Requirements specification and architecture design for internet-based control

Requirements Specification and Architecture Design for Internet-based Control Systems

S H Yang,
Department of Computer Science,
Loughborough University,
Loughborough, Leicestershire,
LE11 3TU UK
Email:s.h.yang@lboro.ac.uk

L S Tan,
Department of Computer Science,
Central China Normal University,
430079 Wuhan, PR China

X Chen
Department of Computer Science,
Loughborough University,
Loughborough, Leicestershire,
LE11 3TU UK

Abstract

The Internet is playing an important role not only in information retrieving, but also in industrial processes manipulation. This paper describes an approach to writing requirements specification for Internet-based control systems and to deriving architecture for this new type of control systems according to the requirements specification. Specification is described in terms of a functional model and then extended into information architecture. Distinct from the functional model, the information architecture gives an indication to the architecture of the Internet-based control systems. An integrated-distributed architecture has been derived from the functional model and the information architecture as a case study.

1. Introduction

Internet-based control system is a new concept, which has been received much attention in past years. This new type of control systems allows remote monitoring and adjustment of plants over the Internet. With the prevalence of the Internet, plants stand to benefit from the ways of retrieving data and reacting to plant fluctuations from anywhere around the world at any time.

To date, most research work on Internet-based process control has resulted in small-scale demonstrations [1-4] like Sun Microsystems and Cyberonix, Foxboro, and Valmet. A few companies try to produce Internet control systems as a control device [5]. Some researchers in this area, from higher education institutions, focus on developing web-based virtual control laboratories for distance learning purposes [6-11]. They allow a remotely-located user to conduct experiments in their control engineering laboratory via the Internet. IFAC has held the first workshop on Internet Based Control Education in Spain (2002). The ScadaOnWeb system [12] funded by the European Council from September 2001 to August 2003 targets Internet-based protocols enabling process, monitoring and optimisation via the web. It is hoped that the specific web-based approach towards the development of an online framework will eventually result in the adoption of ScadaOnWeb as an industry standard for transporting large volume of process data online.

The design process for this new sort of control systems includes requirement specification, architecture design, control algorithm and interface design, and safety analysis. The key issues are overcoming the Internet latency and ensuring the system safety [4,13].

Specifying requirements for Internet-based control systems is the first task in the design process because different requirements may lead to different control architectures. Architecture design is the second step in the design process of Internet-based control systems. The requirements specification should be met in the architecture design.

Many requirements validation techniques involve building prototypes or executable specifications or waiting until the system is constructed and then testing the whole system. It could be too late and too expensive by that time to make any change in specification for control systems although certainly much can be learned by “testing” a specification. Very little work has so far been done on requirements specification for control systems design.

Our approach is to build a functional model to describe requirements specification and then extend the functional model into information architecture. The information architecture gives an indication to the architecture of the Internet-based control systems.

2. Requirements Specification

2.1 Goals, Constraints, and Requirements

Proceedings of the 26 th Annual International Computer Software and Applications Conference (COMPSAC’02)
0730-3157/02 $17.00 © 2002 **IEEE**

Generally, a system is a set of components working together to achieve some common purpose or objective. For control systems, the goal is to maintain a particular relationship between the input to the system and the output from the system in the face of disturbances in the process [14].

Beside the basic objective implemented by the process, systems may also have constraints on their operating conditions. Constraints limit the set of acceptable designs with which the objectives may be achieved. Constraints may arise from several sources including physical limitations and safety.

Due to system constraints, the goals of a system may not be entirely achievable. These goals cannot be a requirement since it is not possible to achieve. The major task in requirements specification is to identify and resolve tradeoffs between functional goals and constraints that are conflicting or not completely achievable. The second task is ensuring that the specified behaviour of the control system will achieve the goals to an acceptable degree while satisfying the constraints.

Obviously, for Internet-based control systems, the requirements should include process monitoring and control objectives. But some of requirements require a deterministic timing regime, and therefore may not be achievable due to Web-related traffic delay. This sort of requirements, which are not entirely achievable over the Internet, should be excluded. The major task in the requirement specification is to identify tradeoffs between goals and constraints of the system that are conflicting or not completely achievable, and resolve them.

