Evaluation of Web Services Resource Framework for Virtual Organization

Asif Akram, CCLRC Daresbury Laboratory, Daresbury, UK; E-mail: a.akram@dl.ac.uk
Rob Allan, CCLRC Daresbury Laboratory, Daresbury, UK

ABSTRACT

The organization of resources in communities is an efficient way for resource discovery. The effectiveness of proposed communities depends on the technologies selected, their interoperability, platform independence and compliance with standards. This paper outlines the role of different available technologies to sustain the effective working of the community and to provide the general middleware for the community formation. Main focus of this paper is to evaluate the Web Services related specifications particularly Web Services Resource Framework and Web Services Distributed Management; and their effectiveness in the community formation.

1. INTRODUCTION

Emerging distributed computing paradigms, such as Grid Computing [4] and Virtual Research Environments [7] comprise of services/peers which collaborate in one way or another to improve their effectiveness – this cooperation leads to the formation of a "Virtual Organization". Virtual Organizations already exist in e-commerce, science, research and engineering projects – whereby a collection of organizations or scientists come together to solve a single large problem. Enormous development in Information and Communication Technology (ICT) has changed the requirements and expectations of consumer/user leading to tense competition (economically), composite and complicated services (scientific), and demand for diverse and dispersed resources (Grid environment). Extensive collaboration in every aspect in every possible way is required to meet these high user expectations. A Virtual Organization is one such strategic approach to building strong but flexible links to partner organizations with similar interests [6].

The effectiveness of a Virtual Organization relies on the technology selected for the middleware, adherence to standards, and built in support for managing the diverse resources. In our understanding, the middleware provides basic features for community formation; defines protocols for coordination among resources and peers; the mechanism to advertise the capabilities of the resources, the discovery of suitable services and the management of these dynamic resources.

The community formation process and different types of communities are discussed previously and can be found in [1], [2] and [3]. The 'Section 2' discusses the requirements of the VO and suitability of emerging specifications in the Web Services. The 'Section 3' evaluates Web Services Resource Framework (WS-RF) [5, 13] and Web Services Distributed Management (WSDM) [14] for Virtual Organizations and different collaborative environments. The last section explains the existing and proposed implementation details for the community.

2. REQUIREMENTS OF VIRTUAL ORGANIZATION

A Virtual Organization "VO" defines a logical group of members within the grid, which can be geographical distributed. Usually all members have a common interest, for example one institute, a group of scientist or a commercial enterprise. Within these VO's all resources are shared and the use is strictly defined by policies. Selection of any technology to simulate VO requires thorough consideration and analysis of its built in support not only for the formation of VO but its management and functionality. In the following section we will discuss different requirements of VOs in the context of features available WS-ServiceGroup and WSDM. Technical details of WS-ServiceGroup and WSDM are based on the GT4 [8] implementation of WS-RF and Apache implementation of WSDM [9].

2.1 Bootstrapping Service

Virtual Environment provides essential information to a peer on joining the network to discover enough network-based and other resources to sustain itself. Each container hosting Web Services manage

ContainerRegistryService and DefaultIndexService which collects data from various sources and provides a query/subscription interface to that data. Resources are added in the ContainerRegistryService and DefaultIndexService at the time of initialization.

2.2 Maintaining Membership Policy

Each community needs to keep appropriate membership criteria, thus members of community should have resources, capabilities and competencies compatible with the rest of the members. VO manages the membership requirement of community by restricting membership according to a set "Membership Policy". Membership Policy is constrained in the ServiceGroup as MembershipContentRule below is the example:

<wssg:MembershipContentRule MemberInterface="ns2:X" ContentElements=""/>

<wssg:MembershipContentRule MemberInterfaces="ns3:Y" ContentElements="ns3:RP1 ns3:RP2" />

MembershipContentRule in the first statement expects members to implement "ns2:X" portType; where as second MembershipContentRule expects not only implementation of "ns3:Y" but also exposing two ResourceProperties.

2.3 Lease Time

The membership of any ServiceGroup is for specific duration set at the joining time. After membership duration ceases, peers can apply to renew their membership, the outcome of which can be linked to the effectiveness of a peer and its historical effectiveness within the community. VO makes strategic alliances to achieve set goals and there is no need for long term contract which is contrary to the purpose of Virtual Organizations.

2.4 Access Control

Different members enjoy different access rights for different resources available within the community. Nature of access rights depends on the effectiveness and activeness of specific member within the community and relevance of resources to a particular member. Access rights granted to different peers depends on their role within the community and nature of resources; Each Resource has its own access and security policies which are controlled in the form of Grid Map file in GT4.

