Interoperability: Bridging the Gap

Shashi
Department of Computer Science
and Engineering, Thapar University
Patiala-147004, INDIA.

Seema Bawa
Department of Computer Science
and Engineering, Thapar University
Patiala-147004, INDIA.

ABSTRACT

Grid computing is the aggregation of networked computers to form a large-scale distributed system which can be used to solve complex scientific problems. The emergence of open standards has great influence on this computing technology, especially in providing seamless grid interoperability and grid integration facilities. This paper focuses on the need of interoperability between various grids and discusses TUX-INTERO, an interoperability system which is used to collaborate resources in a Grid using web services. It registers varied resources and makes them accessible to any end user.

General Terms

Security, Grid Computing, Distributed Computing, Interoperability, Pervasive Computing.

Keywords

Grid Computing, Interoperability, Resource Description.

1. INTRODUCTION

Grid is a type of parallel and distributed system that enables the sharing, selection and aggregation of geographically distributed "autonomous" resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements [1]. A computational grid can be defined as "a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities." Typically, grid resources are provided by various organizations and are used by people from diverse sets of organizations [2]. A grid may support (or define) a single virtual organization or it may be used by more than one virtual organization. Individual pieces of hardware may be used in more than one grids, and people may be members of more than one virtual organization. Since the idea of Grid computing has gained a foothold both in academia and industry a multitude of Grid environments have been developed. These environments work reasonably well within their own boundaries; however there is little interoperability between them. The main reasons are the diversity of software platforms, the various policies governing the access and use of distributed resources and the apparent lack of adherence to the emerging standards [3].

Over the recent years substantial effort has been invested in the development of Grid standards. There are several internationally recognized bodies that are active in this area that provide significant contributions, in particular the Open Grid Forum (OGF) [4], the Organization for the Advancement of Structured Information Standards (OASIS)[5], the World Wide Web Consortium (W3C) [6] and the Distributed Management Task Force (DMTF) [7] as well as others. A critical requirement in a distributed, multi-organizational Grid environment is how to enable interoperability, which can be interpreted as two parts: one is the definition of service interface and the other one is identification of the protocol that can be used to invoke a particular interface. Software to enable Grid computing has been primarily written for Unix-class operating systems, thus severely limiting the ability to effectively utilize the computing resources of the vast majority of desktop computers i.e. those running variants of the Microsoft Windows operating system.

In this paper, TUX-INTERO- an Interoperability system has been discussed, that enables any end user to use disparate and heterogeneous resources in a Grid. One of the primary aims of TUX-INTERO is to discover and provide heterogeneous resources at one interface. TUX-INTERO uses web service to enable resource description in a grid where heterogeneous resources can invoke web service to become part of a Grid. The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 discusses proposed interoperability system design and Section 4 covers the implementation part. Section 5 discusses conclusion while giving a glimpse of future directions.

2. Related Work

Over the recent years substantial effort has been invested in the development of Grid standards. There are several internationally recognized bodies that are active in this area that provide significant contributions, in particular the Open Grid Forum (OGF) [4], the Organization for the Advancement of Structured Information Standards (OASIS) [5], the World Wide Web Consortium (W3C) [6] and the Distributed Management Task Force (DMTF) [7] as well as others.

Interoperability assumes the use of interfaces described by generally accepted standards. Without compliance with the standards we cannot talk about interoperability. This idea has been strongly supported and systematically implemented by the Grid Interoperability Now (GIN) initiative that is a community activity within the Open Grid Forum (OGF). The idea is to establish so-called interoperability islands that would cover the key services such as Job Submission, Information Services, Storage Management, Accounting, Job Monitoring, Database Access and Virtual Organization Management.

Currently, there are a number of solutions that aim to provide interoperability in grids. Here, we discuss a few representative implementations of portals which aim to interoperate grid resources at various levels.

