

Web-Based Interactions Support for Information Systems

Youcef Baghdadi, Ph.D.
United Arab Emirates University, UAE

y.baghdadi@uaeu.ac.ae

Abstract

Work organization, business innovation and IT have enhanced the distributed nature enterprise information systems. Information systems today are made up of subsystems running on heterogeneous IT platforms with varying implementations of business objects and processes increasing the dual risks of (i) inconsistency of business objects views and (ii) inefficiency of processes.

This paper frames this problem as lack of representation and implementation of interactions among the subsystems and external sources. It proposes an interaction support system to make interactions an explicit element of the Enterprise Information System like data and operations. It describes a solution where the interaction elements are encapsulated into a separate subsystem and located in a web server to be used by other subsystems to exchange and share data and to perform processes with complete transparency.

We argue that such Interaction Support System may provide global, unified and consistent view of business objects and synergy of processes.

Keywords: Information Systems, Interactions, Interactions Support System.

Introduction

Work organization, business innovation and information technology increase the heterogeneity of the components of the enterprise information system. Therefore, the information system is not monolithic. Instead, it is made up of subsystems running on distributed heterogeneous IT platforms with varying conceptual and technical pieces. That is, the subsystems, by their conceptual, organizational and technical specificity, have different representations, schemas, views and implementations of the business objects and processes. Moreover, most of these subsystems are developed and implemented on case-by-case basis to satisfy particular goals (e.g., personal information system, office automation system, group work information systems, etc.), and not as well-designed elements of an enterprise information system. Pieces of data and processes are, therefore, overlapping and replicated, leading to the dual risks of (i) inconsistency of business objects schemas and views, and (ii) inefficiency of processes.

Therefore, *artifacts* are required: (i) to allow global, coherent and transparent views of business objects and efficient distribution of processes, and (ii) to avoid, design of the IS each time new IT is introduced.

This problem has been investigated under different aspects: schema integration of heterogeneous and federated databases (Batini et al., 1986) and (Pitoura et al., 1995), (Konopnicki and Shemueli, 1999), (Castano et al., 2001), information integration (Arens et al., 1993),) and (Calvanese et al., 1998), (Kwan and Fong, 1999), cooperation and coordination paradigm such as Computer-Supported Cooperative Work (CSCW) which tends to motivate and valid GroupWare used to support and enhance activities in which more than one person are involved (Schmidt, 94), (Mills, 1999), GroupWare, (Greif, 1988), (Ellis and Wainer, 1999), Workflow Systems (Casati et al., 1999), cooperative information systems (Papazoglou and Schlageter, 1997, or as a purely technical problem which may be solved by existing technologies such as Intranet/Internet Computing, Client/Server, Middleware (e.g., COBRA, DCOM, RMI, ODBC, JDBC and JSP/Servlets) and Web applications (Umar, 1997), Lewandowski, 1998), (Fraternali, 1999) and (Crestani, 1999).

We frame this problem as an *interactions* problem, that is, a lack of representation and implementation of interactions among subsystems of the information system with each other and with external sources (representing business partners). In fact, the community of information systems researchers has not focused on interactions as a component of the information systems as well as it has on the compo-

Material published as part of this journal, either on-line or in print, is copyrighted by the publisher of Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Editor@inform.nu to request redistribution permission.

Web-Based Interactions

nents of data and operations. Instead, it is considered as technical communication problem so that communication means and tools are implemented in case-by-case basis when required, that is, when breakdowns are observed.

We argue that interaction is an important aspect of the information system as well as business objects and processes. In fact, interactions allow global, unified and consistent view of business objects and synergy of processes, but also make explicit emergent *knowledge*, which has considerable added value for the organization. Yet, interactions do not exist naturally. It is necessary to make them explicit and visible all more so since the exploding popularity of the world wide web (web) allows building distributed interacting subsystems; and accessing a variety multimedia structured, semi structured or unstructured information representing business objects and processes via web browsers.

Therefore, interactions based upon the web must be analyzed, designed and implemented as a separate, orthogonal perspective that co-exists with the two existing perspectives of business objects (data) and processes.

Accordingly, this paper presents a framework for:

- (i) Representing interactions as a separate perspective, and
- (ii) Constructing artifacts of new types called *Interactions Support Systems*.

These new kinds of systems co-exist with the different existing subsystems of the information system. The interaction support system is dedicated to subsystems' interactions with each other and with external sources. It mainly provides these subsystems with business objects-oriented and processes-oriented interactions services. By supporting interactions, the interaction support system is expected to support, at the same time, coordination, cooperation, collaborative human work, information integration, and information access and retrieval. That is, different situa-

tions of interactions, which is an added value to the paradigm of cooperative information systems (Papazoglou, 1997), (De Michelis et al., 1997) and (Arcieri, 1999).

The framework is a conjunction of *information systems*, *interactions* and the *web* as the irresistible implementation technology for this kind of systems. The framework emphasizes answers to questions such as: Why do information systems need to interact with each other? How interactions allow emergent knowledge and therefore added value? What are the situations of interactions with respect to the information systems and consequently the methodological specializations of the interactions? And finally, why the web makes interactions easier and efficient?

Factors Influencing Distribution of the Information Systems

The information system is a technology-based image of the business system. This image is used by the decision system to control (day to day operations, resources management, decision making and problem solving, and business improvement and innovation) the business system that products goods and services. This image is a mosaic of IT (e.g., audio, video, computers and networks and software) as well as traditional (e.g., paper) (figure 1). The information system is therefore, not simple and monolithic. It is rather a set of multiple interrelated subsystems. These subsystems result from two main factors: business's improvement and innovation and IT.

1) Business innovation and improvement structures the business system by functional areas (or departments) having their respective goals, skills and specialized processes, which enhances the distribution of the information system. That is, there are, in the same organization, a proliferation of several subsystems of the information system, which are generally classified into personal information systems (PIS), workgroup information systems (WIS) and legacy systems. Moreover, organization is more and more inter-

