COMPUTER-BASED INFORMATION SYSTEMS AS A MEANS OF AUTOMATING THE USE OF KNOWLEDGE ABOUT INFORMATION PROCESSING IN COMPLEX 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

Computer-based information systems play an increasingly important role in the way in which complex adaptive systems such as modern organizations process information. The effectiveness with which computer-based information systems support the functions of a system is therefore a basic, underlying design goal. In this paper we explore the role that some basic principles for the design of computer-based information systems stated earlier play as a guide for the development of computer-based information systems that provide effective function support and contribute to adaptability. We discuss some design factors on the light of these principles.

One of the design principles that we stated earlier (the function support principle) aims at the development of computer-based information systems that provide effective function support and contribute to adaptability (Kampfner, 1997, 2002). Another related design principle (the architecture design principle) aims at reducing subsystem interdependence in the system that the computer-based information system will support (Kampfner, 2008). In this paper we explore the idea that some of the information processing knowledge that the computer-based information system needs can be transferred to it only when it needs that knowledge. This, of course, should be done in a manner consistent with the design principles mentioned above, that is, in a manner consistent with the effective support of function and adaptability.

Keywords: computer-based information systems, information processing knowledge, information processing, dynamics, human information processing, meta-knowledge

INTRODUCTION

Information processing can be considered an aspect, indeed as an integral part, of the dynamics of systems (Kampfner, 1998). Any system uses information in order to control, and coordinate the processes that perform its functions. It is in this broad sense that the dynamics of a system, which comprises the processes that perform its functions, can be seen as the specific way in which a particular system processes information. Information processing in a system or, equivalently, the information processing aspect of its dynamics, can be seen as a manifestation of the 'knowledge' about information processing that the system possesses. The notion of information processing described above is indeed very broad. It applies not only to information processing as performed directly by humans or by devices or systems of human design, such as computers and computer-based information systems. It applies to any dynamic system since any such system needs to process information in order to perform, control, and coordinate the processes that perform its functions and achieve its goals. This view confers to information processing a fundamental role in the dynamics of a system, hence in its ability to perform its functions and to achieve its goals.

The dynamics of complex systems such as modern organizations is composed of a variety of processes including physical, human, social, and economic ones. As an aspect of their dynamics, information processing is correspondingly complex: it takes a variety of forms that act concurrently with the other aspects of the dynamics. Especially important to us is the fact that it combines human, computer-based, and other forms of information processing. Human information processing is capable of representing, organizing, transforming, manipulating, and communicating information in a variety of ways, both explicitly and through the actions that we take and the processes that we control. Human information processing is capable of exercising judgment, creativity, and general problem solving capabilities. In addition to that, we humans can acquire, develop, and use knowledge about information processing and can process information in a conscious manner.

The way a system changes its behavior can be seen as a manifestation of the knowledge that it has of how to evolve and to adapt to changes in its environment. This knowledge can be seen as inherent to the system. Part of this knowledge is implicit in its structure and dynamics in the sense that it is not stated in any explicit form. We mentioned earlier that the information processing aspect of the dynamics of a system can be seen as a manifestation of the information processing knowledge that it possesses. The information processing knowledge that a system possesses is, however, only an aspect of the knowledge that it has about all of its dynamics and, consequently, it can be considered an integral part of this broader knowledge. In the systems of interest to us, this knowledge is constantly changing, which means that these systems are constantly developing new ways of processing information. Part of the knowledge that a system has about information processing is meta-knowledge, that is, knowledge about information processing knowledge. This meta-knowledge includes the ability of a system to change its information processing knowledge and, consequently, the way in which it processes information. In the systems of interest here parts of the information processing knowledge and the corresponding meta-knowledge (that is, the knowledge of a system about how it processes information) usually exist in an explicit form. This includes the knowledge that people have about computer-based information systems, and the knowledge that is expressed in the documentation of their functions, of the information systems development processes, and of their operation and maintenance. Other parts of their meta-knowledge, however, may exist only implicitly in their structure and dynamics.