Various grid portals have been developed till date, for specific applications. The NEESgrid Portal [10], the Alliance Portal [8], the GENIUS Portal for the European Data Grid [9], and the IeSE (Integrated e-Science Environment) [10] are some of application based grid portals. There have been some efforts to build Grid middlewares also, such as GridPort [11] which is a middleware toolkit to facilitate the development of Grid computing portals. NSF NMI's OGCE project [12] is considered as a more flexible portal. The Open Grid Computing environments (OGCE) Release 2 consists of a core set of Grid portlets. Java Specification Request 168 Portlet Specification (JSR-168) [13] is a portlet specification which enables interoperability between portlets and portals, and defines a set of APIs for portal computing addressing the areas of aggregation, personalization, presentation and security. JSR-168 has been supported many research groups and projects.

P-GRADE [14] is a web based, service rich environment for the development, execution and monitoring of workflows and workflow based parameter studies on various grid platforms. It hides low-level grid access mechanisms by high-level graphical interfaces, making even non grid expert users capable of defining and executing distributed applications on multi-institutional computing infrastructures. Workflows and workflow based parameter studies defined in the P-Grade are portable between grid platforms without learning new systems or re-engineering program code. Technology neutral interfaces and concepts of the P-GRADE Portal help users cope with the large variety of grid solution It has been a step further in allowing various scientists to submit workflows to grid and achieve interoperability.
There have also been efforts such as DISCOVER [15], [16] and [17][18] to enable interoperability in Grids.

3. TUX-INTERO: Proposed Interoperability System

TUX-INTERO is an interoperability system whose primary function is to allow users access to heterogeneous resources located in various administrative domains. Secondly, the goal of this work is to provide resources to an end user on the basis of trustworthiness and reputation values of past transactions.
The key functions of the TUX-INTERO are

(a) plugging the resources available,
(b) invoking web services,
(c) storing and retrieving resource information, and
(d) presentation of resource information.

Plugging the resource available: Resources can be plugged into Grid by using the application designed in .Net. The responsibility of the interface is to invoke a web service so as to provide information to the TUX-INTERO on the resource available by giving resource description as depicted in Figure 1. Its role is to gather information from each of the resource and to publish that information for use by other requesters.

Invoking Web Services*:* Web Services can be invoked from incongruent applications and responses can be submitted to the TUX-INTERO system. Any Web Service can interact with any other Web Service. SOAP, the standard protocol is supported by all of the major vendors. As Web Services can be written in any language and any platform, developers do not need to change their development environments in order to produce or consume Web Services.

Storing and Retrieving Information: The idea is to maintain a cache of the resources which are available instead of storing all the available resources. We are making use of SQL database to store the information.

Presenting Information: The TUX-INTERO System provides the user interface for various heterogeneous resources to be accessed in a grid

3.1 Design of TUX-INTERO

TUX-INTERO [21] has been enhanced to use .Net Framework to enable resources on Windows Platform to become part of Grid by using Web Services. There are three components in the TUX-INTERO enhanced architecture includes three roles: service provider, service broker and service requester. These parties perform three fundamental operations: publish, find and bind. These concepts are explained as following:

3.1.1 Roles in TUX-INTERO architecture

Service provider: The owner of the service is known as Service Provider. From an architecture perspective, this is the platform that hosts access to the service.

Service requester: The consumer of the service is known as Service Requester. From an architecture perspective, this is the application that is invoking or initiating an interaction with a service. A program or other Web Services can play this role.

Service registry: This is a service description repository where service providers publish their service descriptions and service requester find services and obtain binding information.

3.1.2 Operations in TUX-INTERO architecture

These are three operations in TUX-INTERO**.**
Publish: In order to be accessible, a service description needs to be published so that the service requester can find it. The published information can vary from service description.

Find: The service requester retrieves a service description from the service registry in order to find the required resource. The find operation can be involved in two different phases for service requester: at design time to retrieve the service's interface description for program development, and at runtime to retrieve the service's binding and location description for invocation.

