

E-learning data warehouse maintenance system for collaborative learning availability resources optimization

Jalel Akaichi
University of Tunis, Tunisia

ABSTRACT

An E-learning Data Warehouse (EDW) is constituted of information collected from heterogeneous, distributed, and autonomous E-learning Information Sources (EISs). EDW, which is fed by view definitions' built upon EISs, is frequently interrogated by e-learning actors such as educators, learners and decision makers for many reasons going from consultation to analysis. The above view definitions, which represent considerable informational resources, can become undefined when EISs change their schemas. This obviously decreases e-learning resources availability and consequently affects analysis efficiency.

In this paper, we propose to study the issues of using agent's based architecture to achieve e-learning data warehouse maintenance under schema changes by repairing automatically affected view definitions. This implicitly optimizes e-learning resources availability by automatically finding replacements for affected components belonging to view definitions.

Key-Words: *Data Warehouse, Maintenance, Collaborative e-learning, Schema changes, Agents.*

INTRODUCTION

The e-learning data warehouse is built by gathering shared data from e-learning information sources and integrating them into one personalized deposit according to e-learning users needs in order to have persistent information about the e-learning system, to avoid networks traffic saturation and resources unavailability. One of the significant tasks of an e-learning Data Warehouse Maintenance System (e-DWMS) is to update the materialized views during the changes of e-learning information sources data. Moreover, beyond the updates of data, we note that the changes of schema are also rather frequent in the distributed applications such as in the e-learning domain (Breslin, et al 2006; Saliah-Hassane, 2006; Sheehy, et al 2006). A change of schema could occur for many reasons and at any time during participating e-learning information sources life cycle. In fact, the e-learning information sources in such environments can change continuously not only contents but also their schemas which may render view definitions built among e-learning data warehouse undefined, and consequently decrease significantly e-learning resources unavailability.

Our solution has the goal to preserve the maximum number of view definitions instead of being completely bewildered with each information source schema change, while allowing implicitly view materialization by proposing a prototype solution to automate view definitions rewriting thanks to Meta knowledge about information space formed by information sources, to Meta knowledge about user space constituted by evolving view definitions, and view synchronization algorithms (Rundensteiner, et al 1997; Zhang, et al 2001).

Adding to that, the increase in user demand of a quick reach of distributed information large volume, the more autonomy requirements, and the need to avoid network saturation and to minimize communication costs have led to the adoption of the new rising techniques such as

agents, resulting from research in distributed artificial intelligence, to solve several problems such as e-learning data warehouse maintenance by decreasing the maintenance process time, and increasing e-learning data warehouse maintenance system components availability .

The remainder of this paper is organized as follows:

- In section 2, we detail the problem and foundations of view definition maintenance under schemas changes. We also illustrate the maintenance process by some running examples showing how e-learning resources availability may be improved.
- In section 3, we describe our solution based on static agents. We also explain agents' roles and interaction leading to view definitions survival and implicitly to resources availability enhancement.
- In section 4, we improve the solution by integrating agent mobility model which takes into account the system components partition.
- In section 5, and before to conclude, we describe and discuss the implementation.

VIEW DEFINITION MAINTENANCE UNDER SCHEMA CHANGES

In order to increase e-learning information resources availability, we propose a solution to solve the problem of view inflexibility. This solution has the goal to preserve the maximum number of affected view definitions by the occurrence of e-learning information sources schema changes. This implicitly allows the view definition evolving which is, in former work, carried out by the developers. We assume that information sources are integrated in the e-learning system via a wrapper which translates their structures into a relational common model. They are supposed to be heterogeneous and autonomous which join, or change dynamically their capabilities such as their schema.

E-DWMS includes two basic modelling tools: a model that permits to user the expression of view definition evolution via an extended SQL called Evolvable SQL (E-SQL) and a model for the description of the information sources (MISD) (Rundensteiner, et al 1997) and the relationships between them. This model of e-learning information sources description can be exploited for seeking a suitable substitution for the affected view definition components (attributes, relations, and conditions). The View Knowledge Base (VKB), which contains views definition expressed with preferences by users and described by E-SQL and the Meta Knowledge Base (MKB) revealed by MISD, represent the base for any operation of view rewriting or view synchronization process.

The Meta Knowledge Base (MKB)

The e-DWMS constitutes an intermediary between the e-learning user space called E-learning Data Warehouse and the information space including the participating e-learning data sources. When an e-learning information source joins the structure of e-DWMS, it provides its structure, its data model and eventually it's content. This information, also called descriptive knowledge, are stored into the MKB with respect to the MISD, or any other knowledge representation models such graphs.

As well, the relationships between e-learning information sources, also called substitution rules, have to be added by the e-DWMS administrator and/or generated automatically, then inserted into the MKB. This information constitutes the key platform for finding affected view definitions components substitutions.

