How Distributed Data Mining Tasks can Thrive as Knowledge Services

Domenico Talia [1,2] and **Paolo Trunfio** [1]
[1] DEIS - University of Calabria
[2] ICAR-CNR
Rende (CS), Italy
{talia,trunfio}@deis.unical.it

Computer science applications are becoming more and more network centric, ubiquitous, knowledge intensive, and computing demanding. This trend will result soon in an ecosystem of pervasive applications and services that professionals and end-users can exploit everywhere. Recently, collections of IT services and applications, such as Web services and Cloud computing services, became available opening the way for accessing computing services as public utilities, like water, gas and electricity.

Key technologies for implementing that perspective are Cloud computing and Web services, semantic Web and ontologies, pervasive computing, P2P systems, Grid computing, ambient intelligence architectures, data mining and knowledge discovery tools, Web 2.0 facilities, mashup tools, and decentralized programming models. In fact, it is mandatory to develop solutions that integrate some or many of those technologies to provide future knowledge-intensive software utilities. The Grid paradigm can represent a key component of the future Internet, a cyber infrastructure for efficiently supporting that scenario.

Grid and Cloud computing are evolved models of distributed computing and parallel processing technologies. The Grid is a distributed computing infrastructure that enables coordinated resource sharing within dynamic organizations consisting of individuals, institutions, and resources. In the area of Grid computing a proposed approach in accordance with the trend outlined above is the Service-Oriented Knowledge Utilities (SOKU) model [10] that envisions the integrated use of a set of technologies that are considered as a solution to information, knowledge and communication needs of many knowledge-based industrial and business applications. The SOKU approach stems from the necessity of providing knowledge and processing capabilities to everybody, thus supporting the advent of a competitive knowledge-based economy. Although the SOKU model is not yet implemented, Grids are increasingly equipped with data management tools, semantic technologies, complex workflows, data mining features and other Web intelligence approaches. Similar efforts are currently devoted to develop knowledge and intelligent Clouds. These technologies can facilitate the process of having Grids and Clouds as strategic components for supporting pervasive knowledge intensive applications and utilities.

Grids were originally designed for dealing with problems involving large amounts of data and/or compute-intensive applications. Today, however, Grids enlarged their horizon as they are going to run business applications supporting consumers and end-users [3]. To face those new challenges, Grid environments must support adaptive knowledge management and data analysis applications by offering resources, services, and decentralized data access mechanisms. In particular, according to the service-oriented architecture (SOA) model, data mining tasks and knowledge discovery processes can be delivered as services in Grid-based infrastructures.

Through a service-based approach we can define integrated services for supporting distributed business intelligence tasks in Grids. Those services can address all the aspects that must be considered in data mining and in knowledge discovery processes such as data selection and transport, data analysis, knowledge models representation and visualization. We worked in this direction for providing Grid-based architectures and services for distributed knowledge discovery such as the Knowledge Grid [4,6], the Weka4WS toolkit [13], and mobile Grid services for data mining [12].

Here we describe a strategy and a model based on the use of services for the design of distributed knowledge discovery services and discuss how Grid frameworks, such those mentioned above, can be developed as a collection of services and how they can be used to develop distributed data analysis tasks and knowledge discovery processes using the SOA model.

On Grids and Data Mining

The main aim of Grid computing is to give organizations and application developers the ability to create distributed computing environments that can utilize computing resources on demand. Grid computing can leverage the computing power of a large numbers of server computers, desktop PCs, clusters and other kind of hardware. Therefore, it can help to increase the efficiency and reduce the cost of computing networks by decreasing data processing time and optimizing resources and distributing workloads, thereby allowing users to achieve much faster results on large operations and at lower costs.

Data mining algorithms and knowledge discovery applications demand for both compute and data management facilities. Therefore the Grid is a good candidate offering a computing and data management infrastructure for supporting decentralized and parallel data analysis. The opportunity of utilizing Grid-based data mining systems, algorithms and applications is interesting to users wanting to analyze data distributed across geographically dispersed heterogeneous hosts. For example, Grid-based data mining would allow corporate companies to distribute compute-intensive data analysis among a large number of remote resources. At the same time, it can lead to new algorithms and techniques that would allow organizations to mine data where it is stored. This is in contrast to the practice of selecting data and transferring it into a centralized site for mining. Centralized analysis is often difficult to perform because data is becoming increasingly larger, geographically dispersed, and because of security and privacy considerations.