Bind: The service needs to be invoked. In the bind operation the service requester invokes or initiates an interaction with the service at runtime using the binding details in the service description.

In service-oriented architectures, service descriptions and metadata play a central role in maintaining a loose coupling between service requesters and service providers. The service description, published by the service provider, allows service requesters to bind to the service provider. The service requester obtains service descriptions through a variety of techniques, from the simple "e-mail me the service description" approach to various other approaches by vendors.

TUX-INTERO facilitates resource information publication, management and browsing. It allows resource providers and consumers to use a web browser as a simple graphical client to access the web service.

4. Implementation of TUX-INTERO

There is rapidly emerging interest in Grid computing from commercial enterprises to researchers. A Microsoft Windows-based Grid computing infrastructure will play a critical role in the industry-wide adoption of Grids due to the large-scale deployment of Windows within enterprises. This enables the harnessing of the unused computational power of desktop PCs and workstations to create a virtual supercomputing resource at a fraction of the cost of traditional supercomputers. However, there is a distinct lack of service-oriented architecture-based Grid computing software in this space. To overcome this limitation, we have developed a web service based interoperability solution for Grid implemented on the Microsoft .NET Platform.
While the notion of Grid computing is simple enough, the practical realization of Grids poses a number of challenges. Key issues that need to be dealt with are security, heterogeneity, reliability, application composition, scheduling, and resource management. The Microsoft .NET Framework provides a powerful toolset that can be leveraged for all of these, in particular support for remote execution (via .NET Remoting [4] and web services), multithreading, security, asynchronous programming, disconnected data access, managed execution and cross-language development, making it an ideal platform for Grid computing middleware.

1. The foremost requirement is to discover a web service.
2. After locating it, the second thing required is to invoke it.
3. A web service can be only invoked after knowing the description of it, so it should describe itself (i.e. tell us how exactly we should invoke it).
4. Now, knowing where the Web Service is located and how to invoke it. The invocation itself is done in a language called SOAP. Therefore, first a *SOAP request is send*.
5. The Web Service replies with a *SOAP response* which includes the response, or maybe an error message if our SOAP request was incorrect.

6. The Web Service replies in a language called WSDL.

The desktop application TMS adds reference to TMSService as web reference to invoke ProviderService web service. Figure 1 presents desktop application TMS.

At the Resource Provider End

A desktop application is designed and implemented in ASP. Net and which supports and invokes these basic SOAP invocation methods. Figure 3 depicts the web methods as described here.
(1) Online () – Publishes system and providers information and updates the information in the in the database which is further updated in web application. It shows the Provider is available to volunteer services and become part of the Grid. WSDL (Web Services Description Language) is generated when Online web method is invoked. Provider's Id is passes as a parameter to the web service.
(2) Offline () – This web method deletes system information to indicate provider has been shutdown or is not willing to share its information. System name is provider as parameter to this web method. WSDL is generated when Offline web method is called.
Figure 4 presents the response of passing arguments to Web service ProviderService. Figure 5 shows in Online column by showing status of services available as TRUE.

At the Service Consumer End`

Authenticated Consumers can retrieve list of resources available on system connected to Grid from status database by specifying its requirements. The services are provided based on the trustworthiness values of service requesters using TUX-TMS[19][20]. The services are evaluated for trustworthiness and reputation. Figure 5 depicts sequence diagram of TUX-INTERO as integrated with TUX-TMS.
The TUX-INTERO system is used in integration with TUX-TMS, a reputation based system for Grid.

5. Conclusion

Web service provides interoperability and has been propagated as a solution for enabling next generation massive infrastructures such as Grid and Cloud. TUX-INTERO has been redesigned further and implemented using web services. TUX-INTERO can be enhanced further to incorporate various web services and provide other features of resource management, job management, data management and load balancing in a Grid. Here, resource description and resource discovery feature has been discussed and presented. TUX-INTERO integrated with TUX-TMS provides secure services for end users accessing from various platforms. It enables users to become provider by simply invoking web services and installing the desktop application designed in .Net which is compatible with Linux platforms too.

