Towards a Domain Specific Application Development Environment for the ELeGI architecture: the Software Factories approach

Angelo Gaeta, Pierluigi Ritrovato, Matteo Gaeta

Towards a Domain Specific Application Development Environment for the ELeGI architecture: the Software Factories approach

Angelo Gaeta, Pierluigi Ritrovato
Centre for Research in Pure and Applied Mathematic
{agaeta, ritrovato}@crmpa.unisa.it

Matteo Gaeta
DIIMA – University of Salerno
{gaeta}@unisa.it

The Next Generation Grids (NGG) expert group has pioneered the vision of "Invisible Grid" whereby the complexity of Grid systems and architectures is hidden to both developers and users. In this new vision, the Grid has a different role: it will not more provide a virtual computing environment but it will be the basis of a more complex service oriented, knowledge-based and collaborative environment suitable for the specific domains in which citizen and organizations have to operate. Grid community main effort is still related to definition and creation of middleware upon which the development of Grid architectures and applications remains a straight difficult task that can be done only by expert researchers and developers. This is a factor, among other ones, that will prevent the fulfilment of this vision. In this paper we try to contribute to a challenging research topic: how can we easy develop applications for Grid architectures and environment? To this aim, we will show an approach to the creation of development environment for Grid applications and we will see how our approach affects the ELeGI software architecture.

INTRODUCTION

Since from its early days, the Grid computing [1] has raised many expectations in both industrial and academic communities that immediately identified the benefits and advantages of using Grid technologies. The computational paradigm proposed by the Grid went beyond the classical distributed computing paradigm. Foster et al. [5] present the specific problem that underlies the Grid concepts as coordinated resource sharing and problem solving in dynamic, multi-institutional Virtual Organizations (VO) where the distinctive features with respect to the other distributed computational paradigm was the VO concept. In those days Grid development of middleware and applications was driven by the increasingly request of computing power from the science community
The issues of definition and development of Grid middleware and applications are being studied by researchers since from the beginning and a good survey about Grid technologies, middleware, applications and world wide projects was provided by Baker, Buyya, and Laforenza in [6] that concluded outlining that "*It is difficult to predict the future in a field such as information technology where the technological advances are moving very fast. Hence, it is not an easy task to forecast what will become the 'dominant' Grid approach. Windows of opportunity for ideas and products seem to open and close in the seeming 'blink of the eye'*". They were right.
Grid paradigm is evolved and the major milestone of its evolution is the definition of the Open Grid Service Architecture (OGSA) [2] marked by the adoption of a service oriented approach in designing Grid architectures. Integrating Grid technologies and Web Service ones, the Grid community aims to promote a wide adoption of Grid technologies in both e-business and e-science domains trying to move Grid form a niche technology to a commodity one. The key concepts of OGSA together with the raising of the Semantic Web vision [8] and its related technologies have provided new exiting challenges and new research priorities.
In [7] it is established a vision of a "*Grid-enabled, seamless, collaborative knowledge and services environment for all"* in which "*the role of the Grid will change from providing a virtual computing environment to being the basis of a knowledge fertilisation and communication environment"*. The Grid 'dominant' approach today seems to be the so-called "Grid for all" moving beyond computing-on-demand model towards a model integrating semantic and knowledge technologies, characterised by domain verticalisation (Grid domain specific) through the definition of specific services, development environment tailored on the domain need and inter-grid interoperability and widely adoption in specialized domains.
The European Learning Grid Infrastructure (ELeGI) project [22], since its conception, has foreseen this "Grid for all" vision. Through the adoption of a service-oriented model that is deeply intertwined with the use of semantic tagging, and it is aligned with the global community developing trends such as OGSA, the Web Services Resource Framework (WSRF) [24] and the Next Generation Grid (NGG) [3] [4], ELeGI aims to define and design a Grid software architecture in order to provide a specific verticalizion for the e-Learning domain.

