THE ARCHITECTURE OF COMPUTER-BASED INFORMATION PROCESSING AND THE EFFECTIVENESS AND ADAPTABILITY OF SYSTEMS

Roberto R. Kampfner
Department of Computer and Information Science
College of Engineering and Computer Science
The University of Michigan-Dearborn
Dearborn, Michigan 48128

Abstract

The effectiveness of a computer-based information system can be defined in terms of its ability to support effectively the functions of the system it serves. In fact, a basic principle for the design of computer-based information systems stated earlier (Kampfner, 1997) asserts that in order to provide effective function support, a computer-based information system must be compatible with the structure, dynamics, and adaptability of the system it supports. The effect of computer-based information processing on adaptability can be associated with its effect on the interdependence of the subsystems of the system it serves. In fact, a basic architecture design principle favors the adaptability of the system as a whole (Kampfner, 2008) by appropriately reducing subsystem interdependence (Conrad, 1983). In this paper we propose a top-down approach to the design of computer-based information systems in which the architecture design principle is applied first in order to find an architecture that favors the adaptability of the system being supported by the information system. The effective support of function support must then be achieved on the basis of this architecture.

Keywords: architecture, computer-based information processing, adaptability, subsystem interdependence, functional subsystems, effective function support.

Introduction

Earlier we discussed two basic principles for the design of computer-based information systems. One of these principles refers to the compatibility that must exist between a computer-based information system and the structure and dynamics of the system it supports and contribute to its adaptability in order for the support it provides to be effective (Kampfner, 1997). The other design principle asserts that the architecture of computer-based information processing must reduce the interdependence between the subsystems of the system it supports in a manner consistent with its adaptability (Kampfner, 2008). Other design principles for the architecture of information systems have been discussed (see for example, Erl, 2008). The principles discussed here, however, are more general in nature. In this paper we argue that these two design principles are consistent with each other and that they can be applied jointly using a top-down approach to design that starts with architecture design. A good architecture design is important. According to Rozanski and Woods (2005) a given architecture or architectural style expresses a fundamental structural organization scheme for software systems. Once an architecture that contributes to adaptability has been found, a computer-based information system that provides effective function support can be designed on the basis of this architecture.

The view of information processing as an aspect of the dynamics of systems (Kampfner, 1998) is basic to our approach. This view allows us to consider computer-based information processing an integral part of the dynamics of the system it supports. Because of the influence that the structure of any system has on its dynamics, the architecture of a computer-based information system, which is an expression of its structure and an integral part of the structure of the system it supports, necessarily contributes to the influence that the structure of a system exerts on all the processes that participate in its dynamics. An important consequence of this is that the architecture of a computer-based information system influences not only the dynamics of computer-based information processing but it also influences all the other aspects of the dynamics of the system it supports, including the human information processing aspect of this dynamics and other aspects that do not directly involve information processing. It is important to notice that the architecture of a computer-based information system participates in the influence that the structure of the system it supports exerts on all the information-processing processes that constitute its dynamics, whether they are computerized or not, as well as on all the other aspects of this dynamics.

The abstraction-synthesis methodology (ASM) of information systems development (Kampfner, 1987, 1997) provides conceptual tools and methods for the analysis and design of computer-based information systems in a manner consistent with the view of information processing as an aspect of the dynamics of systems. In a nutshell, the ASM helps to determine the information needs of the functions to be supported by the computer-based information system on the basis of the dynamics underlying those functions. In the ASM, determining the information needs of these functions in fact amounts to understanding and describing the information processing aspect of their underlying dynamics.

The ASM also allows for the analysis of the requirements that a computer-based information system capable of fulfilling the information needs of a particular set of functions must meet. In the ASM the requirements of a computer-based information system are in fact requirements for the effective support of functions in a manner consistent with the adaptability of the system as a whole.

The ASM follows a synthetic approach to the design of information systems. This synthetic approach to design is a predominantly top-down process in which the architecture of the computer-based information system can be design first under the guidance of the architecture design principle. The design can proceed on the basis of this architecture under the guidance of the design principle for the effective support of function. With the conceptual tools and methods it provides, the ASM allows for the design of computer-based information systems that effectively support the functions of a system and also contribute to its adaptability.

We illustrate this in the context of an order processing subsystem of a manufacturing organization. We show how the information needs of the functions to be supported can be expressed as an aspect of the underlying dynamics, and how a set of computer-based information system requirements derived from these information needs can be used as a basis for the design of effective function support. Then we show how these requirements provides a basis on which a computer-based information system capable of providing effective function support and adaptability can be designed with the guidance of the design principles mentioned above.

Determining the Information Needs of Hierarchical, Distributed Control Systems

The structure of any system influences its own dynamics. In a computer-based information system this dynamics is the way in which it receives, stores, processes, and communicates information. The structure of a system also influences its adaptability. An important part of this influence is the effect that the structure of a system has on the interdependence between its subsystems. The architecture of a computer-based information system expresses its structure in terms of variables representing structural features that the information system designers can manipulate. Architecture design is therefore a process in which the structural features of a computer-based information system that influence dynamics in a manner consistent with the effective support of function and adaptability can be determined. According to the function support principle, however, the compatibility of the architecture of the information system with the structure and dynamics of the system it supports is essential to its ability to provide effective function support and to contribute to adaptability. This compatibility, however, can only be achieved on the basis of a suitable description of the structure with which the computer-based information system must be compatible. The OCSM formalism which allows for the description of the hierarchical, distributed control structures needed for adaptability provides such a suitable description of the system's structure.

