SFIDA: interoperability in innovative c-business models for SMEs through an enabling Grid platform

M. Aldinucci[1], S. Gusmeroli[2], M. Vanneschi[3], M. Villa[4]

[1] Inst. of Information Science and Tech. (ISTI – CNR), 56124, Via Moruzzi 1, Pisa, Italy
aldinuc@di.unipi.it

[2] TXT e-Solutions, via Frigia 27, 20126, Milano, Italy
sergio.gusmeroli@txt.it

[3] Computer Science Dept. – University of Pisa, 56127, Largo B. Pontecorvo 3, Pisa, Italy
vannesch@di.unipi.it

TXT e-Solutions, via Frigia 27, 20126, Milano, Italy
matteo.villa@txt.it

Abstract. This position paper describes the objectives of project "SFIDA" (co-funded by the Italian Government), aiming at developing a GRID-based interoperability platform able to support next generation Supply Chain Management applications specifically addressing the needs of SMEs belonging to industrial districts and dynamic supply networks. Next generation SCM applications are intended in SFIDA to be based on componentization (e-services), intelligence (mining), collaboration (c-business) and customer business-processes orientation. The platform and the next generation SCM applications running on top of it will be tested in various typical industrial cases, spanning from automotive, textile, food, white goods and media retail.

1 Introduction

This position paper summarizes the main objectives of "SFIDA", a project co-funded by the Italian Ministry of Research, University and Instruction (MIUR). The project started on 1st January 2005, so it is currently only possible to describe its goals and to show how they address the area of interoperability of enterprise software and applications.

1.1 The Context

Project SFIDA is driven by a long-term view in which SMEs belonging to industrial districts and supply chain will be fully enabled to establish and run collaborative business relationships with their partners, beyond technological, economic, cultural and political barriers. This "plug & do business" view represents an ambitious challenge to the IT industry, requiring it the capability to provide new solutions and enabling platforms allowing SMEs to synchronize the business processes and the inter-operability of their systems.

Recent progresses and efforts in the systems integration area proved that without considering also the business context in which they are inserted, and without considering also knowledge management, use and sharing issues, it would be extremely difficult to have them accepted by final users [21].

Notwithstanding this model, even if made inter-operable, it is still based on IT products licensing, which is valid for medium-big companies, but which represents a great barrier for SMEs, due to the fact that very often SMEs need only a small portion of the functionalities that big "monolithic" solutions offer, while they have to pay for the entire product. On the other hand "ad-hoc" solutions require a long set-up and configuration process that are often more expensive than a license.

Moreover SMEs communities have specific classes of problems that currently are not being addressed, like:

1. they are not able to overcome the problems related with the establishment of faithful relationships amongst various actors when a direct human contact is not present
2. the lack of IT-based added value services to support cooperation amongst SMEs, in particular to model their applicative domain
3. the lack of efficient co-operative models for the "virtual" world
4. their inability to use existing technologies to improve inter-dependencies inside and outside the company

1.2 The Process-Oriented Collaborative Model

SFIDA is planning to support the collaborative business processes of SMEs, having identified the following steps as necessary in order to achieve a "collaborate to compete" objective [22]:

1. to **focus** on "strategic competences" (organizational, procedures, technical, human capital, etc…) through "self-awareness" mechanisms
2. to **expose** such competences in a structured way, to other (possible) partners
3. to **compose** competences exposed by other partners according to semantic interoperability rules, ensuring knowledge sharing independently of a specific cultural or linguistic context

4 to **manage** a collaborative process, by measuring and monitoring performances parameters, and by supporting business operations by processes synchronization, knowledge sharing and exchange, cooperative work, and IT systems interoperability

Anyway, even if such points are basically accepted and felt relevant by SMEs, there are still economic, social and cultural barriers [23] that are slowing down the adoption of such model.

For this reason SFIDA is planning to develop a **new methodology**, specifically targeted to SMEs belonging to Italian industrial districts and supply chains, leveraging on cultural similarities deriving from geographic proximity, and by providing specifications for IT systems to support SMEs in the process of "becoming collaborative". More in particular, the idea is to define a new IT solutions development paradigm based on a new IT service-based model oriented to collaborative processes.

The they key aspects of such development paradigm are:

1 a components-based view of enterprise applications
2 a components seamless and flexible composition and integration
3 de-centralized architectures with central (or hierarchical) coordination
4 the adoption of an "e-business Grid Computing" platform.

