Grid Technologies and c-Business for SMEs

Marco ALDINUCCI[1], Marco VANNESCHI[2], Matteo VILLA[3]
[1]*Inst. of Information Science and Technologies – CNR, Via Moruzzi 1, Pisa, Italy*
[2]*Dept. of Computer Science – University of Pisa – Largo B. Pontecorvo 3, Pisa, Italy*
[3]*TXT e-Solutions Spa, via Frigia 27, Milano, 20126, Italy*
Tel: +39 02 257711, Email: matteo.villa@txt.it

Abstract: We describe the objectives of project SFIDA, aiming at developing a Grid-based interoperable platform able to support next generation applications specifically addressing the needs of SMEs. We sketch the architecture of the platform under development in SFIDA, which will support componentization (e-services), intelligence (mining), collaboration (c-business), and customer business-processes orientation concepts on top of ASSIST, a Grid-aware high-level programming environment. The SFIDA project outcomes will be validated on Supply Chain Management applications matching various typical industrial cases, spanning from automotive, textile, food, white goods, and media retail. Finally we show what business benefits it is expected to bring.

1. Introduction

This paper describes the objectives and the first results obtained from project SFIDA, and Italian National research project (co-funded by the Ministry for University and Research) started in January 2005, and shows how they are still valid and applicable at an International scale.

Project SFIDA is driven by a long-term view in which SMEs belonging to industrial districts and supply chains 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 [11].

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.

2. Objectives

Project SFIDA is developing an innovative enabling platform to promote a new collaborative business model to SMEs belonging to industrial districts and supply chains. Such model will be based on services provision to support collaborative industrial processes. More in detail, SFIDA is developing:

1. The technology for a multi-layer development environment, to implements the process-driven paradigm for e-business.
2. Tools for supporting processes life cycle: ranging from processes modelling, to planning, execution and performances monitoring.

The integration of these two technologies will form the enabling platform for supporting c-Business for SMEs. Such platform will be based on Grid computing paradigm, according to its more recent evolutions, emphasizing in particular aspects such as interoperability, adaptability, dynamicity and Quality of Service (QoS) for distributed applications, integrating standard instruments such as Web Services, and supported by commercial hosting environments for distributed components. The generic nature of the Grid computing approach allows also to customise the functionalities of the platform according to specific requirements and to different hardware-software technologies.

3. Methodology

SFIDA is planning to support the collaborative business processes of SMEs, having identified the following steps as necessary in order to achieve the "collaborate to compete" objective ("c-Business") [10]:

1. to focus on strategic competences (organizational, 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 [9] 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 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.

4. Technology Description

The idea in SFIDA is to define a new IT solutions development paradigm based on a new IT service-based model oriented to collaborative processes. 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.

In particular, Grid technology is at the base of the success of the entire project. Indeed, Grid technology addresses the problem of the “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations”, where the sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering [12]. At the fabric level, a Grid system is a geographically distributed collection of possibly parallel, interconnected platforms that all run some kind of common Grid middleware (e.g. Globus services). In general such platforms are characterized by heterogeneity of nodes, and by dynamicity in resource management and allocation.

In c-Business processes the competition is no longer between individual companies, but between value chains that aim to deliver products and services more cost effectively and profitably. This requires an operating environment in which the partners act together as a single, unified entity, with transactions passing seamlessly between them. Grid computing technologies aim at dynamically bridging a suitable subset of resources and services to support the execution of large-scale applications. Therefore, Grid naturally represents the technology infrastructure that tears down organizational barriers and pulls together business units to work as one.

Current Grid-aware applications are developed as custom solutions on existing software infrastructures, such as Globus, by developers who are experts on Grid software implementation. Although some useful scientific applications have been produced this way, this homemade approach may hardly support the additional complexity of enterprise applications realizing collaborative processes. These applications will be developed by different companies or outsourced, they will have to work flawlessly and uninterrupted, and very often they should provide a contractually specified QoS in terms of performance.

