WEB- BASED MODELING OF COAL MINE- CFTPP VIRTUAL CONSORTIUM. PART ONE- FROM IT TO KNOWLEDGE MANAGEMENT FOR A BEST REPRESENTATION OF AN ENTERPRISE

by
Mircea Rîşteiu and Iosif Andras

Project abstract
The main objective of the proposed research stage is to simulate and animate on the web interface a ground lignite mine which works correlated with a coal-fired thermal power plant (CFTPP) and transportation systems. Deeper goals: building internet and intranet objects for lignite mine and CFTPP, which might be separate configurable and separate compiled, for stand alone running; linking configured objects for complex simulation and animation with zoom and mouse-over facilities; web publishing of the simulators; building tools for user scenario; report- generator.

Paper abstract
In the first part of the stage we want to write a state-of-art in the field of the discrete mining simulation: simulation aware groupware tool versus web simulation. The purpose of this part is to identity how can we put together the experience and results witch have been done until now for answering to the reality questions. In this part we will be strongly focused on multi-user manner. It means that they will be used modular design of the simulation for extensible integration and for joining together groupware and simulation. The main results of this part will be the manner of identification of the horizontal and vertical structure of the models. The obtained schema opinion related to the simulation process is to reproduce at the highest level the practical situation (even the educational or other social influences if possible).

1.Introduction

Placing a real enterprise in a virtual context has many common senses. [1] resumes very well:

- applying the best know-how while performing core tasks- leaving things to serendipity will not achieve the benefits;
- there are not unlimited resources which means codifying that which is known;

Some tools and methods could be used: expertise profiling, knowledge mapping, information audits/inventory, but not always results are optimally.
An enterprise is a collection of infrastructures, processes, organizational structures and legislation and rules. An architecture of the enterprise is a formal representation of this collection. Enterprise architecture has the purpose effectively to align the strategies of enterprises with their business processes and their resources. An

enterprise architecture for an organization combines and relates all architectures describing particular aspects of that organization. META Group, for example, defines the enterprise architecture to consist of the enterprise business architecture, the enterprise information architecture, the enterprise-wide technology architecture, and the enterprise application portfolio.
The main goal of enterprise architecture is to provide insight in the organizational structures, processes and technology that make up the enterprise, highlighting opportunities for efficiency improvements and improved alignment with business goals.
In [2] some terms have to be fixed and:

- what is enterprise architecture and why is it important?
- what is the current state of practice in enterprise architecture?
- why should enterprises consider moving to a service-oriented enterprise architecture?
- what are the implications of service orientation for enterprise architecture?
- what support is required for doing service-oriented enterprise architecture?
- what roadmaps exist for moving to a service-oriented enterprise architecture?

Frameworks provide structure to the architectural descriptions by identifying and sometimes relating different architectural domains and the modeling techniques associated with them. [2] and (ITU, 1996) define framework for architecture specification of large distributed systems. It identifies five viewpoints on a system and its environment: enterprise, information, computation, engineering and technology. In this way some views might be identified: Business Architecture, Information System Architecture and Technology Architecture.

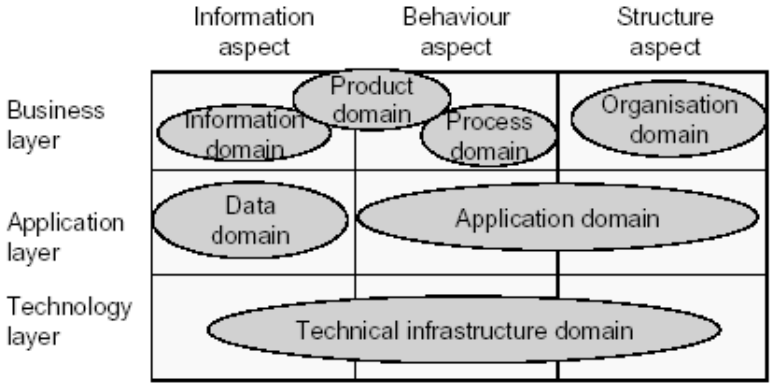

Figure 1. Enterprise architecture layer summarized by the three KM views layer

These non-homogenous and overlapped layers permit to draw trajectories in the frame of enterprise architecture:

- **FROM** tools that try and replicate human thinking **TO** those that augment it
- **FROM** providing passive information **TO** delivering active intelligence
- **FROM** reasoning with text **TO** visualization and pattern discovery
- **FROM** logical proof **TO** felt experience.

