Interaction Support System for Unlocking Computational and Informational Resources

Youcef Baghdadi
United Arab Emirates University, Al-Ain, UAE

y.baghdadi@uaeu.ac.ae

Abstract

Due to business innovation and IT, information systems (IS) are made up of distributed autonomous subsystems running on heterogeneous IT platforms. These subsystems implement differently same business objects and processes. Thus pieces of data and processes are overlapping and replicated. Moreover, informational and computational resources are locked i.e. they do exist, but they are not accessible.

This paper specifies an interaction-dedicated subsystem of the IS called interaction support system (ISS). It is a support for i***nteractions.*** It aims to: (1) provide subsystems of IS with a unified and consistent representation of business objects, (2) coordinate processes, and (3) lock data and processes.

An implementation of a web-based business object-oriented ISS, a specialization of the ISS, is made up of four elements: 1) browser used by subsystems to browse and query business objects over the Internet/Intranet, 2) a web server on which run the logic of the ISS, 3) a metadata representing distribution of business objects and processes, and 4) business objects implementations over the subsystems willing to interact.

Keywords: Information System Distribution, Interactions, Cooperation, Coordination, Unlocking Data, Interaction Support System, Web-Based Business Object ISS.

Introduction

Due to business innovation and IT, information systems (IS) are made up of distributed autonomous subsystems running on heterogeneous IT platforms. These subsystems may have heterogeneous representations (e.g., schema) and implementations (e.g., database table, file) of the same business object or different implementation (e.g., function, object) of the same process (or a composition of the process i.e. subprocess). For example, the business object customer may have two different schemas and implementations within the subsystem of the representative and the accounting subsystem. This means risk of (i) inconsistency of business objects schemas and views and (ii) inefficiency of processes. While, all types of control of a business system require transparent and consistent data related to business objects and efficient processes. Besides informational and computational resources (data and processes) are locked i.e. they do exist somewhere but they are not accessible by all the subsystems as they differ by their platforms.

This problem is considered under different aspects: schema integration of heterogeneous and federated databases (Batini, et al. 1986), (Pitoura et al. 1995) and (Castano et al. 1998), information integration (Arens et al. 1993), (Arens el al. 1996), (Calvanese et al. 1998) and (Papastavrou et al. 2000), cooperation and coordination paradigm such as in computer-supported cooperative work (CSCW) which tends to motive and valid GroupWare used to support and enhance activities in which more than one person are involved (Winograd 1988), (Schmidth 1994) and (Mills 1999), GroupWare (Greif 1988), Workflow Systems (Casati et al. 1999), or purely technical problem which may be solved by existing technologies such as Internet computing, client/sever, middleware (e.g., COBRA, DCOM, RMI, ODBC and JDBC) and web applications (Umar 1997), (Lewandowski 1998), (Fraternalli 1999) and (Avedal et al. 2000).

We consider this problem as ***interactions*** problem. That is, a lack of representation and implementation of interaction across the enterprise (among subsystems of the IS) and outside (with external sources representing partners, customers and suppliers). In fact, interaction is an important aspect of the IS as well as business objects and processes. Interaction allow not only a unified and consistent view of business objects and processes, but also ***knowledge emerging from the unlocked data and processes*** which is an add value for controlling the business system.

However, interaction and consequent emerging knowledge do not exist naturally. It is necessary to make them tangible using the world wide web (web), which is a universal mean for network information systems. The web allows distributed interacting subsystems to be built over it and to access a variety of multimedia structured, semi structured or unstructured information representing business objects via web browsers.

This paper presents a framework and an architecture:

(1) For interaction representation which consists of developing a third axis insulated or integrated with data and processes axes.

(2) For the artifact which is a new kind of systems so-called I***nteractions Support System (ISS)***. It co-exists with the different existing subsystems of IS (e.g., legacy systems). The ISS is dedicated to the subsystems interactions each other and with external sources. It mainly provides these subsystems with business objects-oriented and processes-oriented request.

The framework is a conjunction of ***information systems***, ***interaction support system*** and the *w**eb*** as the most adequate implementation technology for this system. The framework emphasizes answers to questions such as: Why do ISs need to interact with each other? How interactions allow unlocking data and processes and therefore emerging knowledge? What are the situations of interactions with respect to the IS? And finally, why the IT makes interactions easier and efficient.

The remainder of this paper is organized as follows: Section 2 presents factors influencing and increasing distribution and heterogeneity of the IS. Section 3 specifies interaction concept and situations of interaction with respect to IS. Section 4 presents structural and functional architectures of the ISS. Section 5 presents an architecture implementation of business object-oriented ISS, a specialization of the ISS, based on JSP/Servlets, Java Beans and JDBC. Finally a conclusion presents the results.

Factors Influencing the Distribution of the IS

The information system is a technology-based image of the business system. This image is made up by a mosaic of information technology (figure 1). Therefore, the IS is rather a set of multiple autonomous subsystems resulting from three main factors which are work organization, business's improvement and innovation and IT.

