Intelligent Agents in the Architecture of Unified Information Space of a Virtual University

Vadim A. Ermolayev, Vjacheslav A. Tolok

Zaporozhye State University, 66, Zhukovskogo st., 330600, Zaporozhye, Ukraine

E-mail: {eva, tolok}@zsu.zaporizhzhe.ua

The implementation of enterprise/corporate wide software systems and networks today is probably one of the most complex problem for Information System (IS) designers and Information Technology (IT) specialists. From one side the Universe of Discourse we examine has natural tendency to be rapidly changed within its life cycle. From the other — there is always a severe dichotomy between the conception of system designer and the way the end user understands the functions and the roles of software he is using in his everyday work. The most common user perception of software system may be demonstrated by the following opinion: "I don't understand nothing in computers except the fact it should think for me…". As far as we probably have no chance to change the majority of users to software system designers this semantic gap will exist for a long time. Moreover, this entanglement of different aspects of information retrieval causes additional training costs, programming delays and deficiences in program reliability and flexibility. Therefore, the main task for IS and IT developers is to continue deluding their clients and to supply software with virtual intellectual properties. The most evident capabilities we might need for the purpose are: IS should offer the means for getting the resources and solving the problems the user might have an intention to receive or solve, IS should possess the interfaces intuitively clear to the user with navigation native to his environment, IS should transparently and adaptively absorb all the programs, tools and another components which are already "thinking for the user".

PROJECT OUTLINE

The outline of the presented research project may actually be denoted as bridging the gap between the mentioned common user perceptions and the lower layer IS components performing formally defined tasks and functions and thus providing a user with a virtual native looking media to instruct the enterprise IS what to “think about” and which task to perform. The project is aimed to implement mentioned capabilities in a kind of an enterprise-wide information media based on a CORBA like architecture - the Unified Information Space (UIS). The Unified Information Space is developed as a logical buildup upon the University – wide Network. As any other enterprise level computer environment the University – wide Network is a complex heterogenious distributed system comprising various types of hardware and software components on different levels. The main task of Unified Information Space is therefore to serve as a homogenizing virtual media providing transparent access procedures and visual navigation tools for the user, desired properties of adaptability, flexibility and reusability for system designer and effective control means for system administrator.

VISUAL NAVIGATION, OBJECTS AND INTERFACES

Keeping in mind that the major effort should be applied to the upper side of the iceberg — the closest to the user, we are paying strict attention to the interface of user - computer interaction. For the moment we are in the process of the design of Unified Visual Intranet Interface (UVII). Its main task is to provide the means for visual navigation in enterprise model. It's obvious that today's computerized enterprise environment is so complicated that attempting to overview all the information is a despairing task. The

way we are trying to solve it is to design visual interfaces which naturally imitate the structure and appearance of real objects — a University, a building, a department or an office, a server or a workstation, a group or a person, a function or an application. For the moment we are almost certain that this object set is finite and, moreover, the number of elements seems to be not too big. Thus, we have an opportunity to design an interface object collection and relate its elements to the appropriate objects of the real world. This is the first step of virtualization on our way of narrowing existing semantic gap.

While modeling an enterprise we most evidently shall examine the real world elements and relationships from several points of view. The first one is topological and it should answer the questions like:

How do the building and floor plans look like?

What is placed to the room N on a certain floor plan?

What are the network nodes presented by the plan of room N?

What are the shared resources of workstation S?...

Whereas the second point of view is structural or organizational and is aimed to resolve the following problems:

Where is department D: buildings, floors, rooms?

Who is the head of department D and what does his/her staff comprise?

What is the structure of department D and the relationships to another departments?...

And one more point of view is definitely functional. The tasks to be solved in this projection are likely to be as follows:

What are the services and information resources provided by room N, department D or server S?

Which enterprise elements will supply us with the answer to the request R?

What information we may and what information we may not expect to receive?...

We classify interface objects by belonging to a certain modeling scope: topological, organizational or functional. For the moment we have worked out and submitted to publishing the topological subset and have partially completed the organizational one.