The OCSM formalism is a structure of sets and relations that models the structure of a system S in terms of the functions it performs and the relationships that exist between them. More precisely, the OCSM formalism models a system S as a structure (F, R), where $F= \{F_C, F_O\}$ is the set of functional subsystems (or functions) of the system S. These functions are of two types: control and operational. F_C is the set of control functions of S and F_O is the set of operational functions of S.

R = {INCLUDES, CONTROLS, COORDINATES} is a set of binary relations that specify the relationships between the functions that the system S performs, that is, between the functions in F.

The INCLUDES relation is the binary relation INCLUDES $\subseteq$ F $\times$ F. It is defined as follows.

INCLUDES = $\{<f_n, f_m> \mid f_m, f_n \in F$, and f_m performs part (or all) of the function $f_n\}$.

The INCLUDES relation relates all the functional subsystems, or functions, $f_i \in F$, to each other because any such function, with the exception of those at the lower level, includes one or more functions $f_i \in F$, and any of these functions, with the exception of the system as a whole, is included in some other function. However, the OCSM formalism imposes specific constraints on the types of functions that can be included in a particular function of S. An important constraint, for example, is that each operational subsystem can include only one control function at the next lower level.

The binary relation COORDINATES $\subseteq F_C \times F_O$, where $F_C, F_O \subseteq F$, relates each control subsystem fc_i to the operational subsystems it controls. It is defined as follows.

COORDINATES = $\{<fc_{i0}, fo_{ij}> \mid fc_{i0} \in F_C, fo_{ij} \in F_O$, and fc_{i0} sets goals and parameters to fo_{ij} and coordinates its operation with that of the other operational functions fo_{ij} , $j = 1, 2, \ldots , m_i$ toward the achievement of the goals of their parent system}.

The COORDINATES relation captures the relationship that the OCSM formalism defines between a control function fc_{i0} and each of the operational functions fo_{ij} that it controls. It is important to notice that in the OCSM, any operational function INCLUDES at the next lower level one or more operational functions and only one control function that COORDINATES them.

The CONTROLS relation is the binary relation CONTROLS $\subseteq F_C \times F_C$, $F_C \subseteq F$, defined as

CONTROLS = $\{< fc_{i0}, fc_{j0} > \mid fc_{i0}, fc_{j0} \subseteq F_C$, and fc_{i0} verifies whether fc_{j0} achieved its goals and, if not, takes actions that take it closer to the achievement of its goals}.

Any operational function, with the exception of those at the lowest level, includes, at the next lower level, a control function and one or more operational functions. This control function CONTROLS each of its operational siblings, that is, each of the operational functions that its parent operational function also includes.

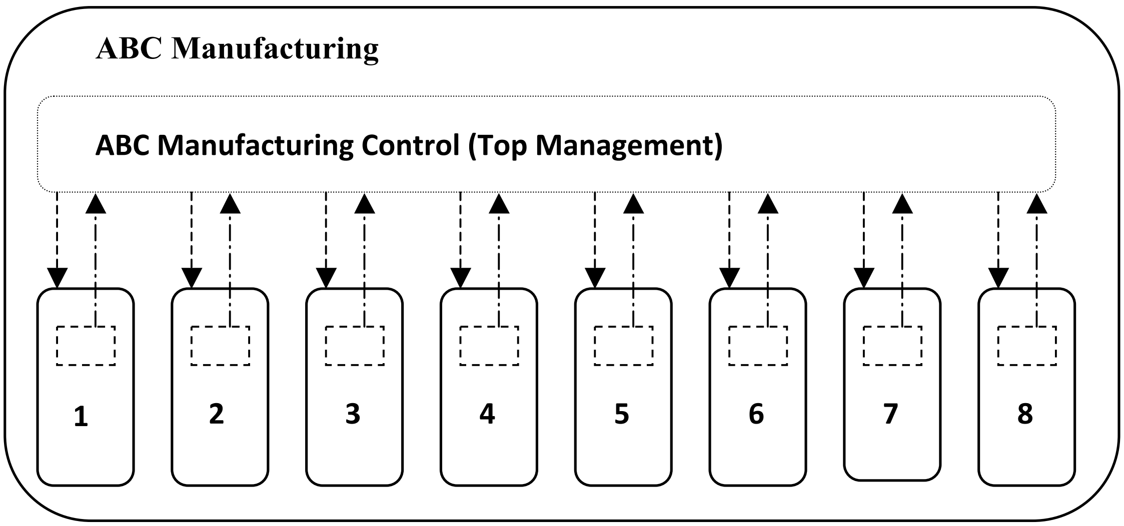
We write CONTROLS(fc_{i0}, fc_{j0}) to mean $<cf_i, of_j> \in$ CONTROLS, that is, that the ordered pair $<cf_i, of_j>$ belongs to the CONTROLS relation. Similarly, we write INCLUDES(f_m, f_n) to mean $<f_m, f_n> \in$ INCLUDES, and COORDINATES(fc_{i0}, fo_{ij}) to mean $<fc_{i0}, fo_{ij}> \in$ COORDINATES.