1.3 GRID Computing as the Enabling Technology

Grid Computing concerns essentially a range of technologies intended to abstract and support resource cooperation and data sharing between groups of platforms [1]. These may be complex distributed architectures, whose nodes are parallel machines of any kind, including PC/workstation clusters. In general such platforms are characterized by heterogeneity of nodes, and by dynamicity in resource management and allocation. A key issue of Grid computing technologies is the ability of dynamically bridging a suitable subset of resources and services to support the execution of large-scale applications. Among them, "naturally" distributed applications are of major interest. These applications are composed of different software modules, which exploit different needs in terms of geographic localization and disclosure proprieties of data and code, hardware architecture, O.S., and software developing tools.

Enabling technologies for Grid-aware application include:

✍ A high-level programming environment and toolkit supporting the development of configurable code and the binding of legacy code. This environment should enforce the minimization of developing cost by enabling the (static and dynamic) reconfiguration of the application to target different customer scenarios, while exploiting an optimal/maximal performance/cost ratio over a broad class of hardware platforms. The programming environment should also cope with legacy code by providing the programmer with a suitable set of adaptors for standard software technologies (Corba, CCM, Java Beans, DCOM, etc.)

✍ An advanced run-time support for developed code able to seamlessly adapt the application structure to the current platform status. It should transparently manage faulty events and performance degradations of the underlying platform, which should be considered the standard behavior of a large-scale distributed platform. The run-time support should cope with the complexity, which entangle complex system management and optimization by equipping their parts with *autonomic* facilities: self-configuration, self-healing, self-optimization, self-protection, and, as a combination of all, self-management [2]. Even if truly autonomic systems are years away, although in the nearer term, autonomic early autonomic system may threat the five "selves" as distinct aspects, with different solutions that address each one separately.

These aspects are currently addressed also within the Italian national project *Grid.it* concerning enabling technologies for Grid Computing. The *Grid.it* project is coordinated by CNR (by means of the ISTI–CNR institute and the university of Pisa, who are also SFIDA partners), and embodies the excellence among research centers in the area of Grid Computing.

International standardization bodies have a clear strategic relevance in Grid Computing, has shown by the interest of major IT manufacturers (IBM, SUN, Microsoft, HP), and by the growing relevance of the Global Grid Forum (GFF) activity (involving SFIDA partners, such as ISTI–CNR, ICAR–CNR, and University of Calabria). Major European strategic projects, such as the Network of Excellence CoreGRID, and the special support action GridCoord also involve SFIDA partners in key positions.

Within the *Grid.it* project, the ASSIST programming environment evolved from its very first version, only targeting workstation clusters, to the current version, targeting Grids and solving many critical problems related to expressive power, flexibility, inter-operability and efficiency. ASSIST (A Software development System based upon Integrated Skeleton Technology) [3, 4, 5, 6, 7] is a programming environment oriented to the development of parallel, distributed, high-performance, Grid-aware applications according to a unified approach. It provides the application programmer with a high-level language and its compiler, providing the need glue to effectively bridge different component technologies (e.g. CCM, Web Services). The compiler relies on an advanced run-time support dynamically enforcing Quality of Services constraints on the application by means of self-configuration and self-optimization features.

In project SFIDA, the e-services platform will be based on a Grid computing-based platform, stressing its specific features on **inter-operability**, **adaptability**, **dynamicity**, and **Quality of Service** in distributed applications. The generality of the Grid computing approach will allow to optimize the e-services platform according to different application needs and to different hardware/software systems. This approach will bring various benefits:

1 to establish a tight correlation between services composition and inter-operability and QoS indices
2 the configurability of the tools for knowledge management, either at data, processes and competences level, through distributed learning mechanisms
3 a complete independency from the underlying software and hardware platforms

2 The GRID-based platform

Currently, Grid applications are often designed according to a low-level approach (i.e., by relying on the Middleware services directly, possibly through a Grid portal) and, in many cases, they consist in single jobs or in limited forms of job composition (e.g. DAGs). Parallelism, where present, is limited inside single jobs, in a way that does not affect the overall workflow of the application (e.g. a job may be a MPI program). The result is that Grid applications are rarely highly optimized for high-performance.

As discussed in the previous section, our point of view is radically different. It is based on the definition and realization of a programming model with the following features:

1. applications are expressed as compositions of high performance components,
2. a uniform approach is followed for distributed and parallel programming: in general components exploit internal parallelism and are executed in parallel with each other,
3. the strategies to drive the dynamic adaptation of applications are expressed in the same high-level formalism of the programming model.