Information processing can be seen as an aspect of the dynamics of systems (Kampfner, 1998) and as a manifestation of the knowledge about information processing that it possesses. In complex adaptive systems such as modern organizations the knowledge about information processing exists in a variety of forms, including the form the knowledge takes in the minds of people, such as the designers, implementers, and users of the computer-based information systems. Information processing knowledge is also described in documents and media accessible to people, and in documents and files that computers can process such as data files, databases, and source code and object code of computer programs. To be manifested, or used, the information processing knowledge of a system needs to become part of the information processing aspect of its dynamics, that is, it needs to be embodied in the processes that perform the functions of these systems.

The structure of a system influences its dynamics at all levels. As part of the information processing aspect of the dynamics, the way in which the knowledge about information processing is manifested, or processed, is also influenced by the structure of the system. In this paper we focus on the influence that the architecture of a computer-based information system, as an expression of the structure of the larger system, exerts in the way in which this larger system uses its information processing knowledge and how this influence can be directed toward and increase of adaptability in a manner consistent with the effective support of function. More specifically, our focus is on a design that allows for some of the information processing knowledge to be transferred to the computer-based information system only when it is needed.. With this architectural feature, the responsibility of storing, updating, and maintaining this knowledge on a permanent basis falls outside the system that uses it on a temporary basis. This raises the issue of how the knowledge to be transferred can be stored, updated, and maintained in the larger system and how this can be done in a manner that effectively supports the function and adaptability of the larger system. Although this issue is outside the scope of this paper, some of the results obtained here apply to it in an indirect manner.

Transferring knowledge to a computer-based information system only when it is needed helps reduce subsystem interdependence, hence adaptability. Section 2 discusses the effect of subsystem interdependence on the adaptability of systems from the standpoint of M. Conrad's formal theory of adaptability. Section 3 discusses the advantages and disadvantages of importing information processing knowledge to a computer-based information system only when this knowledge is needed. The emphasis is on the effect that architecture design has on the ability of the computer-based information system to provide effective function and to contribute to the adaptability of the system it supports. Section 4 discusses some of the factors and tradeoffs involved in the design of computer-based information systems with the capability of using knowledge transferred to them. Section 5 presents our conclusions and some ideas for future research in this area.

INFORMATION PROCESSING KNOWLEDGE, SUBSYSTEM INTERDEPENDENCE, AND THE ADAPTABILITY OF SYSTEMS

Computer-based information systems support the functions of the systems they serve on the basis of the information processing knowledge they possess. This knowledge becomes available to them through the computer-based information systems development process which, in this sense, can be thought of as a process in which information processing knowledge is transformed into a form that the computer-based information system being developed can use. This typically consists of algorithms that the computer programs perform, interfaces through which the computer-based information system exchanges information with its users and other external systems, and procedures that specify how the computer-based information system should be operated and maintained. To be effective this body of knowledge must always reflect the information system requirements of the system being supported. Consequently, it needs to be updated whenever the changes that occur in the system being supported affect these requirements. Transferring some of the information processing knowledge that it needs to the computer-based information system frees it from the need to store, update, and maintain this knowledge. An important advantage of this is that it reduces the interdependence between subsystems that arises form the need of making correlated modifications to the representation of this knowledge when it changes. This, in turn, tends to increase the adaptability of the computer-based information system and the adaptability of the system as a whole.

Adaptability is the ability of a system to cope with the uncertainty of its environment. According to Conrad (1983) a property of systems that has a significant effect on adaptability is subsystem independence. Conrad also asserts that subsystem independence is good for adaptability as long as the cost, in terms of an increased need of coordination, does not upset the benefits. An increase of subsystem independence or, equivalently, a reduction of subsystem interdependence, causes an increase in adaptability because it increases the modifiability of the system which makes it more able to cope with the uncertainty of the environment. It does, however, increase the cost of coordinating the more independent subsystems that result.

