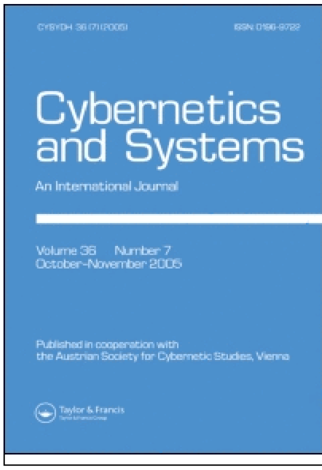

Cybernetics and Systems
An International Journal
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713722751

SMART SYSTEMS INTEGRATION: TOWARD OVERCOMING THE PROBLEM OF COMPLEXITY

Edward Szczerbicki [a]
[a] Gdansk University of Technology, Gdansk, Poland

Online Publication Date: 01 February 2008

To cite this Article: Szczerbicki, Edward (2008) 'SMART SYSTEMS INTEGRATION: TOWARD OVERCOMING THE PROBLEM OF COMPLEXITY', Cybernetics and Systems, 39:2, 190 — 198

To link to this article: DOI: 10.1080/01969720701853475
URL: http://dx.doi.org/10.1080/01969720701853475

Cybernetics and Systems: An International Journal, 39: 190–198

ISSN: 0196-9722 print/1087-6553 online
DOI: 10.1080/01969720701853475

SMART SYSTEMS INTEGRATION: TOWARD OVERCOMING THE PROBLEM OF COMPLEXITY

EDWARD SZCZERBICKI

Gdansk University of Technology, Gdansk, Poland

Information is seen as one of the main resources that systems analysts try to use in an optimal way. In this short article, we show how this resource can be used in integration issues. We introduce the problem of information-based integration, propose a solution, and discuss briefly future trends in this area.

INTRODUCTION

Systems become increasingly complex. Their decomposition into smaller units is the usual way to overcome the problem of complexity. This has historically led to the development of atomized structures consisting of a limited number of autonomous subsystems that determine their own information input and output requirements—that is, can be characterized by what is called an information closure. In a real-world context, autonomous subsystems consist of groups of people and/or machines tied by the flow of information both within a given subsystem and between this subsystem and its external environment (Szczerbicki 2006; Esteve 2002; Kowalczuk and Orlowski 2007). Autonomous subsystems can still be interrelated and embedded in larger systems, as autonomy and independence are not equivalent concepts. These ideas are recently gaining very strong interest in both academia and industry, and the atomized approach to complex systems analysis is an idea whose time has certainly come (Yang and Reidsema 2007; Orlowski 2002).

Address correspondence to Edward Szczerbicki, Gdansk University of Technology, Gdansk, Poland. E-mail: Edward.Szczerbicki@zie.pg.gda.pl

Complex systems (e.g., manufacturing) are often viewed as sets of components (e.g., agents, subsystems) supporting separate functions. Many organizations operate in this highly compartmentalized manner. It appears that the general direction of systems in the future, however, is toward linking together function-specific agents into fully integrated entities. An integrated system is a system that consists of agents/subsystems that efficiently contribute to the task, functional behavior, and performance of a system as a whole. It is believed that such an integration can be achieved through the flow of information. *Integration*, as used in this article, should not be confused with integration at a physical level by means of computer networks or computer buses. Rather, the semantics of integration is addressed—the information that subsystems should share.

While structuring the approach presented, the first consideration was to design some tools that can be easily implemented as components of an intelligent system supporting development of system configurations integrated by the flow of information. Due to the complexity and creativity associated with the early stages of such a development, it is quite clear that the way a practicing analyst solves a system configuration problem cannot be easily implemented. This explains the need for new tools and approaches that solve the problem but at the same time can be supported by a computer. Elements of such an approach are presented in this article.

BACKGROUND

We propose a three-stage approach for the development and analysis of complex systems. The involved stages are system decomposition, subsystem modeling representation, and integration at the level of information flow. Figure 1 depicts the general underlying idea behind this approach.