The e-learning system knowledge is stored into the MKB and constitutes two main classes:

- The descriptive knowledge class composed by a set of e-learning information sources schemas as given in Table 1.
- The substitution knowledge class describing relations between e-learning information sources and composed by a set of rules or constraints such as Join Constraints as given in Table 2, Type Integrity Constraints as given in Figure 1, Figure 2, and Figure 3 and Partial/Complete Constraints as given in Figure 4.

In the following, we give an example (Table 1) of a collaborative e-learning system composed by three institutes collaborating to provide common e-learning resources for e-learning large community. Each institute forms an E-learning Information Sources (EISi) having its proper schemas and contents. This example will be used in the rest of this paper.

Table 1: *E-learning Information Sources*

EIS ID	Relation provided
EIS 1	Course (Course-Id, Name, Duration)
	Professor(Prof-Id, Name, Phone-Number, Address, Title, Diplomas, E-mail)
	Student (Stud-Id, Name, Level, Nationality)
	Course-Ref (Course-Id, Ref-Id)
	Stud-Course (Stud-Id, Course-Id)
	Teach-Course(Prof-Id, Course-Id)
	Reference (Ref-Id, Authors, Availability, Subject)
EIS 2	Course (Course-Id, Name, Duration)
	Professor(Prof-Id, Name, Phone-Number, Address, Title, E-mail)
	Student (Stud-Id, Branch-Id, Name, Level, Nationality)
	Stud-Course (Stud-Id, Course-Id)
	Teach-Course(Prof-Id, Course-Id)
	Branch (Branch-Id, Name, Duration)
	Course-Med (Med-Id, Course-Id)
	Media (Med-Id, Type, Marque)
	Publication (Pub-Id, Publication-Date, Subject, Volume)
Prof-Pub (Pub-Id, Prof-Id)	
EIS 3	Course (Course-Id, Name, Duration)
	Professor(Prof-Id, Name, Phone-Number, Address, Title, E-mail)
	Stud-Course (Stud-Id, Course-Id)
	Teach-Course(Prof-Id, Course-Id)
	Student (Stud-Id, Branch-Id, Name, Level, Nationality)
	Media (Med-Id, Type)
	Course-Med (Med-Id, Course-Id)

Table 2: Join Constraints

JC	Join Constraints
JC1	EIS1.Course.Name=EIS2.Course.Name
JC2	EIS1.Course.Name=EIS3.Course.Name
JC3	EIS1.Professor.Title=EIS2.Professor.Title
JC4	EIS1.Professor.Title= EIS3.Professor.Title
JC5	EIS2.Media.Type=EIS3.Media.Type
JC6	EIS1.Professor.Prof-Id= EIS2.Professor.Prof-Id

TCEIS1.Professor (Prof-Id, Name, Phone-Number, Address, Title, Diplomas, E-mail) = Professor (Prof-Id, Name, Phone-Number, Address, Title, Diplomas, E-mail) \subseteq Prof-Id (Number) \times Name (String) \times Phone-Number (Number) \times Address (String) \times Title (String) \times Diplomas (String) \times E-mail (String)

TCEIS1.Course (Course-Id, Name, Duration) = Course (Course-Id, Name, Duration) \subseteq Course-Id (Number) \times Name (String) \times Duration (Number)

TCEIS1.Student (Stud-Id, Name, Level, Nationality) = Student (Stud-Id, Name, Level, Nationality) \subseteq Stud-Id (Number) \times Name (String) \times Level (String) \times Nationality (String)

TCEIS1.Course-Ref (Course-Id, Ref-Id) = Course-Ref (Course-Id, Ref-Id) \subseteq Course-Id (Number) \times Ref-Id (Number)

TCEIS1.Stud-Course (Stud-Id, Course-Id) = Stud-Course (Stud-Id, Course-Id) \subseteq Stud-Id (String) \times Course-Id (String)

TCEIS1.Teach-Course (Prof-Id, Course-Id) = Teach-Course (Prof-Id, Course-Id) \subseteq Prof-Id (Number) \times Course-Id (Number)

TCEIS1.Reference (Ref-Id, Authors, Availability, Subject) = Reference (Ref-Id, Authors, Availability, Subject) \subseteq Ref-Id (Number) \times Authors (String) \times Availability (String) \times Subject (String)