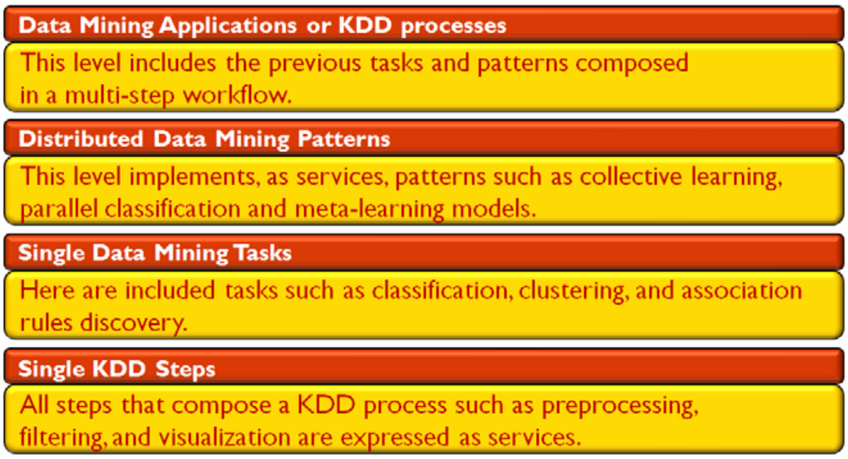

Data Mining Applications or KDD processes
This level includes the previous tasks and patterns composed in a multi-step workflow.
Distributed Data Mining Patterns
This level implements, as services, patterns such as collective learning, parallel classification and meta-learning models.
Single Data Mining Tasks
Here are included tasks such as classification, clustering, and association rules discovery.
Single KDD Steps
All steps that compose a KDD process such as preprocessing, filtering, and visualization are expressed as services.

Figure 1. Four classes/levels of services to facilitate the implementation of distributed data mining tasks and KDD processes.

Some research frameworks currently exist for deploying distributed data mining applications in Grids. A subset of them are general environments supporting execution of data mining tasks on machines that belong to a Grid, others implement single mining tasks for specific applications that have been "gridfied," and some others are implementations of single data mining algorithms. Considering the general purpose environments, besides the systems we describe here, a few of those that exploit the SOA model for supporting Grid-based data mining service are Discovery Net [1], Gates [5], GridMiner [2], DataMiningGrid [11], and the Services Oriented Framework proposed by Wang et al. [14].

As the Grid becomes a well accepted computing infrastructure in science and industry, it is necessary to provide general data mining services, algorithms, and applications that help analysts, scientists, organizations, and professionals to leverage Grid capacity in supporting high-performance distributed computing for solving their data mining problems in a distributed way.

The Grid community has adopted the Open Grid Services Architecture (OGSA) as an implementation of the SOA model within the Grid context. In OGSA every resource is represented as a Web service that conforms to a set of conventions and supports standard interfaces. OGSA provides a well defined set of Web service interfaces for the development of interoperable Grid systems and applications. The WS-Resource Framework (WSRF) has been adopted as an evolution of early OGSA implementations. WSRF defines a family of technical specifications for accessing and managing stateful resources using Web services. The composition of a Web service and a stateful resource is termed as WS-Resource.

The possibility to define a state associated to a service is the most important difference between WSRF-compliant Web services, and pre-WSRF ones. This is a key feature in designing Grid applications, since WS-Resources provide a way to represent, advertise, and access properties related to both computational resources and applications. Statefulness is important in complex applications where is needed to store (in the service state) the results of previous stages of computation, because they must be used in the following ones. This is particularly important in distributed data mining applications that generally are complex, multi-stage, and long running.