2.5 Resource Monitoring Service

Optimized usage of resources means maximum productivity of the resources; which boosts overall performance of the community. Overloading resources or minimum usage of resources with respect to their capacity results in declined standards, which can be achieved by better governance and scheduling. ServiceGroup implementation provides QueryResourceProperties operation which can be used to search members matching certain criteria i.e. CPU with 50% load, resource with 1 GB memory etc. Resource monitoring leads to fault detection, performance analysis, performance tunneling, performance prediction and scheduling.

2.6 Performance Controller Service

The WSDM-MUWS describes the manageability of the resources. The manageability capability is best thought of as an interface to the internal workings of the Service. The role of the manageability capability is to list the available operations on a resource so that management software can get a list of the available operations. These capabilities are defined using XML and can be aligned with ServiceGroup membership content rules by including ResourceProperties essential for monitoring and management in the membership content rules.

2.7 Scheduling Manager Service

The combination of WS-ServiceGroup and WSDM-MUWS provides best Resource Querying and Monitoring mechanisms; on top of which any level of complex scheduling service can be implemented. Scheduling of tasks involves co-ordination between member peers and co-operation with other communities, which leads to complex and demanding workflows. Both WS-ServiceGroup and WSDM-MUWS subscribe to the changes in the Resource Properties of the members through Web Services Notification (WSN) and effective use of WSN provides real time monitoring capabilities to schedule tasks to member peers either in parallel or series.

2.8 Security Manager Service

The Community provides a secure environment to its members for co-ordination. The community as a whole ensures the protection of data through its unique ability to adapt, resist and protect data by scattering the multiple copies within the community boundary. Members can access local resources and communicate with other members without any additional authorization and authentication once they are known to the local authentication authority. Most of WS-RF implementation provides different level of security, message level and transport level security along with delegations of credentials.

2.9 Community of Communities

In WS-ServiceGroup each VO itself is a stateful Web Services with ResourceProperties encapsulating the member's details. WS-ServiceGroup can be arranged in hierarchical manners to create communities within community as long they all full membership criteria. Community of Communities is essential to pool similar services within the community and group different communities to from super VO according to the requirement of workflow.

3. WS-RF AND COMMUNITY FORMATION

The community formation is to achieve the relevance of resources and services with respect to each other or according to the business requirements. The suitability of resources and services is the basis of different type of communities, which is crucial for the collaboration among participants from business, commercial or academic domains can be achieved. The services are units of work that map closely to the activities in a business or scientific process, such services are immediately relevant to analysts who can readily participate in their creation and use them to define complicated new processes thus enabling Service Driven Environment. The architectural implementation of the community based on WS-RF ServiceGroup has many advantages and some are listed below:

3.1. Governance

Sharing of services/resources offered by participating Peers is central to our Community approach and is fundamental of VO. The ability to rapidly assemble the applications or orchestrate the processes based upon the availability of services from participating members. The sharing of different resources, by definition requires efficient governance. Monitoring and Discovery System (MDS) in Globus Toolkit 4, is a suite of Web Services to monitor and discover resources and services on the Grid. The implementation of MDS allows resource discovery and monitoring in the standard way. The query and subscription interfaces in MDS to arbitrarily detailed resource data can be configured with trigger interface for appropriate actions under specific conditions.

3.2. Reusability

Conventionally developers are geared to tackle each application as an independent problem to be solved, with little possibility of the code reuse. Web Services based communities are all about application development with reuse in the mind, both in term of the existing code and in terms of planning new code to be used in future applications. Different service/s provided by different members, can be reused by other members to support their services or to develop composite services i.e. workflow.

3.3. Rapid Application Development

Peers are categorized in Communities based on the service/s they tend to share; these services are reusable fragment of code, which can be assembled in many different ways to develop bigger and sophisticated applications. The community focus on each activity as a service, which can be useable in different applications and applications are no more isolated code with limited applicability outside their problem domain. These services are orchestrated through a "workflow" based on the available resources, competencies, strengths/weaknesses of members; which may require collaboration among communities to "buy in" services missing within the community.

3.4. Manageability

Sharing of service/services offered from different resources even when these resources are in different Communities leads to endless possibilities. Thus development of any complicated applications is more or less orchestration of independent services, resulting in more manageable application development process with respect to time and development cost. Modular software is designed to avoid failures in large systems, especially where there are complex user requirements. The possibility to replace similar services particularly at run time under un-predictable circumstances gives new dimension to the management of final goal according to the user requirements.