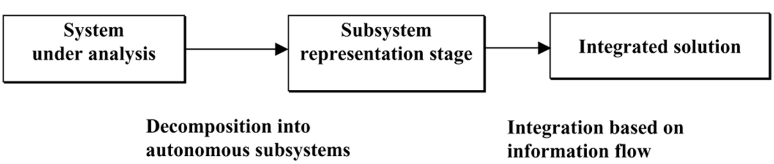

Figure 1. Overview of a three-stage approach to complex systems integration.

The essence of the information-based integration problem can be formulated as below:

Given the informational inputs and outputs of autonomous subsystems, find the overall system being designed that meets the desired functions and is integrated through the flow of information.

The theoretical framework to provide support for systems integration as outlined above includes the following fundamentals:

1. The syntax for connections of autonomous subsystems
2. The mechanism for guiding the generation of such connections
3. The integration algorithm

The aim of this article is to briefly outline these fundamentals.

The author, together with his collaborators, has been researching information-based integration issues since the early 1990s, starting at the University of Iowa (Kusiak, Szczerbicki, and Vujosevic, 1991), Iowa City, USA; continuing at the GMD FIRST, Berlin, Germany (Szczerbicki 1994); and later working on these issues at the University of Newcastle, Newcastle, Australia, and Gdansk University of Technology, Gdansk, Poland (Szczerbicki 2003, 2006). The aim of this brief report, which is based on previous research publications by the author, is to overview the integration problem from the perspective of the author's experience and to place it historically among the work of others.

The integration problem has been gaining very strong interest in both academia and industry since the 1990s. Model development and synthesis (which resembles autonomous systems development and integration) is frequently based on the general systems theory, and it uses hierarchical structures and a number of model-based concepts (Esteve 2002; Wyzalek 1999; Rolstadas and Andersen 2000; Yang and Reidsema 2007). In the work of Raczkowsky and Reithofer (1998), the development of a hierarchical communication model for coordination of a set of agents performing several functions was addressed probably for the first time. The problem of coordination of multiagent manufacturing systems developed to fulfill their functional requirements advocating a decentralized approach in which each agent has relative autonomy over its own actions was first discussed by Pacholski (1998). The role of the flow of information in the process of integration was discussed in Prakken (2000) and Orlowski (2002).

Traditionally, information system analysts have been solving the integration problem in an ad hoc manner. What we propose through our research is a formal integration approach suitable for computer implementation.

The information-based integration problem should be seen as one of the challenges within the broader, more general aim to enhance systems performance through the flow of information. Engineering, operations research, information science, and management science use scientific and engineering processes to design, plan, and schedule increasingly more complex systems in order to enhance their performance. One can argue that systems have grown in complexity over the years mainly due to increased desire for resource optimization combined with a greater degree of vagueness in the system's environment. Information is seen as one of the main resources that analysts try to use in an optimal way in complex systems. Proper design of information flow, management, use, and maintenance (i.e., information engineering) is critical to systems' abilities to act appropriately in an uncertain environment—in other words, to act intelligently.

INFORMATION-BASED INTEGRATION

Autonomous subsystems are matched using informational input and output defined at the representation stage (Fig. 1). For example, in the domain of environmental engineering, information may represent geographical positioning, urban planning, sources of pollution, traffic data, and the like. In the domain of manufacturing systems, informational inputs and outputs may represent material availability, tool availability, machine availability, number of parts produced, and number of products assembled. After the matching has been accomplished, the informational input variable of a given subsystem represents the value of the informational output variable of the subsystem to which it has been connected. For example, if the input variable X of autonomous subsystem AS2 is matched with the output variable Y of AS1, then the syntax of this connection would be given as

$$\text{AS1.Y} - - \rightarrow \text{AS2.X} \tag{1}$$

Similar simple syntax can be used for all structures that can be produced during the integration process. These structures are enclosed into higher-level subsystems using ports. The informational input and output ports provide an interface to the subsystem environment. This interface is used to develop hierarchical structures. With the syntax of autonomous subsystem connections in place, the mechanism for guiding the generation of such connections is required. The mechanism needs to represent qualitative system theoretical knowledge, so it is based on "IF . . . THEN" production rules.