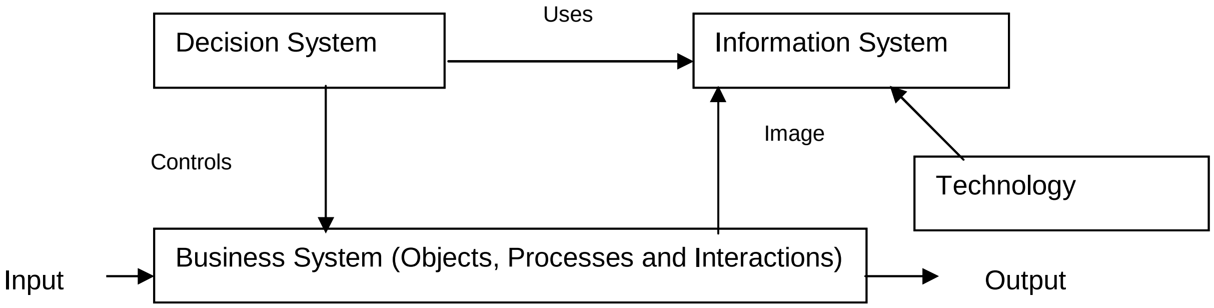


Figure 1: Information System is a Technology-Based Image of the Business System.

1) Work organization structures the business system by functional areas or departments having their respective languages. That is, their own information system with some conceptual and linguistic breakings (e.g., different perception of the same business objects).

2) IT allows readily implementation of local and specific subsystems of the IS characterized by their own IT platforms. Office automation systems provided with browsers are examples of proliferation of technology such as personal computers, workstations and networks (local area networks and intranets).

3) Decision-making and business's improvement and innovation require respectively new types of data modeling (portals, data warehouse, data mart, data mining) and new kinds of IS to support group decision support systems (GDSS), business process reengineering (BPR), business network reengineering (BNR), business scope redefinition (BSR), e-commerce, and corporate portals.

In fact, the IS is an aggregation of subsystems as shown in Figure 2.

This natural structuring of IS leads to data and processes locking (data and processes do exist but they are not accessible), overlapping and replicating of data and processes. Therefore, higher are the consumed resources, and lower are the consistency of data and the efficiency of the processes, which is paradoxical. At the enterprise level, information needed for any action notably decision-making is in part only processed in the local subsystems and a simple data integration is still insufficient. Furthermore, the new kinds of IS supporting: WIS, GDSS, EIS (Executive IS), IOS (Inter-Organizational IS), e-commerce and enterprise portals are mainly based upon interactions as they involve actors from across and outside organization.

All these subsystems share business objects and processes. They cannot perform operations without interactions.

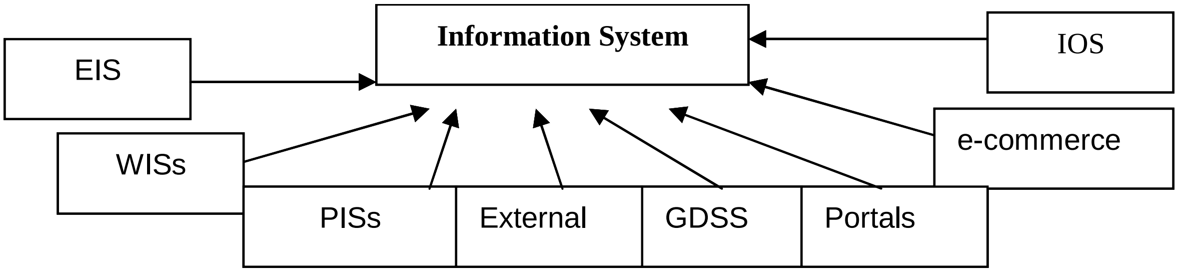


Figure 2: Information System is an Aggregation of Multiple Types of Systems.

Interaction Specification

This section emphasizes representation of the interactions as a third axis that co-exists with the business objects and processes representation axes.

Interaction Definition

Interaction is 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).
- To satisfy lack of informational and computational resources and to unlock data and processes.
- To allow consistent view of business objects and synergy of processes.
- To solve conflicts.
- To improve and innovate business processes.
- To allow ***emerging knowledge*** by unlocking data and processes (which do exist but are not accessible) which is an add value. In fact, local knowledge is static; it is not representative of the dynamics of the organization (subsystems including the internal and external interactions). Moreover, knowledge resulting from interactions is more intelligible, relevant and complete than a simple data integration.

We consider a definition that takes into account interaction aspects such as ***dynamics***, ***tim***e, ***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 and Ferber 1994). They are made by a set of reciprocal actions performed by the subsystems of the IS and partners 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 the question ‘Why do subsystems interact?’ The concepts of interactions and situations of interactions have been used in different disciplines (Linguistics, Organization, CSCW, DAI...) (Winograd 1988), (Bannon and Schmidth 1991), (De Michelis and Grasso 1994) and (Matthes 1997) and defined by many authors. This paper concerns with the situations of interactions with respect to the IS.

Situations of Interactions with Respect to IS

With respect to IS, subsystem may interact with its 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 and Alquier 1996) where the situations are subsequent to the nature of IS which contain informational (data related to business objects) and computational (processes) resources. That is, two obvious situations called respectively ***business object-oriented interactions*** and ***business processes-oriented interactions***.

Business Object-Oriented Interactions is a situation of interactions that deals with the consistency and the semantics of schemas, views and pieces structured 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 data-oriented interactions. For example, a situation in which a subsystem needs to query or multi-query a schema/implementation of a business object (e.g, customer) over heterogeneous platforms.

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 process-oriented interactions. For example, the situation in which a subsystem needs to invoke another subsystem for performing/running a particular task/process that it cannot perform (with its own resources) is a process-oriented interaction. It deals also with situation in which we need to articulate distributed business process activities (e.g., Workflow Systems).