Conrad (1983) defines the adaptability of a system in terms of the uncertainty of its transition scheme and how it relates to the uncertainty of the transition scheme of its environment. Formally, this is expressed as follows:

$$U(S') - U(S'|E') + U(E'|S') \geq U(E') \qquad (1)$$

where S' is the transition scheme of system S and E' is the transition scheme of the environment E. U(S'), the potential uncertainty of the behavior of the system S, is the entropy of its transition scheme, defined by Shannon and Weaver (1962) by the expression $U(S') = -\Sigma_{ij}(p_{ij}\log(p_{ij}))$, where S' is the transition scheme of system S, and p_{ij} is the probability of a transition of system S from state i to state j. The other entropies are defined in a similar manner. U(E') is the potential uncertainty of the behavior of the environment. There are two conditional entropies. One is U(S'|E'), which measures the uncertainty of the transition scheme of the system given the state of the environment. In fact, U(S'|E') measures the inability of the system to anticipate the behavior of the environment.

The anticipation component of the adaptability of system S, given by U(S') – U(S'|E'), measures the uncertainty of the behavior of S that is not potentially decorrelated from the behavior of the environment E. The other conditional entropy, U(E'|S') measures the uncertainty in the behavior of the environment given the state of the system. This is also interpreted as the indifference with respect of the behavior of the environment that the system can afford. Notice that for Eq. (1) to be meaningful in terms of statistical properties inherent to particular systems, U(E') must be considered as the maximum uncertainty of the environment that is tolerable by the system (Conrad, 1983). The uncertainties on the left-hand side of Inequality (1) consist also of maximum potential uncertainties.

To study the effect of subsystem interdependence on adaptability we need to consider the structure of a system in terms of the subsystems composing it and the relationships that exist between them. In the case of complex adaptive systems such as modern organizations the organizational control systems model or OCSM (Kampfner, 1987, 2002b), is a formalism that helps us do that. The OCSM allows us to express the structure of a system as a hierarchy of self-controlling organizational functions, or functional subsystems. In this type of structure, each functional subsystem can be decomposed into a control subsystem and a set of smaller functional subsystems. The control subsystem monitors and coordinates the operation of the operational subsystems of its parent system, i.e. its operational siblings. In the OCSM modeling formalism all the functional subsystems that are considered to be adaptive consist of a control subsystem that represents its adaptive capabilities, and the operational subsystems it controls. A simplified system S with a control subsystem S_0 and two functional subsystems, S_1 and S_2 is shown in Figure 1.

The adaptability of a hierarchical, compartmental system corresponds in fact to the product of the adaptabilities of the transition schemes of its subsystems (Conrad, 1983). We can represent the adaptability of system S described in Figure 1, for example, as follows:

$$U(S_0'S_1'S_2') - U(S_0'S_1'S_2'|E') + U(E'|\, S_0'S_1'S_2') \geq U(E') \qquad (2)$$

Using effective entropies, the term can be expanded as

$$U(S_0'S_1'S_2') = U_e(S_0') + U_e(S_1') + U_e(S_2') \qquad (3)$$

here the expansion of $U_e(S_1')$ in Eq. (3) for example, is given by

$$U_e(S_1') = (1/3)[U(S_1') + (1/2)U(S_1'|S_0') + (1/2)U(S_1'|S_2') + (1/2)U(S_0'|S_2') + U(S_1'|S_0'S_2')] \qquad (4)$$

In Eq. (4), the modifiability term $U(S_1')$, that represents the modifiability of subsystem S_1, should be sufficiently large. The conditioned modifiability terms, also called independence terms, $(1/2)U(S_1'|S_2')$ and $U(S_1'|S_0'S_1')$, on the other hand, express the extent to which the modifiabilities of the subsystem S_1 and the other subsystems, S_0 and S_2 will be correlated.

The OCSM representation is particularly useful for the analysis of the effect of transferring information processing knowledge to a computer-based information system so that it can use, but it doesn't need to update and maintain this knowledge. The concept of effective entropies helps us formalize the effect on the adaptability of a system taking into account not only the contribution of the effect of the variability of its subsystems, but also of the effect that the nature and the magnitude of their interactions have on its modifiability. This concept is particularly convenient when we want to compare the effect on the adaptability of a system, say S, of transferring knowledge to the computer-based information system (say, S_1) when it needs it, as opposed to having to store, update, and maintain this knowledge on a permanent basis. Transferring knowledge to S_1 reduces the independence terms of Eq. (4), that is, $U(S_1'|S_0')$, $U(S_1'|S_2')$ and $U(S_1'|S_0'S_1')$, to the extent that S_1, does not need to correlate the updates to the changes in the information processing knowledge that it uses with corresponding changes in subsystem S_2. This reduction of interdependence is shown schematically in Figure 2.