Generation of connections between elements in the model base of autonomous subsystems is guided by the following production rules (Szczerbicki 2003; Kusiak, Szczerbicki, and Vujosevic 1991):

Rule 1
IF there is only one element left
THEN do not generate connections
Rule 2
IF a single element that is left includes boundary inputs and outputs only
THEN it is an overall system
Rule 3
IF there is more than one element
THEN select a connection for an input boundary element
Rule 4
IF there are elements other than the boundary elements
THEN do not specify any connections that involve boundary elements only
Rule 5
IF an element is an input boundary element
THEN it cannot accept an input from any other element
Rule 6
IF an element is an output boundary element
THEN it cannot provide an input to any other element
Rule 7
IF two elements have identical output and input variables
AND there are no production rules that prevent one from connecting them
THEN specify the connections for these elements
Rule 8
IF there are no elements with identical input and output variables
AND there are elements with partially identical input and output variables
AND there are no production rules that prevent one from connecting them

THEN specify the connection for these elements beginning with the closest match
Rule 9
IF a connection for an input boundary element has been specified
THEN continue with selecting connections for elements that have not been listed in the specifications
Rule 10
IF there are boundary elements only
THEN specify connections between them

The above production rules are domain independent and were structured using the underlying general systems theory. The analyst may, however, add domain-based production rules. They may follow, for example, the safety requirements, emission data, traffic data, or other constraints imposed by the analyst.

The last tool needed for simulation of the information-based integration process is an integration algorithm. The algorithm guides the simulation process across various levels of integration. It was developed with the assumption that in order to enter the next level of integration, it is enough to generate just one connection in a given step. Elements taking part in integration that are not matched at integration level i are considered for matching at level $i+1$. The algorithm terminates at the level at which it will no longer be possible to match subsystems into pairs (i.e., no connections will be generated). At each integration level, the production rules presented above are fired during the simulation process to generate connections between remaining integration elements. The integration algorithm is the last element of the fundamentals of a framework to support systems integration process. The algorithm includes the following simple steps (Szczerbicki 2003; Kusiak, Szczerbicki, and Vujosevic 1991):

Step 1. Define database of autonomous subsystems; set level $= 1$.
Step 2. Generate connections between elements at current level.
Step 3. If no connections are generated, stop; otherwise, match elements in database into pairs using the existing connections.
Step 4. Define informational input and output variables for subsystems generated by the matching process.
Step 5. Remove from the database all elements that have taken part in the matching process.

Step 6. Add to the database all subsystems generated by the matching process.
Step 7. Set level = level + 1 and go to step 2.

The presented framework for information-based integration has been applied for numerous real-life cases in which simulation was used to arrive at an integrated solution. Some particularly successful applications included system modeling and integration for a coal mine (Szczerbicki and Charlton 2001), integrated agile manufacturing strategy (Szczerbicki and Williams 2001), modeling for steel processing (Szczerbicki and Murakami 2000), integration of maintenance services (Szczerbicki and White 2003), plan integration for intelligent knowledge management platform development (Sanin and Szczerbicki, 2005), and model integration for complex economic systems (Szczerbicki and Waszczyk 2006). In all these cases, all three stages of system decomposition, representation, and integration were present (see Fig. 1), integration representing the last step in the process of arriving at a systems structure integrated by the flow of information.

FUTURE TRENDS AND CONCLUSION

Information-based system integration is gradually becoming one of the main challenges of the information age. The most prominent future trend in this area is focused on using Web-based technologies to tackle problems of integration. Within this trend, we have a rapid increase of research efforts toward developing metamodeling architectures and interoperability of Web-enabled information flows (Terrasse, Becker, and Savonnet 2003), semistructured data integration (Liu and Ling 2004), schema integration (Castana et al. 2003), and Web-based aggregation architectures (Bussler 2003). The explosive popularity of the Web makes it an ideal integration tool, as it opens the possibility to integrate geographically distributed systems. Also, in the future the Web may become a universal platform to synthesize a number of integration approaches (e.g., the one presented in this article) into one universal Web-based integration interface.

Information flow integration is one of the major activities of the design process of an integrated system. The outcome of the integration process is integration of the overall system through the flow of information.