3.6. Improved Quality

MDS monitors the status of different registered resources and service/s with "self-cleaning", each resource has a lifetime and is removed from the Index of available resources if it is not refreshed before it expires. Monitoring capability can be coupled with set of conditions to control the Quality of Service (QoS) [17] of individual resource i.e. advertise only those resources with processor load less than 50%, resources with memory size of at least I Gig. Tools can discover similar type of services and select any service from the pool of similar services according to the Quality of Service (QoS) constraints and the requirements of final goal. The module design of services means they can be replaced at run time without compromising the quality. The replacement of semantically equivalent services can be automatically initiated by the community (in case of Co-Operating community) or can be manually done by the client.

3.7. Diversity

Very few applications can survive in the isolation. Most often applications have to be integrated with other applications inside and outside the enterprise. This integration is usually achieved using some form of "middleware". Middleware provides the "plumbing" for data transport, data transformation, routing etc. [15]. No sizeable IT organization operates with a single programming standard, and with emergence of Web Services as the de-facto standard the architecture of the applications is changing with no reliance on single programming modal. Modular applications developed by integration of different of services deployed on variety of platforms and environments can exploit the specialization of individual programming language to achieve maximum quality. This Technology-to-Technology" (T2T) gives new dimension to integration of application development, which requires changes in programming style and our understanding to the problem.

4. SEMANTIC AND COMMUNITY MANAGEMENT

The recent trends in industry connected with enterprise integration demand solid technology to provide effective Knowledge Management within and across industrial enterprises [16]. Combination of Semantic Web and Peer-to-Peer technologies provides many attractive and powerful features for this domain, but these great possibilities are not yet fully explored. There is a strong interest in the development of reliable platform for support of cooperative knowledge management and flexible integration of various applications, Web Services and industrial resources in the Grid Environment.

WS-RF is described as a set of protocols for manipulating WS-Resources, and deals with application data (resource properties) in the form of the XML. Additional data is defined within WS-RF itself –e.g. TerminationTime in WS-ResourceLifetime and service group entry WS-Resources in WS-ServiceGroup. It is easy to attach semantic information with the service through optional resource. Any service can work on multiple resources and one resource can be managed by multiple services, adding additional resource declaring the semantic information related to the service, data types and operation is more feasible. Tools can query the semantic resource of the service, if it exists before integrating the service in the workflow. WS-Resources can have reference to other WS-Resources in the form of EPR, which gives an alternative possibility of keeping the semantic information separate from the WSDL and integrating it with the service as a WS-ResourceProperty. WS-RF mandates the use of XML Schema for the application specific data and use of languages like XPath or XQuery for querying purposes. The application data can be anything in XML format and WS-RF doesn't constrain the XML Schemas used to describe the data which make it possible to annotate the data with semantics.

Different implementations have different mechanism to collect the information which can be from different sources. GT4 has three different possibilities to populate Index Service (i) to get information from resources, (ii) through subscription/notification mechanism and (iii) through any external program. WS-RF is described as a set of protocols for manipulating WS-Resources, and deals with application data (resource properties) in the form of the XML. Additional data is defined within WS-RF itself –e.g. TerminationTime in WS-ResourceLifetime and service group entry WS-Resources in WS-ServiceGroup. External program provides maximum flexibility to collect the data and apply local Knowledge Base to extend the WS-ResourceProperties. This approach requires the flexibility in the data set with attached semantic, to organize the services based on the semantic nature of the data consumed by the service and semantic information related to the operations in the form of pre and post-conditions.

The WS-ServiceGroup specification can express group, membership rules, membership constraints and classifications using the resource property model from the WS-ResourceProperties. Groups can be defined as a collection of members that meet some constraints as expressed through resource properties. Membership Policy is constrained in the WS-ServiceGroup as MembershipContentRule defined either in the form of WSDL portType or WS-ResourceProperties. The use of XML Schema and WS-ResourceProperties facilitates further elaboration the membership policy by coupling semantics with MembershipContentRule. Thus WS-RF communities can be formed on the basis of semantic defining the application data, application operations or both of them without any additional support from the WS-RF engines. The default reliance of WS-RF on XPath as query language and adding the semantic in the form of WS-ResourceProperties make it easier to query the semantic information and relate it to the local knowledge base.

5. PROTOTYPE

Currently available P2P systems tend to use protocols which are proprietary and independent of other networks, incapable of leveraging their services. For example, Gnutella [20] defines a generic file sharing protocol and Jabber [21] defines an instant messaging protocol, but none of these protocols are interoperable. Each system creates its own P2P community, duplicating efforts in creating software primitives required by P2P systems, such as managing underlying physical network. Initial prototype for community formation [3] is developed in JXTA [23]. Details and result of our initial prototype are discussed in [3].