The transfer of information processing knowledge: Architectural considerations

We have seen the development of computer-based information systems as a process in which some of the information processing knowledge of a system is automated. In general this entails a transformation of the knowledge that the system has about how to process information in order to perform its functions. Typically, as this knowledge is transformed, it is in fact embodied in the structure and dynamics of the computer-based information system and it is used or manifested through its dynamics, that is, the processes that it performs. The design option that we are considering here allows for some of the information processing knowledge that the computer-based information system uses to be transferred to it only when it is needed. With this architectural feature, the responsibility of storing, updating, and maintaining this knowledge on a permanent basis falls outside the system that uses it on a temporary basis. Some of the advantages of this approach from the standpoint of adaptability were discussed in the previous section. In this section we discuss the effect of this architectural feature on the ability of the information system to provide effective function support and point to the main factors and tradeoffs involved.

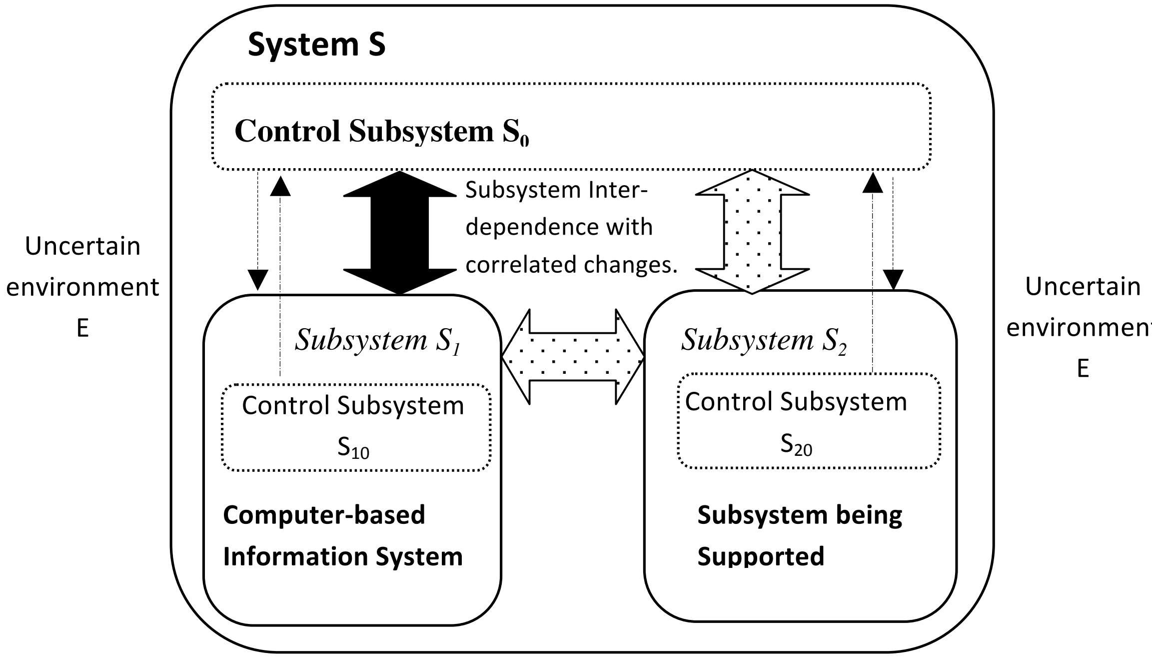

Figure 1 Subsystem interdependence that includes the need of correlated changes of information processing knowledge. The hierarchy diagram shows a system S with 3 subsystems: S_0, S_1, and S_2. The diagram shows the structure of system S following the OCSM modeling formalism (Kampfner, 1987).