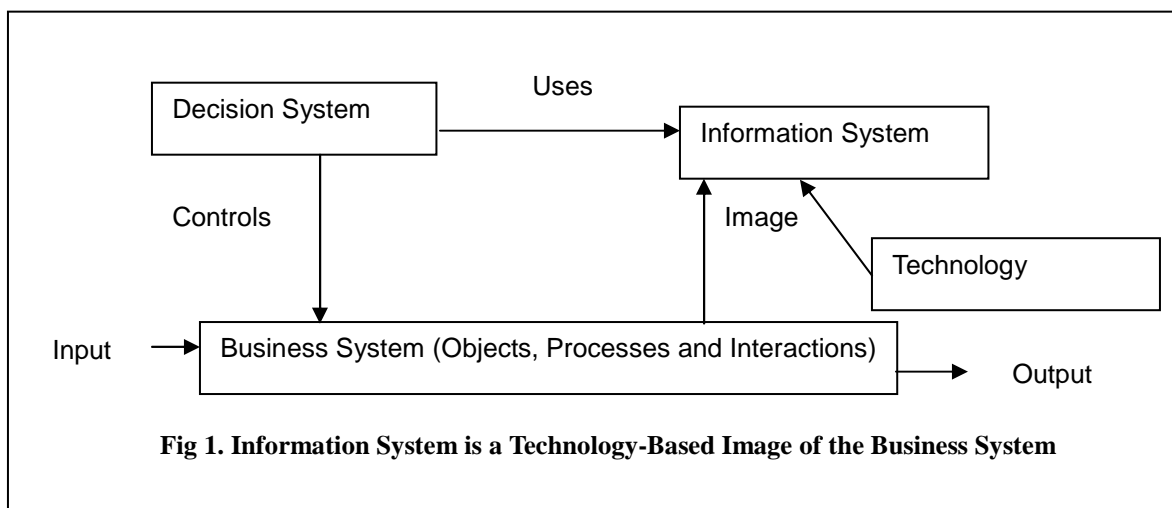


Fig 1. Information System is a Technology-Based Image of the Business System

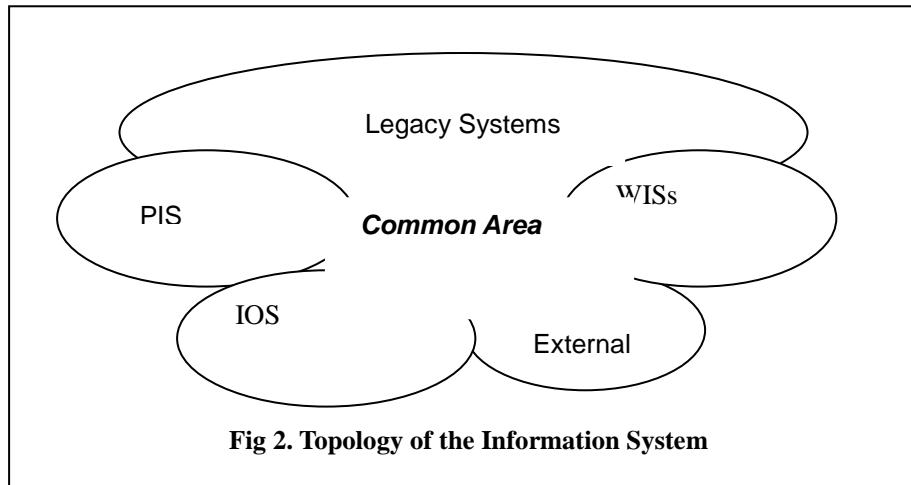


Fig 2. Topology of the Information System

acting with its environment helped by inter-organizational system (IOS). PIS support knowledge of the individuals, especially individual data. WIS support formalized knowledge of departments (or groups) including specialized processes. Legacy systems support collective and formalized knowledge of the whole enterprise, and OIS automate inter-organizational information flow among organizations to support all the stages of products and services (planning, design, development, production and delivery).

Figure 2 shows the information system as set of subsystems. Decision-making, problem solving and business's improvement and innovation require respectively new types of data modeling (portals, data warehouse, data mart) and new kinds of information systems supporting respectively group decision support systems (GDSS), corporate portals (EP), e-commerce, business process reengineering (BPR), business network reengineering (BNR), and business scope redefinition (BSR). For instance, enterprise portals require a metadata for the whole enterprise, which is impossible without interactions. The common area (in which we are interested) is the interactions space. This common area does not exist naturally because it is not automatically implemented and used. It will enable structured messages dealing with shared or overlapping business objects and processes and locked data. Implementing it requires an environment, that is, a computational infrastructure enabling low-level communication

protocols as support of high-level interaction protocols.

2) IT consisting of computer-based tools is a support for information-processing tasks (e.g., capture, create, convey, store and communicate information) and enabler of innovation and strategies (e.g., just in time, teamwork, information partnership, virtual organization and transnational firm enforcing organization horsepower (Haag et al., 1999). IT supports information processing of local and specific subsystems of the information system. Office automation systems (with their communications techniques and tools) implementing PIS, groupware for WIS, Internet/intranets/extranet supporting OIS and e-commerce, are examples of proliferation of such technology as personal computers, workstations and networks (local area networks, intranets and extranets). IT supports the flow of information among all involved parties (product manufacturers, service suppliers, distributors, and wholesalers, retailers and customers) gathered to do business in virtual market place supported by e-commerce. At last but no least, e-commerce, which is a new methodology that consists of performing business in electronic fashion, involves interaction between an enterprise and its costumers (B2C) or partners (B2B).

These two factors have enhanced the distributed nature and the heterogeneity of the components of the information system: pieces of data representing business objects, applications related to business processes, and hard-

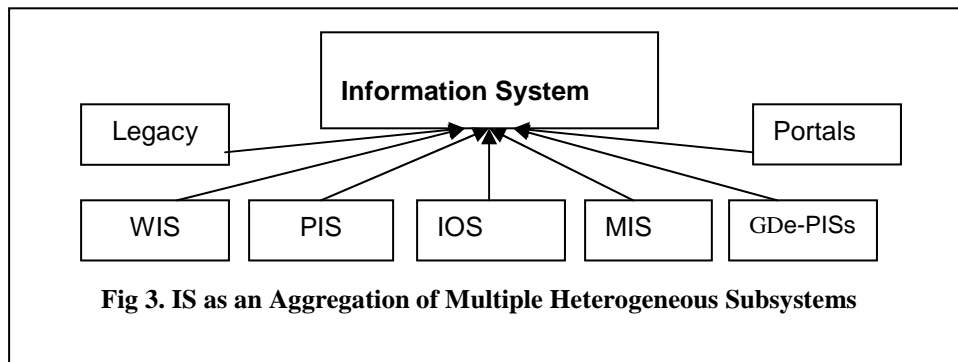


Fig 3. IS as an Aggregation of Multiple Heterogeneous Subsystems

Web-Based Interactions

ware/software platforms on which these subsystems are running. In fact, the information system is an aggregation of subsystems (figure 3).

This structuring of the business and the resulting distribution of the organization knowledge is certainly relevant for local autonomy, reliability, availability and performance. However, data and processes are locked, pieces of data related to same business objects are overlapping and same processes are performed redundantly. Therefore, higher the consumed resources, lower are the consistency of data and the efficiency of the processes, which is paradoxical. Moreover, this situation leads to constraints and even conflicts at both local and enterprise levels while infrastructures for communication protocols and interactions protocols to offset it are neglected. In fact, at the local level, individuals, groups and departments may have informational and computational resources constraints (e.g., pieces of data). These constraints are due to their specificity and their partial perception of business objects and processes. At the enterprise level, information needed for any action notably decision-making is in part only processed in the local subsystems and simple data integration is still insufficient (even though data do exist but not accessible). Furthermore, WIS, DSS, GDSS, EP, e-commerce and IOS are mainly based upon interactions among local subsystems and especially inter-enterprises (with the partners). In new business IT providing communication and interaction are primordial. Organization is highly interacting with its environment, which is in turn an important and non-negligible source of information. All these subsystems share business objects and processes. They cannot exist without interactions that allow them global and consistent view of their business objects and processes and those of their partners.

Interaction Specification

This section emphasizes representation of the interactions as a separate orthogonal perspective that co-exists with the business objects and processes representation perspectives. It introduces, defines and specifies the concept of interactions and situations of interactions with respect to the information system.

Interaction Definition

Interactions are considered at first sight as relationships between two or more subsystems. However, we consider them fundamental:

- To allow internal and external exchanges as well as coordination, cooperation, collaboration and therefore re-organization (process reengineering, network reengineering and scope redefinition) and e-commerce.
- To satisfy the lack of informational and computational resources.
- To allow consistent view of business objects.
- To allow synergy of processes.
- To make decision (e.g., supporting groups, portals and data warehouse)
- To improve and innovate business processes.
- To allow **emergent knowledge**. In fact, local knowledge is static; it is not representative of the dynamics of the organization notably the internal and external interactions. Moreover, knowledge resulting from interactions is more relevant and more complete than simple data integration.

Accordingly, we propose a definition that takes into account interaction aspects such as **dynamics**, **time**, **space** and **situations**.

Dynamics. Interactions may be a fixed or dynamic relationship between subsystems. Work organization generally stipulates a fixed relationship (e.g., mechanisms of coordination used to offset planned task allocation). Yet, interactions are generally dynamic (Erceau, 1994). They are made by a set of reciprocal actions performed by agents (subsystems of the information system) having a certain degree of autonomy and freedom.

Time. Action-reactions of subsystems are temporal, synchronous as well as asynchronous, that is subsystems are able to interact at any time.

Space. Interactions may involve local as well as distributed and remote subsystems.

Situations. Interactions also have different situations considered as answers to questions such as ‘Why do subsystems interact?’ The concepts of interactions and situations of interactions have been used in different disciplines (e.g., Linguistics, Work Organization, CSCW and DA) (Winoograd, 1988), (De Michelis, 1994), (Matthes, 1997) and (Weiss, 1999). This paper concerns with the situations of interactions with respect to the information systems discipline.

