Grid Database - Management, OGSA and Integration

Florentina Ramona PAVEL (EL BAABOUA)
Academy of Economic Studies ROMANIA, Bucharest
pav_florentina@yahoo.com

The problem description of data models and types of databases has generated and gives rise to extensive controversy generated by their complexity, the many factors involved in the actual process of implementation. Grids encourage and promote the publication, sharing and integration of scientific data, distributed across Virtual Organizations. Scientists and researchers work on huge, complex and growing datasets. The complexity of data management within a grid environment comes from the distribution, heterogeneity and number of data sources.

Early Grid applications focused principally on the storage, replication and movement of file-based data.. Many Grid applications already use databases for managing metadata, but increasingly many are associated with large databases of domain-specific information. In this paper we will talk about the fundamental concepts related to grid-database access, management, OGSA and integration.

Keywords: *grid, data grid, grid computing, database, management, integration.*

1 Introduction

A data grid is a grid computing system that deals with the controlled sharing and management of large amounts of distributed data. Data grids should provide a low level framework for data management activities.

The size and number of data collections has been growing rapidly (petabytes), the costs of computation and data storage decrease and performances increase.

Data grids allow to store, manage and share large data collections, huge amount of files, geographically distributed databases across virtual organizations.

Data management represents the real challenge for the next generation petascale grid environments. In the last few years, there was an increasing interest in fine grained (database related) grid data management activities and services connected with database access, metadata management, data integration, data transformation, data flow. Grid Services for database access and integration play a strategic role and provide added value to a production grid environment since they allow to aggregate data, join datasets stored at different sites, infer new knowledge by analyzing structured and distributed data, manage monitoring and accounting information.[4]

Research and development activities relating to the Grid have generally focused on applications where data is stored in files.

Grid Services for database access and integration play a strategic role and provide added value to a production grid environment since they allow to aggregate data, join datasets stored at different sites, infer new knowledge by analyzing structured and distributed data, manage monitoring and accounting information.

2 Grid Computing and Common Benefits

Grid computing is a term that has been applied to various architectures designed to deliver the benefits of an IT grid. It is an approach to computing that detaches the software functionality from the specifics of hardware deployment by blending system and storage resources into a continuum of resources that can be allocated to, and deallocated from a particular function or functional locus, in this case, a database.

It enables administrators to assign computing tasks to computing resources, and it assigns data to storage resources in a way that enables such resources to be easily added or removed or tasks and data to be moved as needed.

In the case of database workloads, grid computing contrasts with the classic model that involves dedicated servers associated with dedicated storage in that the servers and storage are fluid. They can be assigned, added, and reassigned as necessary without upsetting the overall topology of the database server environment.

The key benefits of grid computing come in the form of resource flexibility, scalability, and optimization of operations through parallel processing. These benefits are expressed through an architecture that gives users the following capabilities:

- To avoid unnecessary hardware, power, and staffing costs of overprovisioning IT systems, commonly done to avoid capacity upgrades.
- When capacity upgrades are necessary, to scale incrementally by adding (or in some cases, redeploying) system and storage resources without expensive "forklift upgrades" or time-consuming and error-prone upgrade procedures.
- To ensure continuous availability through the provisioning of redundant resources, ensuring automatic failover when necessary.
- To increase transaction throughput through parallelization of tasks.

All these benefits combine to enable better business agility in responding to changes in load or business priorities.

3 Metadata – importance in Grid

The use of metadata in Grid applications tends to be quite simple it is mainly for mapping the logical names for datasets into the physical locations where they can be accessed.

Metadata will be important for many Grid applications, in the following activities:

- Management or scheduling through provision of system and administrative information.
- Data discovery or interpretation through provision of data structure and content information.
- Resource or access method selection, through indexes or summaries.
- Data selection or evaluation, to inform human judgements about the data. [13]

Almost all aspects of metadata can have components that are application specific. [13]

Many applications involve portals, workflows or bespoke code that first examines metadata according to user requirements and then uses these metadata to locate the data, describe which data are accessed, determine what transformations are necessary, to steer analyses and visualizations, and to carry forward information into automatically generated metadata associated with result sets.

As users and developers develop more sophisticated applications, more sophisticated metadata systems and tools will be required.