Figure 1: EIS1 Type Integrity Constraints

TCEIS2.Professor (Prof-Id, Name, Phone-Number, Address, Title, E-mail) = Professor (Prof-Id, Name, Phone-Number, Address, Title, Diplomas, E-mail) \subseteq Prof-Id (Number) \times Name (String) \times Phone-Number (Number) \times Address (String) \times Title (String) \times E-mail (String)
 TCEIS2.Course (Course-Id, Name, Duration) = Course (Course-Id, Name, Duration) \subseteq Course-Id (Number) \times Name (String) \times Duration (Number)
 TCEIS2. Student (Stud-Id, Name, Branch-Id, Level, Nationality) = Student (Stud-Id, Name, Level, Nationality) \subseteq Stud-Id (Number) \times Name (String) \times Level (String) \times Branch-Id (Number) \times Nationality (String)
 TCEIS2. Stud-Course (Stud-Id, Course-Id) = Stud-Course (Stud-Id, Course-Id) \subseteq Stud-Id (String) \times Course-Id (String)
 TCEIS2.Branch (Branch-Id, Name, Duration) = Branch (Branch-Id, Name, Duration) \subseteq Branch-Id (Number) \times Name (String) \times Duration (Number)
 TCEIS2.Teach-Course (Prof-Id, Course-Id) = Teach-Course (Prof-Id, Course-Id) \subseteq Prof-Id (Number) \times Course-Id (Number)
 TCEIS2.Publication (Pub-Id, Publication-Date, Subject, Volume) = Publication (Pub-Id, Publication-Date, Subject, Volume) \subseteq Pub-Id (Number) \times Publication-Date (Date) \times Subject (String) \times Volume (String)
 TCEIS2. Media (Med-Id, Type, Marque) = Media (Med-Id, Type, Marque) \subseteq Med-Id (Number) \times Type (String) \times Marque (String)
 TCEIS2. Course-Med (Med-Id, Course-Id) = Course-Med (Med-Id, Course-Id) \subseteq Med-Id (Number) \times Course-Id (Number)
 TCEIS2.Prof-Pub (Pub-Id, Prof-Id) = Prof-Pub (Pub-Id, Prof-Id) \subseteq Pub-Id (Number) \times Prof-Id (Number)

Figure 2: EIS2 Type Integrity Constraints

TCEIS3.Professor (Prof-Id, Name, Phone-Number, Address, Title, E-mail) = Professor (Prof-Id, Name, Phone-Number, Address, Title, Diplomas, E-mail) \subseteq Prof-Id (Number) \times Name (String) \times Phone-Number (Number) \times Address (String) \times Title (String) \times E-mail (String)
 TCEIS3.Course (Course-Id, Name, Duration) = Course (Course-Id, Name, Duration) \subseteq Course-Id (Number) \times Name (String) \times Duration (Number)
 TCEIS3. Student (Stud-Id, Name, Branch-Id, Level, Nationality) = Student (Stud-Id, Name, Level, Nationality) \subseteq Stud-Id (Number) \times Name (String) \times Level (String) \times Branch-Id (Number) \times Nationality (String)
 TCEIS3. Stud-Course (Stud-Id, Course-Id) = Stud-Course (Stud-Id, Course-Id) \subseteq Stud-Id (String) \times Course-Id (String)
 TCEIS3. Media (Med-Id, Type, Marque) = Media (Med-Id, Type, Marque) \subseteq Med-Id (Number) \times Type (String) \times Marque (String)
 TCEIS3. Course-Med (Med-Id, Course-Id) = Course-Med (Med-Id, Course-Id) \subseteq Med-Id (Number) \times Course-Id (Number)

Figure 3: EIS3 Type Integrity Constraints

PC EIS1.Professor,EIS2.Professor = $(\pi \text{Prof-Id, Name, Phone-Number, Address, Title, E-mail (EIS1.Professor)} \subseteq \pi \text{Prof-Id, Name, Phone-Number, Address, Title, E-mail (EIS2.Professor)})$
PC EIS3.Media, EIS2.Media = $(\pi \text{Med-Id, Type (EIS3.Media)} \subseteq \pi \text{Med-Id, Type (EIS2.Media)})$
PCEIS1.Student,EIS2.Student= $(\pi \text{Stud-Id, Name, Level, Nationality(EIS1.Student)} \supseteq \pi \text{Stud-Id, Name, Level, Nationality(EIS2.Student)})$

Figure 4: Partial/Complete Information Constraints

The View Knowledge Base (VKB)

E-SQL language allows including user preferences into SQL view definition. E-SQL is an extension of SELECT-FROM-WHERE SQL enriched by specifications defined by the developer in charge of the view definitions in order to indicate how those latter can evolve after schema changes. This latter can take the following values: attributes, relations or e-learning information sources deletion; attributes, relations or information sources addition; attributes, relations or information sources renaming. The E-SQL defined views are then stored into a structure called View Knowledge Base and respect the following syntax:

```
CREATE VIEW V [(Local_Column_List)][(VE =['⊇' | '⊆' | '≡' | '≈' ] AS
SELECT Attribute_Name [(AD = [true | false], AR = [true | false])]
[, Attribute_Name [(AD = [true |false], AR = [true | false])] .. . ]
FROM Relation_Name [(RD = [true | false], RR = [true | false])]
[,Relation_Name [(RD = [true | false], RR = [true | false])] . . . ]
WHERE Clause_Primitive [(CD = [true | false], CR = [true | false])]
[, Primitive_Clause [(CD = [true | false], CR = [true | false])] . . . ]
```

As indicated in the E-SQL view, each view definition component (Attribute, Relation, or Condition) has two evolution parameters which are detailed in Table 3.