Interactions Attributes	Explanation
Interaction ID	Identifier for an interaction.
Interaction Date	Date and time of the interaction.
Subsystem Trigger	Subsystem trigger browsing business object.
Business Object Name	Business object trigger is interested in.
Interaction purpose	Locating business object. Schema of the business object on different subsystems. Querying pieces of data related to business object (views). (Re)Structuring.
Involved Subsystems (1..n)	Subsystems chosen by the subsystem trigger to interact with.
Effect (Re-action)/ Involved Subsystem Side	Actions performed by the subsystems involved in the interaction. Requesting authenticity. Presenting business object schema. Performing query and return data.
Effect (Reaction) /Trigger Subsystem Side	Actions performed by the subsystem receiving reply from involved subsystem which may be: Updating its business object schema. Querying business object on the subsystem receiver. Sending its business object schema to the subsystem receiver.

Table 1. Business Object-Oriented Interaction Structure.

Situations of Interactions with Respect to Information Systems

With respect to information systems, each subsystem has a certain degree of autonomy and freedom within relationship. It has the ability to commit to participate or to quit the relationship. Subsystem may interact with the environment for different purposes namely for coordinating, for collaborating, for cooperating, or simply for transmitting messages. We enhance definitions given by (Alquier, 1993) and (Baghdadi, 1996) with respect to information systems by considering two formal situations. These situations are subsequent to the nature of information systems which contain informational (pieces of data related to business objects) and computational (processes) resources. That is, these two obvious situations are referred to respectively *business object-oriented interactions* and *business processes-oriented interactions*. This is a methodical specialization of interactions that facilitates the specification, analysis, design and implementation of artifacts that support interactions according to the user requirements.

Business Object-Oriented Interaction is a situation of interactions that deals with the consistency and the semantics of schemas, views and pieces of data related to business objects. That is, any exchange between two or more subsystems dealing with the schema and views (pieces of data) related to business objects is considered as business

object-oriented interactions. For example, a situation in which a subsystem needs to query or multi-query schema/implementation of a business object on a given platform is considered as business object-oriented interactions. Querying distributed databases or integrating and unifying shared data are examples of such a specialization of the interactions.

Table 1 shows the business object-oriented interactions specification (structure).

Business process-oriented Interactions is a situation of interactions that deals with the efficiency and the semantics of the business processes. That is any exchange between two or more subsystems dealing with processes is considered as business process-oriented interactions. For example, the situation in which a subsystem needs to invoke another for performing/running a particular task/process that it cannot perform (using its own resources) is a business process-oriented interaction. It deals also with situation in which we need to articulate distributed business process activities (e.g., Workflow Systems).

Table 2 shows business process-oriented interactions specification (structure).

Interactions Attributes	Explanation
Interaction ID	Identifier for an interaction.
Interaction Date	Date and time of the interaction.
Subsystem Trigger	Subsystem trigger invoking business process.
Process	Invoked process.
Parameters	Parameters required performing the process.
Interaction purpose	Locating process. Invocation. Re-engineering.
Subsystems Involved (1,n)	Subsystems chosen by the trigger to interact with.
Effect (Re-action)/ Involved Subsystems Side	Actions performed by the involved subsystems. Requesting authenticity or more parameters. Performing the process. Return results.
Effect (Reaction) / Sender Side	Actions performed by the trigger subsystem: Updating its process. Making a new vision of the process. Sending its process to the subsystem receiver.

Table 2. Business Process-Oriented Interaction Structure

Representation of the Interactions Aspect

Traditional information systems development life cycle, methods (structured or object-oriented), models and techniques usually represent two main aspects which are data related to business objects and functions related to business processes (separated or integrated) at both logical and physical levels.

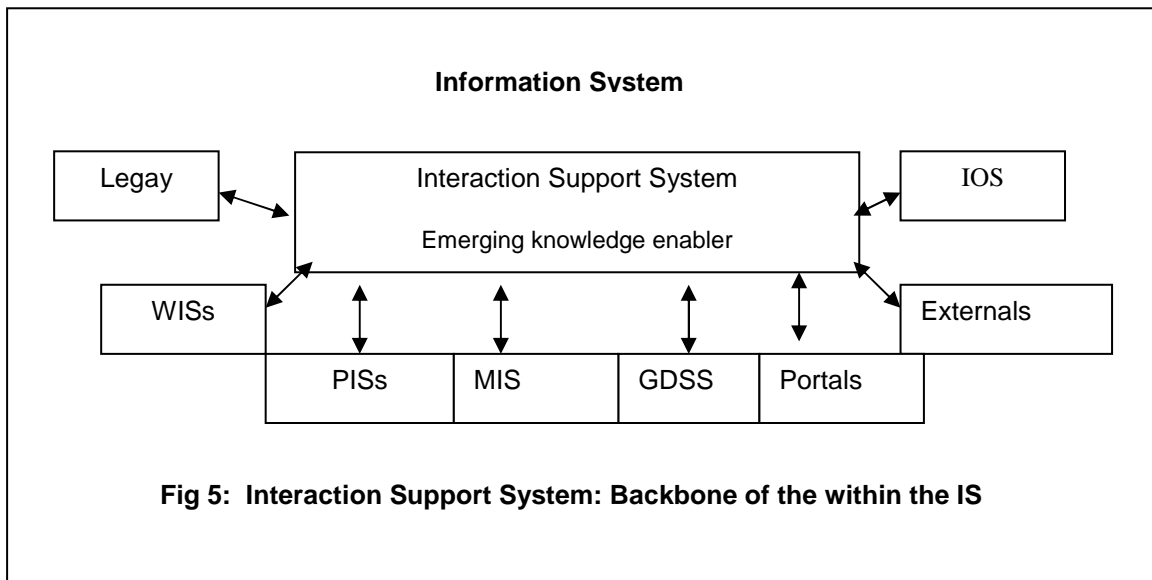
We add interactions as a third perspective which is as relevant as the two traditional perspectives (Figure 4). This perspective will be specified at the logical levels and implemented using the adequate technology at the physical level.

Interaction Life Cycle

Each interaction has a life cycle. That is, an interaction has an identity, it involves both subsystem trigger and involved subsystems that act and re-act. Interactions are supported by a logical mediator which may have distinct implementation according to the nature and situations of interactions (asynchronous/asynchronous, local/remote and data-oriented/process-oriented). The mediator is a kind of broker of business objects and processes that allows locating business objects and processes over the organization space and its partners. Moreover, pertinent interactions are registered in an *interactions log* to keep track of interactions status and results at each step of the life cycle. This interaction log will be used after to summarize, integrate or unify business objects and processes.

<i>Aspects</i> <i>Abstraction</i>	<i>Business Objects</i> <i>(Data)</i>	<i>Business Processes</i> <i>(Functions)</i>	<i>Interactions</i> <i>(Network, Intercation Protocols)</i>
Logical Level	Logical representation used by traditional methods, modeling & software engineering (e.g., E~R, UML diagrams)		Representation of interaction by Nodes: Subsystems (e.g., packages) Links: Requests and Responses, commitments and conventions
Physical Level	Data and Functions, Objects, Software Components		Communication Protocols, Developed Components (Interaction Support Systems), or existing Middleware (CORBA, DCOM, RMI, ODBC, JDBC) XML, or a combination

Fig 4. Interactions Perspective within Information System Development Life Cycle



The mediator tasks may be implemented by each side: subsystem trigger, involved subsystem, or by a specific interaction support according to the nature of the interactions. The next section specifies the mediator as an interaction support system that co-exists with the subsystems of the information system (Figure 4).

Interaction Support System

This section presents a definition, a structural and functional architecture of the interaction support system.

Interaction Support System Definition

An Interaction support system is a subsystem of the information system. It is made up of *hardware, software components, interactions* (its core elements), *procedures* and *people*. The goal of this system is to facilitate, allow and manage interactions (irrespective to their situations) among other types of subsystems and their environment.

Interactions are made possible by means of the components of the interaction support system (hardware, software components, procedures and people).

Interaction support system provides subsystems of the information system with business objects and processes oriented communication facilities namely communication services and semantic services. The communication services implement the interactions and the semantics services solve the heterogeneity. The framework considers that the semantic services are user-oriented not system-oriented, that is, the end user is an important agent responsible for solving problems related to the heterogeneity between subsystems by means of easy to use interfaces.

The interactions support systems that represents global vision of interaction intra-organization and inter-

organizations is well suited as artifact to support both individual and collaborative work. It aims to enable different situations of interactions among a large number of pre-existing information sources including legacy systems, local subsystems and external sources.