☞ The task of functional interface objects subset design is still in line and we'll appreciate any advises and cooperation on the point.

Anyway, more deep we go — more complexity and challenges we meet on the way. The first thing to be mentioned is that outlined model projections are very tightly linked to each other and form intersections on numerous directions.

☞ Another challenging problem is to design the mediators, the agents and the executives for the "ferry". The user coasts are more or less defined, but what about the opposite side and the boat itself?

We are trying to solve the problem by designing a multilevel architecture for the Unified Information Space.

ARCHITECTURE OF THE UNIFIED INFORMATION SPACE

The first level of Unified Information Space architecture is the User or the External Level. The interfaces of this level are: the request the user passes to Unified Information Space and the result Unified Information Space returns in the form of HTML code to the user's browser. Next level is the one of Enterprise Information System. The task of this level is to transform the user's request and to generate the virtual query to Enterprise IS. After the query is processed and the result of the query is returned back the turn of one more level component comes. Query Result Processor generates appropriate HTML code from the results of the query and returns it to the upper (User) level of the architecture. The architecture of External and Enterprise IS Levels is schematically shown on Fig 1.

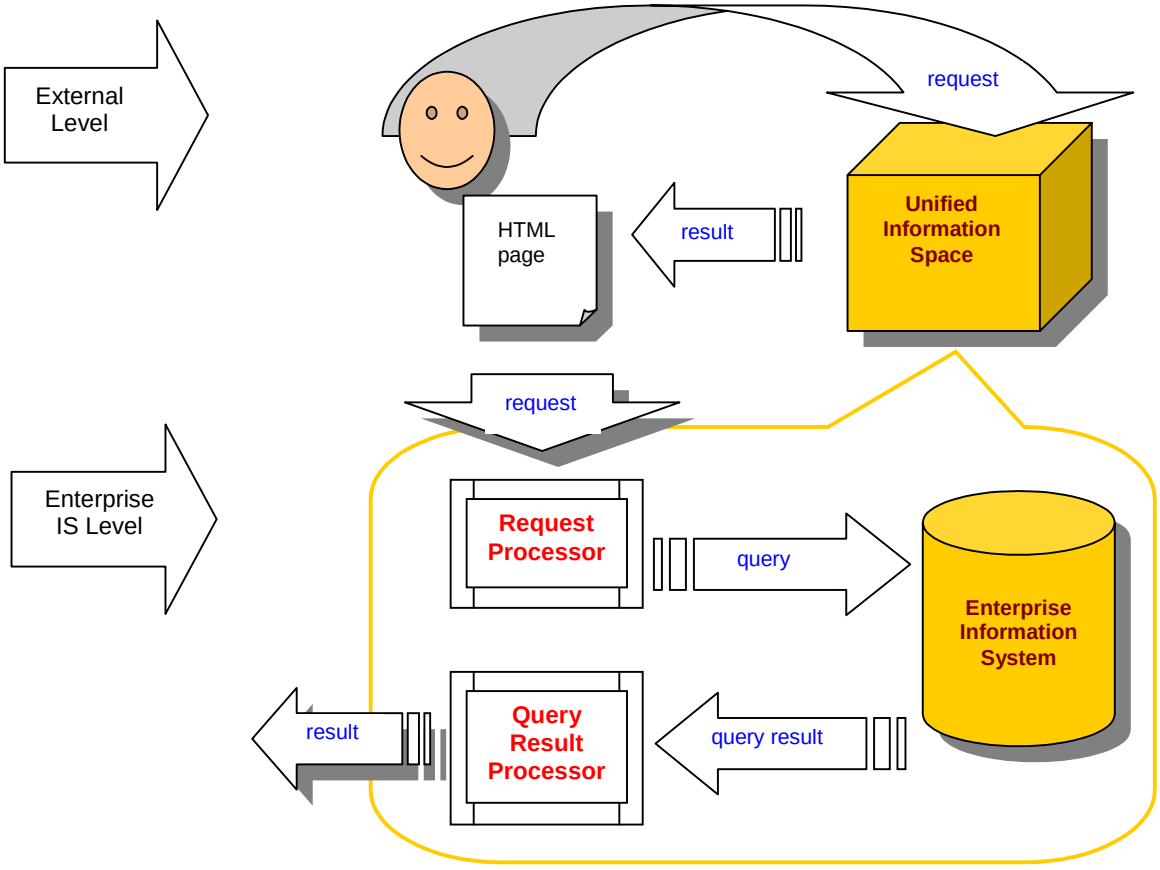

Fig 1. External and Enterprise IS Levels of Unified Information Space Architecture