The use of metadata to locate data has important implications for integrating databases into the Grid because it promotes a two-step access to data . [13]

In step one, a search of metadata catalogues is used to locate the databases containing the data required by the application. Those data are then accessed in the second step. A consequence of two-step access is that the application writer does not know the specific resource that will be accessed in the second step. The application must be general enough to connect and interface to any of the possible resources returned in step one. [13]

Ideally the two-step approach requires that all resources should provide the same interfaces, but variation in facilities, interfaces and representations is inevitable. [13]

OGSA-DAI services provide metadata about the DBMS, e.g. whether it is an Oracle, DB2 or MySQL, DBMS (Database Management System), system that are being exposed to the Grid. Also metadata are provided about the capabilities of the DBMSs that are being exposed to the Grid through the service interfaces as well as any inherent capabilities of the services themselves. The connection technology that is employed to connect to the databases is also exposed for clients capable of using such information.

The metadata may be provided statically, that is when the service is configured, or dynamically, which may require additional coding. On the whole the static metadata model is extensible so that communities that employ OGSA-DAI to access databases within a Grid context can provide community-specific metadata for the databases they expose to the Grid.

4 The need for databases in Grid environments

Early Grid applications were often closely associated with devices or tools that read and/or generated flat files. Consequently, support for files rather than for the management of structured data had the highest profile in the early Grid toolkits.

However, over time, the file management systems and registries associated with Grid toolkits themselves became complex, and database management systems (DBMSs) were increasingly used to store Grid metadata. Contemporaneously, the requirements of the scientific computing community have become more sophisticated with, for example, biological and astronomical communities generating large quantities of data that increasingly use databases for storage and retrieval. Similarly, engineering, medical research, healthcare and many governmental systems can also take advantage of Grids that access and integrate multiple and distributed collections

of structured data. [13]

Researchers, initially led by Globus and IBM, began in 2001 developing new Grid standards and technology. The result was the Open Grid Services Architecture (OGSA). OGSA presents a picture of the Grid where Grid resources and services are represented by instances of Grid services.

Grid services, are stateful service instances supporting reliable and secure invocation, lifetime management, notification, policy management, credential management, and virtualization. The OGSA-DAI project is developing Grid services that represent data resources, where, by a data resource, we mean any physical or logical entity that is able to source or sink data. These underlying data sources and sinks, together with any associated management infrastructure, are referred to as physical data resources. The term data resource is then used to represent the aspects and capabilities that are exposed to the Grid.

If the OGSA is to support a wide range of communities, then database integration is vital. As the Grid becomes commercially important, database vendors will embed the middleware functionality directly into their products to provide support for OGSA Grid integration.

Similarly, it is vital that those designing standards for Grid middleware take into account the special requirements to easily integrate databases across a Grid. One of the motivations for OGSA-DAI is to expose and formulate such requirements. Together, these converging developments will reduce the amount of middleware required to integrate databases into the OGSA Grid.

OGSA-DAI has designed, developed and released a collection of services for integrating database access and description with the core capabilities of OGSA, this allowing structured data resources to be seamlessly integrated into OGSA Grid applications.

Relational database vendors support the integration of their products with Web Services from within SQL queries, the creation of Web Services from stored procedures, and the publication of Web Services based on database requests.

The European Data Grid has developed Spitfire, in a Grid settings, a Web Service interface to relational databases for metadata management. Spitfire has developed an infrastructure that allows a client to query a relational database over GSI-enabled HTTP(S). An XML-based protocol is used to represent the query, and its result. Provide a number of facilities for automating the management of data and its referential integrity.

OGSA-DAI currently only provides interfaces to relational and XML database management systems. [3]

OGSA-DAI allows developers to define their own activities and make them available to consumers. This feature has been exploited by a number of research groups and could, for example, be used to expose specialist functionality such as local data mining capability to database consumers. There is clearly also a relationship between OGSA-DAI and other data Grid functionalities. [3]

The prime goals of OGSA-DAI were:

- Provide controlled exposure of physical data resources to the Grid.
- Support access to heterogeneous physical data resources through a common interface style.
- Provide base services that allow higher-level data integration services to be constructed.
- Leverage emerging Grid infrastructure for security, management, accounting etc.
- Standardise data access interfaces through the GGF DAIS WG.
- Provide a reference implementation of the DAIS specification.

OGSA-DAI should be seen as one of a range of components that together support access, sharing, management and coordinated use of data on the Grid. [3]

5 Operations on a Data Resource

Managing the interaction between a data resource and the Grid involves defining the operations that may be performed on a physical data resource and the data requirements for these operations.