In this research report, the integration problem is discussed from a historical point of view. It is formulated as follows: given the informational inputs and outputs of autonomous subsystems, find the overall system being designed that meets the desired functions and is integrated through the flow of information.

REFERENCES

Bussler, C. 2003. Application service provider aggregation architecture. In *Web-enabled systems integration: Practices and challenges*, edited by A. Dahanayake and W. Gerhardt. London: IDEA, pp. 81–98.

Castana, S., Antonellis, V. D., Capitani S., and Melchiori, M. 2003. Data schema integration in Web-enabled systems. In *Web-enabled systems integration: Practices and challenges*, edited by A. Dahanayake and W. Gerhardt. London: IDEA, pp. 125–139.

Esteve, Y. V. 2002. Notional logic of systems. *Cybernetics and Systems* 33: 189–202.

Kowalczuk, Z. and Orlowski, C. 2007. Knowledge management based on formal models of software project management. *Cybernetics and Systems: An International Journal* 38: 587–610.

Kusiak, A., Szczerbicki, E., and Vujosevic, R. 1991. Intelligent design synthesis: An object oriented approach. *International Journal of Production Research* 29: 1291–1308.

Liu, M. and Ling, T. W. 2004. Towards semistructured data Integration. In *Web-enabled systems integration: Practices and challenges*, edited by A. Dahanayake and W. Gerhardt. London: IDEA, pp. 559–576.

Orlowski, C. 2002. Soft modelling support for managing knowledge-based information technology projects. *Cybernetics and Systems* 33: 401–412.

Pacholski, L. 1998. Fuzzy logic application in ergonomic renewal of multiagent manufacturing systems. *Cybernetics and Systems* 29: 715–728.

Prakken, B. 2000. *Information, organization and information systems design*. Berlin: Kluwer.

Raczkowsky, J. and Reithofer, W. 1998. Design of consistent enterprise models. *Cybernetics and Systems* 29: 525–552.

Rolstadas, A. and Andersen, B. 2000. *Enterprise modelling*. Berlin: Kluwer.

Sanin, C. and Szczerbicki, E. 2005. Set of experience: A knowledge structure for formal decision events. *Foundations of Control and Management Science* 3: 95–113.

Szczerbicki, E. 1994. Flow of information—The semantics of integration. *Systems Analysis, Modelling, Simulation* 16: 229–243.

Szczerbicki, E. 2003, June. *Knowledge-based integration for multi-agent systems in information rich environments*. Proceedings of International Symposium on

Information Technologies in Environmental Engineering, ITEE'2003, Gdansk, Poland (proceedings on CD-ROM published by ICSC-NAISO Academic Press, Canada).

Szczerbicki, E. 2006. Knowledge base for intelligent information management. *Cybernetics and Systems: An International Journal* 37: 673–683.

Szczerbicki, E. and Charlton, G. 2001. System modelling and simulation for a coal mine. *Systems Analysis, Modelling, Simulation* 40: 111–123.

Szczerbicki, E. and Murakami, R. 2000. Simulation-based decision support platform for steel processing. *Systems Analysis, Modelling, Simulation* 39: 461–478.

Szczerbicki, E. and Waszczyk, M. 2006. Modelling of complex economic systems: Toward conceptual platform. *International Journal of Enterprise Information Systems* 2: 66–77.

Szczerbicki, E. and White, W. 2003, September. Simulation modelling for managing complex systems: Case study for maintenance services. *Proceedings of International Conference on Information Systems Application and Technology* (pp. 15–22), ISAT'2003, Wroclaw, Poland.

Szczerbicki, E. and Williams, J. 2001. Developing agile manufacturing strategy with AweSim simulation platform. *Cybernetics and System* 32: 719–736.

Terrasse, M.-N., Becker, G., and Savonnet, M. 2003. Metamodeling architectures and interoperability of Web-enabled information systems. In *Web-enabled systems integration: Practices and challenges*, edited by A. Dahanayake and W. Gerhardt. London: IDEA, pp. 329–345.

Wyzalek, J. 1999. *Systems integration success*. London: Auerbach.

Yang, Q. and Reidsema, C. 2007. Design information handling in a knowledge based intelligent design system. *Cybernetics and Systems: An International Journal* 38: 549–574.