Figure 1 summarizes their interrelationships, which form the conceptual frameworks on which we found our research approach. A programming model implementing this set of features will be able to automatically implement proper and effective optimizations that depend on the kind of target architecture (nodes) considered, thus relieving the programmer from such a heavy and error prone duty. As both parallel execution of components and internally parallel components are taken into account, our "high performance" keyword does not (only) refer to the classical high performance computing framework, but rather it refers to a pervasive attention to the implementation of very efficient mechanisms supporting Grid-aware application execution.

Feature 1 is based on the proper exploitation of the component technology [8,9]. In our view, components are the basic mechanism to achieve compositionality by guaranteeing software interoperability and reuse.

Achieving high-performance in component technology is currently an important research issue [10, 11, 12, 13]. Currently, we are evaluating how the existing standards (CCA [14], Java Beans [15], CCM [16], Web Services [17,18]) can be assumed as starting points to define and realize a robust component-based high-performance programming model, that can be widely accepted and that is able to interoperate in many application areas.

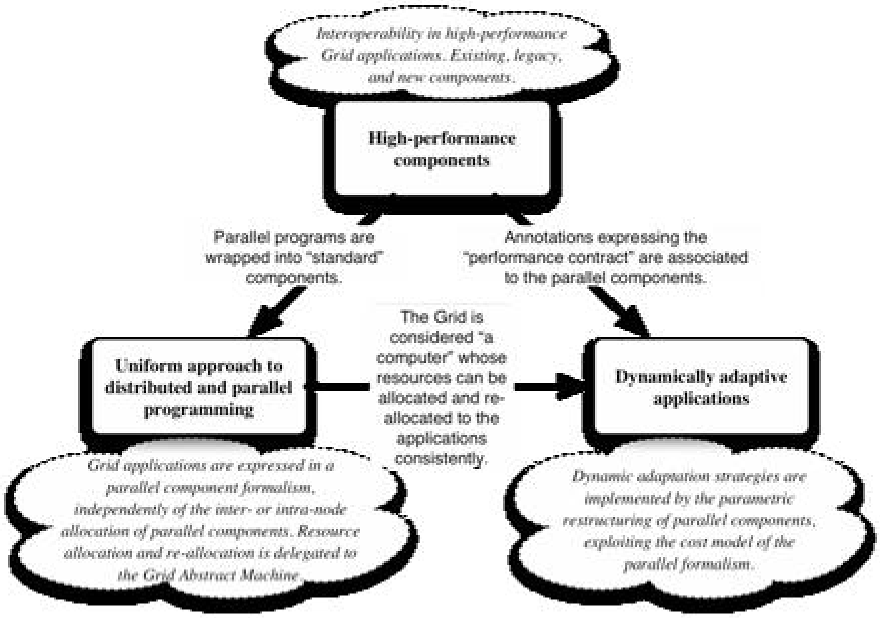

Figure 1. The conceptual framework for Grid-aware programming environments Grid-aware applications as compositions of high-performance components

ASSIST provides the abstraction of high-performance components and high-performance composition of components, independently of any commercial standard. This allows us to understand the basic features that high-performance components should possess, in particular from the point of view of computation structuring, parallelism exploitation and modularity. These features will be properly merged with one or more commercial standard, or their future high-perfomance versions, in order to achieve extensive interoperability and reuse. The merging of high-performance programming and component technology must allow the designer to structure the application as the proper composition of components, i.e. some of which may be already existing/legacy (possibly in binary form), and other ones are programmed from scratch (e.g. written in ASSIST) or as the combination of existing software into new parallel structures.

The current version of ASSIST (ASSIST 1.2) supports heterogeneity and the interoperability with several currant standards, in particular the *CORBA* interoperability [19, 20]: that is, an ASSIST program can act as a client of a CORBA server, but also ASSIST programs can be easily defined as, and automatically transformed into, CORBA servers invoked by any CORBA client. The performance penalties introduced by the CORBA infrastructure, with respect to pure ASSIST programs, are quite acceptable for many applications with reasonable granularity. Though referred to as an object-oriented approach, this experience proves that interoperability features can be merged into the ASSIST model, in order to design applications as composition of components, some of which are possibly parallel.

2.1 SFIDA project software architecture

The general architecture on which SFIDA relies on is sketched in Figure 2. The approach assumes the clear distinction between teams involved in business processes modeling and specialist in Grid technologies. Following this approach, the overall environment shall provide business experts a complete toolkit to model, design, develop, tune, and deploy effective solutions without any concern of the underlying software technology. They are required to focus on business model only, and finally to express required level of Quality of Service of the designed solution.

The Process Management Environment layer is build on top of a Grid platform, which in turn is composed of two tiers: development (programming model), and middleware.