Representation of the Interactions Aspect

Traditional IS 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 (insulated or integrated) at both logical and physical levels. We add interaction as a third axis which is as relevant as the two traditional axes (figure 3). This interaction axis will be specified at the logical level and implemented using the adequate technology at the physical level.

Aspects ***Abstraction***	***Business Objects*** ***(Data)***	***Business Processes*** ***(Functions)***	***Interactions*** ***(Middleware, Network)***
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
Physical Level	Data and Functions, Objects, Software Components		Specific Mediator (Web-based Interactions Support System, Developed Component) or existing Middleware (CORBA, DCOM, RMI, ODBC, JDBC), XML

Figure 3: Interactions Axis within Information System Development Life Cycle.

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 (ISS) is a subsystem of the IS. 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 heterogeneous sub-systems and their environment.

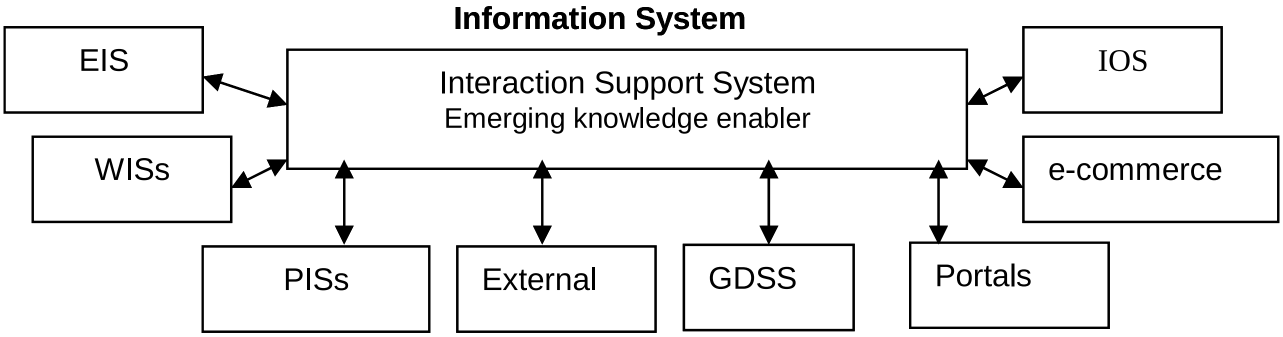


Figure 4: Interaction Support System within the Information System Information.

The ISS represents global vision of the interactions across and outside organization. It aims to enable different situations of interactions among a large number of pre-existing information sources including legacy systems and external sources. It is a subsystem of the IS. It co-exists with other subsystems (figure 4) allowing them to interact and to produce emerging knowledge (information and processes) by unlocking data and processes. This emerging knowledge is used by the subsystems having insufficient informational and computational resources to achieve their respective goals.

Interaction Support System Structural and Functional Architecture

The ISS is specified by a structural and function architectures.

Structural Architecture

A structural architecture of the ISS consists of: a set of functionality of supported by a ***metadata*** that describes the distribution and the implementations of the business objects and processes of the IS, an ***interaction log*** to keep track of the pertinent interactions which will be reused, the management of the metadata and the interaction log (figure 5).

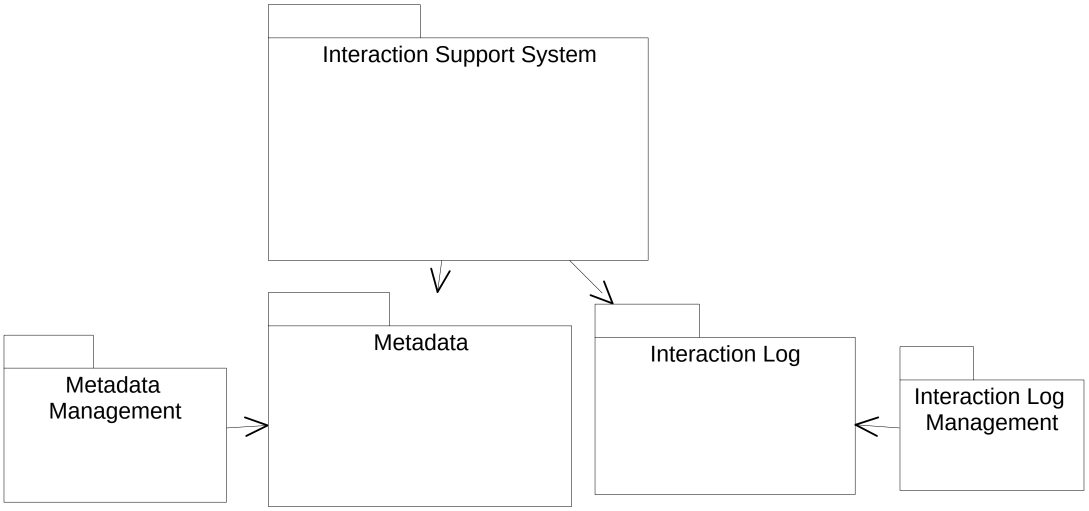


Figure 5: UML Packages Representing the Structural Architecture of the ISS.

The ISS functionalities are responsible for:

- Presenting interfaces to the subsystems of IS.
- Checking the login of each subsystem.
- Registering subsystems general profiles, and their business objects and processes profiles.
- Browsing business objects (locations, synonyms, implementations and schema).
- Browsing business processes (locations, hierarchy, implementations and parameters).
- Querying, viewing business object and restructuring business objects.
- Invoking, reusing and reengineering business processes.