2.2 Functional Modelling of a Control System

The functional modelling consists of making a specification of a control system as expected by the user, which expresses what the control system has to do to allow the process to be controlled. These user's needs are usually represented by Data Flow Diagrams (DFD) identifying control functions on the basis of the {B, I, C} triplet [15] where

- **B** defines the behaviour of the function. It is modelled by function algorithms;
- **I** defines information produced or consumed by the control system of the function. It is modelled by the function interfaces;
- **C** defines communication. It is modelled by means of data flows connecting the functions with other functions within the system.

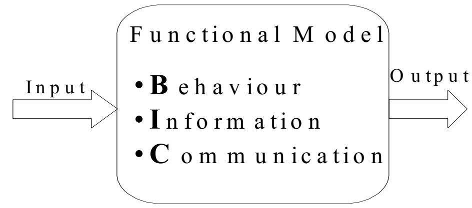

Figure 1. System functional modelling.

The structuring of the Data Flow Diagrams should allow designers to identify an actuation channel, to which an agent (e.g. an operator) can send requests to fulfil the system objective, and a measurement channel, which reports the effects of these requests on the system.

2.3 Functional Modelling for Internet-based Control Systems

The objective of establishing Internet-based control systems is to enhance rather than replace ordinary computer-based control systems by adding an extra Internet level in the control system. A block diagram of an Internet-based control system is drawn in Figure 2. The Internet-based control system [4,13] provides a way of monitoring and adjustment of process plants from any point in the Internet using standard web browsers and PC. Obvious advantages include:

- Access to the monitoring functionality independent from location.
- Use of zero cost software (standard web browsers) on the client side to access the information.
- Internally or externally collaboration between participants.

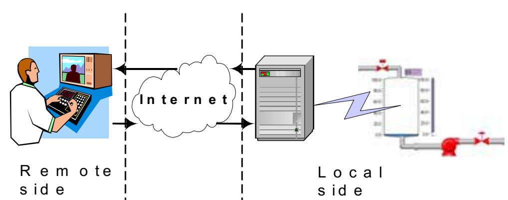

Figure 2. The block diagram of Internet-based control systems.

In order to describe the functions of Internet-based control systems in terms of the {B, I, C} triplet a Process Control Event Diagram (PCED) is adopted here and modified for the functional modelling of Internet-based control systems. The PCED was firstly introduced by the author [16] as a means to symbolically represent the flow of information between a process, actuating, measuring and controlling devices as well as operators. The original PCED illustrates the interaction between six different

Proceedings of the 26 th Annual International Computer Software and Applications Conference (COMPSAC'02)
0730-3157/02 $17.00 © 2002 IEEE

layers *(Lay)* of a controlled process, from top to down: *Operator (Op)*, *Human Input Device/Display (HID)*, *Communication (Com)*, *Computer (Comp)*, *Sensor/Actuator (S/A)*, and *Process (Proc)*. It is used to represent the exchange of information between different layers on a qualitative time scale. Assigned to these layers are three different types of nodes (Nod), the *object nodes* (Nod_{obj}), the *I/O-nodes* ($Nod_{I/O}$), the *computation nodes* (Nod_{comp}): An object node is used to specify that information is retrieved from the process through a sensor, or passed on to the process through an actuator, or that information is exchanged between the operator and the human input device (HID). The I/O-nodes mark the exchange of signals between the controller (i.e. the computer-layer) and the communication layer or the sensor/actuator level. Finally, the computation nodes represent an action of the controller: This is either a jump to another computation node or the evaluation of the controller logic in order to compute a control signal that is delivered to an actuator. The triangles specify the direction of the signal flow that is denoted by arcs (*Edg*) that connect two nodes. The sequence of the nodes in horizontal direction (from left to right) corresponds to the temporal order in which information is processed.