After we've converted user's request to query we immediately gain the opportunity to describe UIS as well as UVII in the terms of data model, data entities/objects and relationships and process the requests as queries by means of DBMS. This transition from HTML and request to query gives us pretty much for formalizing the interfaces, data retrieval and maintenance routines as far as we receive data model and DBMS as quite powerful tools of automation. For the implementation of this level of architecture we need just data model for the description of interface elements, a DBMS, user request processor and query result processor. For today the interface data model is partially defined (topological and organizational aspects) and soon be published, the algorithms for user request and query result processors are designed and waiting for implementation.

Unfortunately more problems arise when we start analyzing virtual query processing. Enterprise IS is actually quite a complex aggregation of the set of Local Information Systems and Functional Servers distributed over the Enterprise Network and handling local information, applications and resources. The data and functional characteristics of the applications and the resources often have semantic overlaps. One of the known methods of resolving the problem of this semantic intersection is the use of Federative Data Models and DBMSs. The methods of federalizing data are quite well developed for the moment and are widely discussed. For now we plan to use the method and the model developed in Magdeburg Otto-von-Guericke University by the group of prof. G. Saake enhancing it with our own Active Data Dictionary (ADD) approach.

☞ We anyway feel that the room is steel free for agent-based means for query rooting and semantic overlap resolution in federated media and will appreciate discussion and cooperation on the point.

Thus, the Federation Level of Unified Information Space architecture may be schematically presented by Fig 2. This level comprises two layers with corresponding interfaces - the Federation Layer with functions very much like the External Level of federative data model and the set of local information resource servers, interacting with the Federative Layer by means of query based interface. As outlined on Fig 2, main functions of the Federation Layer are the control of transparent aggregation of Local Level data models, incoming query translation distribution and redirection as well as processing local query results, which are in general case coming from more than one local Information System and/or local