An architecture method is a structured collection of techniques and process steps for creating and maintaining an enterprise architecture. Methods typically specify the various phases of an architecture's lifecycle, what deliverables should be produced at each stage, and how they are verified or tested.

2.Service concept as a leading engine

From the overview of enterprise architecture in the previous section, one can conclude that there are two main issues in enterprise architecture today:

1. The problem of alignment between the various architectures;
2. The lack of a guiding principle for overall optimization w.r.t. an enterprise's goals.

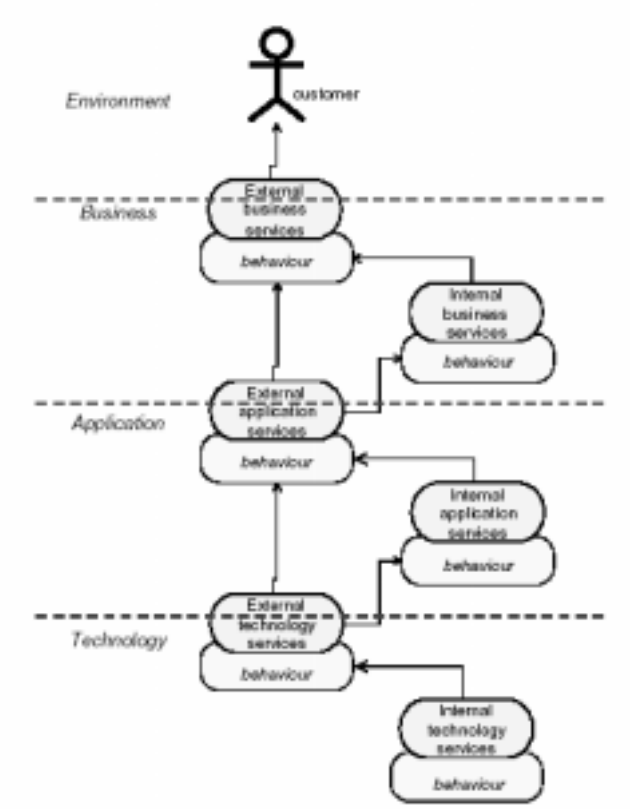

Figure 2. The representation of an enterprise as operational services {2].

In our search for a possible solution to these problems we ran into the service concept. The idea of systems (applications or components) delivering services to other systems and their users is really starting to catch on in software engineering, witness for example this book.
However, in other relevant disciplines there is an increasing focus on services too. In fact, economic development is to an increasing extent driven by services, not only in traditional service companies but also in manufacturing companies and among public service providers (Illeris, 1997). In the service economy, enterprises no longer convert raw materials into finished goods, but they deliver services to their customers by combining and adding value to bought-in services. As a consequence management and marketing literature is increasingly focusing on service design, service management and service innovation (e.g., see Fitzsimmons and Fitzsimmons, 2000, or Goldstein et al., 2002).
The resulting hierarchy corresponds to the architectural layers (business, application and technology) defined in the architectural framework in Figure 1. Each layer makes its external services available to the next higher layer. The external services of the higher layer may depend on services in the same or lower architectural layers. Business services, for example, may depend on external application services. Internal services are used within the same architectural layer; for instance, an application component may use services offered by another application component. External business services could also be called 'customer services', i.e., services offered to the (external) customers of the enterprise.