Table 3: E-SQL View Evolution Parameters

Evolution Parameter	Semantics	Default Value
Attribute Dispensable (AD)	true : dispensable	false
Attribute Dispensable (AD)	false : indispensable	
Attribute Replaceable (AR)	true : replaceable	false
Attribute Replaceable (AR)	false : non replaceable	
Condition Dispensable (CD)	true : dispensable	false
Condition Dispensable (CD)	false : indispensable	
Condition Replaceable (CR)	true : replaceable	false
Condition Replaceable (CR)	false: non replaceable	
Relation Dispensable (RD)	true : dispensable	false
Relation Dispensable (RD)	false : indispensable	
Relation Replaceable (RR)	true : replaceable	false
Relation Replaceable (RR)	false : non replaceable	

It should be noted that the replacement view extent can be equivalent, a superset or a subset of the initial view extent. To represent this characteristic, a parameter called VE (View Extend) is used in each E-SQL view definition. VE Values are presented in Table 5.

Table 4: E-SQL View Extent Parameters

View extends parameters	Semantics
\equiv	V' is equal to V
\supseteq	V' is a super set of V
\subseteq	V' is a subset of V
\approx	indifference

To illustrate the use of E-SQL view definitions for querying e-learning information sources in order to gather information for the e-learning Data Warehouse, we present some examples showing the impact of schema changes on resources unavailability and the solution to this problem performed by view synchronization algorithms, thanks to knowledge stored in the MKB and VKB.

Example 1

Suppose that some users need to have names and emails of professors belonging to EIS1 in order to recruit some of them, or to organize with their help, a kind of a seminar. The following view definition permits to query a common EDW including information gathered from collaborating e-learning information sources. It states that users want the list of assistant professors whatever they teach.

```
SQL view1:
CREATE VIEW V AS
SELECT P.Name, P.E-mail
FROM EIS1.Professor P
WHERE P.Title = "Assistant Professor";
```

Knowing that collaborating e-learning information sources are autonomous, EIS1 administrator may delete "Title" and/or the "E-mail" attributes from "Professor" relation, for a design or security reasons. Obviously, SQL view1 becomes undefined, couldn't work anymore, and decrease resources availability.

The solution increasing resources availability comes from view synchronization process which have the role to find replacements for disappeared components, and rewrite view definition thanks to Knowledge stored in the MKB and preferences embedded in view definitions based E-SQL (see E-SQL view 1).

The semantic of the following example, states that users may accept the replacements of EIS1 assistant professors by assistant professors, either from EIS2.Professor, or EIS3.Professor, if EIS1.Professor is deleted. They would also accept professors E-mail from EIS2.Professor, if the E-mail attribute is deleted.

E-SQL view1:

```
CREATE VIEW V1 VE = '⊇'AS
SELECT P.Name, P.E-mail (AD=false, AR=true)
FROM EIS1.Professor P(RD = false, RR=true)
WHERE (P.Title = "Assistant Professor") (CD = false, CR = true);
```

Example 2

The following query (E-SQL view 2) permits to obtain list of students learning the course "algorithmic", from the relation EIS1.Student. However they would accept information belonging to EIS2.Student in the case of unavailability information from the first source.

E-SQL view 2:

```
CREATE VIEW V2 VE = '⊆'AS
SELECT S.Name (AD=false, AR=true), S.Stud-Id (AD=false, AR=true)
FROM EIS1.Student S(RD = false, RR=true), EIS1.Stud-Course SC(RD = false,
RR=true),
EIS1.Course C (RD = false, RR=true)
WHERE (SC.Stud-Id = S.Stud-Id) (CD = false, CR = true) and (C.Course-Id =
SC.Course-Id)
(CD = false, CR =true) and (C.Name = " Algorithmic") (CD = false, CR = true);
```

Example 3

The following query permits to get the list of courses taught using a media type ="X" from EIS2.Media, and may accept to have this information from EIS3.Media in the case of EIS2.Media deletion.