With this architectural feature, the responsibility of storing, updating, and maintaining this knowledge on a permanent basis falls outside the system that uses it on a temporary basis. As an expression of the structure of the larger system, however, the architecture of a computer-based information system must achieve the goal of increasing adaptability in a manner consistent with the effective support of function. This raises the issue of how the knowledge to be transferred can be stored, updated, and maintained in the larger system and how this can be done in a manner that effectively supports the function and adaptability of the larger system. Although this issue is outside the scope of this paper, we will deal here with an important part of it, that is, the issue of how this transfer of knowledge can be done in a manner consistent with the effective support of function. Two distinct but related problems need to be solved before this design decision can be successfully made. The first one is to determine the information processing knowledge that is to be transferred. The second one is to determine the basic architectural features and the technologies used to transfer and use the knowledge. Let us consider in more detail each of these problems.

Determining the information processing knowledge to be transferred Concerning the type of knowledge to be transferred, the more structured the knowledge the more amenable that it is for the computer-based information system to use it. One reason for this is that structured knowledge can be represented using more structured processes and functions. This makes the more structured knowledge more likely to be chosen as the knowledge to be transferred to the computer-based information system when it needs it. Structured knowledge includes the use of algorithms that produce specific results or trigger specific actions in specified situations or conditions. In a customer order processing system, for example, this kind of knowledge involves procedures to determine discounts to specific types of customers depending on the amount and type of products they order. Less structured knowledge can sometimes be divided into more structured and a less structured parts. When this can be done, the more structured parts can be transferred to the computer-based system for processing while the less structured parts can be handled by the users of the computer-based system. The less structured parts of this knowledge may deal with the control and coordination of the more structured ones. In this case, the users of the computer-based system, representing the subsystems being supported, would control and coordinate the execution by the computer-based system of the more structured processes. Needless to say, the use of the transferred knowledge may involve the interaction between the computer-based system and its users.

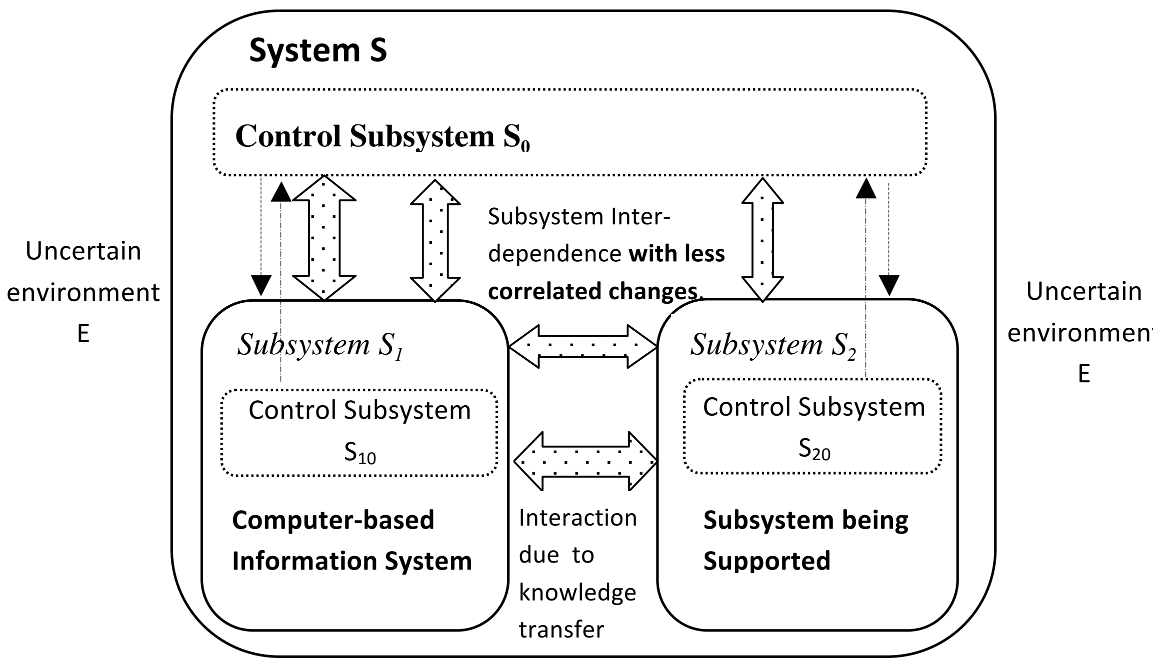

Figure 2 Subsystem interdependence with less need of correlated changes information processing knowledge. The hierarchy diagram shows a system S with 3 subsystems: S_0, S_1, and S_2. using the OCSM modeling formalism. The diagram also depicts the tinteractions due to the transfer of information processing knowledge to the computer-based information system.