Figure 1 shows a hierarchy diagram that shows the structure of a system as described using the OCSM formalism.

The meaning of the relations of the OCSM is illustrated in Figure 1 that shows the top-level hierarchy diagram of a manufacturing organization called ABC Manufacturing. Figure 1 shows the eight top level functions of ABC Manufacturing. The functions into which one of these top level functions is subdivided, the Marketing function, are shown in Figure 2. The hierarchy diagram of Figure 2 shows that the Marketing function is divided into Sales, Advertising, and Market research functions, and that the Sales function is subdivided into the functions of Order preparation and verification and Order processing. Notice that the OCSM description of the structure of a system can be done in a flexible, modular way by describing specific functions and their components in separate diagrams if desired.

The OCSM description of a system shows its basic structure in terms of the functions it performs and the way they relate to each other. This view of the system's structure is important because it focuses on the structure of its functions and such it can be directly associated with the dynamics underlying this functions and, more specifically, with the role of information processing in this dynamics. In order to provide an effective support to a particular set of functions in a system must be compatible with the dynamics underlying these functions. This compatibility entails for it to be an integral part of a dynamics that allows for the functions to achieve their goals. In more detail, this compatibility requires for the computer-based information system to provide the information that is required from it so that the functions it supports are successfully performed. This clearly requires for the computer-based information system to have the necessary information processing capacity, to produce precisely the information that is required from it, and to interact with its users, that is, the people and systems that participate in this dynamics, in a manner that allows for the information to be exchanged in an appropriate manner.

The information needs of the functions should be specified in a manner that allows for the requirements of the information system to be specified in a manner consistent with the effective support of function. In the ASM, the information needs specification is based on the informational interactions between the functions of the system, especially the function that the computer-based information system will support. These interactions are determined by the functions involved in the interactions and the role that information plays in their interrelationships. Important in this respect is the influence that the structure of a system, as expressed by the relationship between its functions, exerts on its dynamics. More specifically, what needs to be determined on the basis of this structure is the information that the functions must exchange between them and with the external environment in order for the functions and the system as a whole to achieve their goals. These informational interactions constitute an important part of the information processing aspect of the dynamics underlying these functions and, consequently, specify an important part of their information needs. The informational interactions diagram for the functions participating in the order-fulfillment process of ABC Manufacturing is shown in Figure 3.



LEGEND:

COORDINATES relation: CONTROLS relation:

1. Accounting and Finance **2. Marketing**
3. Manufacturing **4. Physical Distribution**
5. Purchasing and Inventory **6. Information Systems**
7. Personnel **8. Research and Development**

Figure 1 Top-level hierarchy diagram of ABC Manufacturing; it shows the main control subsystem and eight major operational subsystems are shown at this level. The relations of the OCSM are represented as follows:
The INCLUDES relation is represented by placing the boxes that represent the included functions within the box that represents the function that includes them. For example, Notice that, among others, INCLUDES(ABC Manufacturing, Accounting and Finance) and INCLUDES(ABC Manufacturing, Marketing) are elements of the INCLUDES relation, and that COORDINATES(ABC Manufacturing Control, Marketing) and COORDINATES(ABC Manufacturing Control, Physical Distribution) are elements of the COORDINATES relation. The representation of the CONTROLS relation is made explicit in Figure 2.

Once we know the information flows that must be maintained in order for a system to perform its functions, the question as to how the computer-based information system will effectively contribute to maintaining these information flows in a manner that allows these functions to achieve their goals naturally arises. This is an important question, for it is the question as to how the effective support of function can be built. The answer to this question is given by the first design principle mentioned above, namely that the computer-based information system must be designed in a manner compatible with the structure and dynamics of the system it supports and contribute to its adaptability (Kampfner, 1987, 1997). In the ASM, the requirements of the information system express the conditions that have to be met in order to achieve the compatibility with the structure.

