An Integration Framework for Developing Interactive Robots

Jannik Fritsch and Sebastian Wrede

Applied Computer Science, Faculty of Technology, Bielefeld University, Germany
{jannik,swrede}@techfak.uni-bielefeld.de

1 Challenges in Interactive Robotics Research

In recent years there is an increasing interest in building personal robots that are capable of a human-like interaction. In addition to multi-modal interaction skills, such a robot must also be able to adapt itself to unknown environments and, therefore, it has to be capable of knowledge acquisition in a lifelong learning process. Moreover, as humans are around, reactive control of the robot’s hardware is important, too. Consequently, researchers aiming to realize a personal robot have to integrate a variety of features. Due to the very different nature of the necessary capabilities, interactive robotics research is thus a truly interdisciplinary challenge.

A natural design decision is to take a modular approach to build such a complex system [Ros95]. This allows the different researchers to focus on their respective task and makes integration easier. Because integration is an iterative process in large-scale research systems, it must be possible to incorporate new modules and functionalities in an interactive robot as they become available. Such a continuous extension of the robot’s functionality not only results in new or extended data structures provided by the added components but usually also requires modifications of the control flow in the integrated system. This poses questions of how data and control flow can be expressed in a way that the effort and the complexity for continuous integration remains controllable during the integration process [CS00].

As described in Chapter Trends in Robotic Software Frameworks, these requirements make the framework-based approach well suited to the development of software applications for Interactive Robots. In this context, a fundamental task for integration frameworks is the ability to distribute modules across different computing nodes in order to achieve the reactivity needed for human-machine-interaction [KAU04]. This applies especially to large-scale systems like personal robots. However, most researchers are no middleware experts, prohibiting the native use of, e.g., CORBA-based solutions. Consequently, the communication framework needs to provide a restricted but suf-
icient set of functionality that enables interdisciplinary researchers to easily integrate their components into a distributed robot system.

Knowledge acquisition and the ability to adapt to unknown situations imposes two additional functional requirements that need to be addressed. An application may need some kind of memory so data management services must be provided by the integration framework. Since adaptation and processes like e.g. attention control induce dynamics into the system, dynamic (re-)configuration of components running in the integrated system is essential, too.

In order to support the envisioned incremental development of a personal robot, not only the functional requirements modularity, communication, module coordination, as well as knowledge representation and acquisition and dynamic (re-)configuration must be supported by the system infrastructure. Additionally, several non-functional requirements play an important role that are discussed in the following.

Taking into account that the people carrying out research on interactive robots are usually concentrating on single topics and not the overall integration, the developed framework must be very easy to use in order to gain a wide acceptance. The approach we will outline in this chapter tries to tackle this by focusing on simplicity and exploiting standards compliance.

Another important non-functional requirement we try to address is to provide a framework that enables rapid prototyping. Consequently, iterative development should not only be supported for single modules but also for the integrated system. Erroneous directions in system evolution can more easily be identified if integration is performed on a regular basis starting at an early stage even when components are still missing and need to be simulated. For large-scale systems, software engineering research has shown that decoupling of modules is very important. Thus, the framework should support low coupling of modules. This facilitates not only independent operation of components but also minimal impact of local changes on the whole system. With a framework that adheres to low coupling, debugging and evaluation of a running system architecture can be supported more easily.

The contribution starts with an introduction into the concepts of the XCF framework we developed along these requirements to enable high-level integration and coordination of interactive robots. The process of robot development with this approach and the lessons learned thereby are described in Sect.s 3 and 4, respectively. A conclusion on the presented approach is drawn in Sect. 5.

2 The XCF SDK

Taking into account the requirements outlined in the previous section, we developed a software development kit termed XCF consisting of a set of object-oriented class libraries and the required framework tools to develop, debug and run a distributed robotic system. The concrete implementation of the XCF