In order to fulfil this challenging vision, there are many research issues to solve. The one we try to address in this paper is concerned with the definition of Domain Specific Languages, tools and development environments that have to hide Grid complexity and simplify the development of applications in the e-Learning domain.
To this aim, we propose a methodology based on the classical Domain Modelling approach, and a development environment suitable for development of Grid applications. We will argue how the proposed development environment, based on the new Microsoft Software Factories [17] strategy for development of distributed component-based applications, will be able to capture, by exploiting some specialized Domain Specific Languages, many of the desired feature of a Grid application for the e-Learning domain.
The paper is organized as follow: section 2 presents some works related to the development of Grid applications, section 3 summarizes the ELeGI strategy to progress human learning and provides an overview of its Grid software architecture, section 4 presents the software Factories approach and our proposal to adopt this approach in a Grid environment, section 5 presents the impact that our approach potentially has on the ELeGI project. Finally, we present our conclusions and future works.

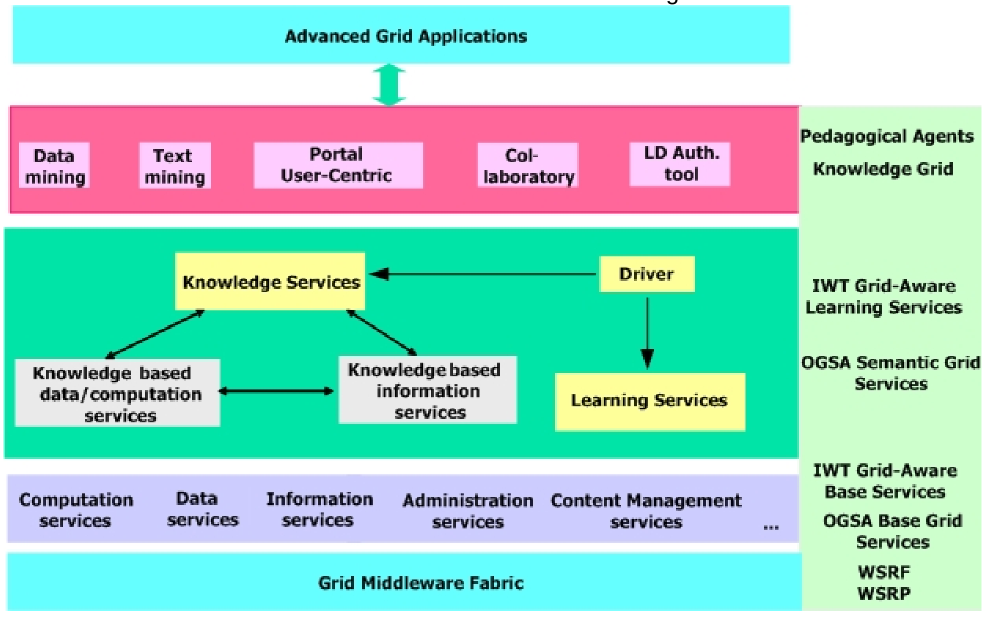
RELATED WORKS

Of course, the development of Grid applications, the creation of Grid application programming environments as well as the redesign of distributed applications in order to be Grid-Aware, they are topics addressed by many research projects. For instance, among the project specialized for Grid application development, we remember the Grid Application Development Software (GrADS) project [9] and its extension Virtual Grid Application Development Software (VGrADS) [10].
Fundamentally, these project share the basis idea of the NGG that is to hide Grid complexity and allow end-user to [13] "*be able to specify applications in high-level, domain-specific problem solving languages and expect these applications to seamlessly access the Grid to find required resources when needed*". One of the main results of the GrADS project is its execution framework for adaptive Grid application based on the concepts of configurable object program [12] [13].
Another project sharing the same philosophy is the GridLab project [11] that "*aims to provide fundamentally new capabilities for applications to exploit the power of Grid computing, thus bridging the gap between application needs and existing Grid middleware*" and its main result is the Grid Application Toolkit (GAT) providing access to services needed by the Grid applications [15].
Even if they share our purpose, these projects are focalized on the computational Grid vision and they are mainly related to address issues of dynamic resource management and configuration in order to allow adaptivity and reach some desired performances especially for scientific applications. They have obtained good results, both from methodological and technical viewpoints but they can not be directly applicable to the development of Grid applications based on the "Grid for all" vision provided in [7]. They lack, for instance, in providing seamless access to existing knowledge, persons and groups anytime and anywhere, and to support the creation, encapsulation and exploitation of new joint knowledge. In our effort to re-think a Grid application to fulfil the innovative role of the Grids, we think these are relevant issues that developers have to face in order to develop Grid applications, obviously together with the concepts of adaptivity and performance contracts well addressed in the aforementioned projects.
A Grid programming environment is also object of study of the Grid.it project [16]. In the frame of the project, it will be investigated an "*innovative programming environments that i) support the programmers in all the activities related to parallelism exploitation, by providing some kind of structured primitives for parallelism exploitation; ii) allow to achieve full interoperability with existing software, both parallel and sequential, either available in source or in object form; iii) support and enforce reuse of already developed code in other applications*" [14]. Laforenza, Vanneschi et al. in [14] present their research studies to design and develop a Grid-Aware application programming environment based on an ASSIST, which is a structured parallel programming environment originally designed to address cluster and network of TCP/IP workstations [18]. In ASSIST the Grid-Aware application can be built as composition of ASSIST components and possibly other existing (legacy) components. Applications are supported by a Grid Abstract Machine including dynamic application management and all the abstractions from the resource, collective and connectivity levels of a Grid middleware. An Application Manager supports static allocation and dynamic reallocation of adaptive applications according to a performance contract, a reconfiguration strategy, and a performance model.