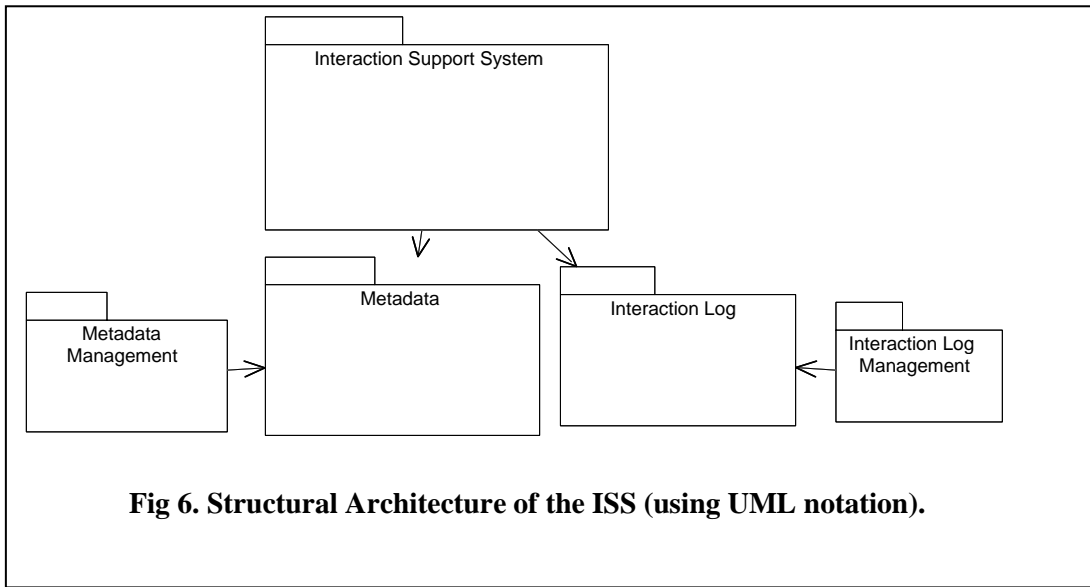
Figure 5 shows that the interaction support system is a subsystem of the information system that co-exists with the other subsystems allowing them to interact and to produce emerging knowledge (information and processes) which is an added value for all of them. This emerging knowledge is used by the subsystems having insufficient informational and computational resources to achieve their respective goals and to participate efficiently to achieve the goal of the enterprise.

ISS Structural and Functional Architecture

This section specifies a structural and a functional architecture for ISS with respect to the adopted interaction protocol.

Structural Architecture

A structural architecture of the interactions support systems consists of: (1) a *component (ISS)* implementing a set of functionality, (2) a *metadata* that describes the distribution and the implementations of the business objects and processes over the subsystems of the information system. The metadata is the core support of the ISS, (3) an *interaction log* to keep track of the pertinent interactions, which may be reused, (4) a *metadata manager*, and (5) an *interaction log manager*. Figure 6 represents these elements as interrelated (arrows) UML packages, that is, they embed the functionality (e.g., services interfaces and implementations).



The ISS component is responsible for presenting services to the subsystems namely:

- Commitment service i.e., registering subsystems general profiles with their business objects and processes profiles.
- Undo commitment service i.e., canceling registration or removing some or all business objects and processes profiles.
- Logging service.
- Browsing services i.e. browsing business objects (locations, synonyms, implementations and schema), or business processes (locations, hierarchy, implementations and parameters).
- Manipulating services i.e., viewing business object and restructuring business objects, or invoking, reusing and reengineering/overriding business processes.
- Sharing services i.e., sharing common space where the business objects and processes may be reengineered.
- Metadata services i.e., services related to the coherent and consistent metadata for the enterprise portal or for data warehouse.

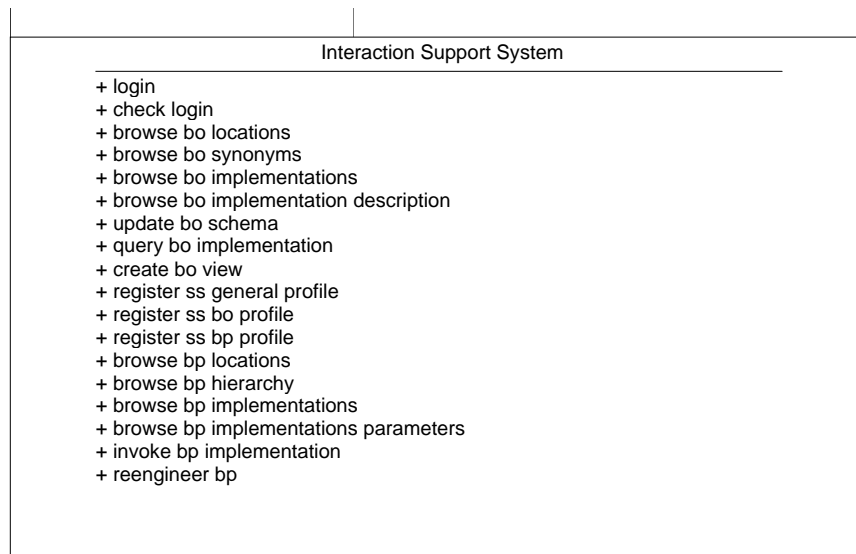


Fig 7. ISS Services packed as UML Package

Legend for figure 7. ss: subsystem bo: Business object bp: Business process

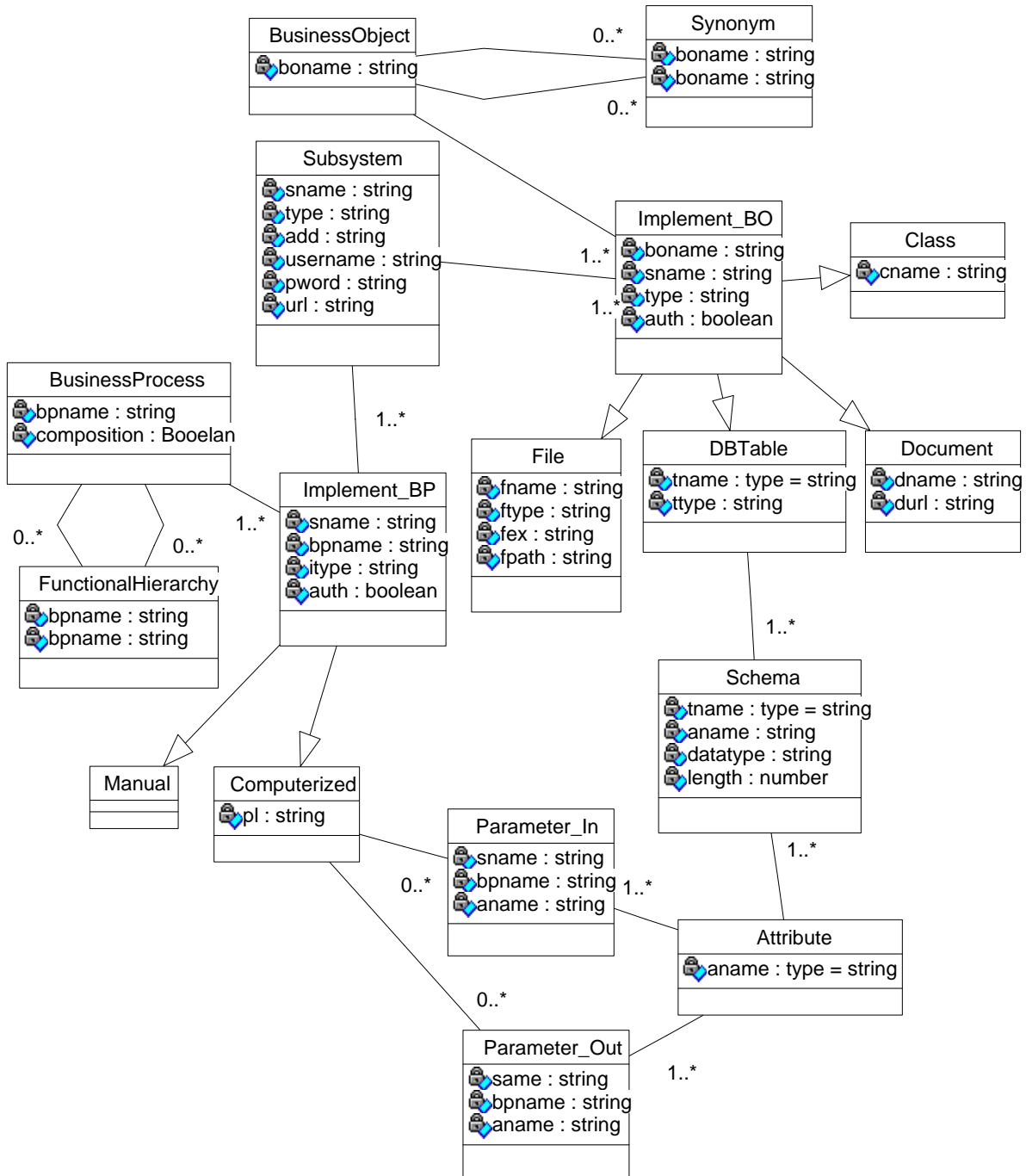


Fig 8. Metadata Class Diagram (with UML)

- The ISS is scalable, that is, more services may be added to the ISS.
- The set of services is packed into a UML package (figure 7) and are considered as functions of a molecular class.