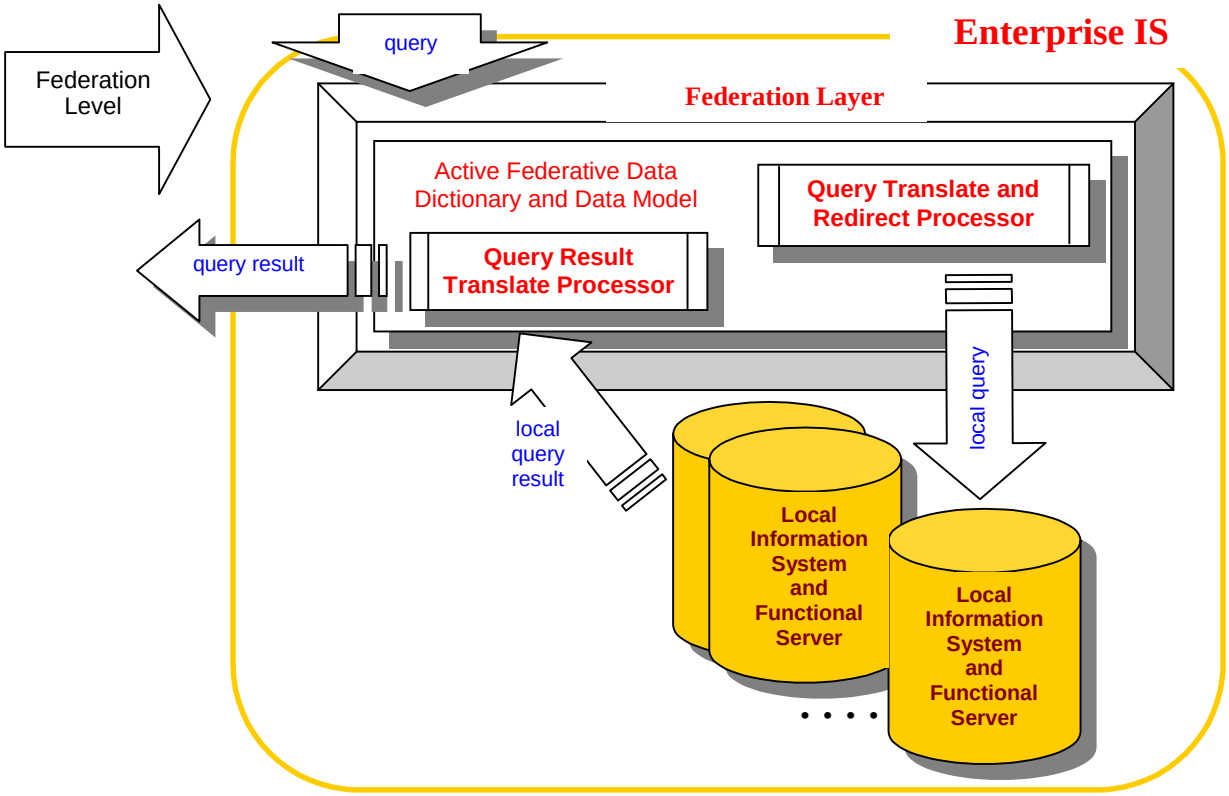

Fig 2. Federation Level of Unified Information Space Architecture

resource server. Thus, the Federation Level of Unified Information Space architecture interprets the incoming virtual query into the set of local queries and appropriately redirects them to the components of the Local Level of the architecture.

The main advantage we have at the Local Level is the method of Local IS design and control. We use Active Data Dictionary as IS control shell and Enhanced Metadata Repository as IS control engine. Local Level comprises three layers: Program Code Layer, Metadata Repository Layer and Data Layer (refer to Fig 3) Data Dictionary with its active behavior function controlled by Data Model serves as an intelligent flexible interface between the upper levels and local data and/or resource. This changes in Local IS architecture definitely affect the technology of IS Design. This modified design procedure is based upon certain properties of our Active Data Dictionary and underlying Enhanced Data Model.

Still there is the problem on the local level been not resolved yet:

☞ We are looking for a model and a tool of creating/generating the agents for execution local IS clients in an internet browser environment.

ACTIVE DATA DICTIONARY AND THE UNDERLYING MODEL

The main function of Active Data Dictionary is to supply the information system with the properties of self-adaptation to data model changes. If we for determinacy purposes restrict the discussion to the UIS Local Level and Client – Server architecture, ADD may be denoted as an intellectual data model controlled mediator between the IS application clients and the database server part. The principal chart of Client-ADD-Server liaison is given on Fig 4.

The basic ADD and Enhanced Metadata Repository model is extended relational. The main enhancement we've made to the relational model is the method we control the IS client code and reflect the changes in IS data model. It is based upon some new formal features of the so-called control attributes (data attributes with some specific properties), which have relationships with program code objects. Thus, the active behaviour of ADD is bi-directional – refer to Fig 4. Data direction exposures are based upon