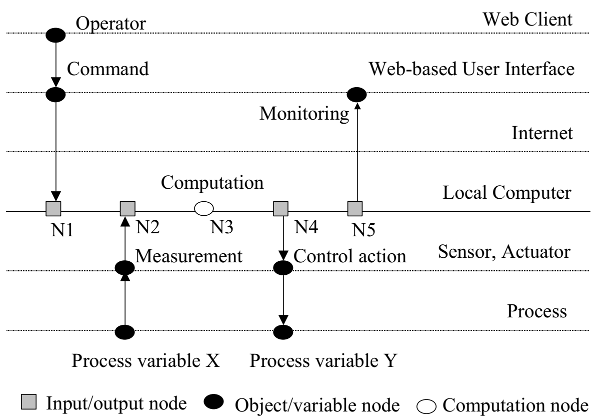

Figure 3. Functional model of Internet-based control systems.

Compared with the original PCED described above the PCED shown in Figure 3 has replaced the communication level with an Internet level. Therefore the modified PCED is designed for Internet-based control systems. If a certain signal is assumed to be emitted from the *Process* layer, the modified PCED represents the behaviour of Internet-based control systems as the flow of data and information through the layers, i.e. as the {B, I, C} triplet.

The PCED can be formally defined as a triplet:

$$PCED = (Lay, Nod, Edg)$$

in which *Lay* denotes the six different layers. INT denotes the Internet layer.

$$Lay = (Op, HID, INT, Comp, S/A, Proc)$$

Nod denotes the three types of nodes.

$$Nod = (Nod_{obj}, Nod_{I/O}, Nod_{comp})$$

Edg denotes a finite set of edges, each of which connects two nodes with data transferred as a label.

$$Edg = \{e_1, \cdots, e_n\}$$

The above formal description includes all the information in the {B, I, C} triplet functional model as follows:

$$B = \{Nod^1_{comp}, \cdots, Nod^n_{comp}\}$$

$$I = \{Nod^1_{I/O}, \cdots, Nod^k_{I/O}\}$$

$$C = Edg$$

The advantages gained from using the PCED in our approach are that the structure of the PCED is easy to be understood and can be further extended into an information architecture which gives an indication of the physical control system architecture.

3. Information Architecture

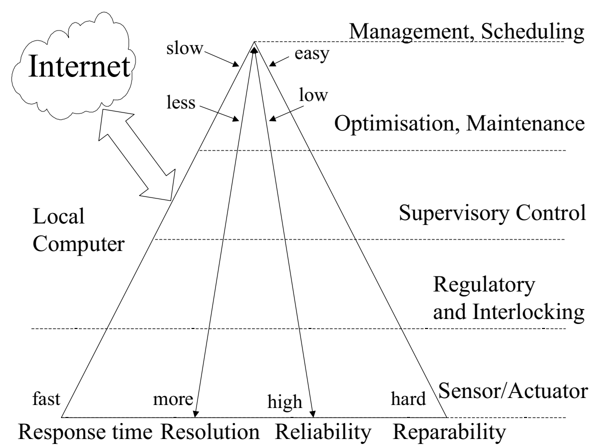

Figure 4. Information architecture.

The local computer level in Figure 3 can be further separated into 4 levels with the sensor/actuator level as shown in Figure 4, which are distinguished from each other by "4 R's" principle criteria [17];

- Response time: as one moves higher in the information architecture, the time delay which can be tolerated in receiving the data increases. For example, at the regulatory (control loop) level, data becomes "stale" very quickly. Conversely, information used at the management & scheduling level can be several days old without impacting its usefulness.

Proceedings of the 26 th Annual International Computer Software and Applications Conference (COMPSAC'02)
0730-3157/02 $17.00 © 2002 IEEE

- Resolution: Abstraction levels for data varies among all the levels in the architecture. The higher the level is, the more abstract the data is.
- Reliability: Just as communication response time must decrease as one descends through the levels of the information architecture, the required level of reliability increases. For instance, host computers at the management & scheduling level can safely be shut down for hours or even days, with relatively minor consequences. If the network, which connects controllers at the supervisory control level and/or the regulatory control level, fails for a few minutes, a plant shutdown may be necessary.
- Reparability: The reparability considers the ease with which control and computing devices can be maintained.

As shown in Figure 4 the Internet can be linked with the local computer system at any level in the information architecture, or even at the sensor/actuator level. These links result in a range of 4Rs (response time, resolution, reliability, and reparability). For example, if a fast response time is required a link to the control loop level should be made. If only abstracted information is needed the Internet should be linked with a higher level in the information architecture such as the management level or the optimisation level.