- Update operation, data must be delivered to the data source.
- Query operation, data may be transported away, via a delivery mechanism, from the data resource.

OGSA-DAI is not defining any new query languages; the GDS is acting as a conduit through which existing query languages may be directed to the physical data resource.

Figure 1 presents primary mode of operation employed by OGSA-DAI: a Grid service presents some view of a data resource, a query document is submitted to the Grid service, and is evaluated to produce a result document, usually returned to the client. The nature of the query document submitted to the Grid service and the subsequent result document depends on the type of the data resource that the Grid service is configured to represent. For example, a relational database may accept SQL queries. [3]

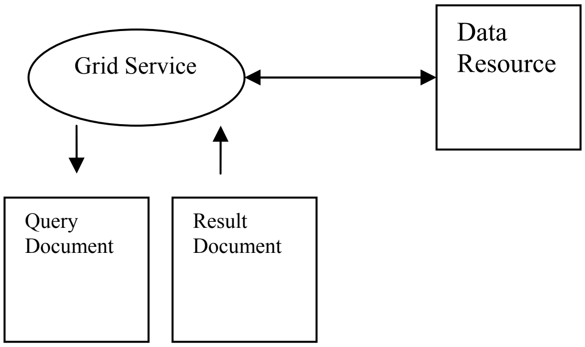

Fig.1. GDS mode of operation

If the data in question is transported somewhere else in the Grid then a GDSF may be used to represent the data at the destination point. Alternatively, the data may be represented in some other non-Grid-enabled storage system in which case it may be referenced using out-of-band techniques.
GDSF (Grid Data Service) is defined to represent the point of presence of a physical data resource on the Grid. It is through the GDSF service instances that a physical data resource's capabilities and meta-data are exposed.

Consider the case where a GDS is used to request data from a physical data resource:

- If the results are anticipated to be small then the client may request that the data is returned synchronously, i.e. in-lined in the response to the original query.
- Out-of-band delivery mechanisms might be used to transfer data resulting from a query. A new GDSF could then be created against the physical data resource to which results have been delivered - see Figure 2. [3]

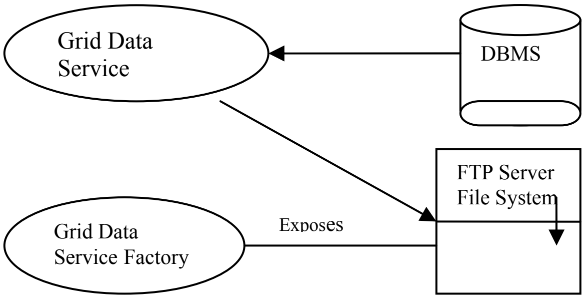

Fig.2. GDS delivery to file system.

- Delivery from one GDS to another may be used as a mechanism for transferring data. The results could then be served by a new GDS.

It is not the intention of OGSA-DAI to build delivery technology or indeed Grid services that represent the data that is being delivered. The interface to delivery and the specification of what is to be delivered across a particular interface is of interest to OGSA-DAI.

6 Grid-DBMS: Key Issues

A grid database management system should provide transparent, secure and efficient management of data sources in a grid environment. Since the beginning of the grid era many efforts were directed towards computational access and storage management. Grid database management was addressed starting from the year 2000 (EDG-Spitfire (Bell et al. 2002), GRelC (Aloisio et al. 2005), and OGSA-DAI (Antonioletti et al. 2005)). In the following we describe some basic elements connected with database management in distributed environments, highlighting how they impact on the application domains and why they are so relevant for end users.
Next subsections will be devoted to the discussion of data representation, data organization, data models, query languages, data access, data integration, access control and data flow. [4]

Data Representation

To be domain-independent, data grids must provide support (in terms of access and management) to every type of data format, structure and representation.

Data can be both structured and unstructured,

characterized by different formats, coding, precision, accuracy and semantics.

