Prestudy of Configuration of a Naval Combat Management System

Klas Orsvärn
Swedish Institute of Computer Science
Box 1263, 164 28 Kista, Sweden

Ola Hansson
CelsiusTech Systems AB
175 88 Järfälla, Sweden
E-mail: klasorsv@sics.se, oaha@celsiustech.se

CTKONFIG is a project with the goal to investigate the potential of developing a system that can semi-automatically configure a highly complex product such as a combat management system, using a tool for development of interactive sales configuration systems called OBELICS (Axling & Haridi 1996). The project is performed in collaboration between CelsiusTech Systems and the Swedish Institute of Computer Science.

This position paper will first of all describe the product line, the nature of the sales configuration task, and the requirements on a support system. This is followed by a brief description of Obelics, and the current demonstration system.

CelsiusTech Systems and its Products

CelsiusTech Systems belongs to Celsius – one of Europe's leading defence industry groups. CelsiusTech Systems product range includes Naval Combat Management Systems, landbased command and control systems for coastal defence, the Air Force and Army. CelsiusTech has been setting the trend in shipbourne Combat Management Systems. Since the 1950's CelsiusTech has sold 170 9LV systems to navies around the world. More than 55 systems are contracted of in orders of the latest 9LVMk3 version.

The Type 9LV Mk3 Combat Management Systems are designed around a stable architecture based on selected standards, rules and guidelines for constructing software components and joining them together. With a software component library of hundreds of functions and several millions of lines of source code, reuse is becoming increasingly important.

In order to facilitate reuse of the building blocks, and to make configuration of new systems easier using existing components, CelsiusTech Systems is now making an effort to find ways to support system configuration.

The application task

When offering a system solution to a potential customer, there is often very limited time available for specification of the system. During this time, the customer requirements on operative functionality and equipment is analysed and mapped onto the set of software components in the 9LV Mk3 library. The system hardware is then configured in a way to meet the performance requirements in terms of board configuration in each node and node configuration on the LAN.

There is a big potential for cost and risk reduction if more accurate and detailed system specifications can be provided in offers to potential customers.

We have identified three types of requirements:

Interfaces to external equipment One kind of requirement is the equipment on the ship that the command and control system is to interface with, i.e. sensors, weapons, and communication equipment.

Functional requirements These are functions that the operators of the system wish to perform. They are called CMS-functions (Combat Management System). Some are mandatory. The optional ones are selected from a structured list of CMS-functions that the current library of software components can provide. There may be preconditions for selecting optional CMS-functions: specific kinds of external equipment may be needed, and some CMS-functions are useless unless other CMS-functions have already been selected.

Non-functional requirements Systems may differ in requirements such as redundancy, response time, and the number of concurrent operators.

The requirements are used to select among components of the following types:

- Software components from the Mk3 9LV library.
- Hardware components (interface boards, CPU, printer, etc.).
- Network nodes, selected from a set of standard types such as gun-control unit, director control unit, multifunction console, standard interface unit.

Component selection

Given customer requirements of the above three types, the selection of components is governed by the following types of rules:

Interface adaption For each interface to external equipment, and for peripheral equipment such as printer, specific software components are needed for adapting the signals to a general interface. Likewise, specific hardware is needed for each interface.

Core system software A large part of the software system is the same in all systems, e.g. the operating system, application support functions, data distribution mechanisms, and mandatory CMS-functions.

Optional software For each optional CMS-function, a specific set of software components is needed for providing it. The same software component may be utilized for different CMS-functions.

Network nodes A set of network nodes, from a set of predefined types, needs to be defined. Some node types are specialized for specific kinds of external equipment, e.g. gun and director, and can thus be determined directly from the requirements. Other node types can host a group of external equipment of any type. Instances of such node types have to be selected in a way that allows an allocation of external equipment and software that meets the non-functional requirements on performance and redundancy. For example, different equipment that is in some sense equivalent, such as different radars, should *not* be allocated to the same node, and software components that have much communication *should* be in the same node. Moreover, each node can have one or more CPU's, and the estimated CPU load should not be too high.

Tool requirements

In addition to the task analysis described above, a few more tool requirements of this application are relevant.

With a product line of this kind, the process of generating a configuration and quotation involves very many decisions, even with a support system. This means that the support system must not require the user to restart configuration whenever significant changes need to be made, either in the current customer requirements, or if new components need to be added in the product line.

If something is changed in the requirements, e.g. deleted, it is essential to find out what other decisions were made by dependency from that, so that they can be retracted where appropriate.

Obelics

OBELICS was developed at SICS during 1992 and 1993, based on the experiences of a fielded system for sales configuration of low-voltage switch-gear at AIEL, a subsidiary of ABB. The objective of OBELICS is to enable very quick development of sales configuration systems, which could be maintained by domain experts. Applications are developed by representing product knowledge in a component oriented modelling language.

A component can be defined to have properties and relations of the following kinds:

- Preconditions, are logical expressions which must be satisfied before an instance can be selected.
- Dependencies to other components.
- Attributes, which may have different values according to either methods in terms of other component attributes, or user selections.
- Sub-classes, i.e. other components are sub-classes of this component.
- Sub-components, i.e. other components are subcomponents of this component. The sub-component relation may be mandatory or optional, and the cardinality may vary.

The knowledge can be entered by editing the component hierarchy graphically. A configuration engine will generate the problem solving behaviour, by inspecting the component models. The end-user interface is also generated on the fly. Configuration begins by trying to instantiate the top-level component in the hierarchy of sub-classes and sub-components, which leads to instantiation of its sub-components, etc., while taking component dependencies into account. OBELICS is interactive in the sense that it will ask the user whenever there is a choice to resolve, which cannot be determined from the component models. Moreover, the user can edit the configuration, while OBELICS keeps track of the dependencies. For a more elaborate description of OBELICS, see (Axling & Haridi 1996).

The demonstration system

On the basis of the task analysis described above, a first demonstration system has been developed using OBELICS. It meets the requirements given above fairly well, and covers all parts of the configuration task, except allocation of components to nodes in the computer network. The latter is a difficult allocation problem that Obelics cannot support today. It is possible to program a solution in Prolog, but we have decided to postpone this. Nevertheless, it is an important part of the configuration task, and will occur in other complex configuration domains as well. In this application, many considerations need to be taken into account in the allocation, e.g. reuse, which may also be conflicting. This implies that the end user may want to edit the allocation, and this editing should be taken into account if the allocation is recomputed.

Acknowledgements Robert Johansson, Celsius-Tech Systems, and Olle Olsson, SICS, have contributed to the prestudy reported here.

References

Axling, T., and Haridi, S. 1996. A tool for developing interactive configuration applications. *Journal of Logic Programming* 26(2):147–168.