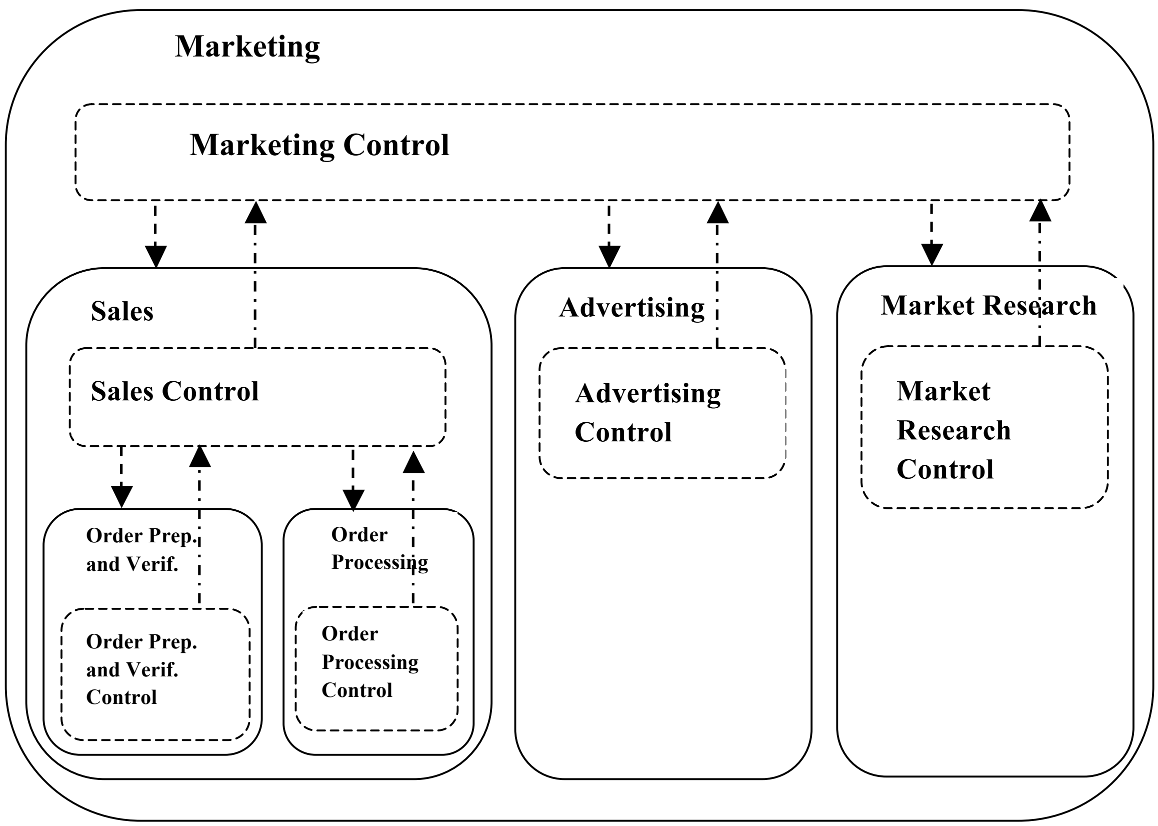


Figure 2. A hierarchy diagram of the Marketing Function of ABC Manufacturing. Three of the functions in which the Marketing function is subdivided (Sales, Advertising, and Market Research) are shown in the diagram. Notice in this figure that INCLUDES(Sales, Sales Control) and INCLUDES(Sales, Order Processing) are elements of the INCLUDES relation. Also, CONTROLS(Marketing Control, Sales Control) and CONTROLS(Sales Control(Order Preparation and Verification Control) are elements of the CONTROLS relation.