Some examples concern bioinformatics (i.e. textual files, relational data sources), astrophysics (i.e. relational DBMS with postGIS extensions), climate scientists (i.e. XML) data banks. [4]

Data Organization, Data Models and Query Languages

Data can be organized following several data models such as relational and hierarchical. Support in terms of relational or XML engines is widely provided by existing systems: Postgres, MySQL, IBM/DB2, as well as XIndice, eXist. Such DBMSs provide full support in terms of database access and management functionalities. Different data models adopt different query languages such as SQL (for the relational one) and XPath and XQuery (for the hierarchical one). Data grids must provide support to all of them. [4]

Data Access

Even if DBMSs provide a lot of functionalities for the management of data sources, they are not fully compatible with existing grid middleware. They can be accessed in grid by using a "grid-DBMS" interface. This grid interface must provide full support to all of the query languages (SQL, XQuery, XPath, etc.) concerning the

target data resources (transparency requirement with regard to the query language). The specific part of the grid-DBMS that makes a data resource accessible in grid (or "grid enabled") is called Grid Database Access Service (Grid DAS).

It must provide secure, transparent, robust and efficient access to heterogeneous and distributed databases exposing standard interfaces to enable interoperability with other grid components and/or services.

Several research projects exploit the service-in-themiddle or front-end approach to provide such kind of functionalities, that is, they focus on the development of a transparent, secure and robust grid interface to existing DBMSs. On the contrary, vendor-specific products (i.e. Oracle 11g) generally exploit an embedded approach providing within the product, software modules to run on a grid environment. [4]

Data Integration

While the Grid DAS is a basic service to expose databases in grid (it provides a first level of virtualization), the Grid Data Integration Service (Grid DIS) is a further necessary building block if we want to provide aggregation capabilities (second level of virtualization).

A Grid DIS can be centralized or distributed and in some cases it is integrated into the related Grid DAS providing what we call a Grid DAIS.

Data integration is strongly challenging since it allows both to integrate data within several application-level domains (bioinformatics, astrophysics, financial, etc.) and system-level distributed environments for monitoring and accounting purposes. [4]

Data Access Control

Data access control is more important to ensure that the confidentiality of the data is preserved/maintained against unauthorized accesses.

The facilities that the Grid provides to control access must be very flexible in terms of the combinations of restrictions, available policies, etc. User-centric and VO-centric (Virtual Organization), data access control allow managing policies at each level of granularity addressing local site autonomy and user-level policies management (in the first case) and flexibility, scalability and manageability in the VO-level policies management (in the second case).

A combined User-VO data access control allows mixing the benefits related to the two approaches (any combination of insert, update, and delete privileges can be defined with the right level of granularity).

The Grid must provide the ability to control access based on user role (as it usually happens for DBMSs). Role based access control is fundamental for collaborative working, when several individuals may perform the same role at the same time and provides a scalable and manageable way to split users in subclasses with specific and well-known privileges.

Granting and revoking activities must be dynamically performed by administrators and should be easily carried out by using high level interfaces such as data grid portals.

Data access policies should be managed at the Grid- DBMS layer, without entirely relying on the back-end framework. This could enable data access control for trivial data resources such as text files and prevent the access attempts to the back-end systems for unauthorized users. [4]

7 **Transparency, Efficiency and Interoperability**

Transparency is a common requirement for grid services and fundamental to make virtualization a reality. There are various possible types of transparency in a distributed environment.

• Physical data location: the physical location of a database in the grid must be hidden/virtualized by the grid service.

• Naming: an application must be able to access a data source without knowing its name or location. These kind of information must be managed by means of mapping, alias, which conceal data that are not relevant to the end-user, such as connection string for the databases, DBMS port, login and password.

• Data replication: replication of data improves performance, availability and fault tolerance. The user must not be aware of the existence/management of multiple physical copies of the same data source, has just to deal with the logical (virtualized) data source name.

• DBMSs heterogeneity: many different RDBMSs exist, such as ORACLE, IBM/DB2, MySQL, SQLite. An increasing number of applications interact with not relational databases such as flat files and XMLbased documents in the bioinformatics and climate change domains. This kind of heterogeneity must be properly handled in order to provide a uniform access interface to different data sources and a grid database access service independent of the back-end systems. [4]

Efficiency

Performance plays a fundamental role in the data grid environment. High throughput, concurrent accesses, fault tolerance, reduced communication overhead, are important goals that must be achieved by exploiting among the others data localization and query parallelism. Efficient data delivery mechanisms can reduce the connection time and the amount of transferred data. [4]

Security is crucial for the management of a database in data grid environment. Data security aims at protecting data against unauthorized accesses by preventing unauthorized users from accessing data and protecting information exchanged in the data grid network. Authentication is strongly required to check user’s identity, authorization concerns privileges and read/write permissions. Users must be able to "log on" (authenticate) just once and then have access to any resource in the Grid that they are authorized to use, without further user intervention.