Table 1. Pros and Cons of Possible Links Between the Internet Control Level and Existing Control Levels

Existing Information Level	Information Exchange	Advantages	Disadvantages
Management level	Commercial data systems	Enable the commercial data to their customers and managers	Not suitable for real time monitoring and control tasks.
Plant-wide Optimisation Level	Global Database	Easily achieving the plant-wide information of process plants.	Not suitable for real time monitoring and control tasks.
Supervisory Level	Process Database	Easily achieving the real-time status of process plants, suitable for implementing advanced control.	Missing management information.
Regulatory Level	PLC, Control Unit	Allowing controllers to directly talk to the Internet	Introducing a high risk of being attacked by malicious hackers.
Sensor/Actuator Level	Smart-Devices	Monitoring and controlling the smart-devices directly from the Internet	Introducing a high risk of being attacked by malicious hackers.

Table 1 shows a simple evaluation for the possible links. This table can be used to guide the selection of the links between the Internet and the existing levels in the information hierarchy. For example, as shown in Table 1, smart-devices (Internet-enabled sensors and/or actuators) can directly communicate with the Internet with their Internet capability. Programmable Logic Controllers (PLC) can be directly integrated with the Internet using a Transmission Control Protocol/Internet Protocol (TCP/IP) card [18] which allows them to talk to the Internet. However, in most cases direct access to a sensor, an actuator, or a controller introduces a high risk of being attacked by malicious hackers and is probably not desirable. Furthermore, information exchange between process plants and Internet–based clients can be achieved through corporate systems – such as commercial data systems, relational databases or real-time databases, instead of control units. For example, information from the corporate system can be wrapped in a self-describing object written in the Java programming language, and seamlessly and efficiently sent to the client's workstation, ready to be published or included in usable formats. The disadvantages of these high level links are missing real-time data of the process plants.