These functionalities are packed into a UML package (figure 6) and are considered as functions of a molecular class.

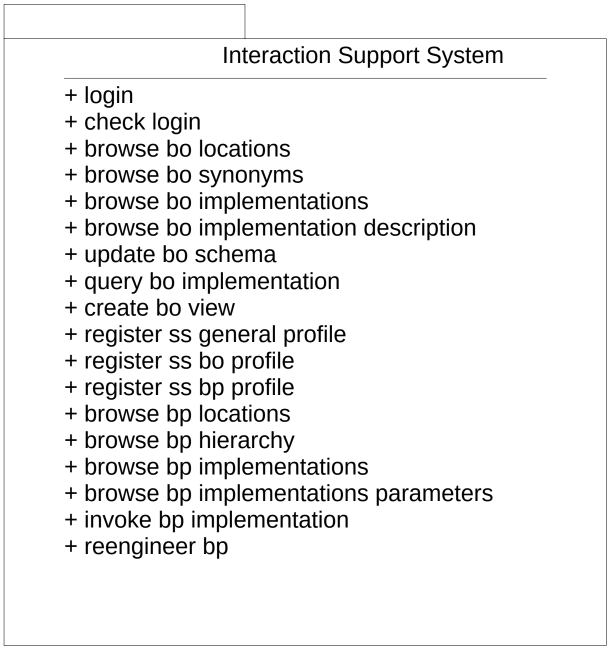

Interaction Support System
+ login
+ check login
+ browse bo locations
+ browse bo synonyms
+ browse bo implementations
+ browse bo implementation description
+ update bo schema
+ query bo implementation
+ create bo view
+ register ss general profile
+ register ss bo profile
+ register ss bp profile
+ browse bp locations
+ browse bp hierarchy
+ browse bp implementations
+ browse bp implementations parameters
+ invoke bp implementation
+ reengineer bp

Figure 6: Interaction Support System Functionality as UML Package.
Where ss : subsystem bo : Business object bp : Business process

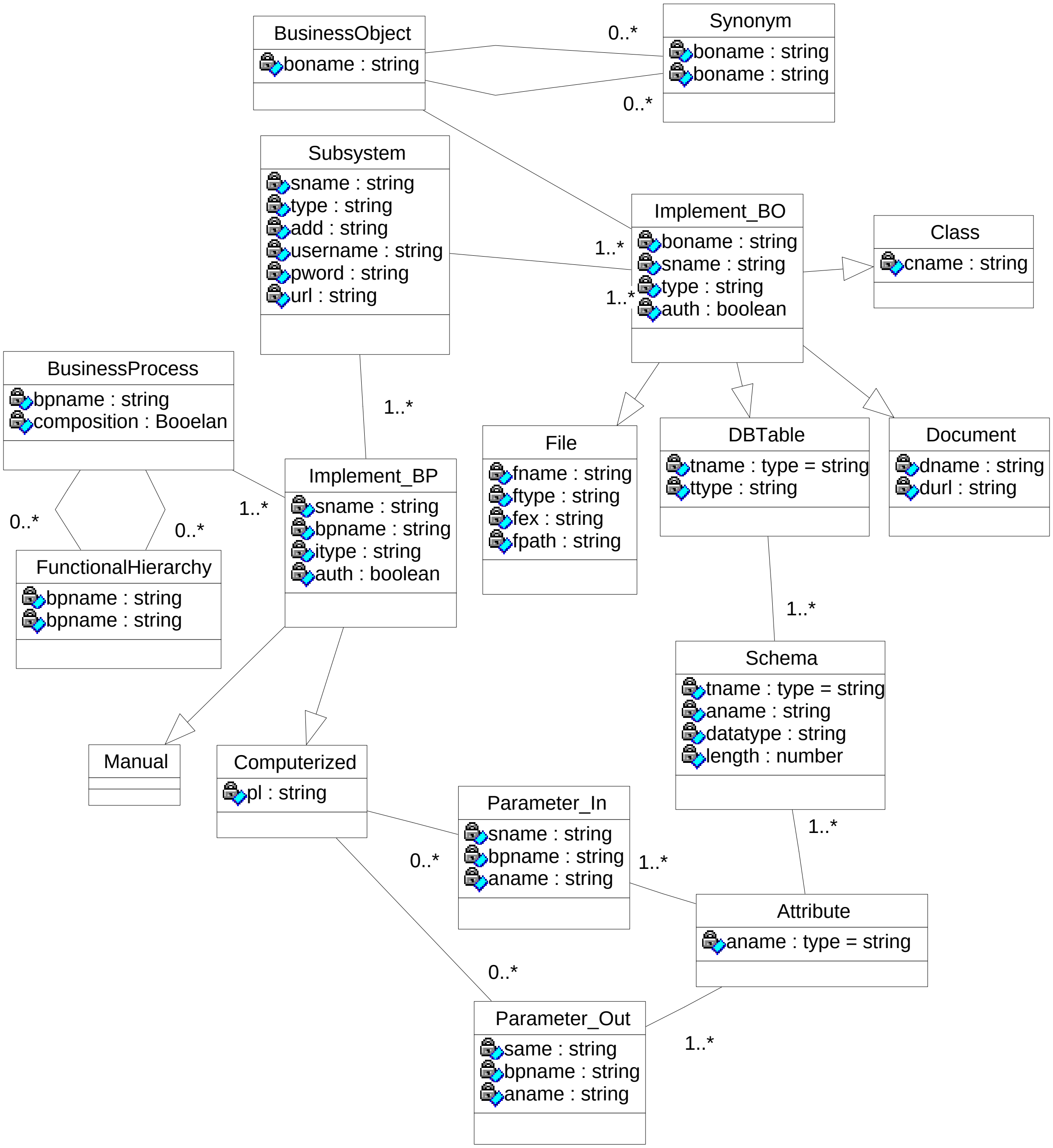


Figure 7: Metadata Class Diagram (with UML).