Therefore, we envision a layered, high-level programming model for the Grid, which is currently pursued by several research initiatives and programming environments, such as ASSIST, eSkel, GrADS, ProActive, Ibis [17,7,16,6,15]. In such an environment, most of the Grid specific efforts are moved from programmers to Grid tools and run-time systems. Thus, the programmers have only the responsibility of organizing the application specific code, while the programming tools (i.e. the compiling tools and/or the run-time systems) deal with the interaction with the Grid, through collective protocols and services.

In particular, project SFIDA inherit the experience made at University of Pisa during the design and the development of the ASSIST programming environment. It includes a coordination language enabling the programmer to describe the logical interaction among software components as in a graph. Both ASSIST native components and several kinds of legacy components may be mixed in the same graph (i.e. an application). ASSIST components are binary units of deployments, wrapping sequential or parallel code, exposing services to other components (provided ports) and requiring a specific context to properly work (used ports). Legacy components may be developed with standard technologies, such as Corba/CCM, Java Beans, DCOM [5,14]. Components interactions may be realized via RPC-like mechanisms and one-way data streams (or event channels), possibly implemented via standard SOAP/WebServices technology. These interactions (wiring, communication) may be established by using both ASSIST native technology or legacy code mechanisms. The ASSIST environment provides the developer with a toolkit to automatically generate suitable adaptors to wire ASSIST to legacy components, and to re-generate ASSIST components [1,3].

ASSIST components also provide non-functional ports, i.e. mechanisms to control components run time behavior with respect to QoS requirements. Through these ports a component may dynamically accept a user-defined QoS contract, i.e. a XML file describing performance and fault-tolerance features the component is required fulfil. QoS contract is transparently managed at run time by the ASSIST framework, which dynamically controls parallelism degree, processes and data mapping and remapping, and possibly their replication [3]. In other words, ASSIST components exhibits a self-optimizing, self-healing autonomic behaviour [13]. Moreover, components may be collectively coordinated by a hierarchy of Application Managers, i.e. components dedicated to QoS contract management ensuring QoS proprieties at the application level. We experimentally shown that both stateless and stateful computations may be suitably reconfigured to fulfil several kinds of QoS contracts, and that these reconfigurations might have negligible cost enabling fine grain control on the application dynamic evolution [4].

We believe that ASSIST underlying technology is an enabling technology for next generation e-services platform because:

- The high-level nature of the ASSIST programming environment enforces the minimization of developing cost by enabling the static reconfiguration of the application to target different customer scenarios, by effectively decoupling developed code from both hardware platform and optimization (due to application QoS constraints). Since the ASSIST compiler is designed to enforce efficient execution on the Grid, it also enforces the same propriety on proper subsets of the Grid, ranging from large SMP to enterprise COWs. This enables the application provider to exploit a good performance/cost ratio over a broad class of hardware platforms through automatic re-generation and recompilation [2,3].
- ASSIST components may seamlessly interoperate with other standard component technologies, thus enabling code outsourcing and reuse [14]. In many cases ASSIST toolkits may provide multi-site deployment of both ASSIST and legacy components thought standard middleware. It also provides middleware extensions to deal with batch schedulers and multi-site co-allocation [8].
- It includes an advanced run-time support for developed code able to seamlessly adapt the application structure to the current platform status. It transparently manages performance degradations (and faulty events, in the near future) of the underlying platform, which should be considered the standard behaviour of a large-scale distributed platform.

In project SFIDA, the e-services platform will be based on a Grid computing-based platform, stressing its specific features on interoperability, adaptability, dynamicity, and QoS 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 interoperability 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.

5. Developments

The general architecture on which SFIDA relies on is sketched in Figure 1. The approach assumes the clear distinction between teams involved in business processes modelling 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 QoS 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: e-Business Process Development Environment (programming model), and the e-Business Process Middleware.