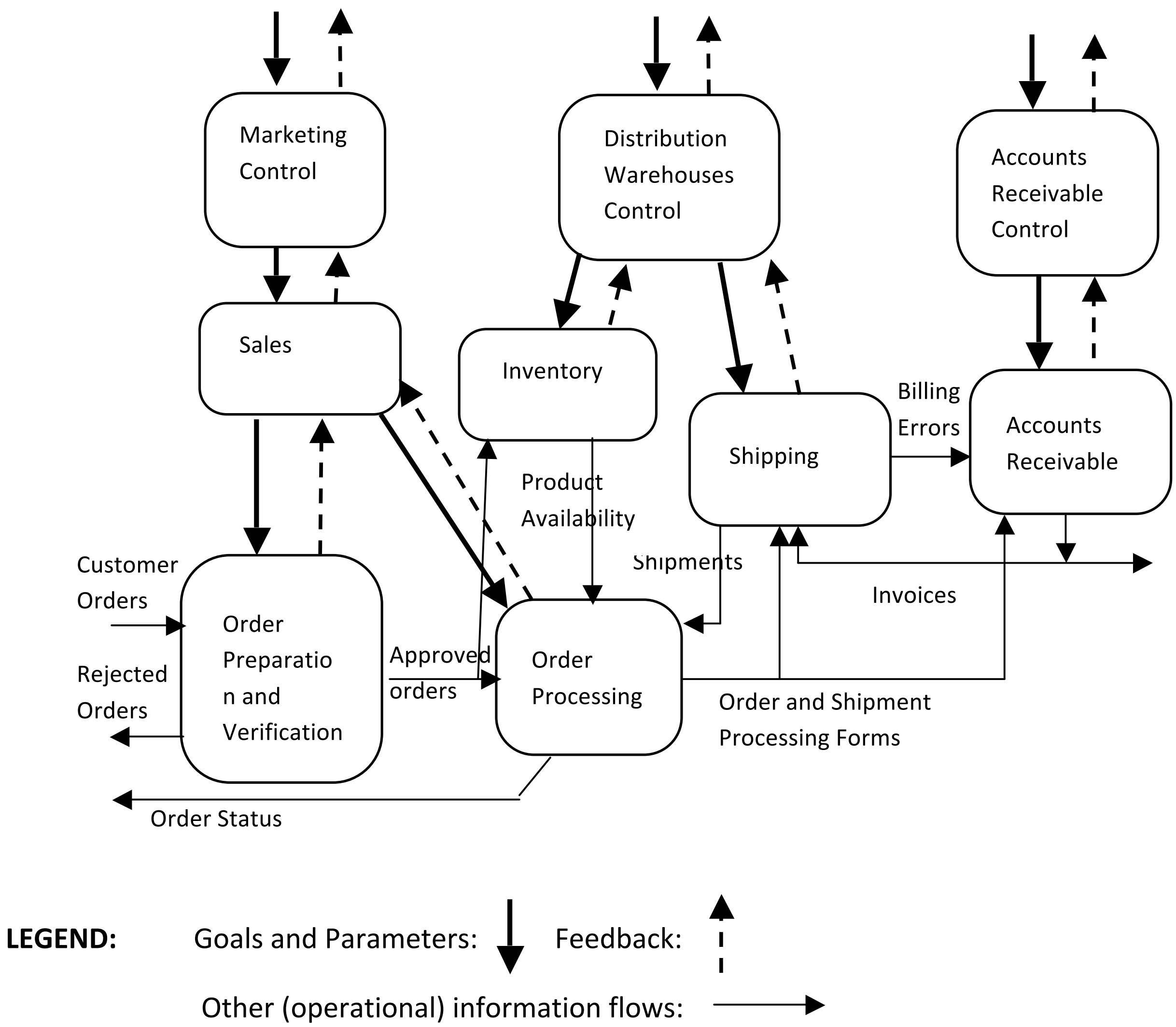


Figure 3. Informational interaction diagram for the order fulfillment system of ABC Manufacturing. The interactions between functions shown in the diagram are determined on the basis of the hierarchy diagrams of the OCSM description.

The OCSM description of a system and the informational interaction diagrams that are derived from this description are essential components of the specification of its information needs. The reason is that they describe the structure and dynamics with which the computer-based information system must be compatible. The informational interaction diagrams in fact describe an important part of the information processing aspect of the dynamics of the system in the context of its structure as provided by the OCSM description. This clearly provides a solid basis for the design of an information system that is compatible with the structure and dynamics of the system it supports; a solid basis on which the requirements of the information system can be determined. More specifically, the analysis of information system requirements determines the information processing functions needed to provide the functions of the system being supported with the information they need to achieve their goals. Since these information processing functions are specified in the context of the dynamics underlying the functions they support, their specification can be done in a manner compatible with this dynamics and, consequently, with the effective support of function.

Central to the requirements of a computer-based information system are the information processing functions (IPFs) that need to be performed in order for the functions of the system being supported to achieve its goals. These IPFs are in fact information transformations of information representations that need to occur in order for the processes that perform the functions of the system to be successfully executed. These IPFs describe the information processing aspect of the dynamics of the functions to be supported by the information system. These IPFs need to be performed in order for the information flows described by the informational interactions diagrams to be maintained. Some of these IPFs will be performed by the computer-based information system, others by the people that contribute to the functions of the system by solving problems, performing tasks, and making decisions. The requirements of the information system describe these IPFs and the conditions that have to be met in order for the computer-based information system to perform some of these functions in a manner compatible with the underlying dynamics.

The LIPS/LIPN formalism (Kampfner, 1985) can be used for the description of the IPFs that are needed to maintain the information flows described by the informational interaction diagrams. The LIPS/LIPN formalism describes in an abstract, design independent manner the structure of the information processing functions (IPFs), the conditions that trigger their activation, and the information flows that they exchange. Moreover, the LIPS/LIPN formalism, relates the IPFs to the functions that perform these IPFs in order to achieve their goals and to the other processes that jointly with them perform these functions. The relationship between the IPFs that process the information and the functions they support is established on the basis of the informational interaction diagrams because the information processing functions that the LIPS/LIPN describes can be related to the functions they support via the information flows that the informational interaction diagrams describe. In the ASM, this relationship is in fact established in the context of the information needs specification of which the information flow diagrams form a part. More specifically, the information flows that the IPFs of the LIPS/LIPN description must maintain are in fact the information flows between the functions of a system that the informational interaction diagrams describe.

The computer-based information system will perform some of the IPFs included in the LIPS/LIPN description. Specifying which those IPFs are is in fact the basis on which the *logical requirements* of the computer-based information system are determined. The ASM considers two other types of information system requirements, the *performance requirements* and the *user interface requirements*. In the requirements specification of a particular information system the performance and the user interface requirements further describe the functions (or, more precisely, the processes that perform these functions) to which the logical requirements refer. The three types of requirements are determined in the ASM on the basis of the information needs specification which, in addition to the OCSM description of the structure of the system and the informational interaction diagrams, includes the description of relevant parts of the structure and dynamics of the functions that the computer-based information system being developed will support.