Less structured knowledge of a domain specific kind can sometimes be transferred to the computer-based system when it is needed in order to be processed. The representation of this knowledge requires the use of artificial intelligence techniques that are appropriate to the problem domain in question. Some aspects of the use of this kind of knowledge, including the definition and specification of the representations of this knowledge, their conversion to an executable form, as well as the supervision of their execution and the evaluation of the results obtained will require some kind of human intervention.

The less structured types of information processing knowledge, including knowledge about unstructured functions and processes and knowledge that is not domain specific, will always require human intelligence in order to be processed. Therefore, the latter kind of knowledge is not the kind that can be transferred to the computer-based information system.

Another important factor to consider when transferring knowledge to the computer-based system about the functions it supports is the amount of resources needed to process the knowledge. An excessive use of resources, such as processing time and storage space, could make the system too costly or unable to meet the requirements of the information system and, consequently, unable to provide effective function support. The way in which the knowledge is stored and organized may cause an excessive use of resources in order to be transferred or make such transfer of knowledge too complicated or inefficient. Capturing this knowledge in an appropriate form represents in this case an activity that would significantly improve the information processing aspect of the functions being supported by the computer-based information system.

Yet another important factor to be considered in order to determine whether certain kind of knowledge should be transferred is the frequency with which the knowledge is used by the computer-based information system. If the knowledge needs to be transferred very frequently, the cost involved should be weighed against the gains in adaptability that the transfer may produce.

The uncertainty of the environment facing the subsystems to be supported by the information system is an extremely important factor to consider when determining whether a certain kind of knowledge should be transferred to the computer-based information system that will use it. This factor is important because the uncertainty of the environment makes the changes that the subsystems being supported need to undergo in order to adapt to the uncertainty of the environment correspondingly uncertain. When the computer-based information system stores and maintains knowledge about these subsystems, the changes in these subsystems will need to be correlated with changes in the knowledge that the computer-based information system needs to store and maintain. The need of correlated changes will certainly increase the cost of the adaptability of the whole system or, simply, make it less adaptable. This clearly makes the option of transferring knowledge to the computer-based information system when it needs it more attractive when the uncertainty of the environment is high. This factor, however, must be weighed against each of the other factors mentioned above. In each case, the tradeoffs involved must be carefully considered. If the transfer of this knowledge is too complex computationally, for example, the cost of transferring the knowledge might become excessive, the ability of the computer-based information system to effectively support function might be compromised. These negative effects might offset the gains in adaptability that the transfer of knowledge would produce.

Determining the architectural features needed to transfer and use the knowledge effectively The architecture of a computer-based information system that successfully allows for the use of knowledge that is transferred to it only when it is needed must provide the capabilities needed for receiving, preparing, and using the transferred knowledge in a manner consistent with the effective support of function, that is, in a manner in which the requirements of the computer-based information system are successfully met.

Let us look in more detail into the capabilities for receiving, preparing, and using the transferred knowledge. In order to receive the transferred knowledge the architecture must allow for a user interface that accepts a description of this knowledge as an input and directs this description to a component that prepares the knowledge for processing and sends it to the component that is responsible for processing (i.e. using) it. The architecture should be able to provide the interfaces needed to integrate the processing of this knowledge with the other processes that the computer-based information system provides, and must be able to accept the outputs that the use of this knowledge provides and to relay them to the users. In order to prepare the transferred knowledge for use, or processing, the architecture may allows for components that receive the knowledge in a form that it is already executable or that can be transformed into procedures that can be executed locally. Alternatively, it may allow for components that use the transferred knowledge to create calls for procedures that can be executed remotely. Processing the knowledge amounts to executing the procedures that embody it. Therefore, the component that calls these procedures must also be able to monitor their execution and receive the results that they produce. Notice that the knowledge that is transferred to the computer-based system must also include the ability to use computing knowledge that is available locally or in a remote system. Notice that the component that is responsible for the use of the imported knowledge does so through the coordination and execution of the procedures embodying it. This requires also the ability to send the results obtained to other processes, or to the user interface component for their conversion into an appropriate output. An architecture that receives the transferred knowledge, prepares it for processing, integrates its processing with other processes ot the computer-based system, and outputs the result of the processing to the users is shown schematically in Figure 3.