The former (development environment) enables 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, QoS 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).

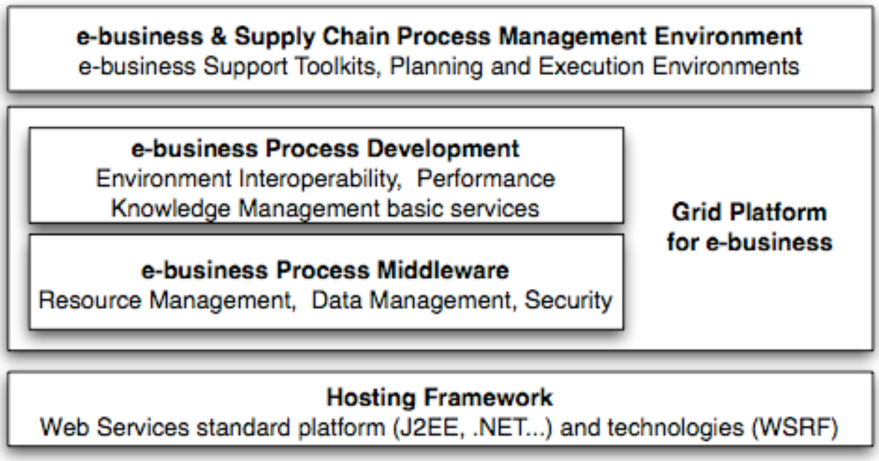

Figure 1: System Architecture.

6. Results

Though the project is started recently, some results have been achieved, even if actual demonstrator will be available only around mid 2006.

In particular, results have been achieved at the Grid Platform level: 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, interoperability and efficiency [1,2,3,4,8,14,17].

ASSIST provides the abstraction of high-performance components and high-performance composition of components, independently of any commercial standard. These features will be properly merged with one or more commercial standard, or their future high-performance versions, in order to achieve extensive interoperability and reuse. Nevertheless, the current version (ASSIST 1.3) supports heterogeneity and the interoperability with several current standards, in particular the CORBA, CCM, and Web Services interoperability [2,9,14]. As an example, ASSIST program can automatically transformed into a CORBA client and/or server. The performance penalties introduced by the CORBA infrastructure, with respect to pure ASSIST programs, are quite acceptable for many applications with reasonable granularity suggesting that interoperability features can be merged into the ASSIST model.

Progresses are being achieved at the Process Management level (see Figure 1), once completed will enable, together with the Grid platform, to start experimenting in different business cases

7. Business Benefits

Business benefits arising from the solutions developed will be demonstrated in a set of different industrial cases, spamming from retail management in the consumer media sector, to distribution chain optimisation in the white goods sector, to collaborative product design in the automotive sector and to supply chain management in the textile sector. It is important to stress that such cases do not have any regional specificity: they represent common business situations that can be found in European supply chains.

Retail Planning: the consumer media sector is probably the most meaningful one: the case consists in the establishment of the collaborative infrastructure between a brand owner and a network of point of sales (POS), in order to improve the quality of the replenishments planning. Currently, all data are gathered by the central brand owner from all the various POS and re-sellers (see Figure 2); the bran owner then performs all the computation necessary for producing a replenishment plan in a centralized way, by adopting a very expensive supercomputer, which is stressed at 100% of its capabilities. This is causing serious performance problems due to the enormous amount of data to be processed: batch executions last for all night time, and there is no further time available for improving algorithms accuracy, which would require more computational time.