Most important production/research grids adopt the de-facto standard for security Globus Grid Security Infrastructure (GSI).

It provides full security support concerning data encryption, data integrity, protection against replay attacks and detection of out of sequence packets. GSI is widely used both in gLite and Globus based grid environments.

The Spitfire (European DataGrid Work Package 2, Project Spitfire) has implemented a security architecture based on transport-level SSL security and mapping of Grid credentials to database roles. [4]

Interoperability can be achieved by standard adoption. To achieve interoperability, using the method of defining and adopting common open standards and architectures is a common approach, which relies heavily on standardization and implementation processes. Because comprehensive implementations and roll-outs spanning different Grid communities are difficult, costly and often politically charged, in certain scenarios a coupling of the architectures is a reasonable alternative.

The loss of interoperability in a Grid of middlewares may lead to problems in the Grid's operation. [4]

8 Grid Integration

The full integration of database technologies with Grid middleware is widely recognized.

There are two main dimensions of complexity to the problem: reconciling implementation differences between server products within a single database paradigm (IBM, Oracle, Microsoft, etc.) and the variety of database paradigms (object, relational, XML, etc.). Each DBMS is the result of many hundreds of person-years of effort to provide a wide range of functionality, valuable programming interfaces and tools, and important properties such as security, performance and dependability.

Grid Data Integration service (GDI), this service provides XML schema mapping utilities for semantically connected XML data sources. To this aim the GDI extends the OGSA-DAI by introducing a new activity devoted to the reformulation of an XPath query by using the XMAP reformulation algorithm.

There is a considerable history in database research of semantic data modelling and data integration techniques, both being dimensions of the problem outlined above for Grid data services required in the earth sciences.

Data modelling has evolved from Codd's relational model through the ER model of Chen to semantic and fully object-oriented models incorporating inheritance, aggregation, and behaviour.

The databases literature also contains a considerable history of data integration methods. These have been developed for a rich range of problems including reverse and re-engineering, schema translation, and database integration.

Proven approaches for DBMS integration that might be examined for their applicability in a file-based data Grid include: data warehousing where data is imported enmasse from legacy databases and transformed into a common data model, and wrapper/mediator architectures where heterogeneous local sources are mapped to a global schema and integrated through middleware.

For integration of heterogeneous file-stores in a data Grid, the warehousing and federation models are impractical. Instead, a wrapper/mediator approach is required, with a common data model exposed through semantic data services.

The requirement for data integration on the Grid has led to a significant amount of activity in the GGF DAISWG, with specifications developed for relational and XML Grid Database Services.

What is really needed in a data Grid is a semantic data integration framework that allows the request on this global geographic dataset.

9 Conclusions

Data services for the Grid have focussed so far primarily on encapsulating data syntax (distributed relational databases, file format and location).

The elements of a generic framework would include:

- A meta-model for constructing semantically-rich domain specific data models independent of storage concerns
- A data storage description language for describing the construction of semantic data object instances.
- A canonical process for serialising semantic data instances in service workflows.

Both implicit and explicit knowledge-bases or ontologies are supported by the general framework.

Data access services may be built on top of a data model constructed according to the framework. These could be exposed through Activity extensions in OGSA-DAI.

10 Acknowledgement

The language used to convey requests to a database service, the database can be SQL for relational database services and XQuery for XML database services.

A single database system may support multiple paradigms.

With Grid Control can be manage a lot of things than just the database. For example, if you run Siebel, PeopleSoft, E-Business Suite, BI-EE, or custom Java application along with Oracle database you can manage them together with Grid Control.

Data grids provide what we need to can manage distributed data, the logical name spaces needed to assemble collections, a common infrastructure base upon which multiple types of data management environments may be implemented.