E-SQL view3:

```
CREATE VIEW V3 VE = '⊆'AS
SELECT C.Name (AD=false, AR=true)
FROM EIS1.Course C (RD = false, RR=true), EIS1.Media M(RD = false, RR=true),
EIS1.Course-Med CM (RD =false, RR=true)
WHERE (C.Course-Id = CM.Course-id) (CD = false, CR = true) and (CM.Med-id =
M.Med-Id)
(CD = false, CR = true) and ( M.Name=" X") (CD = false, CR = true);
```

Example 4

The following query permits the get the list of shared courses between e-learning information sources EIS1, EIS2, and EIS3. It can give only common courses between EIS1 and EIS2 in the case of EIS3 deletion.

E-SQL view 4:

```
CREATE VIEW V4 VE = '≈'AS
SELECT C1.Name (AD=false, AR=true)
FROM EIS1.Course C1(RD = false, RR=false), EIS2.Course C2(RD = false, RR=false),
EIS3.Course C3(RD = false, RR=true)
WHERE (C1.Name = C2.Name) (CD = false, CR = false) and (C1.Name =C3.Name)
(CD = true, CR = true)
```


We mention that, due to the integration of the e-learning resources into the EDW, the above view definitions give results independently on networks traffics, faults or costs. Moreover, affected view definitions by schema changes may survive, thanks to repairing process which takes into account user preferences. These too facts contribute considerably into the enhancement of e-learning resources availability.

Note that e-learning information sources participating to such systems may increase and consequently the possibilities of resources substitutions may also increase. This means that the collaboration between e-learning users' community and availability factors may increase in an e-learning dynamic environment where information sources schemas evolve frequently.

2.3 View synchronization

The view synchronization [2] consists in determining legal rewritings for the affected views, referring to the rules or constraints embodied into the MKB. These rules enable substitutions retrieval for the affected view definition components while respecting preference parameters described into the VKB.

The view rewriting is legal when it is compatible with the current information space. This rewriting have to preserve the information presented by the initial view definition according to preferences parameters associated to the view definition components and the possibilities of substitutions offered by the MISD.

Example 5

In the case of E-mail attribute deletion from EIS1.Professor relation, using partial/complete constraints, view synchronization algorithms may replace the affected attribute by EIS2 E-mail attribute. This is due to the fact that the two attributes have a same type and EIS2.Professor relation is a subset of EIS1.Professor relation. Consequently, the E-SQL view 1 becomes:

E-SQL view1':

```
CREATE VIEW V1' VE = '∩'AS
SELECT P.Name (AD=false, AR=true), P2.E-mail (AD=false, AR=true)
FROM EIS1.Professor P (RD = false, RR=true), EIS2.Professor P2 (RD = false, RR=true)
WHERE (P.Title = "Assistant Professor") (CD = false, CR = true) and (P.Prof-Id=P2.Prof-Id);
```

In the case of EIS1.Professor relation deletion, synchronization algorithms may replace EIS1.Professor relation by EIS2.Professor, thanks to partial/complete constraints established between them.

E-SQL view1'":

```
CREATE VIEW V1'' VE = '∩'AS
SELECT P2.Name, P2.E-mail (AD=false, AR=true)
FROM EIS2.Professor P2 (RD = false, RR=true)
WHERE (P2.Title = "Assistant Professor") (CD = false, CR = true);
```

AGENT-E-DWMS

Previous works (Lee, et al 1997; Rundensteiner, et al 1997), related to Data Warehouse maintenance under schema changes, adopt a centralized framework for maintaining view definitions under schemas changes, which is not suitable for distributed and dynamic environments. In fact, e-DWMS was implemented as a compact module including administration, detection and synchronization. For these reasons, our system evolved to integrate new design, functionalities and technologies. In spite of centralized framework, Agent-e-DWMS framework is distributed using static agents to solve the problem of view synchronization in distributed collaborative learning environments. In Agent-e-DWMS, a static agent is an autonomous entity that pursuits its own goals and objectives, and resides only in the system where it starts execution. If it needs information from another system or needs to interact with agents in another system, it uses a communication mechanism such as messaging or remote procedure calling.

A figure schematizing Agent-e-DWMS framework related to our collaborative learning example is presented in the following figure (figure 9).