2.1 Applying IT techniques for separation of internal and external behavior

Following the definition of the service concept, a service should only define the externally observable behavior of a system, not how that behavior is realized. Such encapsulation has long been a guiding principle in software development (e.g., see Dijkstra, 1968). It provides a mechanism for being truly platform-independent, for substituting different implementations with the same external behavior, or interchange different suppliers of services.
The definition of the external behavior should be in terms of the invoking processes, systems, or users, making the added value and possible uses of the service explicit for service consumers. It encourages cross-domain thinking and design, reducing the semantic gaps between domains. It also facilitates the communication between stakeholders from these different domains, such as business analysts and software architects.
Services should be specified such that (potential) users require a minimum amount of information to understand the external behavior of the service, and with a minimum number of handles to operate the service for these purposes. Minimizing the shared understanding is an enabler for the second guiding principle, as it reduces gaps from different perspectives. These guiding principles provide directions for the development of services and interoperability in all sorts of domains. They hold for the

development of infrastructure services, application services, business processes and business functions.

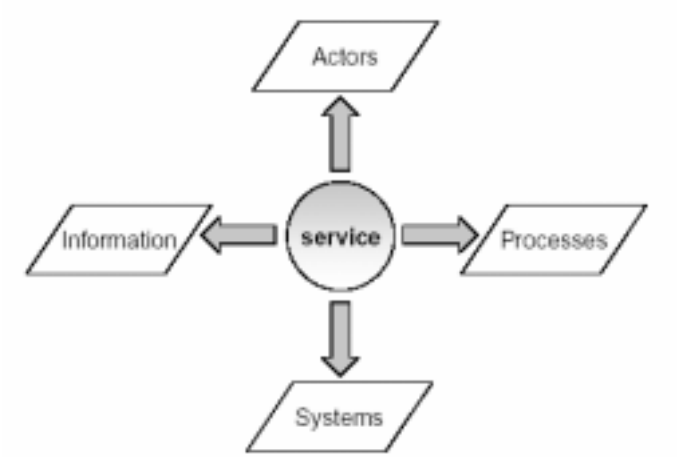

Figure 3. The external behavior is percept as a service [2]

Currently, optimizing an enterprise architecture, i.e., improving performance, quality or cost effectiveness, is typically done locally, within one type of architecture.
For example, an enterprise may decide to change its business processes to improve their efficiency. This change might require changes in the supporting applications and IT infrastructure, and lead to changes in the organization. In this way, the enterprise can at most achieve a locally optimal situation, because the resulting changes could have a detrimental impact on the efficiency of the applications and on the effectiveness of the organization as a whole. Our hypothesis is that services could provide the overall optimizing concept currently lacking in enterprise architecture. This hypothesis still requires validation, but is derived from the reasoning that most enterprises nowadays belong to the service industry and compete on service levels. Therefore their goal generally is to provide the best possible quality of service.
A wide variety of practices and processes are used in this frame. Some of the more common ones (basics of knowledge management) are shown next:
1. Creating and Discovering Creativity Techniques
    - Data Mining
    - Text Mining
    - Environmental Scanning
    - Knowledge Elicitation
    - Business Simulation
    - Content Analysis
2. Sharing and Learning Communities of Practice
    - Learning Networks

- Sharing Best Practice
- After Action Reviews
- Structured Dialogue
- Share Fairs
- Cross Functional Teams
- Decision Diaries

In order to facilitate a service-oriented approach to enterprise architecture, a high-level modeling language is needed in which the different conceptual domains can be described at a sufficiently abstract level.

2.1 Real-time Business Service Management

Tool vendors like BMC4 recognize the importance of integrating real-time IT service management with operational business processes and customer services. They provide tools that propagate events at the IT level to process owners and customers; the other way around, problem reports from users and customers can be propagated to the IT service level. Such integrations should offer operational business-IT alignment giving insight into real-time performance and service levels. These developments create a strong case for service-oriented methods, since they apply service-orientation in real-time operational service management allowing services to be used for on-line decision-making and problem solving.