6. REFERENCES

[1] Foster, I., Kesselman, C. 1999 The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann.
[2] Foster, I., Kesselman, C., Tuecke, S. 2001 The Anatomy of the Grid: Enabling Scalable Virtual Organisations,International Journal of High Performance", Computing Applications 15 (3), 200-222.
[3] Boardman, R., Crouch, S., Mills, H., Newhouse, S. 2007 Towards Grid Interoperability, In: Proceedings of the UK e-Science All Hands Meeting 2007, Nottingham.
[4] Open Grid Forum(OGF). Available at http://www.ogf.org. Accessed on August 12, 2010

[5] Organization for the Advancement of Structured Information Standards (OASIS), Available at http://www.oasis-open.org. Accessed on August 14, 2010.

[6] World Wide Web Consortium (W3C). Available at http://www.w3c.org. Accessed on August 18, 2010.

[7] Distributed Management Task Force (DMTF), Available at http://www.dmtf.org. Accessed on August 14, 2010.

[8] NCSA Alliance Scientific Portal Project. Available at http://www.extreme.indiana.edu/alliance. Accessed on August 12, 2010.

[9] Barbera, R. 2003 The GENIUS Grid Portal, Computing in High Energy and Nuclear Physics.

[10] Allan, R., Integrated e-Science Environment for CLRC. Available at http://www.e-science.clrc.ac.uk/. Accessed on August 14, 2010.

[11] Dahan M., Thomas M., Roberts E., Seth A., Tomislav U. 2004 GridPort: Using Globus for Grid- Enabled Web Portals in Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC'04), 272-273.

[12] Thomas M., Dahan M., Mueller K., Mock S., Mills C. 2002 Application Portals: Practice and Experience. Grid Computing environments: Special Issue of Concurrency and Computation: Practice and Experience, Vol 14. 1427-1444.

[13] Gannon, D., Fox, G. Pierce, M. Plale, B. Grid Portals: A Scientist's Access Point for Grid Services (DRAFT 1). Available at http://www.extreme.indiana.edu/groc/ggf-portals-draft.pdf. Accessed on August 12, 2010).

[14] P-Grade. http://portal.p-grade.hu/(accessed on June 17, 2010).

[15] Vijay Mann and Manish Parashar 2002. Engineering, An interoperable computational collaboratory on the Grid, CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE. Concurrency Computat.: Pract. Exper. 2002; **14**:1569–1593 (DOI: 10.1002/cpe.687).

[16] M. Brooke, J., Fellows, D., MacLaren, J. 2004 Interoperability of resource description across grid domain boundaries, European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004.

[17] Flech, M., Field L. 2008 Grid Interoperability: Joining Grid Information Systems, Journal of Physics: Conference Series 119 (2008), 1-7.

[18] Assuncao, M.D.D.: Provisioning Techniques and Policies for Resource Sharing between Grids. Ph.D. Thesis. The University of Melbourne, Australia.

[19] Shashi, Bawa S. 2010, Reputation Enhancement in a Trust Management System, Springer CNSA 2010, CCIS 89, 440–451.

[20] Shashi, B., Bawa, S. 2010 TUX-TMS: Thapar University Extensible-Trust Management System. International Journal of Security, CSC Journals 4(1), 1–16 .

[21] Shashi, Bawa S. 2010 TUX-INTERO: A Portal for secure interoperation of Grids, International Journal of Engineering, Science and Technology, Volume 2(7) 3335-3343.