Data Mining Grid Services

Through WSRF it is possible to define basic services for supporting distributed data mining tasks in Grids. Those services can address all the aspects that must be considered in knowledge discovery processes from data selection and transport, to data analysis, knowledge model representation and visualization. Some application examples that can be implemented by using data mining services are: the analysis of remotely located genetic data repositories, the mining of financial databases that a bank stores in different branches in a country or a continent for rule discovery, and the classification of Web contents (text, images, and videos) that cannot be stored on a single site.

According to this approach we can provide services for distributed data mining that can be used at different levels from single operations on data to distributed data mining patterns and complete KDD (knowledge discovery in databases) processes running as service-based workflows on a geographically remote set of machines. This can be done by designing four levels of data analysis services (see Figure 1) corresponding to

1. **Single KDD steps;**
2. **Single data mining tasks;**
3. **Distributed data mining patterns;**
4. **Data mining applications or KDD processes.**

It is worth noticing that services provided at one level can be used to implement services in other levels, thus, referring to Figure 1, single step services can be used to implement single data mining tasks or distributed data mining patterns and all these three classes of services can be exploited to develop service-oriented KDD applications. This incremental approach avoids the re-implementation of already available operations, tasks or patterns and provides a collection of services that can be viewed as a "distributed mining engine."

This collection of data mining services constitutes an **Open Service Framework for Grid-oriented Data Mining**. This framework can allow developers to design distributed KDD processes as a composition of single services that are available over Grids. At the same time, those services should exploit other basic Grid services for data transfer and management such as *Reliable File Transfer (RFT*), *Replica Location Service (RLS)*, *Data Access and Integration (OGSA- DAI)* and *Distributed Query Processing (OGSA-DQP)*. Moreover, distributed data mining algorithms can optimize the exchange of data needed to develop global knowledge models based on concurrent mining of remote datasets.

By exploiting this open framework for service-oriented data mining in Grids, Clouds and dynamic distributed infrastructures it is possible to develop data mining services accessible every time and everywhere. This solution can support

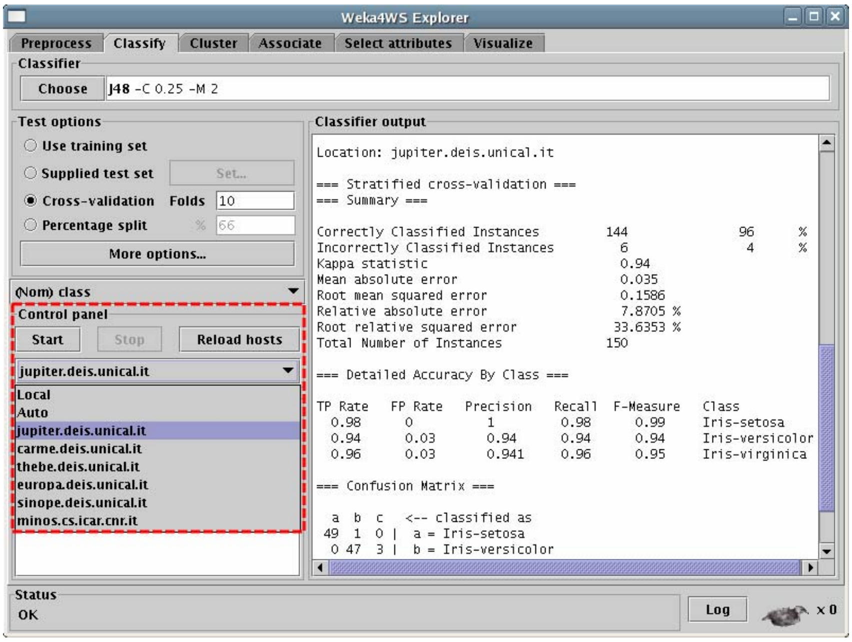
- Service-based distributed data mining applications;
- Data mining services for virtual organizations; and
- Distributed data analysis services on demand.

Therefore, we have a sort of knowledge discovery eco-system composed of a large numbers of decentralized data analysis services that will help users to face the availability of massive amounts of data both in business and science.