The design decisions that need to be made in order to make the use of imported knowledge possible include determining the knowledge that can be transferred only when it is needed, the form in which the knowledge to be transferred can be represented so that it can be used (i.e. processed) by the computer-based information system, the way in which the knowledge is going to be prepared for processing, the way in which it is going to be actually processed and the processing controlled and monitored, and the way in which the results of processing the knowledge are delivered to the users. Clearly, these decisions determine the way in which the components of the architecture shown in Fig. 3 are implemented. In the next section we consider some of the main options for the implementation of this architecture and discuss some of their advantages and disadvantages in specific contexts.

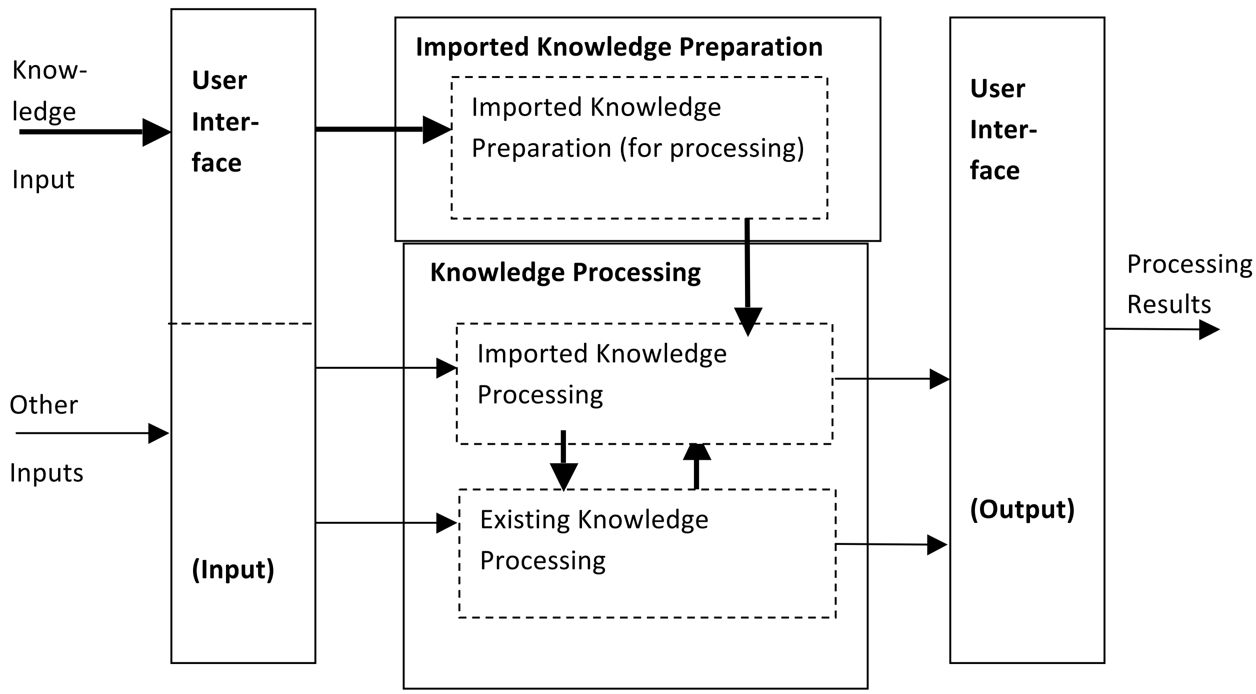

Figure 3 Schematic view of an architecture that imports information processing knowledge only when it is needed.