The ISS uses a metadata (figure 8) that keeps track of three main elements dealing with **what** (list of business objects), **how** (list of processes) and **where** (list of locations) described at different abstraction levels. We consider

three levels: conceptual, logical and physical. At the conceptual level, are described the **business objects**, **business processes** and the **topology of the Information System** (internal as well as external subsystems). At the logical level, are described the data models (e.g., relational, object-oriented or file) used for representing the business objects, the applications architecture and the subsystems (e.g., DFD, logical view). At the physical level, are described, the actual implementations of business objects (e.g., database tables, files, documents); processes (com-

Web-Based Interactions

putational entities e.g., program, function, object, agent); and network architecture (e.g., IP addresses of the subsystems, interfaces) Table explicit these concepts.

The metadata represents all the elements above at different abstract levels. The concepts are specified and then represented as classes according to the UML notation (figure 8). It first represents the conceptual level i.e., business objects, business processes and subsystems as UML classes; and then the logical and physical levels, which are represented as either classes describing more details or sub-classes describing specialized implementations of business objects and processes. Relationships represent the loca-

tions of the business objects and processes.

1. Business Objects. A business object may have some *synonyms*, and different implementations within several subsystems. It is described by *attributes* on each distinct *implementation*. It is then described as follows: name, set of attributes for each implementation on each subsystem (e.g., web document, manual document, database table, file, or object).

For instance, the business object student may be implemented as database table within the registration system, a file within the advising system or an excel sheet within the

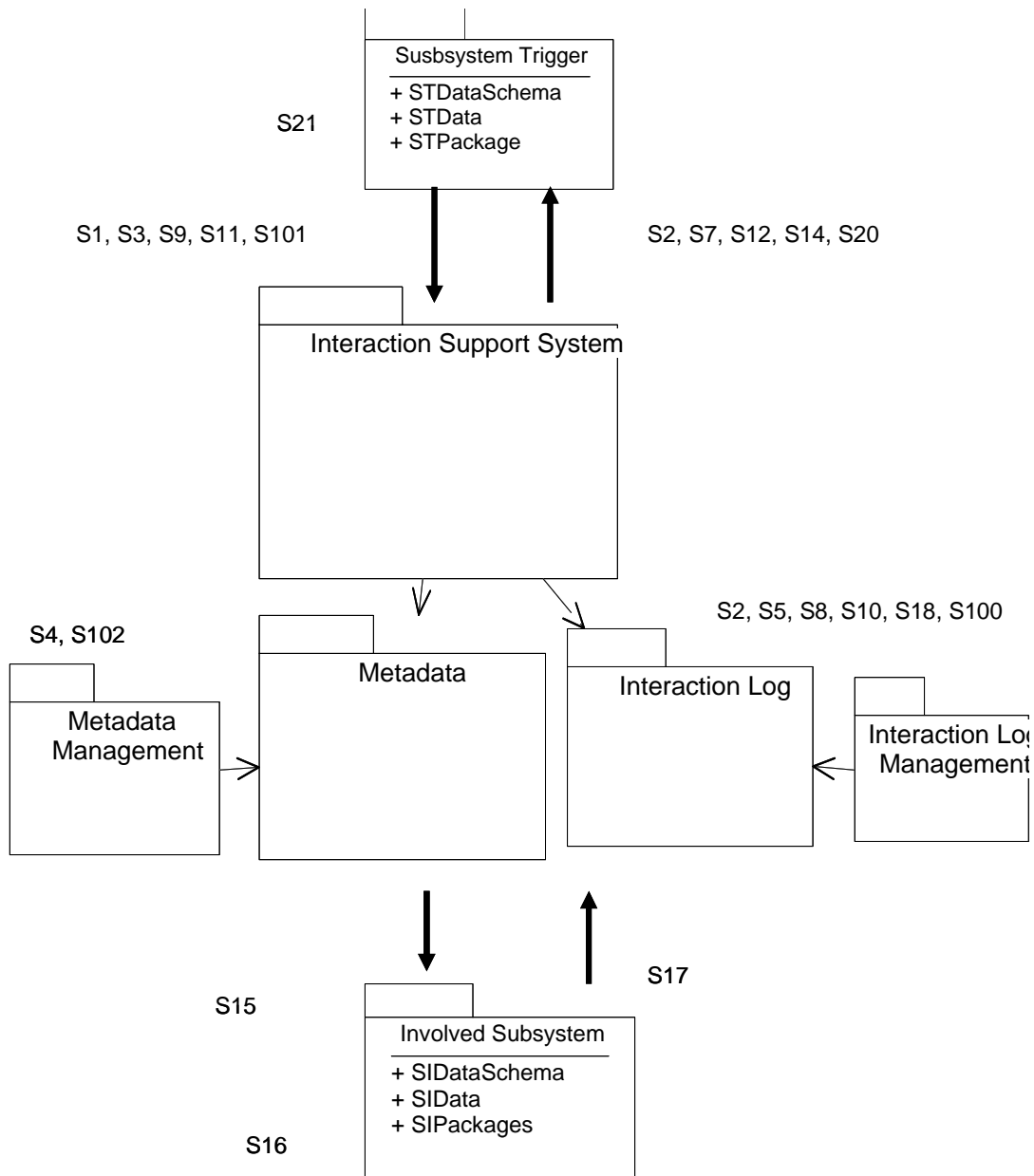


Fig 9. Si (See Table 3) Represents a Step of the ISS Functional Architecture

instructor office automation system. It may be described differently on each subsystem of IS. That is, its attributes differ from one implementation to another.

2. **Business Processes.** A process may be subdivided into sub processes (functional hierarchy). A process may have different implementations on several subsystems. It has attributes as *parameters input* and attributes as *parameters output* for each distinct implementation. A process may have precedence. It is described as follows: name, description, state (current activity), set of activities implementation on each subsystem (an implementation may be: package, application, procedure, method, applet, activeX, Java Bean or any other software component), process required parameters (parameters in and parameters out)

For instance, the student advising process is a process that may be decomposed into many sub processes distributed over the different subsystems which are advising system, department and instructor subsystems.

Each sub process is made up of several activities and has consequently a state. Some sub processes may be identical but implemented differently. Student grade computation process is implemented differently within each instructor subsystem. Moreover, each sub process implementation has its own parameters in and parameters out.

3. **Subsystem** that is described by: Id, name, login password, type (PIS, WIS, EIS or external source), address (e.g., url, username and password) and administrator (user name and password if any to access its subsystem).

Examples of subsystems in education domain are departments, colleges, and registration system, advising system, instructor office.

Functional Architecture

A functional architecture of the ISS is based on the (1) ISS as middle-tier between the subsystems of the information system, (2) a subsystem trigger willing to browse business objects or processes, and (3) one or more involved subsystems that host the business objects and processes implementations. The subsystems of the information systems implement their actual schema, constraints and data related to business objects, with packages implementing business processes (see subsystem trigger and involved subsystems package in figure 9).