This approach also facilitates data privacy preserving and prevents disclosure of data beyond the original sources because it is based on the idea of keeping data at the owner site not requiring to move data to different locations for its analysis. This data-centric approach is mainly based on moving computation to data rather than moving data to computation. Finally, basic Grid mechanisms for handling security, trustiness, monitoring, and scheduling distributed tasks can be used to provide efficient implementation of high-performance distributed data analysis.

Service-based Data Mining Frameworks

After introducing the service-oriented approach for the implementation of distributed data mining on Grids, in the rest of the paper we shortly describe three systems that we developed according to that service-based model. Those systems show the feasibility of the proposed approach.



Classifier: J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

jupiter.deis.unical.it

```
Location: jupiter.deis.unical.it

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        144               96      %
Incorrectly Classified Instances        6                4      %
Kappa statistic                         0.94
Mean absolute error                     0.035
Root mean squared error                 0.1586
Relative absolute error                 7.8705 %
Root relative squared error            33.6353 %
Total Number of Instances             150

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.98      0          1         0.98      0.99     Iris-setosa
  0.94      0.03       0.94      0.94      0.94     Iris-versicolor
  0.96      0.03       0.941     0.96      0.95     Iris-virginica

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
```

Status: OK

Figure 2. A screenshot of the Weka4WS GUI. The red box highlights the panel allowing a user to automatically or manually select the Grid node where to run a data mining task.

Weka4WS

Weka [15] is a widely used open source data mining toolkit that runs on a single machine. *Weka4WS* [13] extends the Weka toolkit by implementing a distributed framework that supports data mining in WSRF-enabled Grids. Weka4WS integrates Weka and the WSRF technology for running remote data mining algorithms and managing distributed computations as workflows. The Weka4WS user interface supports the execution of both local and remote data mining tasks. On a Grid computing node, a WSRF-compliant Web service is used to expose all the data mining algorithms provided by the Weka library.

Weka4WS has been developed by using the Java WSRF library provided by *Globus Toolkit (GT4)* [8]. All involved Grid nodes in Weka4WS applications use the GT4 services for standard Grid functionality, such as security, data management, and so on. We distinguish those nodes in two categories on the basis of the available Weka4WS components: *user nodes* that are the local machines providing the Weka4WS client software, and *computing nodes* that provide the Weka4WS Web services allowing for the execution of remote data mining tasks. Data can be located on computing nodes, user nodes, or third-party nodes (e.g., shared data repositories). If the dataset to be mined is not available on a computing node, it is automatically copied by means of the GT4 data management services. User nodes include three components: *Graphical User Interface (GUI)*, *Client Module (CM)*, and *Weka Library (WL)*. The GUI is an extension of the Weka GUI that supports the execution of both local and remote data mining tasks and the design of knowledge discovery workflows to be run on a distributed infrastructure. Local tasks are executed by directly invoking the local WL, whereas remote tasks are executed through the CM, which operates as an intermediary between the GUI and Web services on remote computing nodes. Through the GUI, a user can start the execution either locally or on a (automatically-chosen or specific) remote Grid node (see Figure 2). Each task in the GUI is managed by an independent thread. Therefore, a user can start multiple data mining tasks in parallel on different computing nodes, this way taking full advantage of the distributed Grid environment. Whenever the output of a data mining task has been received from a remote computing node, it is visualized in a pane of the GUI.

Talia et al. [13] presents a performance analysis of the execution mechanisms described here. The experimental results demonstrate the low overhead of the WSRF Web service invocation mechanisms with respect to the execution time of data mining algorithms on large datasets, and confirms the efficiency of the WSRF framework as a means for executing data mining tasks on remote resources. By exploiting such mechanisms, Weka4WS provides an effective way to perform compute-intensive distributed data analysis in Grid environments (it can be downloaded from http://grid.deis.unical.it/weka4ws).

The Knowledge Grid

The *Knowledge Grid* [4,6] is a Grid services-based environment providing knowledge discovery services for a wide range of high-performance distributed applications. It offers users high-level abstractions and a set of services by which they can integrate Grid resources to support all the phases of the knowledge discovery process.

The Knowledge Grid supports such activities by providing mechanisms and higher level services for searching resources (data, algorithms, and so on), representing, creating, and managing knowledge discovery processes, and for composing existing data services and data mining services in a structured manner, hence allowing designers to plan, store, share and re-execute their workflows as well as managing their results.

