

# A Management Tool for Component-Based Real-Time Supervision and Control Systems

Sandro Santos Andrade, Raimundo José de Araújo Macêdo

Distributed Systems Laboratory (LaSiD)  
Post-Graduation Program on Mechatronics\*

Federal University of Bahia, Campus de Ondina, 40170-110, Salvador-BA, Brazil

***Abstract.** In the last years, many researches have been concerned with the adaptation of software engineering techniques for use in real-time industrial systems. Following this context, we have designed and implemented the ARCOS framework, devoted to the construction of reusable, flexible, and interoperable industrial systems. This paper presents a management tool we have developed to facilitate the specialization of ARCOS into a particular industrial scenario and to provide generic clients for the real-time supervision and control of industrial plants. We briefly discuss the ARCOS server-side component architecture, present two application examples, and describe the provided facilities for the configuration, assembly, and deployment of ARCOS-based applications.*

## 1. Introduction

Component-based system development has been the focus of many researches concerned with the design of platforms that leverage the composability and reusability of software solutions [Heineman and Councill 2001]. By using facilities such as component configuration, run-time resource management, and use of non-functional services, the developer concentrates just on core business logic parts, relying on an efficient and reusable component-oriented middleware infrastructure. The modular and uncoupled approach used for implementing and combining components leads to reusable and easily maintainable systems.

On the other hand, framework technology has increasingly been used for enhancements in productivity, quality, and reusability in the software development process. [Fayad et al. 1999]. As a framework implements partial solutions in a given application domain, it concentrates technical expertise about architectural solutions that are feasible and efficient in that context. Thus, the framework handles the main activities of the system, and performs calls on developer-supplied components which apply the framework into a specific situation of the application domain. This characteristic is usually called "inversion of control" and the points in the framework architecture in which the developer connects the specialized components are usually called "hot-spots".

The development of component-based vertical frameworks, dedicated to specific application domains [Rogers 1997], represents an interesting trend since it introduces the benefits of components into areas that have been demanding solutions for reusability, interoperability, and flexibility. This class of frameworks provides solutions to very specific application domains, such as signal processing, statistical analysis of economic data or numerical analysis.

---

\*Run by the Departments of Computer Science (DCC) and Mechanical Engineering (DEM).

In industry, the requirements of manufacturing management and factory automation systems have driven, in the last years, the adoption of software solutions that are distributed, flexible, and reusable. Another important requirement of industrial systems is that of interoperability, which is due to the use of software and equipments from different vendors, each of them specialized in some industrial activity, such as robot arms, programmable-logic controllers, numerical control machines, etc. Moreover, upgrades in software and hardware usually lead to new systems, usually incompatible with the old ones.

Following this context, we have designed and implemented ARCOS (ARchitecture for CONtrol and Supervision), a component-based framework for real-time industrial systems, initially introduced in [Andrade and Macêdo 2005]. ARCOS provides partially specialized solutions for industrial data acquisition, control, and supervision. The current implementation of ARCOS Framework is built atop CIAO (Component-Integrated ACE ORB) [N. Wang and Subramonian 2004], a real-time implementation of the CCM (CORBA Component Model) specification [OMG 2001]. The CIAO features for assembly, configuration, and deployment of component-based systems, in conjunction with the flexible and reusable framework we purpose, make ARCOS suitable for use in a varied range of industrial supervision and control systems, such as manufacturing, robot arms control, and automation systems.

This paper presents a management tool we have developed to facilitate the specialization of ARCOS into a particular industrial scenario and to provide generic clients for the real-time supervision and control of industrial plants.

The rest of this paper is organized as follows. Section 2 briefly discusses the design choices of the ARCOS Framework. Section 3 presents the management tool, and the related facilities for configuration, assembly, and deployment of ARCOS-based systems, and also two application examples built atop ARCOS: a chemical reactor supervisory and a cruise control system. Finally, section 4 draws some concluding remarks.

## **2. The ARCOS Framework**

In the past, real-time industrial systems were usually characterized by software solutions difficult to maintain, extend, reuse, and integrate with other industrial systems. As hardware and software technologies become more powerful, the demand for flexible, reusable, and interoperable real-time systems, turn out to be unavoidable [Sanz 2003]. However, designing and implementing such a class of systems requires specialized expertise and knowledge. The ARCOS framework is a response for such a demand, providing reusable software solutions for data acquisition, control, and supervision activities. Moreover, such a framework provides an interoperable, distributed, and predictable environment for industrial supervision and control systems.