Figure 9 shows the relationships between these three subsystems. S_i indicates the i^{th} step of the interaction life cycle which is specified in the table 3 (where ST stands for subsystem trigger, ISS stands for interaction support system and SI for involved subsystems). It shows that subsystems (either trigger or involved in interaction) are specified as UML package with schema, data and functions. They use (shaded arrow) the ISS to interact. The ISS is also wrapped as UML package (ISS Package) that encapsulates all its services (figure 7). ISS uses (simple arrow) a metadata (Metadata Package) and interaction log (Interaction Log Package). Both metadata and interaction log are managed by their respectively management systems (Metadata Management Package and Interaction Log Management Package).

Table 3 generalizes Table 2 to specify the role of each of the three involved subsystems: subsystem trigger, subsystems involved and the interaction support system as mediator regardless the situation of interaction.

Step	Specification	SS
S1	Subsystem (ST) requests Interaction Support System (ISS) to open session.	ST
S2	ISS registers the interaction (Interaction_ID, SS_ID, Date and Time).	ISS
S3	ST enters the business object (BO) or Business process (BP) names it searches for.	ST
S4	ISS check for BO or BP existence in its metadata (If BO or BP doesn't exist S5 else S8)	ISS
S5	ISS informs SS of the non-existence of BO or BP and invites the SS to add data BO or BP to metadata	ISS
S6	If ST accept to add data S100 else S7	SI
S7	ISS terminates session and registers the termination in the interaction log (IL)	ISS
S8	ISS displays the locations and implementation type of the BO or BP on each location (sub-systems)	ISS
S9	ST selects any implementation type of the BO or BP	ST
S10	ISS displays the detail of the implementation type of the BO or BP	ISS
S11	ST selects to query BO or to invoke BP: S13, or update its own BO or BP implementation: S21 Or terminate session: S12.	ST
S12	ISS terminates session and registers it in the IL	ISS
S13	ISS prepares the query or the invocation. - case BO query (location, authenticity, attributes and criteria clause). - case PB invocation location authenticity, parameters).	ISS
S14	ISS registers the query or invocation in the IL.	ISS
S15	ISS contacts the selected location (subsystem).	ISS
S16	Selected subsystem performs query or invocation.	SI
S17	Selected subsystem returns results to ISS.	SI
S18	ISS presents the result to SS	ISS
S19	ST Selects a new location: S9, Terminates session: S20 or updates its BO data or reuse BP: S21	ST
S20	ISS terminates session by registering the results as re-action in the I L.	ISS
S21	ST updates its own BO or BP implementation by updating respectively updating BO schema, data, BP reuse.	ST
S100	ISS allows SS to add BO or BP	ISS
S101	ST enters its BO or BP implementation	ST
S102	ISS registers the new BO or BP in the metadata	ISS

Table 3. Interaction Life Cycle

ISS: Interaction Support System; BO: Business Object; BP: Business Process
 ST: Subsystem Trigger (subsystem requesting others subsystems via the ISS).
 SI: Involved Subsystems (subsystems committing to handle the request).
 Si: Ith step in the interaction life cycle (bolded steps are performed by the ISS).

Specializations of the ISS

A good design specializes the interaction support system into *business object-oriented interaction support system* that allows coordination such as distributed data base management, data remote access that make possible the management of the distributed business objects by building their whole semantics. And *business process-oriented*

interaction support system that allows a global workflow of business processes (figure 10).

Implementation Architecture

The interactions support systems rely on technologies such as Internet/intranet, web, object-orientation, client/server and particularly Middleware such as CORBA, DCOM,

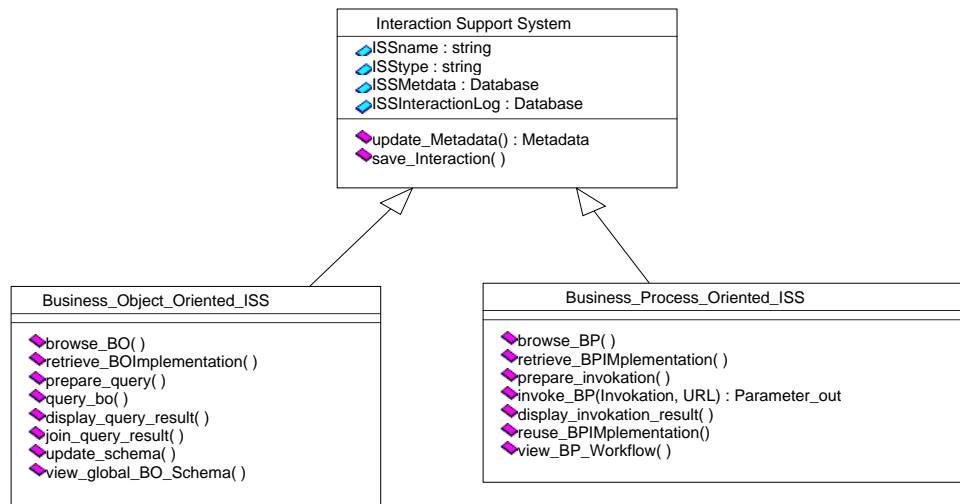


Fig 10. Specialization of the Interactions Support System

RMI, ODBC, JDBC, technologies used to allow an environment (computational infrastructure) allowing different subsystems to communicate and therefore interact. The implementation level deals with deciding implementation architecture according to the criteria of scalability, portability, interoperability and integration.

- Scalability means the possibility to add more services to the ISS (e.g., enterprise portal services, data warehouse services) in order to implement new protocols, and more subsystems without problems.
- Portability means that the ISS component must be decoupled from any particular computing environment.
- Interoperability means working with heterogeneous platforms.
- Integration means that the ISS should require minimal effort to be incorporated

within existing subsystems.

Besides, the architecture implementations must allow us to readily integrate new services, as the ISS functionality will be expanding to take into account new services and new subsystems (Figure 11).

Therefore, the architecture implementation will consist of:

- Interface that presents the interaction support system services to the subsystems committing to interact.
- Service implementation layer contains the implementations for the services of the interface. It includes the logic and control functions
- Data-related services that incorporates data management functionality to access the metadata, and the heterogeneous subsystems business objects implementations. ODBC and JDBC drivers allow access to these heterogeneous implementa-

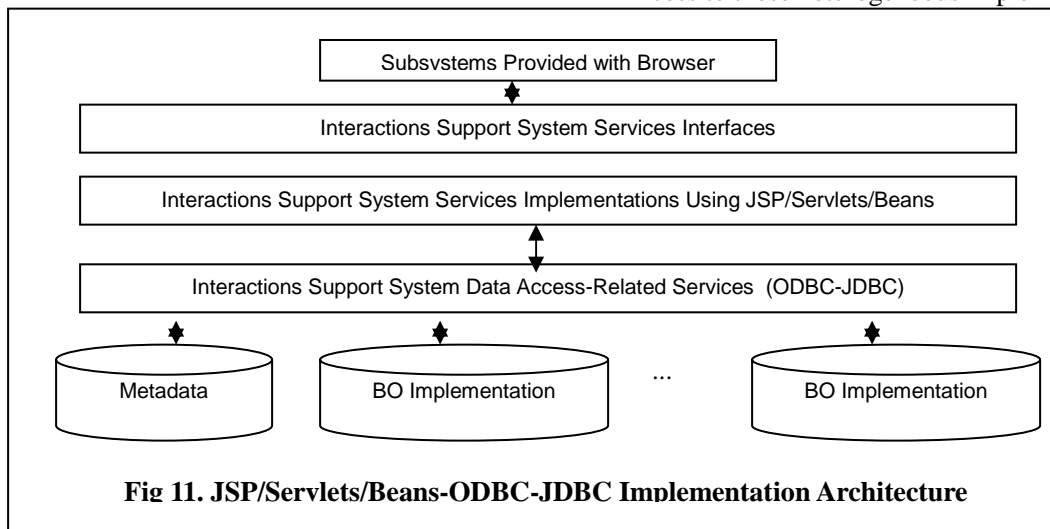


Fig 11. JSP/Servlets/Beans-ODBC-JDBC Implementation Architecture

Web-Based Interactions

tions as all major vendors have dedicated ODBC/JDBC drivers.