The Knowledge Grid architecture is composed of a set of services divided in two groups: the *Resource Management Services* and the *Execution Management Services* (see Figure 3). The first group includes services for data access (DAS), tools and algorithms access (TAAS) and a knowledge directory service (KDS) for metadata management, while the second group includes services supporting a user in designing and executing KDD applications (EPMS and RAEMS), as well as delivering and visualizing the mining results (RPS). Both service groups make use of repositories that provide information about resource metadata (KMR), execution plans (KEPR), and knowledge models (KBR) obtained as result of knowledge discovery applications.

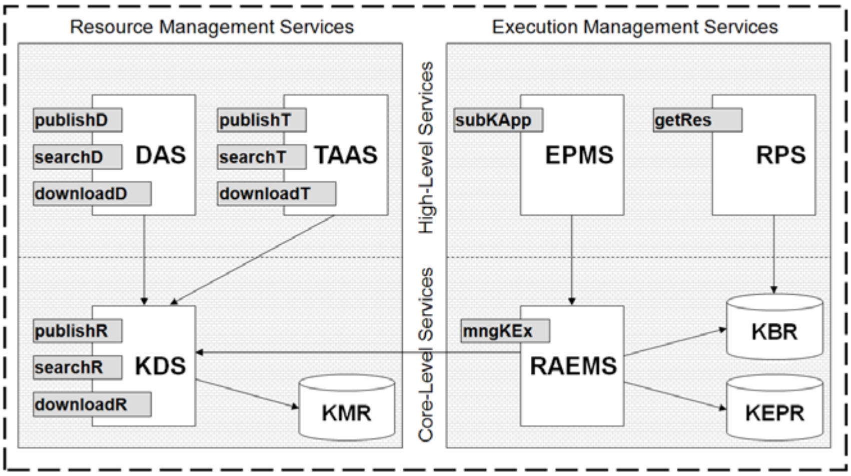


Figure 3. Services composing the Knowledge Grid system.

In the Knowledge Grid environment, discovery processes and applications are represented as workflows that a user may compose using both concrete and abstract Grid resources. Knowledge discovery workflows are defined using a visual interface that shows resources (data and tools) to the user and offers mechanisms for integrating them in a workflow. Information about single or collections of resources is stored using an XML-based notation that represents a workflow (called execution plan in the Knowledge Grid terminology) as a data-flow graph of nodes, each one representing either a data mining service or a data transfer service. The XML representation allows the knowledge discovery workflows to be easily validated, shared, translated into executable scripts, and stored for future executions.

Mobile Data Mining Services

The availability of client programs on mobile devices that can invoke the remote execution of data mining tasks and show the mining results is a significant added value for nomadic users and organizations that need to perform analysis of data stored in repositories far away from the site where end-users are working, allowing them to generate knowledge regardless of their physical location. This section shortly discusses pervasive data mining of databases from mobile devices through the use of Grid Services. By implementing mobile Grid Services we allow remote users to execute data mining tasks on a Grid from a mobile phone or a PDA and receive on those devices the results of a data analysis task. The system [12] is based on the client/server architecture shown in Figure 4. The architecture includes three types of components:

- *Data providers:* applications that generate the data to be mined;
- *Mobile clients:* the applications that require the execution of data mining computations on remote data;
- *Mining servers:* server nodes used for storing the data generated by data providers and for executing the data mining tasks submitted by mobile clients.

Each mining server exposes its functionalities through two Web services: the *Data Collection Service (DCS)* and the *Data Mining Service (DMS)*. The DCS is invoked by data providers to store data on the server. The DMS is invoked by mobile clients to perform data mining tasks. Its interface defines a set of operations that allow for:

- obtaining the list of the available data sets and algorithms,
- submitting a data mining task,
- getting the current status of a computation, and
- getting the result of a given task.

The DMS can perform several data mining tasks from a subset of the algorithms provided by the Weka4WS systems. When a data mining task is submitted to the DMS, the appropriate algorithm of the Weka library is invoked on a Grid node to analyze the local data set specified by the mobile client.