The ISS uses a metadata (figure 7) that keeps track of pieces of data related to different implementations of respectively business objects and processes across the enterprise and outside. While the interaction log records all the relevant interactions to be further used for tracing a global schema of the business object over the subsystems, or for tracing the business process workflow.

The metadata represents the description of the subsystems and the business objects and processes they implement. It is represented as a set of classes according to the UML:

Subsystem which is described by: Id, name, login password, type (PIS, WIS, EIS or external source), address (e.g., URL, username and password) and administrator. Examples of subsystems are stores, representative, billing departments and son on.

Business Objects. A business object may have different implementations within several subsystems. It is described by ***attributes*** on each distinct ***implementation*** and may have some ***synonyms***. It is then described as follows: name, set of attributes for each implementation on each subsystem, synonyms (e.g., an ***implementation*** of a business object may be: web document, manual document, database table, file, or class). For example, the business object customer may be implemented as database table within the accounting subsystem and Excel sheet within representative subsystem. It may be described differently on each subsystem of the IS. That is, its attributes differ from one implementation to another.

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 example, the order entry process is a process that may be decomposed into many sub processes distributed over the different subsystems which are store, representative, billing and accounting 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.

Functional Architecture

A functional architecture of interacting information system is based on interaction support system as middle-tier between the subsystems of the information system ((ii) a subsystem trigger willing to browse business objects or processes and (iii) one or more involved subsystems that host the business objects and processes implementations). The subsystems of the information system implement their own data schema, pieces of data related to business objects and packages implementing business processes.

Figure 8 shows that subsystems (either tiger or involved in interaction) are specified as UML package with data schema, data and functions. They use (double arrow) the ISS to interact. The ISS is also wrapped as UML package (Interaction Support System Package) that wraps all its functionality (figure 6). ISS uses (simple arrow) a metadata (Metadata Package) and interaction log (Interaction Log Package). Both the metadata and the interaction log are managed by respectively management systems (Metadata Management Package and Interaction Log Management Package).