Interoperability is achieved in ARCOS by the implementation of the DAIS (Data Acquisition from Industrial Systems) standard [OMG 1999]. DAIS specifies standardized CORBA interfaces in which a specific data acquisition device technology, such as parallel port or programmable logic controller, can be mapped. By using this standard, any DAIS-aware client can browse and acquire industrial data exposed by the DAIS server, in despite of the underlying programming language, operating system, or communication media.

In order to provide a flexible and uncoupled infrastructure, ARCOS relies on the

CCM component standard and the event-based communication model. The CCM component standard specifies a software environment for hosting components (container), non-functional reusable services (such as persistence, security, and object localization), and flexible mechanisms for component assembly by the use of XML deployment descriptor files. On the other hand, the event-based communication provides an uncoupled, many-to-many, and non-blocking component communication model. The real-time container and real-time event service provided by CIAO [Harrison et al. 1997], an implementation of CCM, address the predictability required by real-time industrial systems.

In order to allow for the use of ARCOS in a range of industrial supervision and control applications, two "hot-spots" were defined, the Dais Provider and the Controller, described below.

**The DAIS Provider.** The component-based DAIS implementation accomplished by ARCOS decouples industrial data sources from the standardized operations for browsing and acquiring these data. For such a decoupled data acquisition, we introduced the concept of the DAIS Provider, implemented as an abstract component. In order to fit ARCOS into a specific data acquisition scenario, the developer needs to extend this abstract component, providing specific implementation for data acquisition. Then, this developer-provided component is connected, by the management tool, to the *DAIServer* component, provided by the ARCOS internal structure. All other operations, such as data selection for acquirement, specification of the acquisition rate, and delivery activities are performed by ARCOS, leveraging the productivity and saving the system from error-prone solutions.

**The Controller.** The automatic control of industrial variables (such as temperature, pressure, and level) is a common activity in industrial systems. ARCOS provides a reusable closed control loop implementation that allows for the use of different control strategies (such as PID, adaptive, or neural-fuzzy control). The job of the developer is reduced to the implementation of a specific Controller component, which is connected, by the management tool, to ARCOS' internal structure.

A more detailed description about the ARCOS internal components, as well as the report of experiments regarding the temporal predictability of ARCOS, can be found in [Andrade and Macêdo 2005].

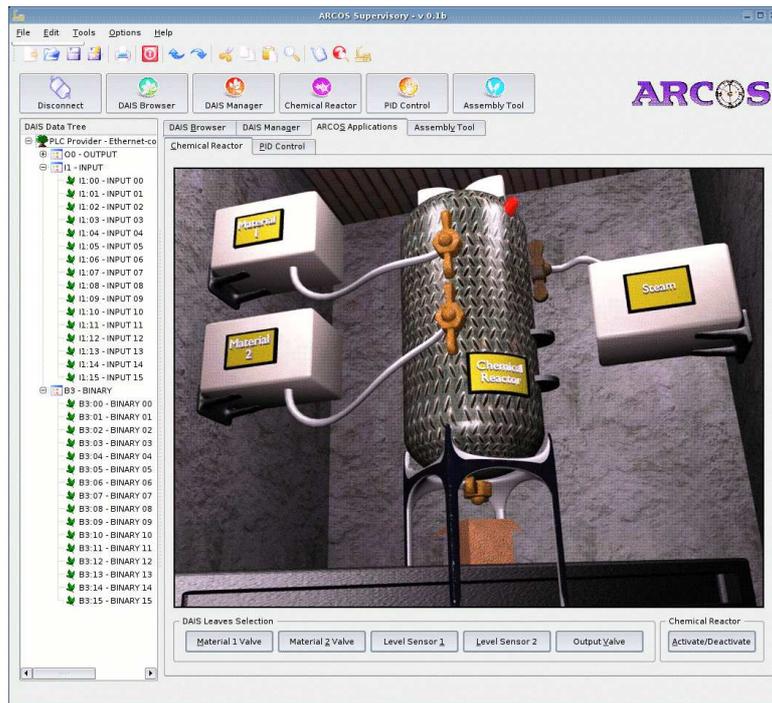
### 3. The Management Tool

The ARCOS Supervisory is a tool that provides a generic DAIS client for data browsing and acquisition, as well as, an assembly tool that facilitates the development of new ARCOS-based industrial systems. The ARCOS Supervisory tool is made up three main modules: the DAIS Browser, the DAIS Manager, and the ARCOS Assembly Tool.