Our approach is close to ASSIST. Like ASSIST, in fact, we propose a component based programming environment, we support reuse of already developed code and like ASSIST we rely upon an abstraction of the Grid middleware functionalities. Our abstractions are modelled using a set of Domain Specific Languages (DSL) for modelling all the features of a Grid environment (e.g. typical resource and information management as well as innovative semantic and knowledge management and specific domain verticalizations) in terms of components and interactions among them. In contrast with ASSIST our approach is not tied to structured parallelism of applications and it is neutral form this viewpoint. As we will see later (see Section 0) our application development environment can be customized on the basis of the specific DSLs adopted and, from a methodological viewpoint, we could define different distributed programming models and techniques. Talia and Lee in [19] provide a dissertation about different Grid programming models and techniques and we observe that, theoretically, our development environment can be customized in order to support one or more of these programming styles. But, as we will se in the later, currently there are many practical issues, mainly due to the early state of the tools allowing the Software Factories approach, that doesn't allow this from a practical viewpoint.

TOWARDS THE SEMANTIC GRID FOR HUMAN LEARNING: THE ELEGI ARCHITECTURE

The aim of ELeGI [22], the European Learning Grid Infrastructure EU funded Integrated Project, is to progress effective human learning by promoting, supporting and demonstrating a learning paradigm shift from the current information transfer paradigm, to one that focuses on the learner and on the knowledge construction using experiential and collaborative learning approaches in a contextualised, personalised and ubiquitous way.
In ELeGI, the learner has an active and central role in the learning process. Learning activities are aimed at facilitating the construction of knowledge and skills in the learner, rather than the memorisation of information. Keeping the learner at the centre of the new learning processes personalisation (exploiting the existing learner's capability and skills) and individualisation (create and adapt learning paths according to learner's preferences) become relevant aspects to be supported by technologies through the creation of the right context.
ELeGI will adopt a service-oriented model, which is deeply intertwined with the use of semantic tagging, aligns itself with the global community developing trends such as OGSA, WSRF and the Next Generation Grid more in general, and its research directions should look at the creation of a Semantic Grid for Human Learning: The Learning Grid.
In [20] we have defined the Semantic Grid for Human Learning as a domain verticalization of the Semantic Grid improved with tools, services, languages, standards and technologies for the Education & Training. To create this particular instance of the Semantic Grid, we foresee: (i) adoption of IMS Learning Design (IMS-LD) specifications and ii) adoption of the Web Services for Remote Portlets (WSRP) standard. As explained in [20], through IMS-LD we add a high-level support to educational models and with WSRP we enhance the low-level service model, allowing the creation of a service model dynamic, stateful and presentation-oriented.
The reference architecture is the Semantic Grid architecture provided by Globe et al in [29] upon which we foresee a set of semantically enriched services typical of the learning domain that we have identified in our previous works [21] and we have defined as IWT Grid-Aware Services. Two sets of services are identified: the Base Services providing functionalities typical of a Learning Management System and the Learning Services providing high-level functionalities for a personalized learning experience, like learning path personalization, student's model evaluation and Driver services for management and delivery of learning experiences. These last one services are WSRP complaint, so they are able to produce and aggregate GUIs. The architecture is depicted in Figure 1

FIGURE 1: The Semantic Grid for Human Learning architecture



For a description of the Semantic Grid for Human Learning, see [20]. In this context, we only emphasize that, as we will see in the next sections, our approach aims to create development environments for learning scenarios production.