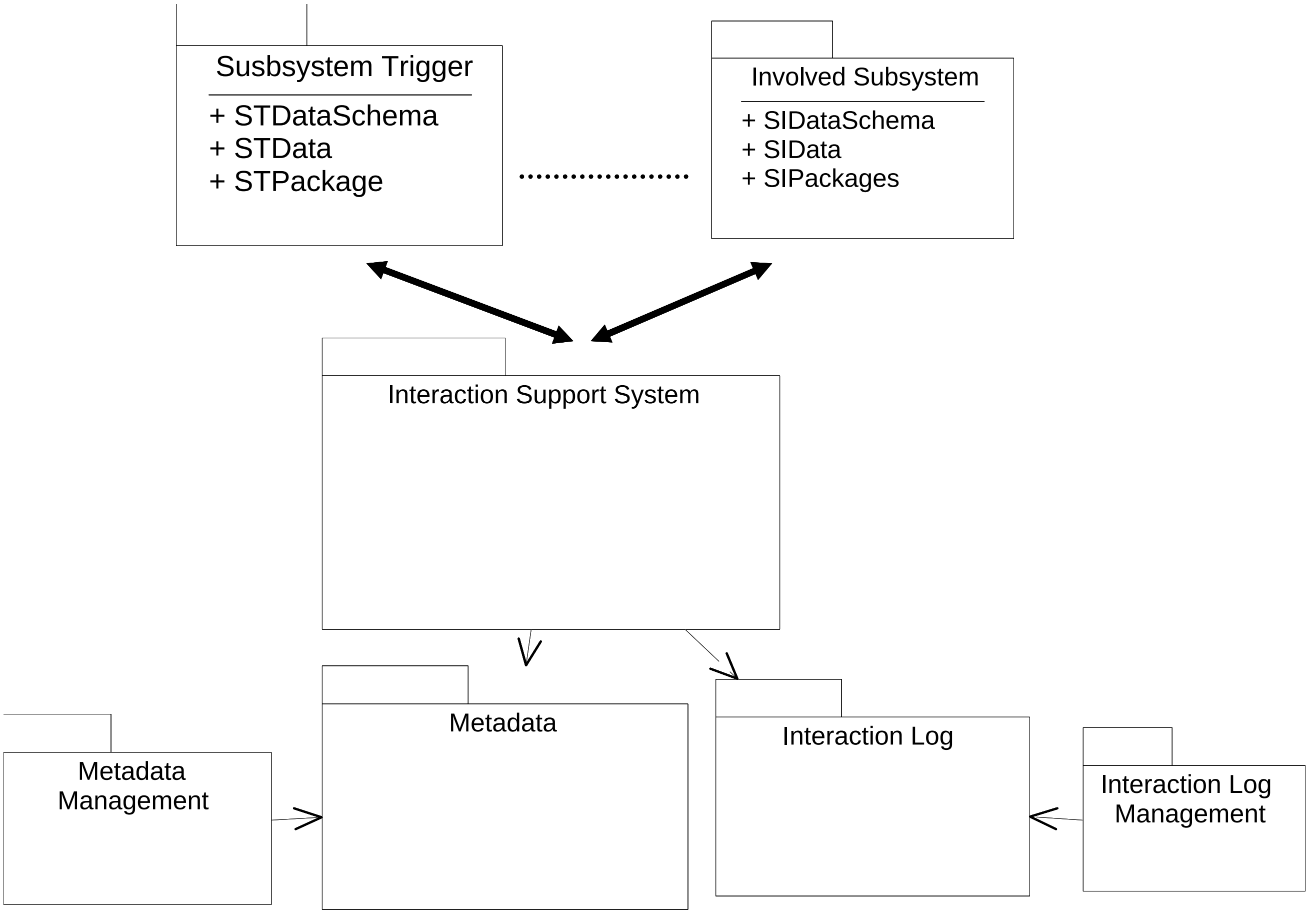


Figure 8: Subsystem Represented as UML Package

Implementation

This paper concerns with the implementation one specialization of the interaction support system. That is a business object-oriented interaction component, which is based on JSP/Servlets, JDBC and RDMBS. JSP/Servlets can easily accommodate this implementation architecture. Figure 9 below illustrates how a JSP page may interact with RDBMS via JDBC.

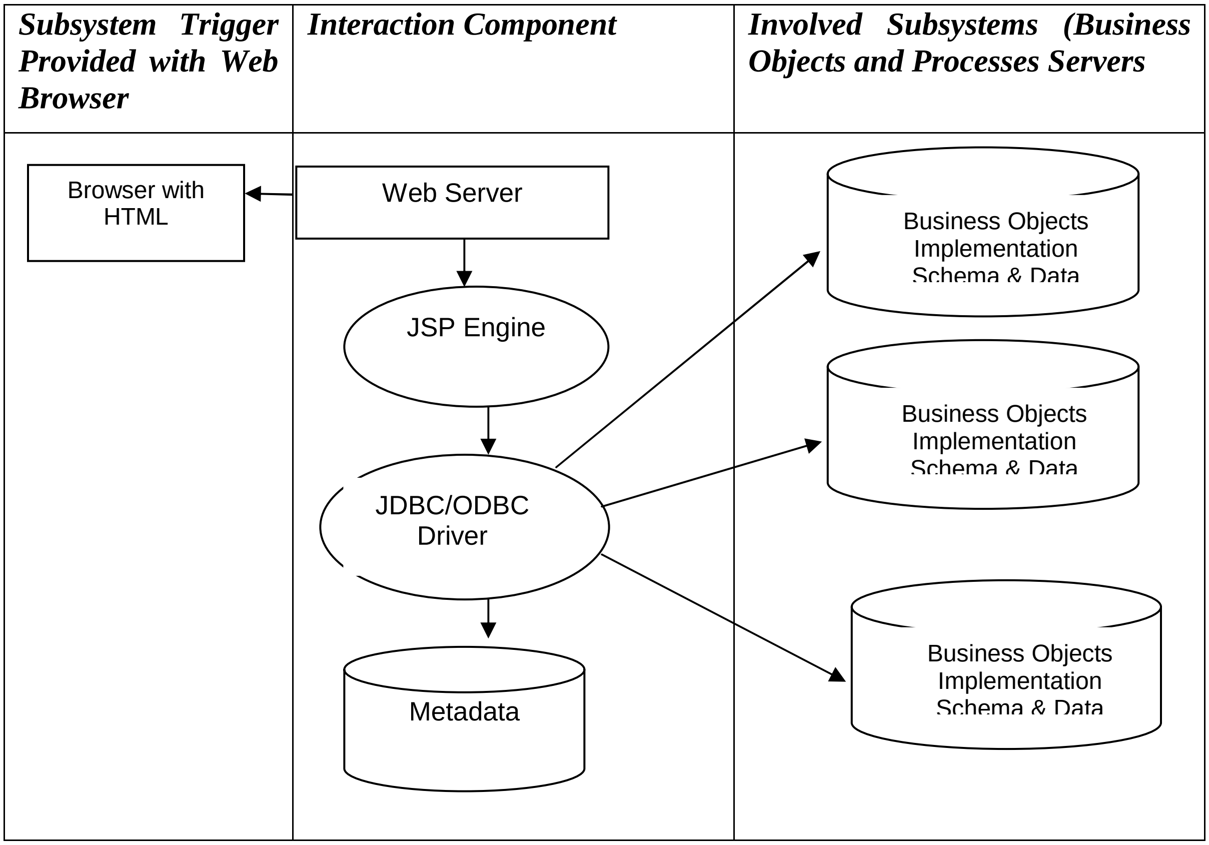
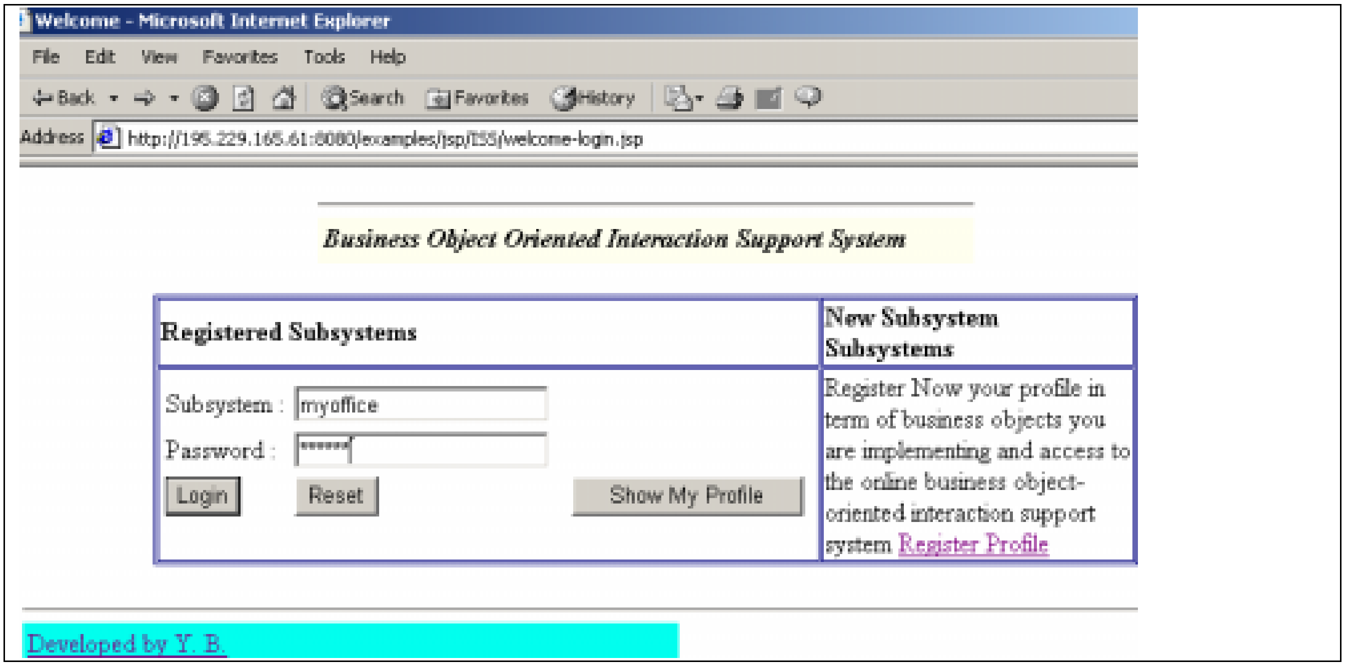