THE TRANSFER OF INFORMATION PROCESSING KNOWLEDGE: SOME DESIGN IMPLICATIONS

The capabilities needed for the transfer of knowledge to a computer-based system for its use when it is needed can be provided in a number of different ways depending on the type of functions that the information system supports, their information needs and the requirements of the information system. Once the knowledge to be transferred as it is needed has been identified, the algorithms or procedures that embody this knowledge have to be defined as well. The form in which these algorithms and procedures are expressed determines the way they can be stored and transferred, and ultimately executed and, if needed, converted into an executable form. This in turn determines the capabilities needed by the computer-based system. Several approaches can be used to implement the various components of the architectural features needed for the transfer of information processing knowledge when it needs it. In this respect, a number of middleware technologies and architecture have been made available. These technologies and architectures, that provide a number of capabilities such as remote procedure calls (RPC) and message queuing include object-oriented middleware such as CORBA, the Microsoft's .NET framework and Java's enterprise java beans (EJB) J2EE technologies. The use of markup languages such as XML in connection with standard communication protocols such as HTTP has made it possible to bridge the gap between applications and systems.

Service-oriented architectures (SOA), for example, provide a set of design options and capabilities (Bean, 2010) that can be used to define a service-oriented scheme for the definition and specification of functional components that can be integrated in a manner that is independent of the place and the form in which the candidate services (information processing knowledge in the form of functional components or information processing functions) can be integrated. The specification of a mix or combination of services can be considered as a scheme for the transfer and use of knowledge. Thomas Erl (2008) provides a series of principles for service oriented architectures.

Web services, therefore, provide an important means of making knowledge that a computer-based information system needs available to it. This knowledge can be given to a computer-based information system in the form of the information needed to identify, locate, and use the assembly of Web Services that embodies this knowledge. Notice that in order for the information processing knowledge embodied in the Web services to be used, the computer-based information system needs to have the information it needs to prepare the specifications of the services that provide the knowledge and, when the services are available, the knowledge needed to interact with these services and to interpret and use the results that it provides. The SOA, on the other hand, must provide the resources and capabilities needed in order to create, discover, transfer and use the Web services that embody the knowledge. These resources and capabilities include those needed for the creation, storage, description, discovery, and transmission of the Web services, and the binding of them to the computer-based information system functions that use them. Several approaches to providing these resources and capabilities have been developed including the .NET and the J2EE platforms, and CORBA. According to Myerson (2002), Web services technology can be seen as the result of evolutionary paths based on the origins of these platforms and supported by Internet protocols and standards such as HTTP, XML, and SOAP.

CONCLUSION

The development of computer-based information systems can be seen as a process in which information processing knowledge is transferred (actually embodied) into the information system being developed. This knowledge, however, needs to be updated when the systems it describes change the way in which they process information. In this paper we suggest an approach in which some of the information processing knowledge that a computer-based system uses is transferred to it only when it needs to use this knowledge. This tends to reduce the adaptability of the whole system because the computer-based system does not have to update the transferred knowledge when it changes.

This architectural feature, however, needs to be used in a manner consistent with the effective support of function, that is, in a manner in which the requirements of the information system are fully met. We mentioned some of the technologies and architectures that make this capability possible. Future research is needed, however, in order to relate specific types and forms of knowledge to specific architectures, designs and technologies in specific situations.

REFERENCES

Bean, James. SOA Web Services Interface Design, Morgan Kaufman Publishers, Elsevier Inc. 2010.

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

Erl, Thomas. *SOA Principles of Service Design*, Hall: Upper Saddle River, NJ, 2008.

Kampfner, R. A hierarchical model of organizational control for the analysis of information systems requirements, Inf. Syst. 3(3): 243-254. 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. "Is there a systemic Scale of Information Processing?," SIG on the Foundations of Information Science, 46th Annual Meeting of the International Society for the Systems Sciences, Shanghai, People's Republic of China, August 2nd to August 6th, 2002a.

Kampfner, R. "Digital and Biological Computing in Organizations," BioSystems, Vol 64, pp. 179-188, 2002b.

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.

Myerson, J. M. *The Complete Book of Middleware*, New York: Auerbach Publications, 2002.

Shannon, C. E., Weaver, W. The Mathematical Theory of Communication, University of Illinois Press, Urbana, 1962.