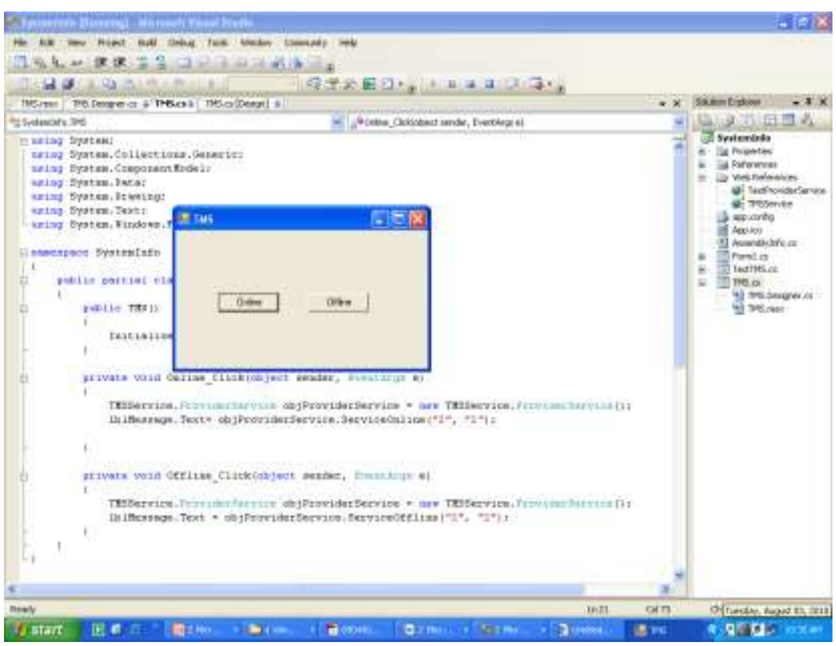

Figure 1. Desktop Application for invoking web service

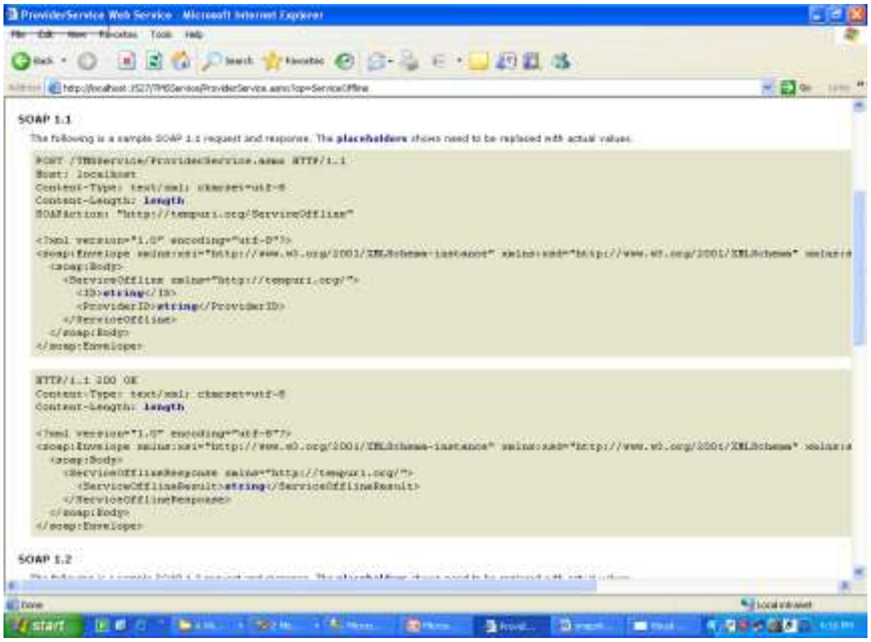