The LIPS/LIPN formalism (Kampfner, 1985) is a structure of P = (X, Z, τ, σ). In this structure, $X = \{x_i\}$ is a set of *information structures*. All the information structures used and produced by the processes that perform the functions of a system form part of the information processing aspect of these processes, that is, of the dynamics underlying the functions that these processes perform. Of the information structures occurring in a system, those amenable to automated processing and that can potentially be handled by the computer-based information system being developed must be included in this set.

$Z = \{p_j\}$ is a set of information processing functions (IPFs). Formally, an IPF p_j is a structure $p_j = (a_j, I_j, O_j,)$, where a_j denotes an effective procedure describing computations performed by p_j; $I_j \subseteq X$ is the set of information structures input to p_j; and $O_j \subseteq X$ is the set of information structures output from p_j. It is important to notice here that the sets X and Z jointly describe an important part of the information processing structures that induce the information processing aspect of the dynamics of a system: The information structures represent forms that information takes in a system, the IPFs that transform them represent the information processing aspect of the processes that concurrently with them perform the functions of the system.

$\tau = \{t_m\}$ is the set of *triggers*. In the LIPS the triggers represent events occurring in a system that result in the *activation* of one or more IPFs.

More specifically, we define a trigger as follows:

Let $C = \{c_m\}$ be a set of predicates specifying the necessary conditions for the activation of one or more IPFs in Z.

$$t_m \in \tau \text{ is a trigger} \Leftrightarrow \exists p_i \in Z \text{ and } \exists c_j \in C, \text{ such that } p_i \text{ activated} \Rightarrow c_j \text{ true.}$$

We will also say that a trigger t_m is on if its associated predicate c_j is true. As conditions that determine the activation of IPFs, the triggers represent an important connection between the structure and dynamics of a system and, more specifically, between the structure of a system and the information processing aspect of its dynamics.

$\sigma = \{\sigma_I, \sigma_O, \sigma_P, \sigma_H, \sigma_T\}$ is a set of relations. These binary relations are described next.

$\sigma_I \subseteq X \times Z$ is the *input* relation. It is formally described as

$$\sigma_I = \{< x_i, p_j> \mid x_i \in X, p_j \in Z, \text{ and } x_i \text{ is an input to } p_j\}.$$

Each pair $< x_i, p_j>$ in the *input* relation relates some information structure x_i to the IPF p_j of which it is an input.

$\sigma_O \subseteq X \times Z$ is the *output* relation. It is formally described as

$$\sigma_O = \{< x_j, p_k> \mid x_j \in X, p_k \in Z, \text{ and } x_j \text{ is an output of } p_k\}.$$

Each pair $< x_j, p_k>$ in the *output* relation relates some information structure x_j to the IPF p_k of which it is an output.

$\sigma_P \subseteq Z \times Z$ is the *precedence* relation. It is formally described as

$$\sigma_P = \{< p_j, p_k> \mid p_j, p_k \in Z, \text{ and } \exists x_i \in X \text{ such that both } \sigma_I(x_i, p_k) \text{ and } \sigma_O(x_i, p_j) \text{ hold}\}.$$

Each pair in the *precedence* relation relates some IPF p_j to an IPF p_k if p_j produces an information structure x_i that is an input to p_k. The *precedence* relation is a characteristic of the structure of information processing in a system. As such it contributes to the influence that this structure has on the dynamics of information processing in that system and more generally on the the dynamics of the system as a whole.

$\sigma_H \subseteq Z \times Z$ is the *hierarchy* relation. It is formally described as follows:

Let $p_i, p_j, p_q, p_l \in Z$ be IPFs, with input sets $I_i, I_j, I_q, I_l \subseteq X$, respectively. Then

$$\begin{aligned}\sigma_H = \{<p_i, p_j> \mid (&\forall x_n, x_n \in I_i, \text{ either } x_n \in I_i \cap I_j \vee \exists p_q \text{ such that } \sigma_P^*(p_q, p_i) \text{ holds} \\ &\qquad \wedge \forall x_m, x_m \in I_q \Rightarrow (x_m \in I_q \cap I_j)) \\ &\wedge (\forall x_k \in O_i, x_k \in O_i \cap O_j \vee \exists p_l \text{ such that } p_i(\sigma_P^*)p_l \text{ holds} \\ &\qquad \wedge \forall x_p, x_p \in O_l \Rightarrow (x_p \in O_l \cap O_j))\}.\end{aligned}$$

Each pair $<p_i, p_j>$ in the *hierarchy* relation relates an IPF p_i to an IPF p_j whenever p_i performs part of the computations that p_j performs. In this sense, p_i can be said to be a sub procedure of p_j.

$\sigma_T \subseteq \tau \times Z$ is the *trigger* relation. It is formally described as

$\sigma_T = \{< t_m, p_k> \mid t_m \in \tau, p_k \in Z, \text{ and } p_k \textit{ activated} \Rightarrow t_m \text{ is on}\}$.

Each pair in the *trigger* relation relates a trigger t_m to an IPF p_k if whenever t_m is on p_k can be activated. The trigger relation is a characteristic of the structure of a system that is very closely related to the dynamics that such a structure induces. In a manner similar to the precedence relation, the trigger relation is a characteristic of the structure of information processing in a system that is very closely related to the information processing aspect of its dynamics.