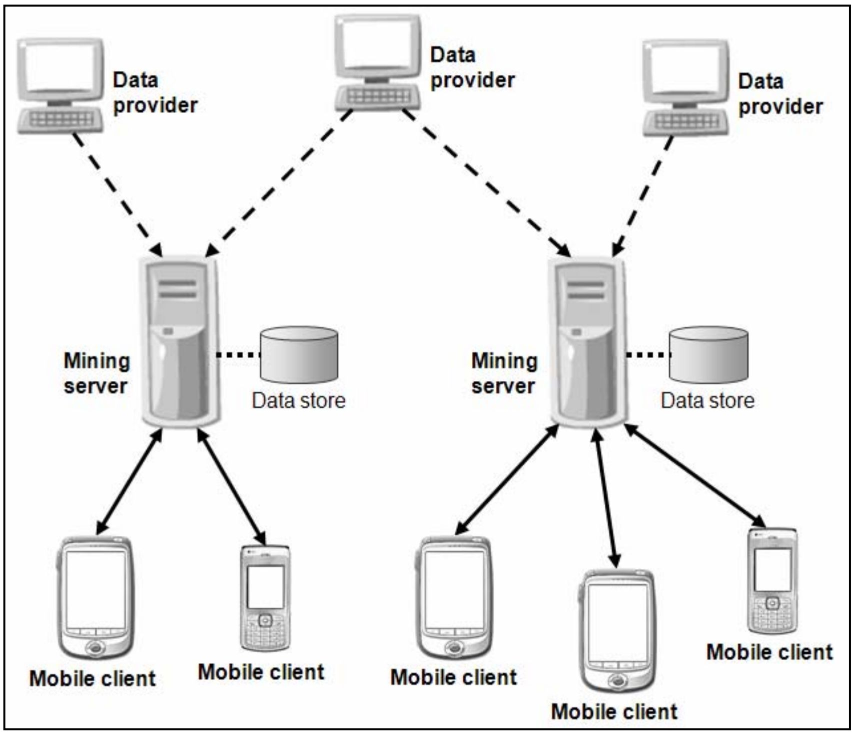
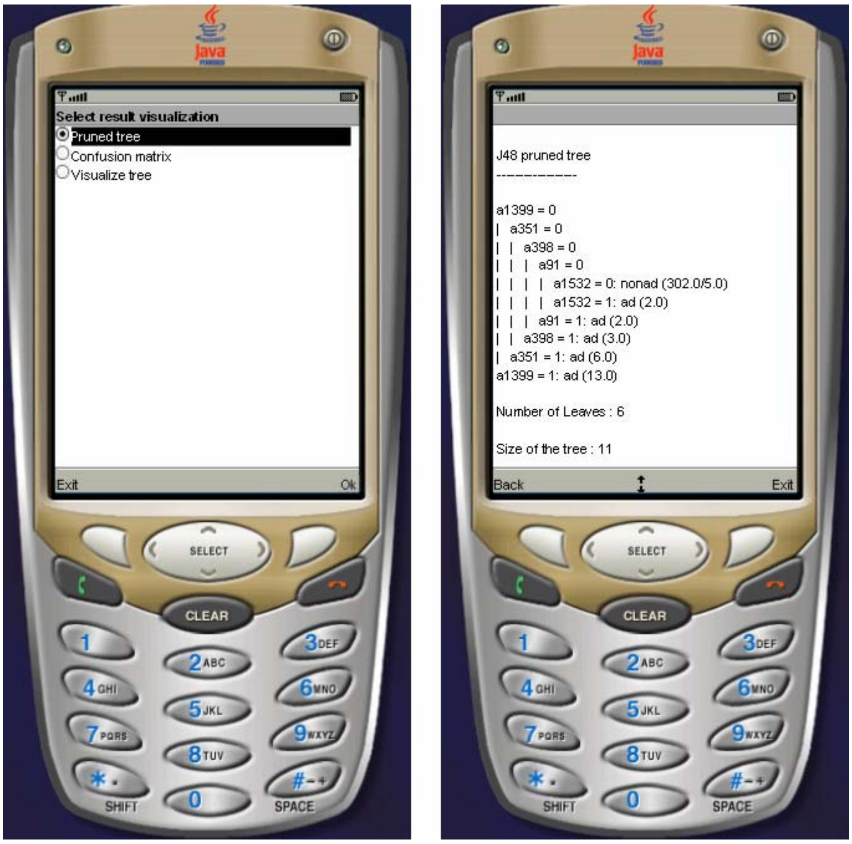