Figure 2.Invoking a web Service

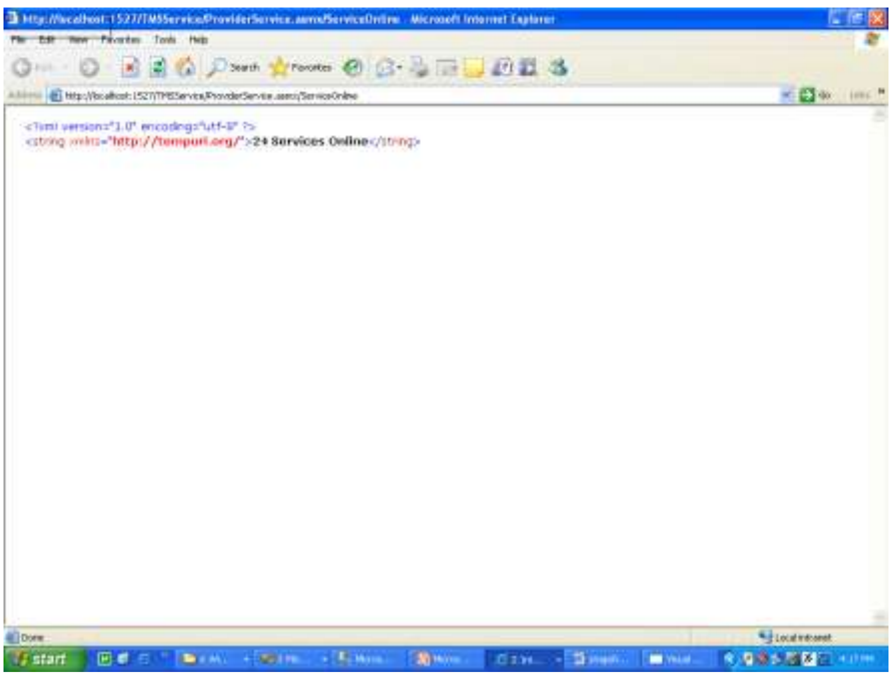

Figure 3. Passing arguments to web service

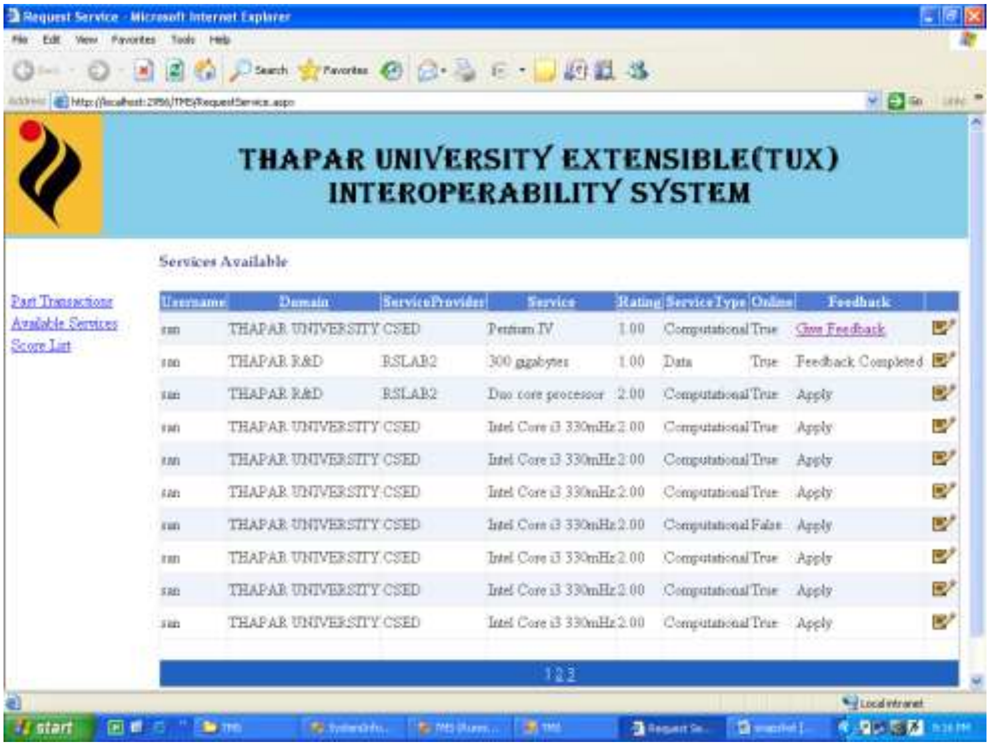