References

[1] Andrew Borley, Neil Hardman, Alan Knox, Simon Laws , James Magowan, Manfred Oevers, Ed Zaluska, “Grid Data Services – Relational Database Management Systems”, Version 1.0 , In: *5th Global Grid Forum*, July (2002), Edinburgh, Scotland. pp. 1-22

[2] Carl W. Olofson, “Grid Computing with Oracle Database 11g” ,Sponsored by: Oracle Corporation, March (2008)

[3] Ali Anjomshoaa, Mario Antonioletti, Malcolm Atkinson, Rob Baxter, Andrew Borley, Neil P Chue Hong, Brian Collins, Neil Hardman, George Hicken, Ally Hume, Alan Knox, Mike Jackson, Amrey Krause, Simon Laws, James Magowan, Charaka Palansuriya, Norman W Paton, Dave Pearson, Tom Sugden, Paul Watson and Martin Westhead, “The Design and Implementation of Grid Database Services in OGSA-DAI” - *Proceedings of UK e-Science* All Hands Meeting (2003) 2-4th September, Nottingham, UK pg 795

[4] Sandro Fiore, Salvatore Vadacca, Alessandro Negro and Giovanni Aloisio, “Grid database management: issues, requirements and future directions”, University of Salento & SPACI Consortium Euro Mediterranean Centre for Climate Change viale Gallipoli, 49 – 73100 Lecce – Italy, February (2004)

[5] Tuecke, S. ,“Grid Security Infrastructure (GSI) Roadmap”, (2001) , Internet Draft

[6] Timo Baur , Rebecca Breu, Tibor Kálmán, Tobias Lindinger, Anne Milbert, Gevorg Poghosyan , Helmut Reiser, Mathilde Romberg, “An Interoperable Grid Information System for Integrated Resource Monitoring Based on Virtual Organizations”, J Grid

Computing (2009) 7:319–333, DOI 10.1007/s10723-009-9134-3

[7] AndrewWoolf, Ray Cramer, Marta Gutierrez, Kerstin Kleese van Dam, Siva Kondapalli, Susan Latham, Bryan Lawrence, Roy Lowry, Kevin O'Neill, "Semantic Integration of File-based Data for Grid Services", Conference: Cluster Computing and the Grid - CCGRID , pp. 182-188, (2005) DOI: 10.1109/CCGRID,1558552

[8] John Wiley & Sons, Ltd. Concurrency Computat.: IBM Systems Journal Pract. Exper. (2005); 17:357–376

[9] Amy Krause (EPCC, University of Edinburgh, James Clerk Maxwell Building, Mayfield Road, Edinburgh EH9 3JZ, UK), Susan Malaika (IBM Corporation, Silicon Valley Laboratory, 555 Bailey Avenue, San Jose, CA 95141, USA), Gavin McCance (Department of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ, UK), James Magowan (IBM United Kingdom Ltd, Hursley Park, Winchester S021 2JN, UK), Norman W. Paton (Department of Computer Science, University of Manchester, Oxford Road, Manchester M134 9PL, UK), Greg Riccardi (Department of Computer Science, Florida State University, Tallahassee, FL 32306-4530, USA. + National e-Science Centre, 15 South College Street, Edinburgh EH8 9AA, UK), "Grid Database Service Specification", GDSS-0.2 4th October (2002)

[10] Moore, R., A. Rajasekar, "Common Consistency Requirements for Data Grids,Digital Libraries, and Persistent Archives", Grid Protocol Architecture Research Group draft, Global Grid Forum, April (2003)

[11] "Open Grid Services Architecture Data Access and Integration", http://www.ogsadai.org.uk

[12] Parent, C. and S. Spaccapietra, "Database Integration: The Key to Data Interoperability", In Advances in Object-Oriented Data Modeling, ed. M.P. Papazoglou et. al., The MIT Press (2000).

[13] Mario Antonioletti, Malcolm Atkinson, Rob Baxter, Andrew Borley, Neil P. Chue Hong1, Brian Collins, Neil Hardman, Alastair C. Hume, Alan Knox, Mike Jackson, Amy Krause, Simon Laws, James Magowan, NormanW. Paton, Dave Pearson, Tom Sugden1, PaulWatson and Martin Westhead, "The design and implementation of Grid database services in OGSA-DAI", Concurrency Computat.: Pract. Exper. 2005, 17:357–376, Published online in WileyInterScience(www.interscience.wiley.com). DOI: 10.1002/cpe.93

Florentina Ramona **Pavel (El Baaboua)** graduated from the Computer Science for Business Management, of the Romanian – American University in 2005. At present she is a PhD candidate at the Academy of Economic Studies and PhD assistant at the Romanian – American University of Bucharest. She is co-author of two books and articles in informatic fields.