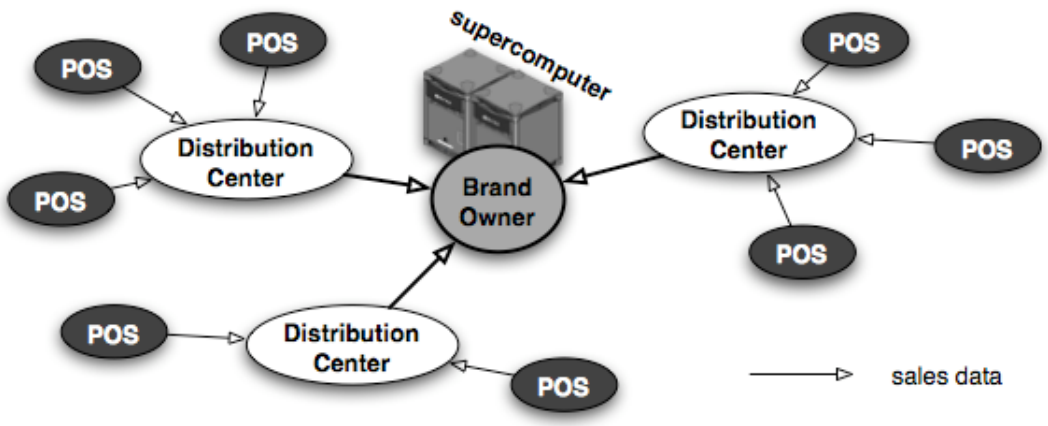

Figure 2: Retail Planning: the current situation.

The adoption of a common Grid-based infrastructure will enable the development of a de-centralized solution, that could take advantage of computational power available at every node to the Grid, thus enabling re-sellers and POS to co-operate in the computation (c-Business), resulting in an overall improvement of the performances and in a better forecast of the sales, eventually in a better retail planning, and above all, avoiding the necessity of the central super-computer. Figure 3 shows how in a Grid-based architecture every actor's computational power can be shared with all the other ones.

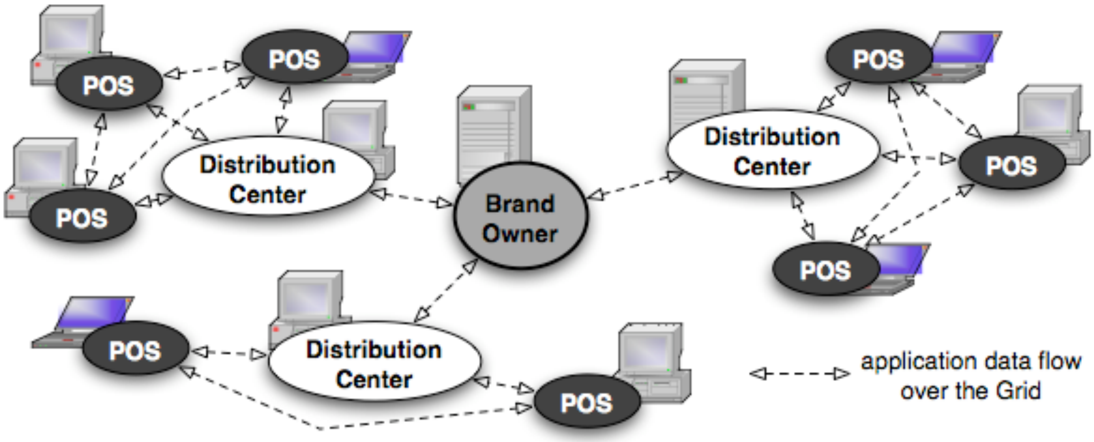

Figure 3:Retail Planning: a Grid-enabled solution.

This scenario shows benefits in terms of reduction of investments, performances improvement, and QoS.
Other business cases will be developed to highlight benefits:

- 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. More in particular, suppliers will make use of the tools for modelling and sharing their competences. This scenario will show benefits in terms of trusted collaboration due to a commonly shared and trusted infrastructure.
- 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.

8. Conclusions

The project is expected to be completed by mid of 2007: 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”. More in particular, the project feels as extremely important the fact of proving the adaptability of its technology to other industrial contexts, in order to let the e-business world fully take advantage of new Grid computing paradigms.