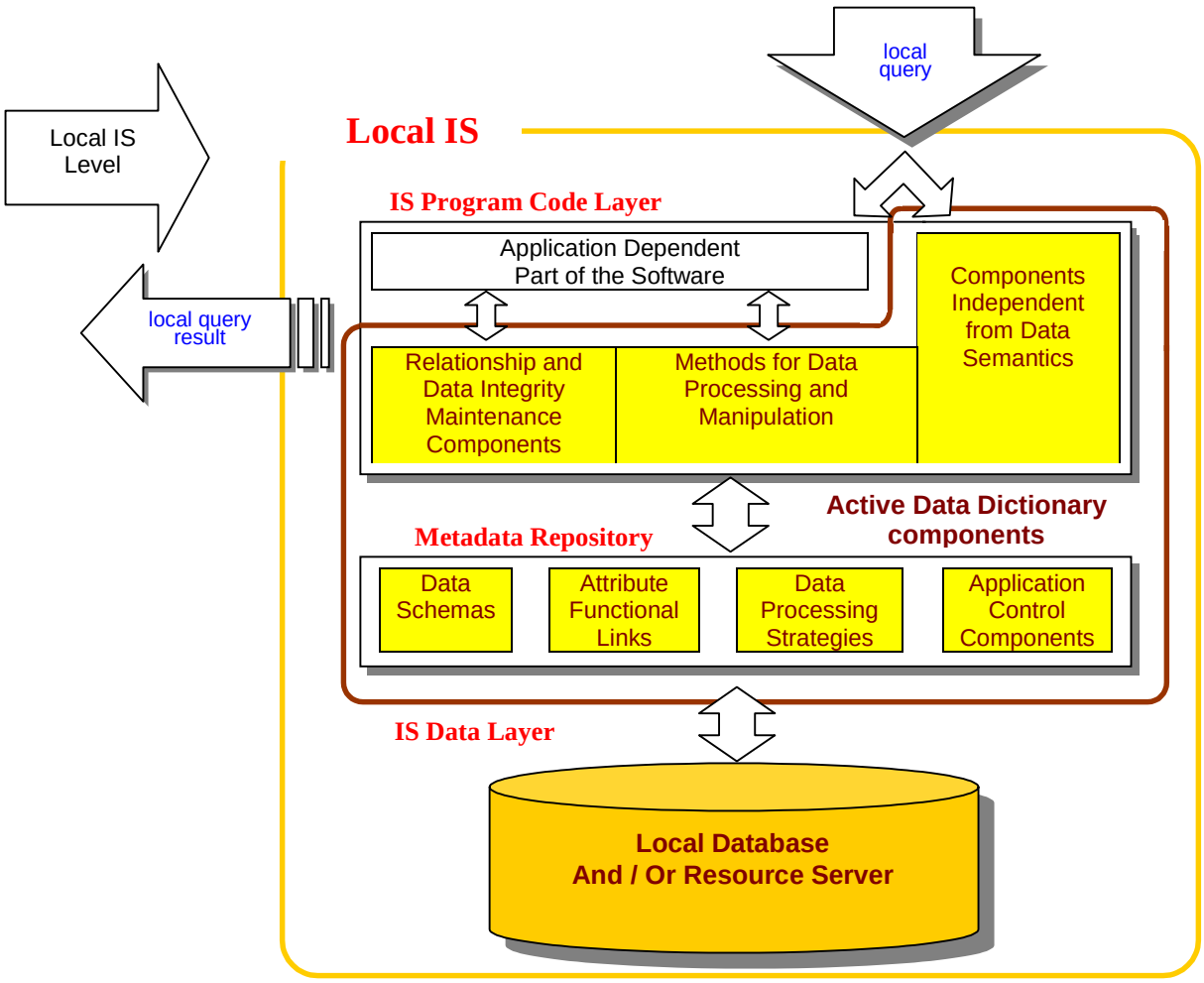

Fig 3. Local IS Level of Unified Information Space Architecture

known ECA rules, whereas program code control is executed by special ADD code embedded into the client under the supervision of ADD Server.