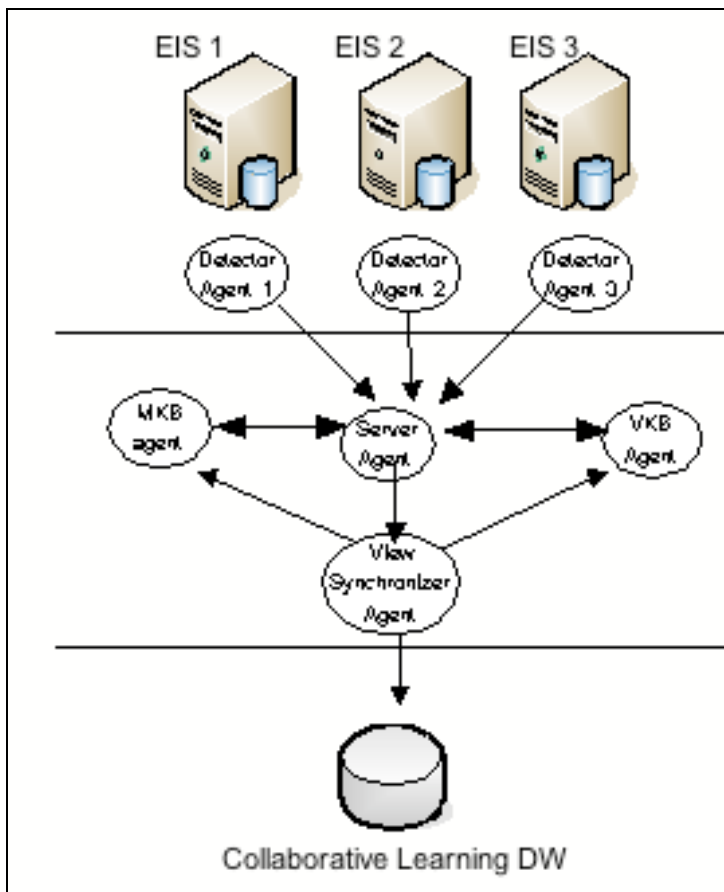


Figure 5: Agent-e-DWMS framework

Agent-e-DWNS architecture is distributed on five entities: the Server Agent, the Detector Agent, the MKB Agent, the VKB Agent and the View Synchronizer Agent, as shown in figure 9. Communication between agents [4] can be ensured either by message sending or by agent migration. In our model communication will be guaranteed by the traditional message sending. In fact, all the e-learning agents of the model know each other directly via their identifier, names and sites. Thus, any agent of the system can communicate directly with any other agent. The different exchanges messages between agents needed to resolve the problem of views synchronization in the collaborative learning system are shown in Figure 6.

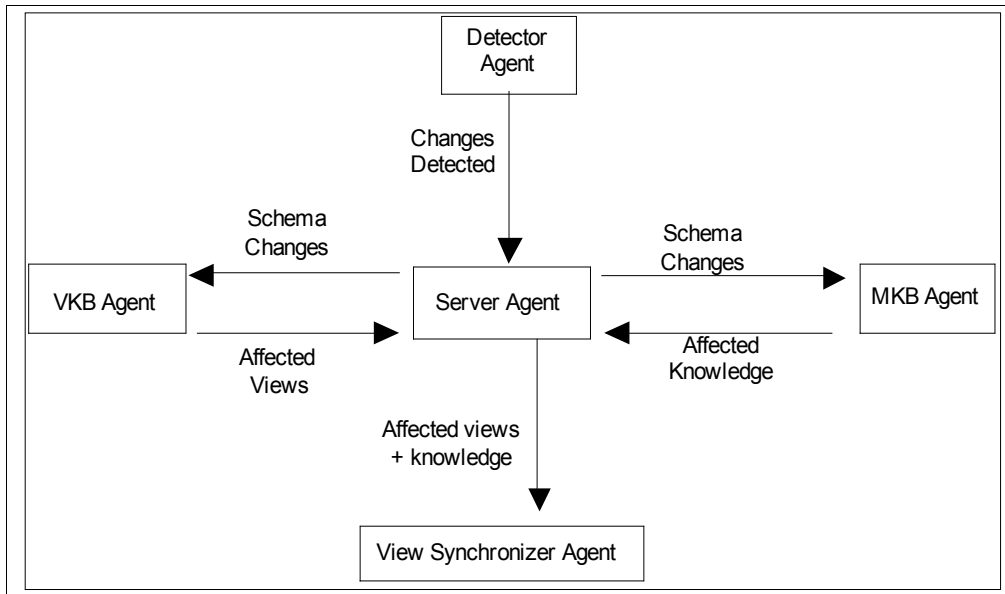


Figure 6: Agents' communication

Dispatching e-DWMS Tasks

The Server Agent is the heart of the e-DWMS. It has the role of initializing the system and dispatching e-DWMS tasks on Detector, MKB, VKB and View Synchronizer Agents. In fact, the Server Agent supervises the correct functioning of all the other agents' instances deciding on their creation, suspension and ending allowing coordination and synchronization between them. In other words, it plays the role of the manager of all the Agent-e-DWMS. When the collaborative learning system is triggered, the Server Agent starts by creating the various agents of the model.

Detecting e-DWMS affected EIS schemas

The Detector Agent is an agent implemented into each e-learning information source. It is responsible of the detection of changes which have occurred on the level of the structures of the participating e-learning information source to the system. Indeed, it traverses an e-learning information source with an aim to detect a change by comparing the schema of the source at moment t and at moment $t-1$. Its mission consists on, transmitting any schema change occurred in the e-learning information source, to the Server Agent.