References

[1] M. Aldinucci, S. Campa, M. Coppola, M. Danelutto, D. Laforenza, D. Puppin, L. Scarponi, M. Vanneschi, and C. Zoccolo. Components for high performance Grid programming in Grid.it. In *Proc. of the Workshop on Component Models and Systems for Grid Applications*, CoreGRID series. Springer, 2005.
[2] M. Aldinucci, S. Campa, M. Coppola, S. Magini, P. Pesciullesi, L. Potiti, R. Ravazzolo, M. Torquati, and C. Zoccolo. Targeting heterogeneous architectures in ASSIST: Experimental results. In *Proc of 10th Intl Euro-Par 2004*, vol. 3149 of *LNCS* pages 638–643, Pisa, Italy, 2005. Springer.
[3] M. Aldinucci, M. Coppola, S. Campa, M. Danelutto, M. Vanneschi, and C. Zoccolo. Structured implementation of component based Grid programming environments. In *Dagstuhl Seminar Future Generation Grid 2004*, CoreGRID series. Springer, 2005. To appear.
[4] M. Aldinucci, A. Petrocelli, E. Pistoletti, M. Torquati, M. Vanneschi, L. Veraldi, and C. Zoccolo. Dynamic reconfiguration of grid-aware applications in ASSIST. In *Proc of 11th Intl Euro-Par 2005: Parallel and Distributed Computing*, *LNCS*, Lisboa, Portugal, August 2005. Springer. To appear.
[5] 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. In *Proc of the Conference on High Performance Distributed Computing,* Redondo Beach, CA, USA, 1999.
[6] F. Baude, D. Caromel, and M. Morel. On hierarchical, parallel and distributed components for Grid programming. In *Proc. of the Workshop on Component Models and Systems for Grid Applications*, CoreGRID series. Springer, 2005.
[7] M. Cole. Bringing Skeletons out of the Closet: A Pragmatic Manifesto for Skeletal Parallel Programming. Parallel Computing, 30(3):389–406, 2004.
[8] M. Danelutto, M. Vanneschi, C. Zoccolo, N. Tonellotto, S. Orlando, R. Baraglia, T. Fagni, D. Laforenza, A. Paccosi. HPC application execution on Grids. In *Dagstuhl Seminar Future Generation Grid 2004*, CoreGRID series. Springer Verlag, 2005. To appear.
[9] EU project COMPANION (IST-2001-37493)
[10] EU projects: EXPIDE (IST-1999-29105), COMPANION (IST-2001-37493).
[11] EU research projects: IDEAS (IST-2001-37368), UEML (IST-2001-34229), ATHENA (IST IP-507849) and INTEROP (IST NoE-508011)
[12] I. Foster, and C. Kesselmann, editors. The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 2003.
[13] J. O. Kephart, and D. M. Chess. The vision of autonomic computing. IEEE Computer 36(1):41-50, 2003.
[14] S. Magini, P. Pesciullesi, C. Zoccolo. Parallel software interoperability by means of CORBA in the ASSIST programming environment. In *Proc of 10th Intl Euro-Par 2004*, vol. 3149 of *LNCS* pages 679–688, Pisa, Italy, August 2004. Springer.
[15] R. V. van Nieuwpoort, J. Maassen, G. Wrzesinska, R. Hofman, C. Jacobs, T. Kielmann, and H. E. Bal. Ibis: a flexible and efficient Java-based Grid programming environment. *Concurrency & Computation: Practice & Experience*, 17(7-8):1079-1107, June-July 2005.
[16] S. Vadhiyar and J. Dongarra. Self adaptability in Grid computing. *Concurrency & Computation: Practice & Experience*, 17(2-4):235-257, 2005
[17] Vanneschi, M.: The programming model of ASSIST, an environment for parallel and distributed portable applications. Parallel Computing 28(12):1709–1732, Dec. 2002.