Figure 4.Web Application showing results updated from web service

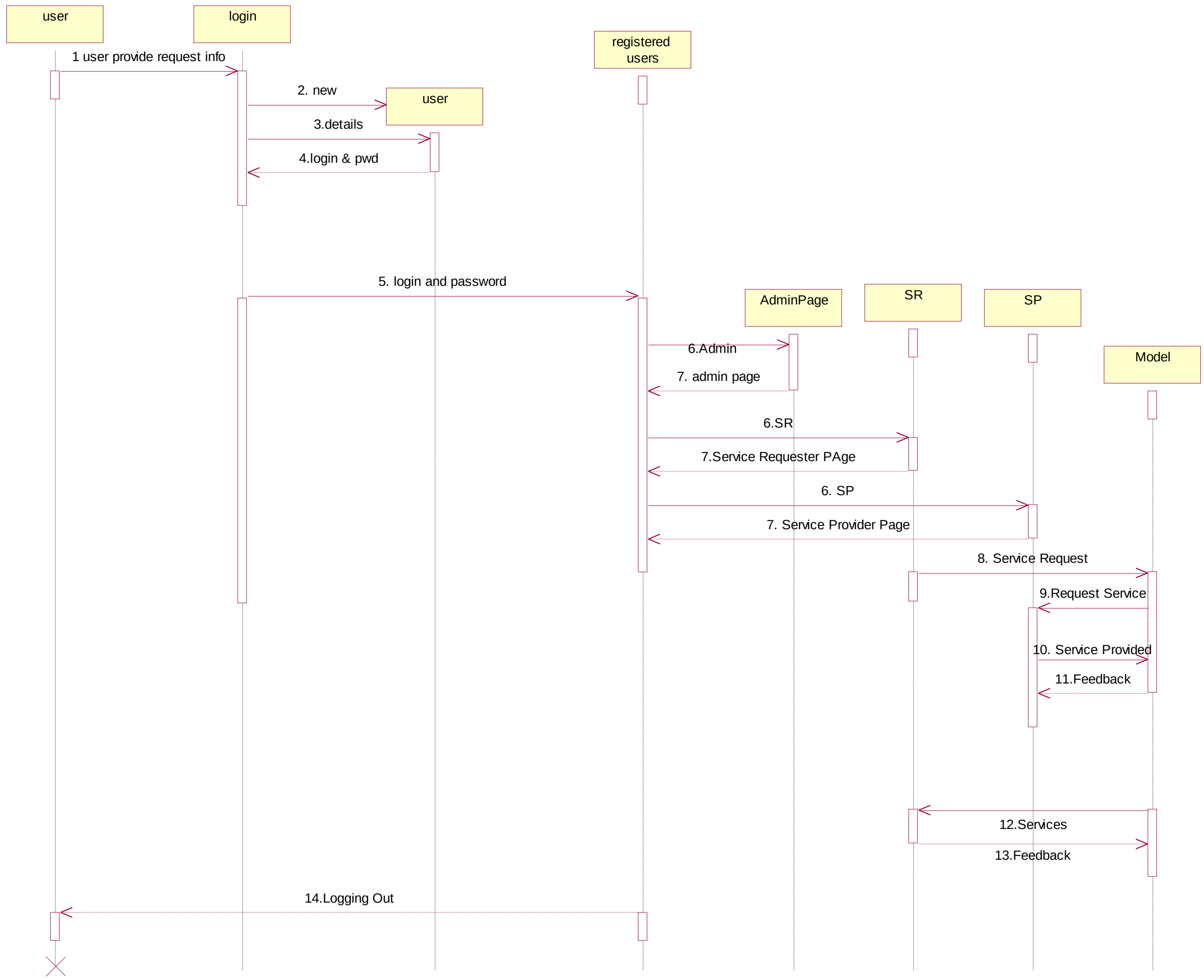

Figure 5. Sequence Diagram of TUX-INTERO