Affected e-learning Rules Determination

The MKB Agent processes the data received from the Server Agent. This latter transmits to it any schema changes occurred into any e-learning information sources. After that, the MKB Agent analyses the Meta Knowledge Base in order to detect the whole unit of affected knowledge or rules by the schema changes received to send them to the Sever Agent.

Affected e-learning Query Computing

The VKB Agent has the role of detecting the subset of views definitions affected by occurred schema changes. In fact, following the changes received, VKB Agent checks within the VKB to determine the set of views definitions which contain one or more components affected by the changes. After that, the VKB Agent transmits the result, composed by the affected views definitions, to the Sever Agent.

Affected e-learning Query Rewriting

After the reception of the affected rules, the affected knowledge and the affected views definitions from the Server Agent, the View Synchronizer Agent starts to check if it is possible to determine a legal rewritings for the affected views in order to create new views definitions compatibles with the current state of the information space. For that, it refers to the e-learning users preferences expressed into the E-SQL affected views received. When the synchronization process is well done, the View Synchronizer Agent transmits its results to the MKB Agent and the VKB Agent in order to update respectively the MKB e-learning rules and the VKB views definitions, according to the new e-learning information space state.

MOBILE E-DWMS

Mobile-e-DWMS evolves Agent-e-DWMS framework to become more adapted to dynamic environments. These environments are characterized by a great number of information sources. Each one has a huge mass of information that can not be implemented in a single machine, but in a local or wide network. In fact, an e-learning information source can be distributed in many machines forming a network. It is the same thing for the VKB and the MKB. Even the collaborative learning data warehouse can be divided into many data marts according to repartition criteria. So, Agent-e-DWMS detector agent, MKB agent, VKB agent must be replaced by mobile ones to become able to move from one machine to another of respectively an e-learning information source, an MKB, and a VKB networks. Concerning the View Synchronizer Agent and the Server Agent, they don't need the mobility characteristic because they don't move from one machine to another of a network to accomplish their tasks. Mobile-e-DWMS keeps the same agents' communication (Figure 6). The following figure (Figure 7) represents a collaborative learning system using the Mobile-e-DWMS framework.

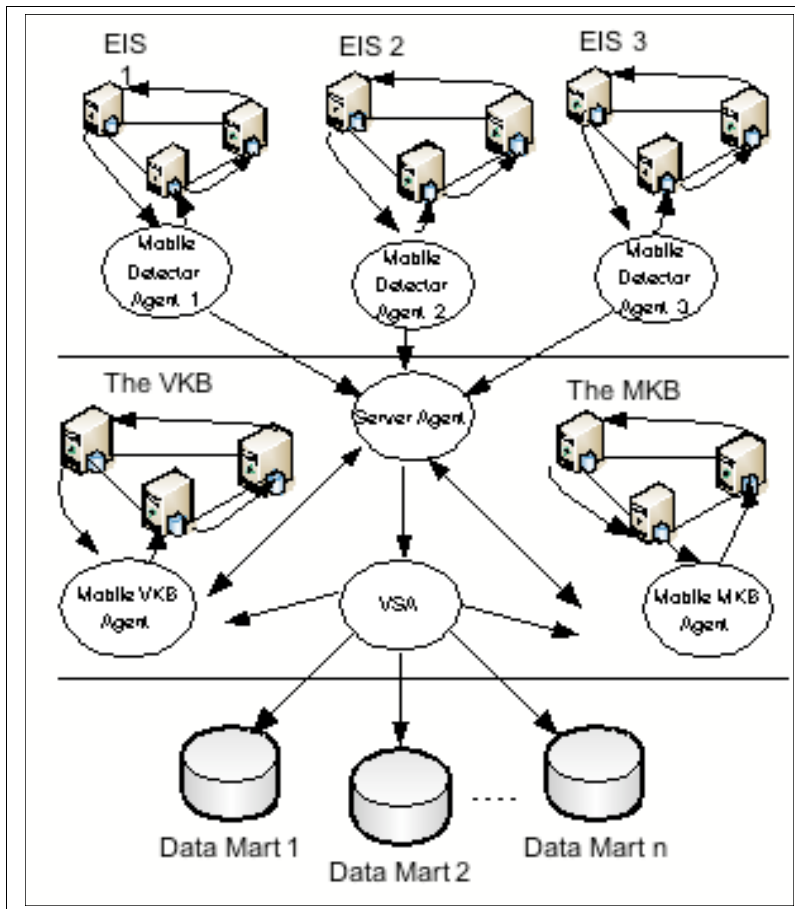


Figure 7: Mobile e-DWMS framework

By adding the mobility paradigm, e-DWMS agents evolve towards new dimensions. In fact:

- As we have seen, an e-learning information source can be distributed in a local or wide network. For that reason, the Detector Agent should become a mobile one able to move from one machine to another of an e-learning information source network to find all schema changes occurred.
- The Meta Knowledge Base can be distributed in a local or wide network. To compute affected knowledge, the MKB agent is replaced by a mobile one capable to look into all MKB components residing in different machines, compute affected rules and knowledge and transmit it to the Server Agent.
- The View Knowledge Base can be distributed in a local or wide network. To compute affected views, the VKB agent becomes a mobile one able to look into all VKB components distributed in different machines, compute affected views and transmit it to the Server Agent.
- The role of the view synchroniser agent states the same as in the Agent-e-DWMS which is to find legal rewritings for affected views using the MKB information and E-SQL views preferences and to transmit it to the Mobile VKB agent and the Mobile MKB agent in order to update respectively the MKB and the VKB.

IMPLEMENTATION AND EVALUATION

To demonstrate the feasibility and the advantages of our mobile e-DWMS, we developed a prototype based on IBM aglets. Agents transformed to aglets become serializable Java objects able to move through the network. When they arrive to destination, they are reloaded as classes and executed. The performed application was implemented and simulated on eleven personal computers linked through a local area network.

The computers were assigned as follows:

- One computer was dedicated to View Synchronization Agent which is responsible for view repairing according to schema changes happened in EISs.
- Two computers were assigned to the MKB and the MKB Agent which migrate from one machine to another in order to determine affected knowledge.
- Two computers were assigned to the VKB and the VKB agent which migrate from one machine to another in order to determine affected view definitions. The separation between the VKB and the MKB permits parallel determination of affected knowledge and view definitions.
- Six computers were dedicated to information sources and the associated Detector Agents. Each information source was partitioned on two computers. This permits parallel schema changes detection by allowing mobile Detector Agent migration and running over distributed EISs' components.

Compared to centralized e-DWMS, mobile e-DWMS solution permits to gain 29 percent of time on each synchronization operation related to one schema change. The estimated time gained for n schema changes is around $0.29*n$ units of time.

CONCLUSION

An e-learning Data Warehouse is constituted by e-learning informational resources collected from heterogeneous, distributed, and autonomous collaborating information sources. It has to be regularly updated and maintained, when e-learning information sources change their contents and/or their schemas, in order to ensure integrity, accessibility, availability and the consistency of the afforded information.

In this paper, as our knowledge, we are the first to exploit data warehousing maintenance techniques to optimize e-learning informational resources availability and sharing. This was ensured essentially by view synchronization concepts which make view definitions, embedding e-learning resources, survive.

We also, studied the issues of using static then mobile agents to achieve e-learning data warehouse maintenance under schema changes. We focus on agents' collaboration leading not only to the availability optimization of e-learning informational resources, but also to the increasing of the system components autonomy, avoiding network saturation and to the decreasing of a variety of costs such as communication and execution times.

REFERENCES

- Breslin, C. Nicol, D., Grierson, H. and Wodehouse, A. 2006. Embedding an Integrated Learning Environment and Digital Repository: Lessons Learned. *CSFIC '06 Workshop*, Alicante, Spain, September.
- Chaib-draa, B., Jarras, I. and Moulin, B. 2001. Systèmes multiagents : Principes généraux et applications. *Agent et systèmes multiagents*. Hermès.
- Lee, A. J., Nica, A. and Rundensteiner, E. A. 1997. Keeping Virtual Information Resources Up and Running. In *Proceedings of IBM Centre for Advanced Studies Conference CASCON 97*, Best paper Award, November.
- Rundensteiner, E. A., Lee, A. J. and Nica, A. 1997. *The EVE Framework: View Evolution in an Evolving Environment. Technical Report WPICS-TR-97-4*, Worcester Polytechnic Institute, Dept. of Computer Science.
- Saliah-Hassane, H. 2006. Building a Repository for Online Laboratory Learning Scenarios. *36th ASEE/IEEE Frontiers in Education Conference*, San Diego, CA, October.
- Sheehy, P., Marcus, G., Costa, F. and Taylor, R. 2006. Implementing e-learning across a faculty: Factors that encourage uptake. In *Proceedings of the 23rd ASCILITE conference, Sydney*.
- Zhang, X. Rundensteiner, E. A. and Ding, L. 2001. PVM: Parallel View Maintenance Under Concurrent Data Updates of Distributed Sources. In *Data Warehousing and Knowledge Discovery, Proceedings*, Munich, Germany, September.

Copyright for articles published in this journal is retained by the authors, with first publication rights granted to the journal. By virtue of their appearance in this open access journal, articles are free to use, with proper attribution, in educational and other non-commercial settings.

Original article at: <http://ijedict.dec.uwi.edu/viewarticle.php?id=177&layout=html>