The former (development) enable the development of complex software supporting process business and supply chain typical formalisms by means of high-performance component technology. This level enforces business processes interoperability and provides the needed tools to cope with Grid hardware performance unsteadiness by means of self-optimizing component implementation. Here, run-time adaptability, Quality of Service managers, and interoperability techniques (such as Web and Grid Services) are exploited.

The latter (middleware) enable the management of resources by means of standard technologies such as OGSA/OGSI platforms.

Eventually, the bottom layer (hosting framework) enforces back-end compatibility with on-the-self component technologies and their evolutions (e.g. J2EE, .NET).

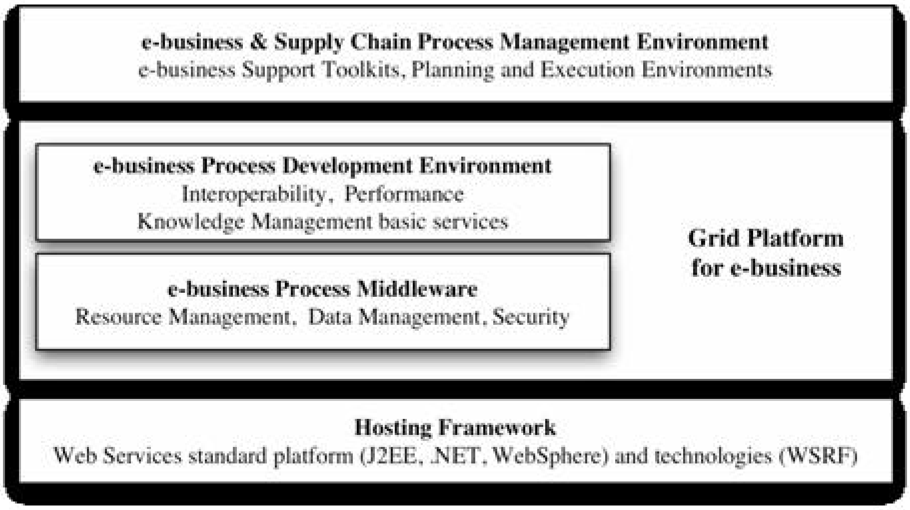

Figure 2. SFIDA project software architecture.

3 Results Validation

The project is expected to be completed by the end of year 2006: the last six months of work will be dedicated to the validation of the results, by testing them in a representative set of industrial test cases, and by analyzing the results through performance criteria.

3.1 Industrial Test Cases

The new business models, the GRID-based platform and the supporting environments will be validated through some important industrial cases. The most meaningful ones will be:

- **Retail Planning** in the consumer media sector: the case will consist in the establishment of the collaborative infrastructure between a brand owner and a network of point of sales, in order to improve the quality of the replenishments planning. In fact, currently there is not a real collaboration, since data are simply collected by the brand owner from the POS, who then performs all the computation in a centralized way. This cause serious performance problems due to the enormous amount of data to be processed. The adoption of a common GRID-based infrastructure will help in de-centralizing the computation, resulting in an overall improvement of the performances and in a better forecast of the sales, eventually in a better retail planning

- **Collaboration with Suppliers** in Product Design phase: this case will be focused in the automotive sector, aimed at creating a pro-active collaboration with second tier suppliers (SMEs). The process will be based on "call for sub-projects" in the scope of framework contracts, where call success criteria will be based also on suppliers capability to certify their processes and their competences in order to ensure the quality of the final product
- **Logistic Management** in a multi-plant environment. The goal of this test case will be the improvement of the distribution network, by directly involving the suppliers SMEs thanks to the common GRID-based infrastructure. In particular the platform will enable suppliers to have access to productive plants data that are necessary to perform the planning of their distribution activities (i.e. progress control), more specifically by making their systems inter-operable and synchronized.

3.2 Performance Criteria

The success criteria of SFIDA will be measured according to the different research areas of the project itself, in particular, regarding the IT tools, it should be considered two levels of abstractions:

4 the GRID-based platform: the objectives here are computational performances, inter-operability, scalability, re-usability, adaptability, security, robustness and fault tolerance, Quality of Service, hosting environments and resources independency

4 c-business supporting tools: the objectives at this level are to have a user-friendly environment, completeness of services and functionalities, configurability, adaptability to different models, generality of the solution.

4 Conclusions

During its 30 months work-plan, project SFIDA is expected to face problems that are felt as extremely relevant in the current e-business scenario, especially in the SMEs area, and to solve them thanks to innovative technologies and methodologies, such as Grid computing as the key enabling technology. The final test-cases will be a key part of the project, since they will constitute the proof that the solutions developed within the project will be really ready to be adopted by the industrial world, and not only a "research exercise".