THE SOFTWARE FACTORIES METHODOLOGY TO CREATE A GRID APPLICATION DEVELOPMENT ENVIRONMENT

A detailed explanation of the Software Factories methodology can be found in [17], in which a software factory is defined as "a software product line that configures extensible tools, processes, and content using a software factory template based on a software factory schema to automate the development and maintenance of variants of an archetypical product by adapting, assembling, and configuring framework-based components".
According to [25], a software product line is a set of software systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and are developed from a common set of core assets in a prescribed way. The advantages and drawbacks of the software product lines are described in [25].
The approach proposed by Microsoft is a software product line and it is based on two central elements: a software factory schema and a software factory template.

A software factory schema is a way to categorize and summarize development artifacts, such as XML documents, models, configuration files, build scripts, source code files, in an orderly way, and a way to define relationships between them, so that we can maintain consistency among them. Once defined the software factory schema, we must implement it, defining the Domain Specific Languages (DSL), patterns, frameworks, and tools it describes, packaging them, and making them available to product developers. Collectively, these assets form a software factory template. Finally, we need and extensible development environment that, when configured with a software factory template, becomes the software factory for the specific product family.
The described methodology is not new and it is based on the convergence of key ideas in systematic reuse, model-driven development, and process frameworks. Much emphasis is in the central role of modelling in the software development processes and in the adoption of specialized DSLs to model specific domain for the product family.
DSLs belong to the class of modelling languages and they are used to build models of a specific domain to address. At the present, there are a lot of modelling technologies relying upon modelling languages. The most famous one is the Model Driven Architecture (MDA) [26] that provides a standardized approach to model driven development on the basis of platform abstractions. The purpose of MDA is to separate the design of application or business logic from the implementation platform and it makes distinction between Platform Independent Models (PIM) and Platform Specific Models (PSM). The first relies upon OMG standards including the Unified Modeling Language (UML), Meta-Object Facility (MOF), XML Meta- Data Interchange (XMI), and Common Warehouse Metamodel (CWM) in order to design a model, while the second relies upon tools to translate the PIM into a particular platform dependent implementation.
The Software Factories and MDA are both methodologies to software development and are both based on the model-driven approach. What are the differences?
Similarities and differences are discussed in [27], in which it is emphasized that the approach of using general purpose modelling languages, as the UML or the MOF, is not often a successful factor in model-driven software development. While the adoption of standards and general purpose modelling language allows portability and platform independence, mostly it is required additional effort in order to allow i) a domain customization and ii) appropriate technology mapping. Domain specificity is the key feature of DSLs.
Now that we have given an overview of the Software Factories methodology, we have to justify our proposal to adopt it in order to create a Grid Application Development environment.
First, we are convinced that raising the level of abstraction of a Grid system is a key factor to allow development of Grid application. To this purpose, we are investigating the capability of using a set of DSLs for modelling all the features of a Grid environment (e.g. typical resource and information management as well as innovative semantic and knowledge management and specific domain verticalizations) in terms of components and interactions among them. All the produced DSLs will be part of a specialized Software Factory Grid Template (SFGT) that, together with the other “Grid products” (supporting tools, orchestrator engines, frameworks), will be deployed in an extensible IDE in order to create a specialized development environment for Grid applications.
The described one is only a proposal and currently we have not tangible results. Even if the software product line’s concept is not new, the Microsoft strategy to address this concept is quite recent. First Tech Preview of tools supporting the Software Factories strategy (e.g. for creating, editing and visualizing DSLs) was released in December 2004.
Anyway, we have decided to move our efforts toward the Software Factories instead of more mature approaches, like the MDA, for the following reasons: i) according to us, the specificity of DSLs, in contrast with the generality of UML, can provide benefits in the modelling phase of the building blocks of a Grid system and of domain-specific applications, as the e-Learning ones, ii) the Software Factories approach is part of the Microsoft strategy to address HPC market with solutions and products that, according to us, are suitable for development of high performance distributed environment and development of domain specific distributed applications.
The advantages and drawbacks of adoption of Microsoft products in order to design and develop Grid systems are described in our previous works as, for example, [28].

