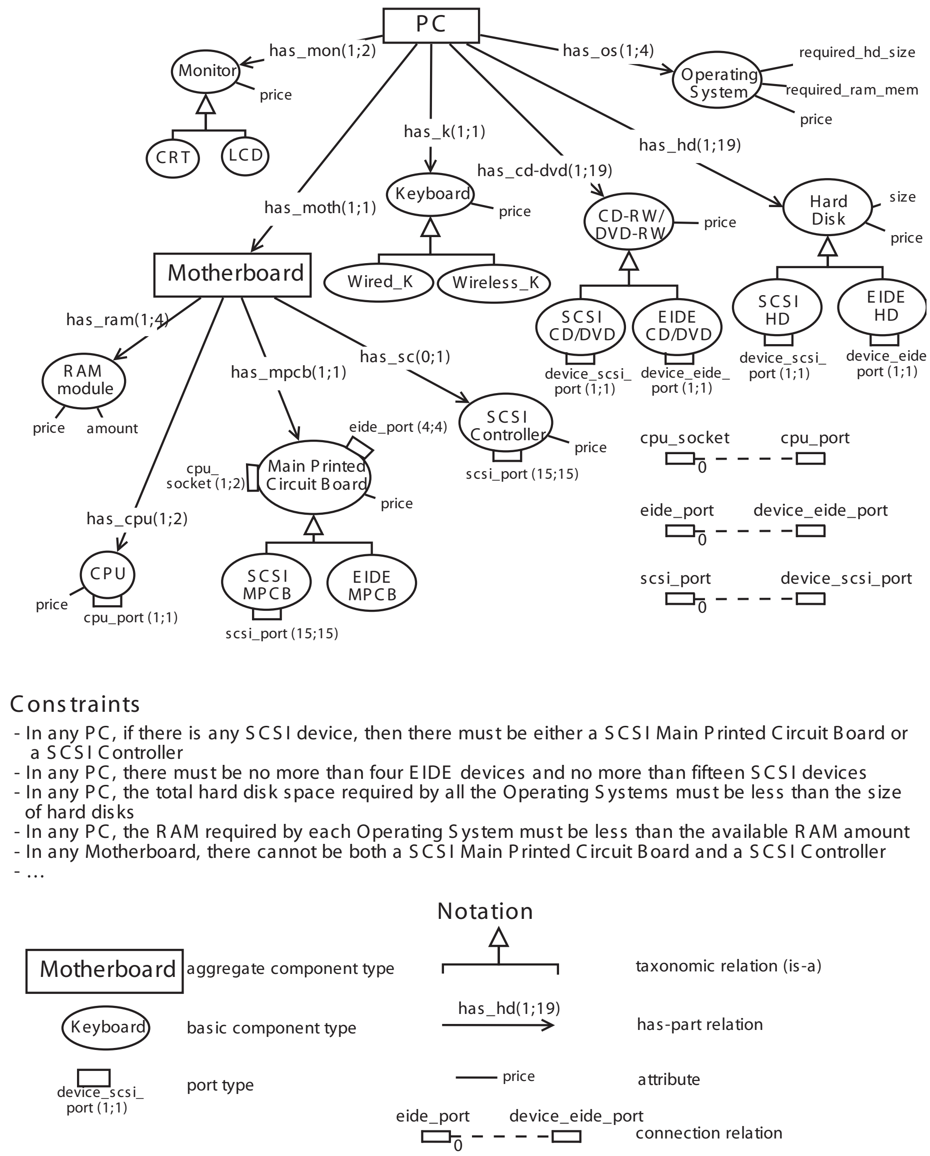
Regarding domain knowledge, there is a general agreement about what the concepts to represent are. In (Soininen, Tiihonen, Männistö & Sulonen, 1998) the authors introduce a widely accepted ***conceptualization*** for configuration problems. This conceptualization includes the concepts of

- *components*, which are the constituents of configurations;

- *parts* to describe the compositional structure;
- *ports* to model connections and compatibilities between components;
- *resources* that are produced, used or consumed by components;
- *functions* to represent functionalities;
- *attributes* used to describe components, ports, resources and functions;
- *taxonomies* in which component, port, resource and function types may be organized in;
- *constraints* to specify conditions that configurations must satisfy.

Figure 1 depicts a simplified fragment of the domain knowledge for PC configuration. It describes all the PC variants valid for the domain. Has-part relations

Figure 1. A fragment of a PC configuration knowledge base



Related Content