4. Possible Implementation of the Information Architecture

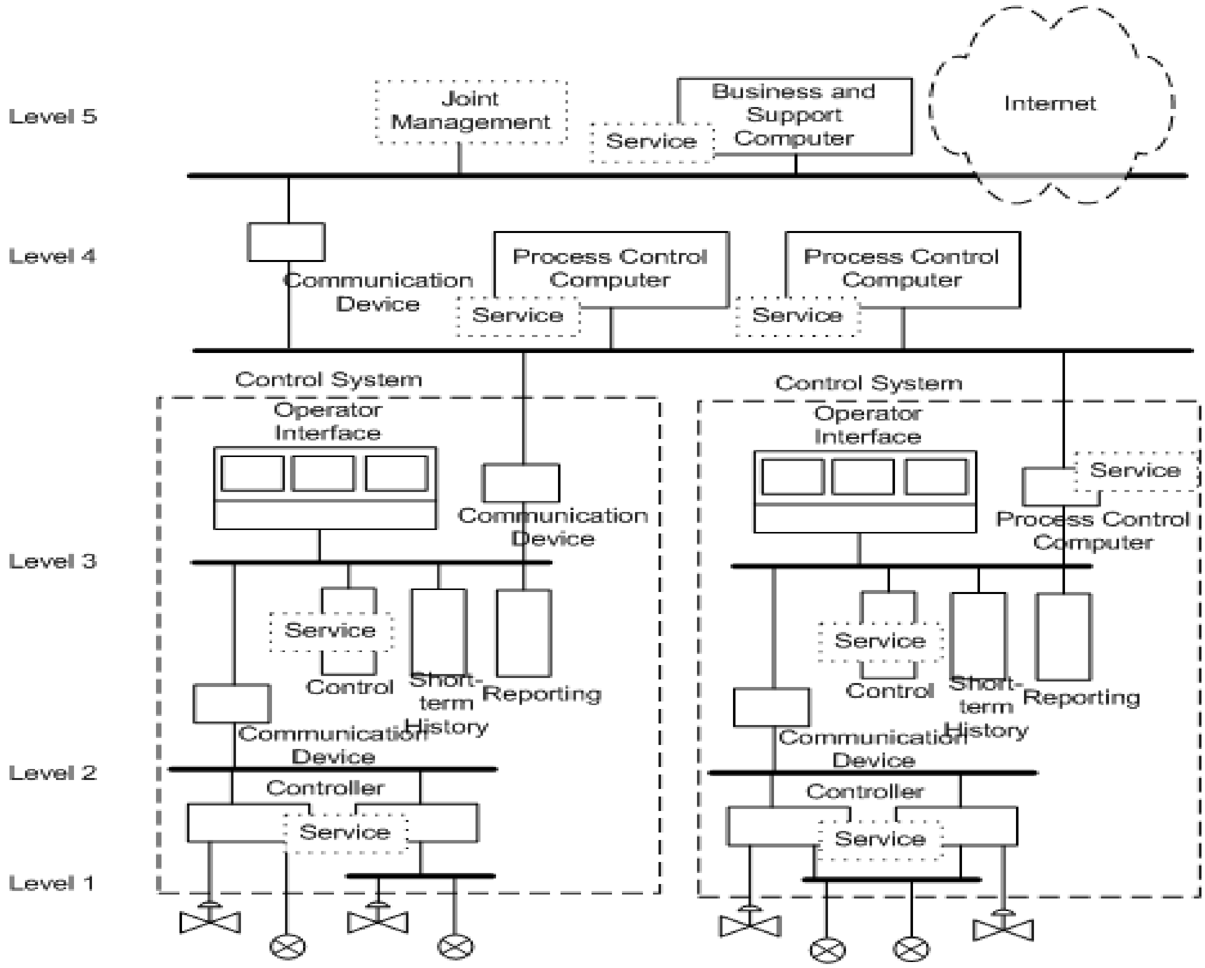

Figure 5. Integrated-distributed control architecture.

Various system monitoring and control requirements are fulfilled by the links with the Internet at the different levels in the information architecture. Considering the case that the links with the Internet at all the information levels are essential in order to meet some requirements, a possible implementation would be the integrated-distributed architecture as shown in Figure 5 discussed below.

4.1 Distributed Services in the Architecture

Nowadays, a lot of control elements have been embedded with Internet-enabled functions, for example, PLC with TCP/IP stack [18], smart control valves with a built-in wireless communication based on TCP/IP protocol, and process control computer (DCS) with an Internet gateway. All these Internet-enabled control elements provide services and form a web-centric distributed architecture for control systems. Furthermore, if some required service is not formally embedded into control elements, often it is practical to facilitate the service by assigning them to a computational support. In that case, the service should be located at the same level as its information source. Therefore, a web-centric distributed architecture of control systems is a reality. Figure 5 shows all the services represented in dotted line boxes at different information levels in a control hierarchy.

4.2 Integrated management

However, it could be turned into a nightmare for the system management and users if too many services are available simultaneously on the Internet and are not integrated properly. The joint management shown in the top level in Figure 5 is a promising way to overcome these problems. Through various physical links with the joint management, the individual services register themselves with the joint management by providing it with a set of parameters describing the service that the service can offer. Once the services have been registered, the joint management serves as the repository of all the services.

The Jini technology [19] can be used to implement the joint management.

There are several reasons for doing the management in this way. Firstly, the joint management is the only agent which directly communicates with web-clients in control systems. All control elements are located behind the joint management. Once malicious hackers are trying to attack control systems they are actually attacking the joint management rather than the control system itself. This structure will reduce the risk of being attacked by malicious hackers. Secondly, the integrated architecture reduces the number of actual links with the Internet into 1, but at the same time provide control elements with unlimited virtual links with the Internet. This structure meets the requirement of linking with the Internet at all the information levels. Finally, most control elements have a very limited capability of networking. The joint management is located at the top level of the control architecture and can be offered a great potential of networking for the control system. Therefore this structure improves the capability of networking and is consistent with congestion control in the Internet.

5. Conclusions

With the prevalence of the Internet, process managers and operators have being offered a virtual global control room for their process plants. This paper contributes on the requirements specification and architecture design for this particular type of control systems. A {B, I. C} triplet based functional model – a PCED description - is used to represent the requirement which is easy to be understood and at the same time the structure information is clearly embeded in it. The information architecture is further derived from the computer level in the PCED and assessed in terms of 4Rs (response time, resolution, reliability, and reparability). The links with the Internet at various levels in the information architecture have been evaluated. A distributed service and integrated management architecture has been derived as a case study in which the links with the Internet at all the information levels are required.