THE IMPACT OF OUR APPROACH ON THE ELEGI PROJECT

We have described our proposal to create an environment for development of Grid applications on the basis of the Software Factories methodology and, in this section, we point out the impact that our approach can have on the ELeGI project.
As stated in the section 0, one of the main results of the ELeGI project is the definition of a service oriented architecture that we have referred as Semantic Grid for Human Learning and we have defined as a domain verticalization of the Semantic Grid improved with tools, services, languages, standards and technologies for the Education & Training.
In [22] and [20], we have justified the adoption of IMS-LD in order to design future learning scenarios that go beyond simple web based LO’s delivery, and we have discussed about the advantages gained from the integration of IMS-LD specifications and Grid technologies. Moreover, we argued that IMS-LD is the best approach to modelling teaching and learning practices that are able to include many different pedagogical models that we have identified in [20] and [23] and we have judged as fundamental in order to create future learning scenarios.
Our aim is to allow the creation of future learning scenarios and applications and we realize the importance of both the modelling of learning and teaching practices and development environments in order to build these scenarios.
We foresee that our approach can have a great impact in the development of authoring tools and, in general, development environment for the creation learning specific Grid applications.

The definition of a SFGT is not enough to create a Software Factory for the development of learning applications for Grid systems. In fact, the SFGT is only able to capture the abstractions and behaviours of a Grid system and, obviously, it is not able to capture all the pedagogical features required for the creation of future learning scenarios. To address this issue we propose the definition of additional DSLs providing the semantics of the base elements of IMS-LD (e.g. Activities, Roles, Environment, …) and providing relationships with the DSLs of the SFGT. These additional DSLs will be part of domain specific SFGT. Eventually, higher-level DSLs can be defined upon the previous ones in order to capture more specific features related to different pedagogies and learning approaches.

CONCLUSION AND FUTURE WORKS

In this paper, we have presented our ideas in order to create development environment for Grid applications. We have presented the Software Factories approach and we have justified its adoption, for our purpose, with respect to more stable methodologies for software development based on the model-driven approach, as the MDA. As pointed out in the previous sections, currently we have not tangible results and this is also due to the early state-of-art of tools for supporting the DSLs. For instance, currently the Tech Preview of DSL tools support only two templates for DSLs design and do not allow the customization of templates.
It is not a simple task to define DSLs able to capture all the necessary Grid abstractions and all the identified pedagogical features in order to create Software Factories Template to support development of Grid applications for the learning domain, but we believe that specificity of DSLs respect to the generality of other modelling languages can be a key features to reach our target. Our first steps will go towards this direction.
Finally, we emphasize that our effort aims to hide the complexity of Grid systems and it is lined-up with the “Grid for all” vision.

REFERENCES.

[1]. *I. Foster and C. Kesselman*: The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999

[2]. *I. Foster, C. Kesselman et al*: The Physiology of the Grid: An Open Grid Service Architecture for Distributed System Integration, 2002

[3]. Next Generation Grid(s) – European Grid Research 2005-2010, Expert Group Report, 16 June 2003

[4]. Next Generation Grids 2 Requirements and Options for European Grids Research 2005-2010 and Beyond. Expert Group Report, July 2004

[5]. *I. Foster, C. Kesselman, and S. Tuecke:* The anatomy of the Grid: Enabling scalable virtual organizations. The International Journal of High Performance Computing Applications, 15(3):200–222, Fall 2001

[6]. *M. Baker, R. Buyya, and D. Laforenza*: Grids and Grid Technologies for Wide-Area Distributed Computing, International Journal of Software: Practice and Experience (SPE), Volume 32, Issue 15, Pages: 1437-1466, Wiley Press, USA, December 2002.