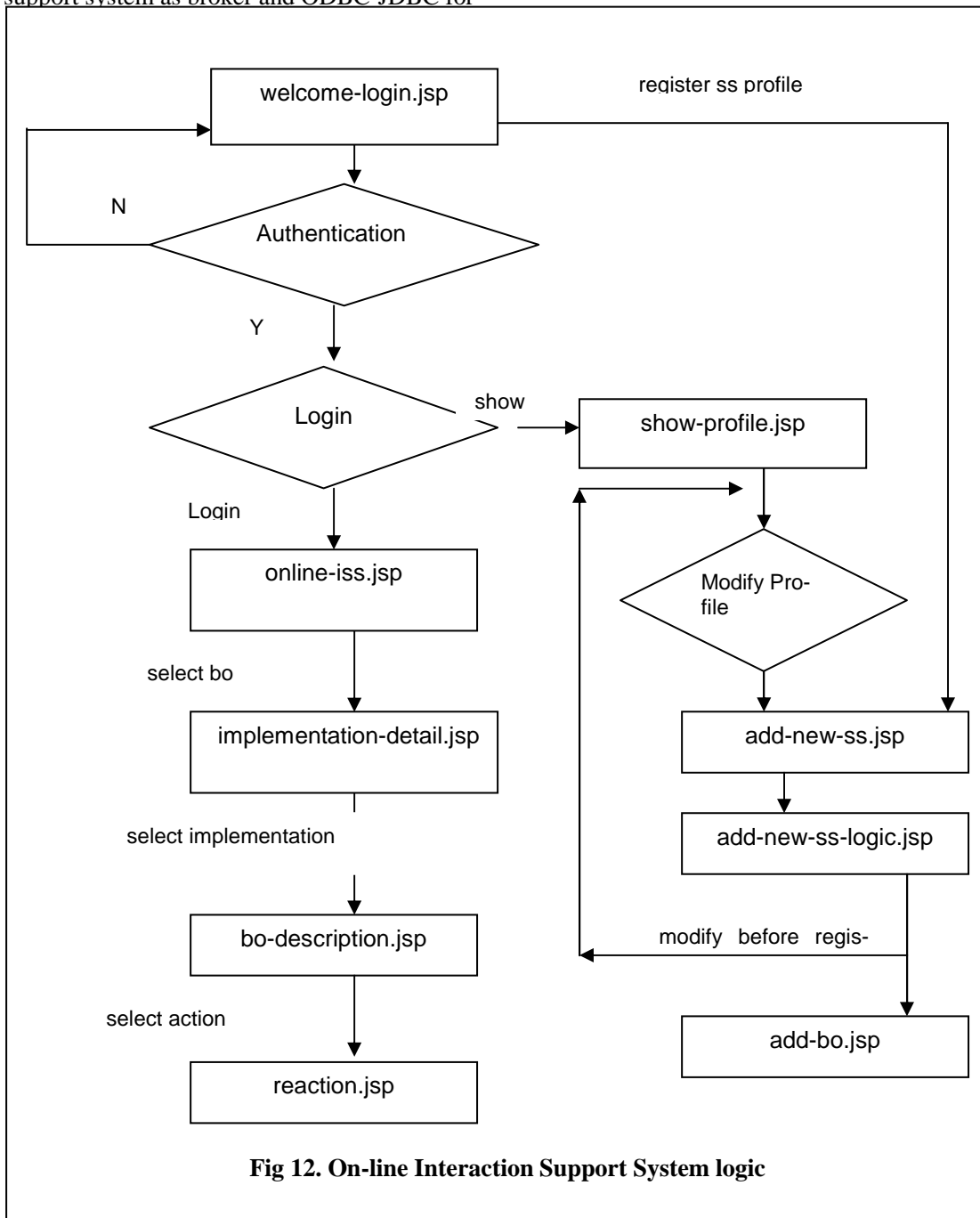
- Subsystems committing to interact.

This work introduces four architecture implementations that may suit our requirement: CORBA architecture, Mobile Agents architecture, XML for exchanging data, and JSP/Servlets/Beans-ODBC-JDBC. All these architectures satisfy criteria of portability, scalability, interoperability and integration.

The implementation architecture based on CORBA involves subsystems wrapped as objects with IDL, the interaction support system as broker and ODBC-JDBC for

action support system as broker and ODBC-JDBC for data-related services. CORBA architecture is well suited when the services of the ISS are distributed over the committed subsystems.

The architecture based on agents (Huhns, 1999) and mobile agents (Papastavrou et al., 2000) requires an environment that allows the execution of the mobile agents (e.g., aglets). Mobile agents may be responsible browsing services of ISS over the subsystems. Mobile agents carry out the query/invoke; have the capability to move, and to be executed on remote subsystems to return the result.



They considerably decrease the network load.

The implementation architecture based on XML assumes the mapping of the subsystems business objects schema and data into XML to be readily communicable through browser supporting XML. It has the advantage to work with structured data (e.g., legacy files or relational databases) as well as unstructured data (e.g., text documents, reports and emails, graphics and images, audio, video resources and web pages).

To validate our work, we choose to implement one specialization of the interaction support system: business object-oriented ISS using an implementation architecture based on Java/Servlet/Beans-JDBC which is the most suitable for the solution we adopted where the interaction elements are encapsulated into a separate subsystem and located in a web server to be used by other subsystems to exchange and share data (Figure 12). Only standard HTTP is used between the web server and clients which are the subsystems committing to interact. Implementing this architecture consists of implementing:

- Metadata that records locations of the implementations of the business objects over the subsystems.
- Business object-oriented interaction component logic that allows any registered (committed) subsystem of the information system to get full online access to the business object interaction support system component, and
- Implementations of some business objects on several subsystems for the validation.

A version can be seen at

<http://faculty.uaeu.ac/~baghdadi/research.htm>

Related Work

There is a large relevant literature on interactions protocols and mechanisms, cooperation, data heterogeneity and integration. We consider the mostly close work to our project namely interactions protocols (Weiss 1999), (Huhns, 1999). Computer-supported cooperative work (Greif, 1988), (Schmidt, 1994), (Mills, 1999), (Ellis, 1999) and (Aiko et al. 2000). Cooperative information systems (Papazoglou, 1997), (De Michelis et al. 1997) and (Arcieri et al., 1999). Heterogeneous Databases (Batini et al., 1986), (Pitoura et al., 1995), (Konopnicki, 1998), (Kwan, 1999) and (Castano et al. 2001). Web information access, retrieval and integration (Arens et al. 1993), (Calavanese et al. 1998) and (Vidal et al. 1998). Agents and Multiagent (Weiss 1999), (Huhns, 1999), and Mobile Agents (Pa-

pastavrou et al.,2000). Reengineering applications and building web-based applications (Umar, 1997), (Lewandowski, 1998) and (Fraternali 1999).

Several interactions protocols, governing the exchange of messages, have been designed. Our framework is an adaptation of the coordination protocol in order to design a computational infrastructure for interacting subsystems.

Computer-supported cooperative work studies how groups work, and how technology can be implemented to enhance group interaction and collaboration. The objective is to develop and implement Groupware which are hardware and software technologies to assist interacting groups. The interaction support system is a kind of groupware to assist artifacts (subsystems) to interact for a global view of business objects and processes they implement.

Cooperative information systems study how to support the human collaborative work. Our framework is rather a touch in this category. However, we use the term of interacting information systems instead of cooperative information systems, as cooperation is only one form of interaction.

Heterogeneous Databases deal with approaches to sharing data among databases designed with different data models and implemented with different DBMS (Network DBMS, relational DBMS or object-oriented DBMS). It concerns with heterogeneity of structured data. While interaction support system deals with structured as well as unstructured data and with process.

Web information integration investigates solution on how to integrate semi-structured data as well as structured data over the web. Its deals with integration while our framework deals with interactions which

Agents and Multiagent are mainly used in distributed artificial intelligence for distributed decision-making and problem solving. Mobile agents are used for accessing distributed database over the web. A framework [Papastavrou et al., 2000) uses the aglet which are agile applets carrying out queries and results. Mobile agents are designed to roam over remote servers provided with an environment to execute the aglets.