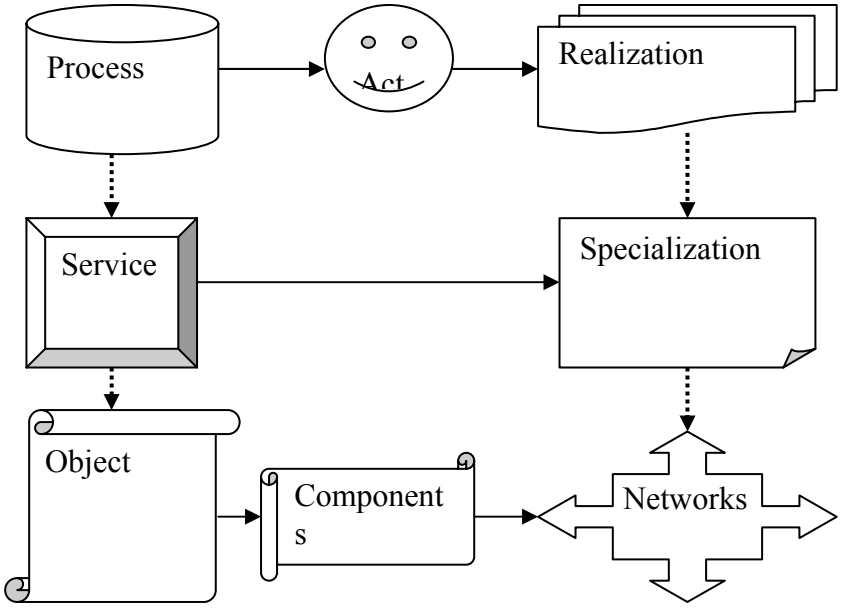

Figure 4 The simplified representation of the architecture of the enterprise

For the Figure 4, some of the next recognized criteria could be focused [3], [4]:

1. **Applications**
   - Enterprise Application Management
   - Application Performance
   - Enterprise Resource Planning (ERP)
   - Customer Relationship Management

2. **Data & Database Management**
   - Enterprise Data Management
   - Data Synchronization
   - Performance
   - Database Performance
   - Querying Performance
   - System Performance

3. **Administration**
   - Administrative Assistant
   - Database Administration
   - Recovery Management

4. **Infrastructure**
   - Infrastructure Management
   - Mainframe Management
   - Application Integration Management
   - Dynamic Batch Scheduling Management
   - Performance Assurance™
   - Infrastructure Monitoring
   - Internet Management
   - Network Management
   - Report Distribution
   - Web Services
   - Server Management

5. **Service**
   - Service Management
   - Advanced Event Management
   - Quality of Service
   - Service Delivery Management
   - Service Impact Management
   - Service Level Management
   - Service Support Management

- Change and Configuration Management
- Services by Category
- Education Services
- Professional Services

This list of criteria covers the knowledge database focused on:

- knowledge in products;
- knowledge in people;
- knowledge in processes .

3. Maturity models

Since both service-oriented development methods and technology and standards surrounding enterprise services are still very much under development, we may ask ourselves what the future of service-oriented architectures will be like, and when all these developments will finally lead to a mature, widely accepted and stable approach for organizations to proceed with such environments.

One outlook that tries to give a realistic prognosis for the near future is that of the Stencil
Group (see Sleeper and Robins, 2002). Their forecast with respect to the growth of the services market emphasize three phases: the first will cover the "organic adoption of services tools and standards" (2001-2003)- which is not effectively done, the second the "systematic deployment of services infrastructure" (2002-2006)- which is in progress and the third the "pervasive use of services in collaborative business processes" (2005 and beyond).

The vision of Sleeper and Robins is partly confirmed by Sprott's (2003) maturity model that identifies four phases on the way of service- oriented architecture (SOA) towards maturity: *early learning* (experimental, mostly internal, focused on better application integration) which is happening now, *integration* (still internal, business process oriented and based on a more mature understanding of SOA) that will start probably in 2004 and will take three years, *reengineering* (services used across organizations, and implemented as part of business products, both internal and external) that will start in 2005 and finally the *maturity* (ubiquitous and federated services, service consumer ecosystems). Nevertheless, Sprott does not predict when we should expect maturity.