[7]. Information Society Technologies FP7 exploratory workshops concerning Emerging Knowledge Infrastructures, Technologies and Applications. Report of the workshop Grid-enabled knowledge organisations and collaborative working environments. September 2004, available at http://www.cordis.lu/ist/grids/pub-report.htm

[8]. *T. Berners-Lee, J. Hendler and O. Lassila*: The Semantic Web. Scientific American, May 2001

[9]. http://www.hipersoft.rice.edu/grads/

[10]. http://www.hipersoft.rice.edu/vgrads/index.htm or http://vgrads.rice.edu/

[11]. http://www.gridlab.org/

[12]. *I. Foster, D. Gannon, K. Kennedy et al.:* Toward a Framework for Preparing and Executing Adaptive Grid Programs. Proceedings of NSF Next Generation Systems Program Workshop (International Parallel and Distributed Processing Symposium 2002), Fort Lauderdale, FL, 2002

[13]. *I. Foster, D. Gannon, K. Kennedy et al.:* The GrADS Project: Software Support for High-Level Grid Application Development. International Journal of High Performance Computing Applications, Winter 2001 (Volume 15, Number 4), pp. 327-344

[14]. *D. Laforenza, M. Vanneschi et al.*: Components for High-Performance Grid Programming in GRID.IT. In V. Getov and T. Kielmann, editors, Component Models and Systems for Grid Applications. Proc. of the Workshop on Component Models and Systems for Grid Applications, June 26, 2004 held in Saint Malo, France. Springer, 2005, to appear. ISBN: 0-387-23351-2

[15]. *G. Allen, K. Davis et al*: Enabling Applications on the Grid: A GridLab Overview**.** International Journal of High Performance Computing Applications: Special issue on Grid Computing: Infrastructure and Applications, to be published in August 2003.

[16]. http://www.grid.it/

[17]. *J. Greenfield and K. Short*: Software Factories: Assembling Applications with Patterns, Frameworks, Models & Tools. September 2004

[18]. *M. Vanneschi*: The programming model of ASSIST, an environment for parallel and distributed portable applications. Parallel Computing, 28(12):1709–1732, December 2002

[19]. *C. Lee and D. Talia:* Grid Programming Models: Current Tools, Issues and Directions. In Grid Computing: Making the Global Infrastructure a Reality, F. Berman, G. Fox and T. Hey (eds.), Wiley, chap. 21, pp. 555-578, 2003.

[20]. *A. Gaeta, M. Gaeta, P. Ritrovato and F. Orciuoli*: Enabling Technologies for future learning scenarios: The Semantic Grid for Human Learning. Proceeding of the Second International Workshop on Collaborative and Learning Applications of Grid Technology and Grid Education CLAG + Grid.edu 2005. To be held in conjunction with the IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005) May 9 - 12, 2005, Cardiff, United Kingdom

[21]. *N. Capuano, A. Gaeta, G. Laria, F. Orciuoli and P. Ritrovato*: How To Use GRID Technology for Building Next Generation Learning Environments. To appear in the book Towards the Learning GRID: advances in Human Learning Services that will be published by the IOS Press in the series: "Frontiers in Artificial Intelligence and Applications"

[22]. *M. Gaeta, P. Ritrovato and S. Salerno*: ELeGI: The European Learning Grid Infrastructure. Proceeding of the Grid Learning Service Workshop at ITS 2004 - August 30, 2004, Maceio, Brazil.

[23]. *C. Allison, S. A. Cerri, P. Ritrovato, M. Gaeta and S. Salerno*:, Human Learning as a Global Challenge: European Learning Grid Infrastructure. Tampere, Finland: University of Tampere Press.

[24]. *I. Foster, J. Frey, S. Tuecke et al.:* The WS-Resource Framework, 2004.

[25]. *P. Clements and L. Northrop*: Software Product Lines: Practices and Patterns. Addison-Wesley, 2001

[26]. *J. Miller, J. Mukerji et al.*: MDA Guide Version 1.0.1. Technical Report omg/2003-06-01, Object Management Group, June 2003

[27]. *S. Cook*: Domain-Specific Modeling and Model Driver Architecture. MDA Journal, January 2004

[28]. *R. Baraglia, A. Gaeta, M. Gaeta, D. Laforenza and P. Ritrovato*: Design of OGSA-Compliant Grid Information Service Using Microsoft Technologies. Proceeding of 6th International Meeting on High Performance Computing for Computational Science – VECPAR04, June 2004, Valencia, Spain – To be published in the post-conference book of VECPAR '04, LNCS

[29]. *Goble C. and De Roure D.*: The Semantic Grid: Myth Busting and Bridge Building. Proc. of the 16th European Conference on Artificial Intelligence, Valencia, 2004