The framework described in this paper is an adaptation of one of the several interaction protocols: the coordination protocol based on commitments and conventions among subsystems of the information systems willing to interact for multiple purposes. It uses the web and the Internet/intranets/extranets as main computational infrastructure. The interaction support system is located on a web server and it is accessible by any subsystem provided with a browser.

Conclusion

Web-based interaction support system, an interaction-dedicated subsystem of the information system, is a solution for the breakdowns due to heterogeneity of the subsystems of the information system; and for today's intensive interactions across and outside the enterprise. The interaction services are encapsulated into a separate subsystem and located in a web server to be used by other subsystems to exchange and share data and to perform processes with complete transparency. It can be readily integrated with the existing subsystems provided with browsers.

This is an issue today where interactions across and outside the enterprise helped by the IT communications become more and more intensive and primordial for a business to survive. Artifacts such as interaction support are required to enable interactions and consequently new methodology to do business: e-commerce.

We will show after an exhaustive list of the services that may be provided by the interaction support system, asset and compare more architecture implementations to decide the best suited to all the functionality of the interaction support system.

References

- Alquier. A. M. (1993). Les Systèmes d'Information: Modèles Coopératifs. Dossier d'habilitation. Université de Toulouse 1.
- Arcieri, F. Cappadozzi, E. Naggari P., Nardelli E. and Talamo. M. (1999). Access Keys Warehouse : a New Approach to the Development of Cooperative Information Systems. Proc. of the 4th Int. Conf. on Cooperative Information Systems (CoopIS'99).
- Arens Y., Chee C. Y. , Hsu C.N. and Knoblock C. A. (1993). Retrieving and Integrating Data from Multiple Information Sources. *Int'l Journal of Intelligent and Cooperative Information Systems*, 2(2), 127-158.
- Avedal K. et al. (2000). *Professional JSP*. Wrox Press Ltd.
- Baghdadi Y. and Alquier A. M.. (1996). Cooperative Information System: a Definition, a Framework and a Modeling, Proceedings of the 2nd International Conference on Cooperative Systems, Juan-les-Pins.
- Bannon L.J. and Schmidt K.. (1991). CSCW : Four Characters in Search of a Context. In J.M. Bowers and S.D. Benford (Eds.): *Studies in Computer Supported Cooperative Work*, Elsevier Sciences Publishers B.V, North-Holland.
- Batini C., Lenzirini M. and Navathe S.B. (1986). A Comprehensive Analysis of Methodologies for Database Schemas Integration. *ACM Computing Surveys*, 18(4), 322-364.
- Bouguettaya, A. Benattallah B., Hendra L. and Ouzani M. (1999). Supporting Dynamic Interactions among Web_based Information Sources, *IEEE Transactions on Knowledge and Data Engineering*, 12(5), 779-801.
- Calvanese D., De Gicomo G., Lenzirini M., Nardi D. and Rosati R. (1998). Information Integration: Conceptual Modeling and Reasoning support, Proc. of the 6th Int. Conf. on Cooperative Information Systems (CoopIS'98), 280-291.
- Casati, F. Ceri S., Paraboschi S. and Pozzi G. (1999). Specification and Implementation of Exceptions in Workflow Management Systems. *ACM Transactions on Database Systems*, 24 (3), 405-451.
- Castano S. and De Antonollis V.(1998). A Framework For Expressing Semantic Relationships between Multiple Information Systems for Cooperation. *Information Systems*. 23(3/4), 253-277.
- Castano S., De Antonollis V. and De C. Di Vimercati S. (2001). Global Viewing of Heterogeneous Data Sources. *IEEE Transactions on Knowledge and Data Engineering*, 13(2), 277-297 .
- Coffey J. T. and Klimes M. (2000). Fundamental Limits for Information Retrieval. *IEEE Transaction on Information Theory*. 26(7), 2281-2297 .
- Crestani, F. and Fraternali P. (1999). Tools and Approaches for Developing Intensive Web Application; A Survey. *ACM Computing Surveys*. 31(3), 227-263.
- De Michelis G. and Grasso M. A. (1994). Situating Conversations within the Language/Action Perspective: The Milan Conversation Model. Proceedings of ACM, Conference on Computer Supported Cooperative Work. 89-100.
- De Michelis G., Dubois E. , Jarke M., Matthes F. , Mylopoulos J., Papazoglou M., Pohl K., Schmidt J., Woo C. and Yu E. (1997). Cooperative Information System: A manifesto. In: M. P. Papazoglou and G. Schlageter (Edts). *Cooperative Information Systems Trends and Directions*. Academic Press.
- Erceau E. J. and Ferber J. (1994). Intelligence Artificielle Distribuée et Systèmes Multi-Agents, In *Intelligence collective*. Hermes.
- Ellis C and Wainer J. (1999) Groupware and Computer Supported Cooperative Work. in *Multiagent Systems: A Modern approach to Distributed Artificial Intelligence*, pp 426-457. edited by Gerhard Weiss, MIT press.
- Fraternali P. (1999). Tools and Approaches for developing Data-Intensive Web Applications: a Survey. *ACM Computing Surveys*, 31(3), 228-261
- Greif I. (1988). Computer-Supported-Cooperative-Work. In *Cooperative-Supported Cooperative Work : A Book of Readings*. I. Greif Edition.
- Haag. S., Cummings M. and Dawkins J. (1999). *Management Information System for Information Age*. Irwin Mac Graw Hill.

- Huhns M.N. and Stephens L. M. (1999). Multiagent Systems and Society of Agents. in *Multiagent Systems: A Modern approach to Distributed Artificial Intelligence*, pp 79-120. edited by Gerhard Weiss, MIT press.
- Konopnicki D. and Shmueli O. (1998). Information Gathering in the WWW: The W3QL Query Language and W3QS System. *Transactions on Database Systems*, 23(4), 369-410.
- Kwan I. and Fong J. (1999). Schema Integration Methodology and its verification by Use of Information Capacity. *Information Systems*, 24(5), 355-376.
- Lewandowski S. M.. (1998). Framework for Component-Based Client/Server Computing. *ACM Computing Surveys*,30(1), 3-27.
- Matthes. F. (1997). Business Conversations: a High Level System Model from Agent Coordination. Proceedings of the Sixth international Workshop on Database Programming Languages, Estes Park, Colorado. Springer-Verlag.
- Mills K. L. (1999). Introduction to the Electronic Symposium on Computer Supported Cooperative Work. *ACM Computing Surveys*, 31(2), 106-115.
- Mendelzon A. O. and Milo T. (1998) .Formal Models for Web Queries. *Information Systems*, 23(8), 615-638.
- Papastavrou S., Samaras G. and Pitoura E. (2000). Mobile Agents for World Wide Web Distributed Database Access. *IEEE Transactions on Knowledge and Data Engineering*, 12(5), 802-820.
- Papazoglou M. P. and Schlageter G. (1997). *Cooperative Information System: Trends and Direction*. Academic Press.
- Pitoura, E. Bukhers O. and Elmagarmid A. (1995). Object-Oriented Orientation in Multidatabase Systems, *ACM Computing Surveys*, 27(2), 141-194
- Schmidh K. (1994). The Organization of Cooperative Work: Beyond the Leviathan Conception of the Organization of Cooperative Work. Proceedings of ACM, Conference on CSCW, October 22-26, 1994. Chapel Hill, North Carolina.
- Umar A. (1997). *Application (Re)Engineering: Building Web-Based Applications and Dealing with Legacies*, Prentice Hall.
- Vidal M. E., Rashid L. and Gruser J. R (1998). A meta-Wrapper for Scaling Multiple Autonomous Distributed Information Sources. Proc. of the Int. Conf. on Cooperative Information Systems (CoopIS'98)
- Weiss G. (1999). Multiagent Systems: A Modern approach to Distributed Artificial Intelligence” edited by Gerhard Weiss, MIT press.
- Winograd T. (1988). A language/Action Perspective on the Design of Cooperative Work, In *Cooperative-Supported Cooperative Work: A Book of Readings*. Edited by I. GREIF.

Biography

Youcef Baghdadi, Ph.D. is an assistant professor in the Department of Mathematics and Computer Science at United Arab Emirates University. His current research interests include interacting information systems, cooperative information systems, federated databases, data warehousing, enterprise portal and e-commerce.