Figure 4. General architecture of the mobile data mining system.



```
J48 pruned tree
------------

a1399 = 0
|   a351 = 0
|   |   a398 = 0
|   |   |   a91 = 0
|   |   |   |   a1532 = 0: nonad (302.0/5.0)
|   |   |   |   a1532 = 1: ad (2.0)
|   |   |   a91 = 1: ad (2.0)
|   |   a398 = 1: ad (3.0)
|   a351 = 1: ad (6.0)
a1399 = 1: ad (13.0)

Number of Leaves : 6

Size of the tree : 11
```

Figure 5. Two screenshots of the mobile data mining client running on the emulator of the Sun Java Wireless Toolkit.

The mobile client is composed by three components: the *MIDlet*, the *DMS Stub*, and the *Record Management System (RMS)*. The MIDlet is a J2ME application allowing the user to perform data mining operations and visualize their results. The DMS Stub is a WSRF service stub allowing the MIDlet to invoke the operations of a remote DMS. Even if the DMS Stub and the MIDlet are two logically separated components, they are distributed and installed as a single J2ME application. The RMS is a simple record-oriented database that allows J2ME applications to persistently store data across multiple invocations. In our system, the MIDlet uses the RMS to store the URLs of the remote DMSs that can be invoked by the user. The list of URLs stored in the RMS can be updated by the user using a MIDlet functionality.

The small size of the screen is one of the main limitations of mobile device applications. In data mining tasks, in particular, a limited screen size can affect the appropriate visualization of complex results representing the discovered model. In our system we overcome this limitation by splitting the result in different parts and allowing a user to select which part to visualize at one time. Moreover, users can choose to visualize the mining model (such as a cluster assignment or a decision tree) either in textual form or as an image. In both cases, if the information does not fit the screen size, the user can scroll it by using the normal navigation facilities of the mobile device.

As an example, Figure 5 shows two screenshots of the mobile client taken from a test application. The screenshot on the left shows the menu for selecting which part of the result of a classification task must be visualized, while the screenshot on the right shows the result, in that case the pruned tree resulting from classification.

Our early experiments show that the system performance depends almost entirely on the computing power of the server on which the data mining task is executed. On the contrary, the overhead due to the communication between MIDlet and Data Mining Service does not affect the execution time in a significantly way, since the amount of data exchanged between client and server is very small. In general, when the data mining task is relatively time consuming, the communication overhead is a negligible percentage of the overall execution time.

Conclusion

The main thesis of this article is that the Grid and service-oriented high performance systems can be used as an effective cyber infrastructure for implementing and deploying geographically-distributed data mining and knowledge discovery services and applications. Future uses of the Grid are mainly related to the ability to utilize it as a knowledge-oriented platform able to run world-wide complex distributed applications. Among those, knowledge discovery applications are a major goal. To reach this goal, the Grid needs to evolve towards an open decentralized platform based on interoperable high-level services that make use of knowledge both in providing resources and in giving results to end-users. The recent emergence of Cloud computing systems could accelerate this process.

Here we discussed a general framework and some systems based on it that implement Grid-enabled knowledge discovery services by using dispersed resources connected through a Grid. These services allow professionals and scientists to create and manage complex knowledge discovery applications composed as workflows that integrate data sets and mining tools provided as distributed services on a Grid. They also allow users to store, share, and execute these knowledge discovery workflows as well as publish them as new components and services. As examples of this approach, we described how the Knowledge Grid, the Weka4WS and the mobile data mining services provide a high level of abstraction of Grid resources for distributed knowledge discovery activities, thus allowing the end-users to concentrate on the knowledge discovery process without worrying about infrastructure details.