The DAIS Browser is a generic client that allows the connection to any DAIS-compliant server. In this client, the user can browser the industrial data exposed by the server, insert data into acquisition groups, and inform the desired acquisition rate.

The DAIS Manager is a monitoring tool that enables the visualization of the current DAIS sessions on a given DAIS server and the data groups created in each DAIS session.

The specialization of ARCOS in order to fit specific data acquisition and control



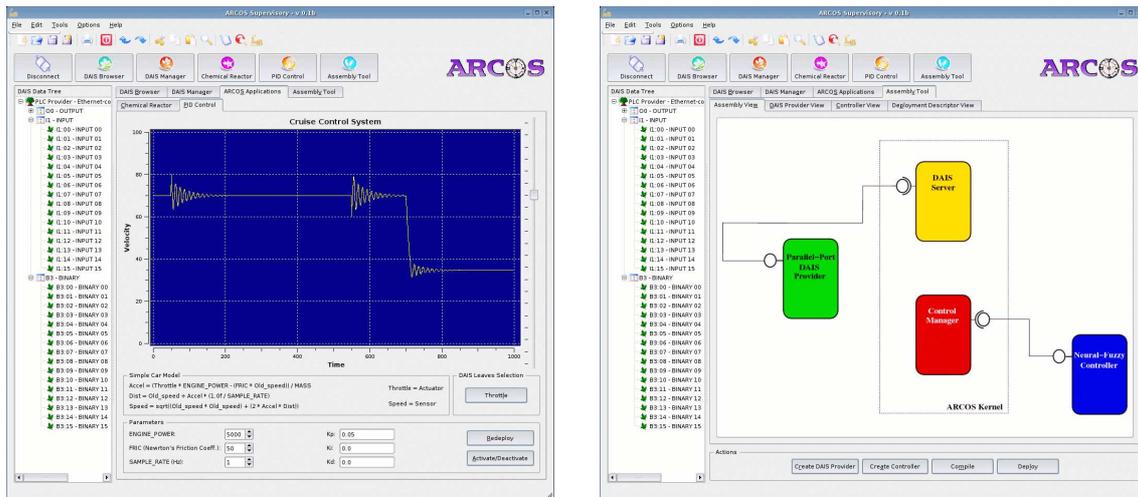
**Figure 1. Supervision client for the chemical reactor simulation kit.**

situations is achieved by the following steps: creation of CCM IDL (Interface Definition Language) and CIDL (Component Implementation Definition Language) files for the DAIS Provider and the Controller components, compilation of these files in order to generate the underlying CORBA objects, implementation and compilation of these new developer supplied components, construction of the related XML deployment descriptor file, and the deployment of the new application. Most of these steps require a specialized expertise, and the construction of the XML deployment descriptor files is boring and error prone. In order to address these issues, the ARCOS Supervisory provides an Assembly tool, which facilitates the generation of new ARCOS-Based applications.

The following two application examples, implemented atop the ARCOS Framework, illustrate the use of the facilities provided by ARCOS. The chemical reactor supervisory, illustrated in figure 1, is a 3D graphical DAIS client that enables the on-line monitoring of a chemical reactor, simulated by an electronic kit, controlled by an Ethernet-connect PLC (Programmable Logic Controller). The automation events, such as the opening of valves and hits of level sensors are simulated by 3D animations accomplished by the ARCOS Supervisory. The cruise control system, presented in the left of the figure 2, exemplifies the use of the PID (Proportional Integral Derivative) Controller provided by ARCOS for the control of the velocity of a car. Both applications have been fully implemented using ARCOS.

As illustrated in the right of the figure 2, with the Assembly Tool the user is able to easily create a new DAIS Provider and Controller, implement and compile these new components, generate the related XML deployment descriptors, and deploy the whole application.

The ARCOS Assembly tool provides four views of the assembly and deployment



**Figure 2. PID control of a simulated cruise control system (left) and the Assembly Tool for new ARCOS-based applications (right).**

process: the graphical assembly view (illustrated in the right of the figure 2), the DAIS Provider view, the Controller View, and the XML deployment descriptor file view.