Subsystem Trigger Provided with Web Browser
Interaction Component
Involved Subsystems (Business Objects and Processes Servers
Browser with HTML
Web Server
JSP Engine
JDBC/ODBC Driver
Metadata
Business Objects Implementation Schema & Data
Business Objects Implementation Schema & Data
Business Objects Implementation Schema & Data

Figure 9: JSP/Servlets-JDBC Implementation Architecture

Implementing this architecture consists of implementing (1) the metadata which records the implementations of the business objects over the subsystems (2) the business object-oriented interaction component logic that allows any registered subsystem of the IS to get full online access to the business object interaction component, and (3) different implementations of some business objects on several subsystems for the validation.

Metadata. Implementing the metadata consists of two four steps: choosing the metadata manager (DBMS or file system), mapping the metadata classes diagram (figure 7) into the DBMS data model or file structure, implementing the schema and populating the metadata from a real world or case study chosen to validate the architecture.

Interaction component logic. The interaction component logic begins by prompting the subsystem to login or to register its profile. A list of cooperative JSP pages allow the user to fully use the business object-oriented interaction component to browse business objects schema and implementations over the registered subsystem. The entire logic is not reproduced here. However the two main pages for registering subsystems, logging for browsing business objects and business objects locations are reproduced in the figure 10 and 11. These figures show that a registered subsystem “myoffice” has logged for searching the business object “student”. It gets the implementations of the business object “student” over all the registered subsystems.