Software frameworks and technologies for the implementation and deployment of knowledge services, as those we discussed in this paper, provide key elements to build up data analysis applications on enterprise or large-scale Grids and Clouds. Those models, techniques, and tools can be instrumented in Grids and Clouds as decentralized and interoperable services that enable the development of complex systems such as distributed knowledge discovery suites and knowledge management systems offering pervasive access, adaptivity, and high performance to single users, professional teams, and virtual organizations in science, engineering and industry that need to create and use knowledge-based applications [7,9].

References

1. Al Sairafi, S., Emmanouil, F.-S., Ghanem, M., Giannadakis, N., Guo, Y., Kalaitzopoulos, D., Osmond, M., Rowe, A., Syed, J., and Wendel, P. The design of discovery net: Towards open grid services for knowledge discovery. *Int. Journal of High Performance Computing Applications 17*, 3, (2003), 297-315.
2. Brezany, P., Hofer, J., Min Tjoa, A., and Wöhrer, A. GridMiner: An infrastructure for data mining on computational grids. *Proc. APAC Conference and Exhibition on Advanced Computing, Grid Applications and eResearch*, 2003.
3. Cannataro, M. and Talia, D. Semantics and knowledge grids: Building the next-generation grid. *IEEE Intelligent Systems 19*, 1, (2004), 56–63.
4. Cannataro, M. and Talia, D. The knowledge grid. *Comm. ACM 46*, 1, (Jan. 2003), 89-93.
5. Chen, L., Reddy, K., and Agrawal, G. GATES: a grid-based middleware for processing distributed data streams. *Proc. IEEE Int. Symposium on High Performance Distributed Computing*, 2004, 192-201.
6. Congiusta, A., Talia, D., and Trunfio, P. Distributed data mining services leveraging WSRF. *Future Generation Computer Systems 23*, 1, (2007), 34-41.
7. Corcho, O., Alper, P., Kotsiopoulos, I., Missier, P., Bechhofer, S., and Goble, C. An overview of S-OGSA: A reference semantic grid architecture. *Journal of Web Semantics 4*, 2, (2006), 102-115.
8. Foster, I. Globus toolkit version 4: Software for service-oriented systems. *Proc. IFIP Int. Conference on Network and Parallel Computing*, 2005, LNCS 3779, 2-13.
9. Gil, Y. On agents and grids: Creating the fabric for a new generation of distributed intelligent systems. *Journal of Web Semantics 4*, 2, (2006), 116-123.
10. NGG3 Expert Group Report: “Strategic Future for European Grids: Next Generation GRIDs based on SOKU - A new paradigm for service delivery and software infrastructure,” Brussels, Dec. 2005.
11. Stankovski, V., Swain, M., Kravtsov, V., Niessen, T., Wegener, D., Rohm, M., Trnkoczy, J., May, M., Franke, J., Schuster, A., and Dubitzky, W. Digging deep into the data mine with DataMiningGrid. *IEEE Internet Computing 12*, 6, (2008), 69-76.
12. Talia, D. and Trunfio, P. Mobile data mining on small devices through web services. *Mobile Intelligence*, Yang L., Waluyo A., Ma J., Tan L., Srinivasan B. (Eds.), Wiley, 2010, 264-276.
13. Talia, D., Trunfio, P., Verta, O. The Weka4WS framework for distributed data mining in service-oriented Grids. *Concurrency and Computation: Practice and Experience 20*, 16, (2008), 1933-1951.
14. Wang, H., Ghoting, A., Buehrer, G., Tatikonda, S., Parthasarathy, S., Kurç, T. M., Saltz, J. H. A services oriented framework for next generation data analysis center. *Proc. IEEE Int. Parallel and Distributed Processing Symposium 11*, (2005), 219b.
15. Witten, H., and Frank E., *Data Mining: Practical Machine Learning Tools with Java Implementations*. Morgan Kaufmann, 2000.

Acknowledgments

This research work is partially carried out under the FP6 Network of Excellence Core-GRID funded by the European Commission (Contract IST-2002-004265) and the TOCAI.IT project funded by MIUR.