When the user decides to create a new DAIS Provider or Controller, the ARCOS Assembly tool performs the steps as follows. First, the user informs the name of the new DAIS Provider or Controller, then the corresponding IDL and CIDL files are created by the ARCOS Assembly Tool and CIAO is invoked in order to compile these generated files. In the next step, the ARCOS Assembly tool generates empty implementations of the DAIS Provider and Controller components, indicating the methods required to be implemented by the developer. The XML deployment descriptor file is also generated and it contains all the component connections and configurations required for the deployment phase. The DAIS Provider, Controller, and XML deployment descriptor file are exhibited in the corresponding views illustrated in the right of the figure 2.

The generation of all required files is performed by the ARCOS Template Engine, a library that generates C++ source-code and XML files in a template-driven approach. This template files specifies the code to be generated and defines variables to be replaced by values provided by the user, like the DAIS Provider and Controller names.

Similar researches have been performed with the aim of providing development tools for the construction of real-time systems. A promising effort is the CoSMIC tool [Schmidt 2003], which provides a fully integrated environment for the implementation of real-time systems by the use of a model-driven approach. Due to the fact that CoSMIC is devoted to generic real-time systems, the ARCOS Supervisory provides a more productive tool when the goal is to construct a real-time supervision and control system.

The ARCOS Framework and ARCOS Supervisory has been fully designed and implemented in LaSiD/UFBA, using a Debian GNU/Linux Operating System, the Standard C++ programming language and the ACE/TAO/CIAO middleware technologies (<http://www.cs.wustl.edu/~schmidt/TAO.html>). The ARCOS Supervisory was implemented using the Trolltech Qt Graphical Toolkit (<http://www.trolltech.com/>) and the KDevelop Integrated Development Environment (<http://www.kdevelop.org>).

Further information about ARCOS Framework and ARCOS Supervisory can be obtained in the ARCOS Project web site (<http://arcos.sourceforge.net>).

#### 4. Conclusions

The benefits introduced by the component and framework technologies have leveraged the productivity and quality of distributed modern real-time systems. In addition, the use of development support tools is a crucial aspect in order to allow the use of these technologies by incipient developers.

This paper presented a management tool that provides generic clients for supervision and control of real-time industrial plants, as well as an assembly tool that facilitates the development of new industrial systems. This management tool is a client-side participant of the ARCOS Framework, a reusable, predictable, and interoperable solution for component-based real-time industrial systems under development at LaSiD/UFBA.

#### References

- Andrade, S. and Macêdo, R. (2005). A component-based real-time architecture for distributed supervision and control applications. In *Proceedings of ETFA2005 - 10th IEEE International Conference on Emerging Technologies and Factory Automation, 2005, Catania*. New York: IEEE Computer Society Press, pages 15–22.
- Fayad, M. E., Schmidt, D. C., and Johnson, R. E. (1999). *Building Application Frameworks: Object-Oriented Foundations of Framework Design*. John Wiley & Sons, Inc.
- Harrison, T. H., Levine, D. L., and Schmidt, D. C. (1997). The design and performance of a real-time CORBA event service. In *Proceedings of OOPSLA '97*, pages 184–200.
- Heineman, G. T. and Councill, W. T. (2001). *Component Based Software Engineering: Putting the Pieces Together*. Addison-Wesley, 1st edition.
- N. Wang, C. Gill, D. C. S. and Subramonian, V. (2004). Configuring real-time aspects in component middleware. *International Symposium on Distributed Objects and Applications - DOA 2004*.
- OMG (1999). *Data Acquisition from Industrial Systems (DAIS), Request for Proposal (RFP), OMG Document: dtc/99-01-02*.  
[http://www.omg.org/techprocess/meetings/schedule/Data\\_Acquisition\\_RFP.html](http://www.omg.org/techprocess/meetings/schedule/Data_Acquisition_RFP.html).
- OMG (2001). *CORBA Component Model*.  
<http://www.omg.org/technology/documents/formal/components.htm>.
- Rogers, G. F. (1997). *Framework-Based Software Development in C++*. Prentice-Hall, Inc.
- Sanz, R. (2003). A CORBA-based architecture for strategic process control. *Annual Reviews in Control*, 27(1):15–22.
- Schmidt, Gokhale, E. K. B. J. B. K. D. T. P. (2003). *Model Driven Middleware: A New Paradigm for Developing and Provisioning Large-scale Distributed Real-time and Embedded Applications*. Elsevier Journal. Special Issue on Model Driven Architecture.