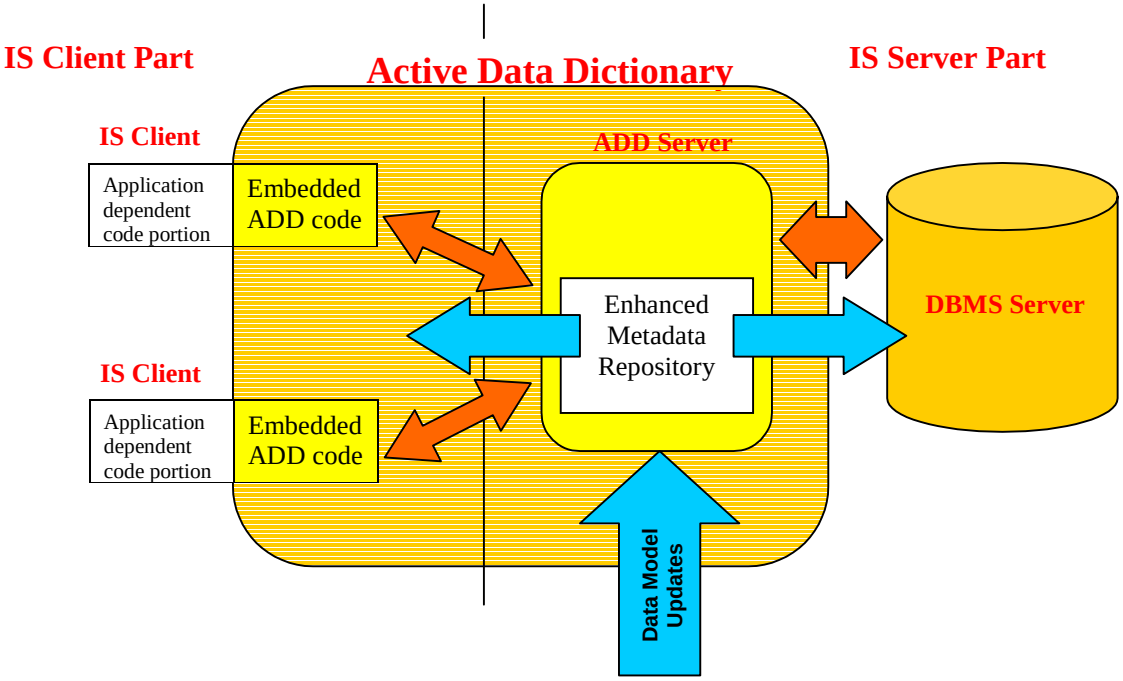

Fig 4. The principal Chart of Client-ADD-Server Liason

SUMMARY AND THE POINTS FOR DISCUSSION

The main task of Unified Information Space is to serve as a homogenizing virtual media providing transparent access procedures and visual navigation tools for the user, desired properties of adaptability, flexibility and reusability for system designer and effective control means for system administrator.

The architecture of the Unified Information Space is multi-level. Each level adds more details to the mechanisms of data processing and retrieval and contributes to bridging the gap between the common user perception of IS and the real thing as it is designed and implemented together with its low level formal interfaces.

The upper level corresponds to the user-oriented Unified Visual Intranet Interface The way we are trying to build it is to design visual interfaces which naturally imitate the structure and appearance of real objects — a university, a building, a department or an office, a server or a workstation, a group or a person, a function or an application. We classify UVII objects by attributing them to a certain modeling scope: topological, organizational or functional. The problems to solve are:

- Design of functional interface objects subset.
- Design of the integral interface object model, comprising the intersections of the objects and relationships between the objects belonging to different modeling scopes.
- Design of the mediators, the agents and the executives for the “ferry”. The user coasts are more or less defined, but what about the opposite side and the boat itself?

Next level is the Federation one. Its functions are: resolving the problem of data, application/agent semantic intersection, query decomposition and rooting, result assembly. For now we plan to use the method and the model developed in Magdeburg Otto-von-Guericke University by the group of prof. G. Saake enhancing it with our Active Data Dictionary approach. In parallel we are analyzing the possibilities to apply

- Agent-based means for query rooting and semantic overlap resolution in federated media.

The local level is the most developed. We use Active Data Dictionary as IS control shell and Enhanced Metadata Repository as IS control engine. Basic Active Data Dictionary model is bi-directional – i.e. ADD Enhanced Metadata Repository maintains data model modifications and appropriate changes are applied both to data and the program code. Open problem for the local level is related to the one we have with functional UVII objects:

- We are looking for a model and a tool of creating/generating the agents for execution local IS clients in an internet browser environment.