References

1. I. Foster and C. Kesselman, editors, *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco, CA, 2003.
2. J. O. Kephart and D. M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003.
3. M. Vanneschi. The programming model of ASSIST, an environment for parallel and distributed portable applications. *Parallel Computing* 28(12), 2002, 1709-1732.
4. M. Aldinucci, M. Coppola, M. Danelutto, M. Vanneschi, C. Zoccolo. ASSIST as a Research Framework for High-Performance Grid Programming Environments. *Chapter in Grid Computing: Software environments and Tools*, Jose C. Cunha and Omer F. Rana (eds), Springer Verlag, 2005. Available as TR-04-09, Dept. of Computer Science, University of Pisa, February, 2004.
5. M. Aldinucci, S. Campa, P. Ciullo, M. Coppola, S. Magini, P. Pesciullesi, L. Potiti, R. Ravazzolo, M. Torquati, M. Vanneschi, C. Zoccolo. The Implementation of ASSIST, an Environment for Parallel and Distributed Programming. *In Proc. of Euro-Par2003: Parallel and Distributed Computing*, LNCS n. 2790, Springer, August 2003.
6. M. Aldinucci, S. Campa, P. Ciullo, M. Coppola, M. Danelutto, P. Pesciullesi, R. Ravazzolo, M. Torquati, M. Vanneschi, C. Zoccolo. A framework for experimenting with structured parallel programming environment design. *In Proc. of Intl. Conference ParCo2003: Parallel Computing*, Dresden, Germany, September 2003.
7. M. Aldinucci, S. Campa, M. Coppola, M. Danelutto, D. Laforenza, D. Puppin, L. Scarponi, M. Vanneschi, C. Zoccolo. Components for high performance Grid programming in the *Grid.it* Project. *In Proc. of Intl. Workshop on Component Models and Systems for Grid Applications*. Held in conjunction with ACM ICS 2004. Saint-Malo, France, July 2004
8. G. Beneken, U. Hammerschall, M. Broy, M.V. Cengarle, J. Jürjens, B. Rumpe, and M. Schoenmakers. Componentware - State of the Art 2003. Background paper for the *Understanding Components Workshop*, Venice, 7-9 October 2003.
9. C. Szyperski. Component Software: Beyond Object-Oriented Programming. Addison-Wesley, 2002.
10. R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. Smolinski. Toward a common component architecture for high performance scientific computing. *Proc. 8th High Performance Distributed Computing (HPDC'99)*, 1999
11. K. Keahey, D. Gannon. PARDIS. a parallel approach to CORBA. *6th IEEE Int. Symp. on High Performance Distributed Computing*, 1997, 31-39.
12. C. Perez, T. Priol, A. Ribes, PaCO++: a parallel object model for high performance distributed systems. *In Distributed Object and Component-based Software Systems*, Hawaii Int. Conf. On System Sciences, IEEE (2004).
13. T. Priol. Programming the Grid with Distributed Objects. *In Proc. of Workshop on Performance Analysis and Distributed Computing* (PACD 2002), 2002

14. *CCA Forum home page*. http://www.cca-forum.org/
15. Sun M.sys. *JavaBeans home page*. http://java.sun.com/products/javabeans.
16. *The CORBA & CCM home page*. http://ditec.um.es/~dsevilla/ccm/
17. I. Foster et al. Grid Services for Distributed System Integration. *Computer*, vol. 35, n. 6, 2002, 37-46.
18. F. Burbera et al. Unraveling the Web Services Web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, vol. 6, n. 2, 2002, 86-93.
19. M. Aldinucci, S. Campa, M. Coppola, S. Magini, P. Pesciullesi, L. Potiti, R. Ravazzolo, M. Torquati, C. Zoccolo. Targeting heterogeneous architectures in ASSIST: experimental results. *In Proc. of the Intl. Conference EuroPar2004: Parallel and Distributed Computing*, Pisa, Italy, LNCS n. 3149, Sept. 2004.
20. S. Magini, P. Pesciullesi, C. Zoccolo. Parallel Software Interoperability by Means of CORBA in the ASSIST Programming Environment. *In Proc. of the Intl. Conference EuroPar2004: Parallel and Distributed Computing*, Pisa, Italy, LNCS n. 3149, Sept. 2004.
21. EU research projects: IDEAS (IST-2001-37368), UEML (IST-2001-34229), ATHENA (IST IP-507849) and INTEROP (IST NoE-508011)
22. EU project EXPIDE (IST-1999-29105)
23. EU project COMPANION (IST-2001-37493)