The LIPS description of the structure of information processing in a system can be done both formally, as a structure of sets and relations and, equivalently, in a graphical form using information flow diagrams. Because of the advantages of a graphical representation the information flow diagrams are an important part of the specification of the logical requirements of an information system. The information flow diagram of Figure 4 illustrates some of the logical requirements of the order fulfillment information system for the manufacturing organization for which the information needs were illustrated in Figures 1, 2, and 3.

Each type of information system requirement plays a specific role in the requirements specification of a computer-based information system. The *logical* requirements specify the computations that need to be performed in order for the functions that the information system will support to achieve their goals. From these computations, those that can be automated can be incorporated into the processes that the computer-based information system will perform. The *performance* requirements of an information system specify the response times, throughput and spatial distribution of the processes that perform the functions to be supported by the information system. These requirements play an important role in determining the processing speed, the storage capacity, and the data communication capabilities that the computer-based information system must provide. The *user interface* requirements specify the characteristics of the interaction between the computer-based information system and its users that the computer-based information system must support. Clearly, a complete requirements specification must include the three types of requirements specified in a manner in which they are consistent with, and complement each other. The performance requirements of the order fulfillment information system of the manufacturing organization, for instance, must provide information about the levels of activity that the processes involved in these operations are expected to undertake including the amounts of information that need to be processed, the response time required from these processes, and the places where the processes that need the information that the information system produces (i.e. people and processes that perform the sales, inventory control, packing, delivery, and invoicing functions) are located. The user interface requirements, on the other hand, must specify the way in which the computer-based information system must interact with its users so that the functions being supported are performed successfully.

The Architecture of Computer-based Information Processing and the Effectiveness and Adaptability of Systems

In the ASM, the compatibility of the computer-based information system with the structure and dynamics of the functions it supports and with the adaptability of the system as a whole can be established by making sure that it meets the logical, performance, and user interface requirements. This, of course, requires that such requirements be consistent with this compatibility. Once a specification of information system requirements that is consistent with this compatibility has been obtained, the goal of architecture design is to find an architecture that is consistent with a structure and dynamics that meets the requirements of the computer-based information system.

The architecture must induce a pattern of distribution of computer-based information processing that supports the flow of information between functions that the informational interaction diagrams of the information needs specification describe. The LIPS/LIPN formalism provides a means of verifying this compatibility in terms of the relations that it establishes between the information processing functions it describes and the information flows that these information processing functions make possible. More specifically, it provides a means of verifying that the IPFs that the computer-based information system performs are consistent with the informational interactions that must exist between the functions (Kampfner, 1985, 1987, 1997). In the ASM the informational interactions that must exist between the functions of a system are determined on the basis of the description of structure that the OCSM provides. Since the LIPS/LIPN formalism describes the information processing structure that is needed to support precisely these informational interactions, it follows that an architecture that is compatible with the LIPS/LIPN description is compatible with the effective support of function.

The architecture design principle mentioned earlier (Kampfner, 2008) aims at achieving a sufficiently low degree of interdependence between the subsystems of the system it supports as a means of increasing its adaptability. It considers three types of subsystem interdependence. The first type of interdependence occurs between the computer-based information system and the subsystems that it directly supports. Because it is an integral part of the structure of the host system the architecture of computer-based information processing influences the first type of interdependence. What we refer to as the second type of interdependence involves subsystems of the host system other than the computer-based ones. As an integral part of the structure of the host system, the architecture of computer-based information processing influences the second type of interdependence as well. An important part of this influence is a reduction of the interdependence between the subsystems of the host system that occurs because a new computer-based information system provides to some of these subsystems information that they formerly received from other subsystems. This clearly reduces their interdependence. It is important to notice, however, that in doing so, the computer-based information system becomes interdependent with the subsystems it support thus creating the first type of interdependence. The third type of interdependence is between the functional components (i.e. functional subsystems) of the computer-based information system. As an expression of the structure of information processing, the architecture of the computer-based information system strongly influences the third type of interdependence. By aiming at a reduction of these three types of interdependence that is consistent with adaptability, the architecture design principle sets the stage for the achievement of the effective support of function. Figure 5 illustrates an architecture that incorporates the IPFs shown in Figure 4. This architecture can be tested as to its consistency with the effective support of function and with adaptability.