4. Conclusions. Future task

There are numerous resources available that discuss what services are, how to implement or use them, what are their benefits in terms of costs, ROI, flexibility and architecture integration.

However, in contrast with the literature addressing classical enterprise and software architectures, one can hardly find well-structured methodologies, frameworks or best practices that might assist enterprise architects during the complex migration process from classic enterprise architectures to SOA and services.

This is not a total surprise, because such instruments are usually developed as soon as the expertise gathered in the application of new ideas and technologies in real environments reach a certain critical mass, which is not yet the case with SOA: there are few examples of fully developed and mature service-oriented enterprise architectures. However, we can refer to a few “roadmapping” initiatives in the area of SOA.

4.1 Future task

The next part of application related to SOA tend to focus on the details, while the best approach to formulating a SOA strategy is to consider all aspects of the IT infrastructure, both inside the enterprise and among the company's partners.

In the traditional "design-build-run" waterfall model lifecycle, architects create models during the design time phase for developers to follow during the runtime phase, and the models are then put on a shelf. In an SOA, however, the model continues to serve a critical role on an ongoing basis. The core model in an SOA models the coarse-grained business Services that encapsulate and expose IT functionality to the business realm. This model acts as the clearinghouse for information about what's going on in the IT environment at any point in time. It tracks current and future business requirements, and follows the current and future versions of the Services available on the designed network.

The Service-oriented approach to accessing legacy data and functionality is different from the traditional, adapter-based approach. Once a company has a requirement to access a legacy system, developers should encapsulate that system with a Services interface that exposes the functionality of the legacy system as atomic Web Services. Once the encapsulation is complete, those Services are now available on the network to meet current as well as future requirements of the business.
This described part will be a distinguished part, will be summarized in a separate work paper. The work will cover web- based modeling medium (Scalable Vector Graphics-SVG) into the frame of Web Based Simulation Center (WBSC) and B2B [9], [10], the manner of building the models for integrating in these concepts.

References

[1] http://www.skyrme.com/updates/u11.htm#Feature

[2] Service-Oriented Enterprise Architecture, M.W.A. Steen, P. Strating, M.M. Lankhorst, H. ter Doest, M.-E. Iacob Telematica Instituut, the Netherlands
[3] www.dbazine.com
[4] http://searchWebservices.techtarget.com/originalContent/
[5] http://sys-con.com/webservices/
[6] http://www.w3.org/2002/ws/arch/2/wd-wsawg-reqs-04012002
[7] A tool integration workbench for enterprise architecture. Integrating heterogeneous models and tools, Diederik van Leeuwen, Hugo ter Doest, Marc Lankhorst Telematica Instituut, P.O. Box 589, 7500 AN, Enschede, The Netherlands
[8] Tool Support for Enterprise Architecture – A Vision, Hugo ter Doest, Marc Lankhorst Telematica Instituut Enschede, The Netherlands
[9] Maps, Agents and Dialogue for Exploring a Virtual World, Anton Nijholt, Job Zwiers and Betsy van Dijk, Centre of Telematics and Information Technology (CTIT) University of Twente, PO Box 217 7500 AE Enschede, the Netherlands
[10] http://isgsim1.cs.uni-magdeburg.de/osterburg/wbmsc.htm, Concept for a Web-Based Mining Simulation Center (WBMSC)
[11] http://www.b2bsim.de/~osterbur/diplom.pdf, A Portal for B2B Simulation , Stefan Osterburg

Authors:

Mircea Rîşteiu-"1 Decembrie 1918" University of Alba Iulia, Department of Computer Science, N.Iorga Street, No. 11, 51009 Alba Iulia, Romania, tel: +40723 774513, +40258 806270, fax : +40258 812630, e-mail: mristeiu@uab.ro, mircearisteiu@yahoo.com

Iosif Andraş-University of Petrosani, Department of Mechanical Engineering, Universitatii Street, No. 20, 330062, Petrosani, Romania, tel: 0254-54.25.80, 0254-54.25.81, 0254-54.29.73, 0254-54.33.82, 0254-54.65.80, e-mail: andras@upet.ro