Acknowledgements

The contribution is part of the work of the EPSRC (Grant No. GR/R13371/01) funded project "design of Internet-based process control".

References

[1]. Atherton, R., "Java Object Technology can be Next Process Control Wave", *Control Engineering*, 45(13), pp. 81-85, 1998.

[2]. Halley, A., and P. Gauld, "Integration Where Java Meets Process Control", *Control and Instrumentation*, 31(4), pp. 57-58, 1999.

[3]. Kellerhals, M.B., B. Kessler, B. Witholt, "Closed-loop control of bacterial high-cell-density fed-batch cultures: production of mcl-PHAs by Pseudomonas outida KT2442 under single-Supervision and cofeeding conditions", *Biotechnology and Bioengineering,* 65(3), pp. 306-315, 1999.

[4]. Yang, S.H., X. Chen, D.W. Edwards, and J.L. Alty, "Development of an Internet-based process control system", *ESCAPE-12*, Netherlands, May, 2002.

[5]. Cushing, M., "Process control across the Internet". *Chemical Engineering,* May, pp.80-82, 2000.

[6]. Aktan, B., C.A. Bohus, L.A. Crowl, and M.H. Shor, "Distance learning applied to control engineering laboratories". *IEEE Transactions on Education*, 39(3), pp. 320-326, 1996.

[7]. Shor, M., and A. Bhandari, "Access to an instructional control laboratory experiment through the World Wide Web". *Proc. American Control Conference,* Philadelphia, PA pp.1319-1325*,* 1998.

[8]. Shaheen, M., K.A. Loparo, and M.R. Buchner, Remote laboratory experimentation. *American Control Conference,* Philadelphia, PA. pp. 1326-1329, 1998.

[9]. Overstreet, J.W., and Tzes, A., "AN Internet-based real-time control engineering laboratory". *IEEE Control Systems Magazine* 19(5), pp.320-326, 1999.

[10]. Ko, C.C., B.M. Chen, J. Chen, Y. Zhuang, and K.C. Tan, "Development of a web-based laboratory for control experiments on a coupled tank apparatus". *IEEE Transactions on education*, 44(1), pp. 76-86, 2001.

[11]. Yang, S.H., J.L. Alty, "Development of a Distributed Simulator for Control Experiments through the Internet", *Future Generation Computer Systems,* 18(5), pp.595-611, 2002.

[12]. Naylor, J., "Online data transfer standard", *The Chemical Engineers*, February, pp. 24, 2002.

[13]. Yang, S.H., X. Chen, D.W. Edwards, and J. L. Alty, "Designing Internet-based control systems for process plants", *The 4th Asian Control Conference*, Singapore, September 2002.

[14]. Levson, N.G., M.P.E. Heimdahl, H. Hildreth, and J.D. Reese, "Requirements Specification for Process-Control Systems", *IEEE Trans. on Software Engineering*, 20(9), 684-707, 1994.

[15]. Iung, B., E. Neunreuther, and G. Morel, "Engineering process of integrated – distributed shop floor architecture based on interoperable field components", *International Journal of Computer Integrated Manufacturing,* 14(3), 246-262, 2001.

[16]. Yang, S.H., O. Stursberg, P.W.H. Chung, and S. Kowalewski, "Automatic Safety Analysis of Computer-controlled Plants", *Computers and Chemical Engineering*, 25, 913-922, 2001.

[17]. Rathwell, G.A., and P.E. Ing, "Design of enterprise architectures", http://www.pera.net/Levels.html

[18]. Maciel, C.D. and C. Ritter, "TCP/IP Networking in Process Control Plants", *Computers & Industrial Engineering*, 35(3/4), 611-614, 1998.

[19]. Ghiassi, M., "Applying Java technology to manufacturing integration and automation", *International Journal of Flexible Automation and Integrated Manufacturing*, 7(3&4), 355-377, 1999.