Welcome - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address

http://195.229.165.61:8080/examples/jsp/ISS/welcome-login.jsp

Business Object Oriented Interaction Support System

Registered Subsystems

Subsystem : myoffice

Password : ******

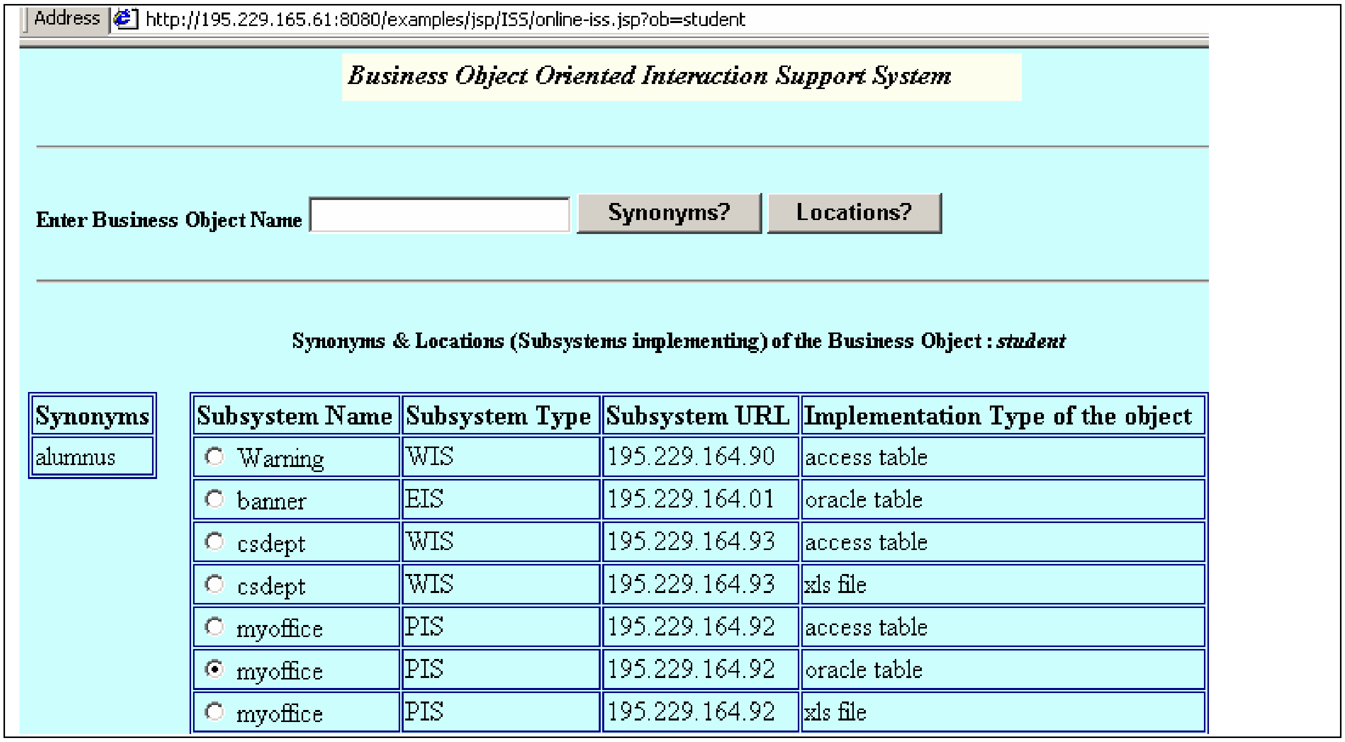
Login
Reset
Show My Profile

New Subsystem Subsystems

Register Now your profile in term of business objects you are implementing and access to the online business object-oriented interaction support system Register Profile

Developed by Y. B.

Figure 10: welcome-login.jsp (user is allowed multiple choices).



http://195.229.165.61:8080/examples/jsp/ISS/online-iss.jsp?ob=student

Business Object Oriented Interaction Support System

Enter Business Object Name

Synonyms & Locations (Subsystems implementing) of the Business Object : *student*

Synonyms
alumnus

Subsystem Name	Subsystem Type	Subsystem URL	Implementation Type of the object
○ Warning	WIS	195.229.164.90	access table
○ banner	EIS	195.229.164.01	oracle table
○ csdept	WIS	195.229.164.93	access table
○ csdept	WIS	195.229.164.93	xls file
○ myoffice	PIS	195.229.164.92	access table
● myoffice	PIS	195.229.164.92	oracle table
○ myoffice	PIS	195.229.164.92	xls file

Figure 11: online-iss.jsp (user searches for a business object implementation).

Conclusion

Interaction support system, an interaction-dedicated subsystem of the IS, can be either an integrator of the subsystems including the legacy systems across and outside the enterprise, support for the collaborative work, or enabler of portal metadata. It allows subsystems to interact via specialized interaction components for structured as well as unstructured data cohesion, processes efficiency and unlocked informational and computational resources.

A web based business object-oriented interaction support system, a specialization of the interaction support system, has been implemented using JSP/Servlets, Java Beans, JDBC and Oracle 8i . It is based on a metadata that describe the different implementations of the business objects over the subsystems willing to interact and to share these business objects.

This is an issue today where organizations try to improve collaborative work and portals because the control of the business requires unlocking consistent and efficient informational and computational resources that are implemented over multiple heterogeneous subsystems running on different IT platforms.

We will show after how to implement all the functionality, including the business process-oriented interaction support system and other usages of the interaction support system.

References

Alquier. A. M. (1993). Les Systèmes d'Information: Modèles Coopératifs. Dossier d'habilitation. Université de Toulouse.

Arens Y., Chee C. Y. , Hsu C.N. et al. (1993). Retrieving and Integrating Data from Multiple Information Sources. *Int'l Journal of Intelligent and Cooperative Information Systems*, 2(2) : 127-158 .

Arens, Y. Knoblock C. A. and Shen W. (1996). Query Reformulation for Dynamic Information Integration, *Journal of Intelligent Information Systems*.

Avedal K. et al. (2000). Professional JSP. Wrox Press Ltd. USA.

Baghdadi Y. and Alquier A. M. (1996). Cooperative Information System: a Definition, Framework and a Modeling, Proceedings of the 2nd International Conference on Cooperative Systems, Juan-les-Pins.

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), pp 280-291.

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* .

Erceau E. J. and Ferber J. (1994). Intelligence Artificielle Distribuée et Systèmes Multi-Agents, In Intelligence collective. Hermes. France.

Fraternalli 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.

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, August.

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. et al. (1997). Cooperative Information System: A manifesto. In: M. P. Papazoglou and G. Schlageter (Edts). Cooperative Information Systems Trends and Directions. Academic Press.

Mills K. L. (1999). Introduction to the Electronic Symposium on Computer Supported Cooperative Work. ACM Computing Surveys, 31(2):106-115.

Papastavrou S., Samaras G., 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., Elmagarmid A. *(1995).* Object-Orientation in Multi database Systems, *ACM Computing Surveys, 27(2): 141-194*

Schmidth 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.

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

Dr. Youcef Baghdadi is an assistant-professor 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. He received his PhD in computer science from University of Toulouse I, France.