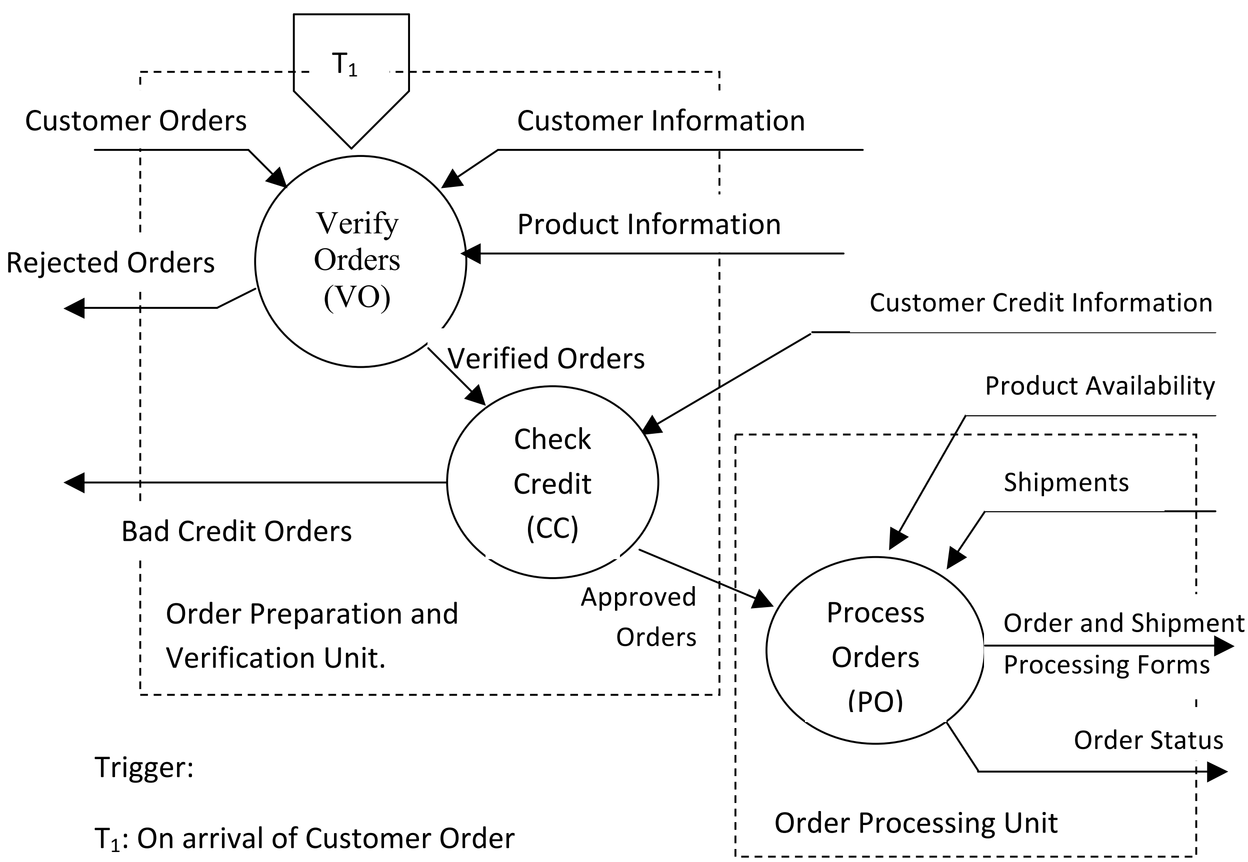


Figure 4. Information flow diagram showing the Information processing Functions (IPFs) Verify Order (VI), Check Credit (CC), and Process Order (PO). The arrows show information flows between these IPFs and between them and the external environment.

Conclusions

Our claim that the architecture design principle is consistent with the function support principle was demonstrated in a constructive fashion because both principles can be applied following a top-down approach to design in which the architecture of the system is designed first. This top-down approach to design was earlier shown to lead to the achievement of the effective support of function and adaptability through an adequate degree of integration of information processing into the functions it supports (Kampfner, 2006). Here we emphasize the need to achieve a degree of interdependence between subsystems that is good for adaptability. This requires that the initial focus of design be the compatibility of the architecture with the structure of the system being supported (as described with the OCSM). This compatibility must be achieved in a manner consistent with a degree of subsystem interdependence that is sufficiently low, so that the contribution of the computer-based information system to adaptability is increased as much as possible. Once an appropriate architecture design is obtained, it provides a solid basis for the development of a computer-based information system that provides effective function support and contributes to adaptability.

References

Conrad, M. *Adaptability*, New York: Plenum Press, 1983.

Erl, T. *SOA Principles of Service design*, New York: Prentice Hall, 2008.

Kampfner, R., "Formal Specification of Information Systems Requirements," Information Processing and Management, Vol. 21, No. 5, pp. 401-414, 1985.

Kampfner, R. “A hierarchical Model of Organizational Control for the Analysis of Information Systems Requirements,” Information Systems, Vol. 12, pp. 143-154, 1987.

Kampfner, R. “Function Support as an Information Systems Development Paradigm,” World Futures, Vol. 50, pp. 689-702, 1997.

Kampfner, R. “Dynamics and Information Processing in Adaptive Systems,” BioSystems, Vol. 46, pp. 153-162, 1998.

Kampfner R. “Adaptability and the Integration of Computer-based Information Processing into the Dynamics of Organizations,” *BioSystems*, Vol. 85, pp. 30-36, 2006.

Kampfner, R. "A basic principle for the architecture of computer-based information processing." *SIG on Information Systems Design and Information Technology, 52nd Annual Meeting of the International Society for the Systems Sciences*, University of Wisconsin, Madison, Wisconsin, July 13th to July 18th 2008.

Rozanski, N. and E. Woods (2005). *Software Systems Architecture*, Addison-Wesley, New York.