For later release we have re-designed the architecture to accommodate Web Services due to broader acceptance of Web Services and WS-* supporting specifications in the Grid Community. WS-RF was the automatic selection for the community formation due to its build in support for VO in the form of WS-ServiceGroup and WS-Resources. The Globus Toolkits has Monitoring and Discovery System (MDS), which defines and implements mechanisms for service and resource discovery and monitoring in distributed environments. MDS extensively uses interfaces and behavoiors defined in the WS-RF and WSN specifications and can be easily integrated with other standard implementations. MDS architecture and the relevant Web Services interfaces allow users to discover resources and services, monitor their states, receive changes in the status and visualize monitoring results. MDS can be used to implement large-scale distributed monitoring and distributed systems. MDS manages ContainerRegistryService and DefaultIndexService; and when new instance of Resource is created through Factory/Instance pattern it is added in the DefaultIndexService. Each Resource itself decides the amount of information it wants to expose; and this information is usually the subset of ResourceProperties in the WS-Resource.

The key issue in selecting the ResourceProperties to be advertised in the VO; is the one which are aligned with the WSDM-MUSE requirements. WSDM-MUSE provides uniform interface for similar WS-Resources; to query their properties and management operations. Each member within the community fulfills the membership requirement and exposes identical ResourceProperties therefore they all can be managed by single WSDM-MUSE interface which is the case with Competing Community. It is also possible that within the community their can be different type of Resources; this is the case when VO has multiple membership criteria in the form MembershipContentRule for Co-operative Community and multiple WSDM-MUSE interfaces are required for each type of resource/service.

WSDM-MUSE operations are specific to the nature of resource; due to varying nature of the resources it was not possible to have generic operations for every resource/service; although it is possible to have single WSDM-MUSE monitoring other management interfaces in a limited fashion. Standard manageable resource definitions create an integration layer between resources (Resources may be using different protocols and programming languages) and users/administrators. WSDM creates a free agent Web services proxy capable of communicating with multiple resources, breaking the classic model of the proxy being a wholly dedicated to single resource/service.

The main benefit of using WSDM is its "Co-relatable Properties" which defines the properties that together determine if two manageable resources with different identities are still the same resource. Concept of "Co-relatable Properties" helps to manage same resource/service which can be in different VO's and is managed by multiple independent policies. Each ResourceProperty itself is the Notification Topic; and VO and management interface subscribe for notification for any update in the state. In case of Community of Communities top level VO gathers information from constituting VOs at regular intervals (retrieving information from its local index); it also subscribes to the changes in the local index. In MDS and WSDM information related to each resource/service is kept as ResourceProperties; which makes it possible to utilize notification mechanism for dynamic discovery and monitoring leading to soft state nature of registration, indexing and monitoring. The extensive use of XML Schema to declare data makes WS-ResourceProperties as the ideal candidate for attaching the semantic to the resource and service. Semantic information is coupled with the service as WS-ResourceProperties means possibility to manage/update the semantic at run time through general purpose WS-ResourceProperties operation life add, update and delete. In the prototype flexibility in the WS-ResourceProperties is achieved by heavy use of data type "xsd:any", to bind any type of data with the WS-ResourceProperties for semantic information.

6. CONCLUSION

In this paper we have presented the concept of categorizing peers in communities on the basis of their expertise and interests. We have suggested Web Services based Community formation due to its broader acceptance, platform independence and interoperability. Vanilla Web Services lack the notion of state and stateful interactions, resource lifecycle management, notification of state changes, and support for sharing and coordinated use of diverse resources in dynamic 'virtual organizations' [4]. WS-RF specifications are designed on top of Web Services specifications to provide missing components. WS-RF along with the WSDM makes the best combination not only for community creation but for community management. Community creation and community discovery is addressed by WS-ServiceGroup Specifications and WSDM provides the missing bit of monitoring and management. WS-ServiceGroup and WSDM is powerful tool to manage dispersed resources; but their effectiveness relies on the other WS-RF specifications and Web Service Notification.

7. REFERENCES

[1] Asif Akram, Omer Rana, Structuring Peer 2 Peer Communities. The Third IEEE International Conference on Peer-to-Peer Computing. (2003) 194-195

[2] Asif Akram, Omer Rana: Organizing Service Oriented Peers Collaboration. The First International Conference on Service Oriented Computing. (2004) 451- 466

[3] Asif Akram, Omer Rana and Rob Allan: Organization of Grid Resources in Communities. First IEEE International Conference on e-Science and Grid Computing (2005) submitted
[4] Ian Foster, Carl Kesselman, Jeffrey M. Nick, Steven Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Downloadable as http://www.globus.org./research/papers/ogsa.pdf, 2002
[5] Web Services Resource Framework, Available at http://devresource.hp.com/drc/specifications/wsrf/index.jsp
[6] H.V.D. Parunak: Distributed Artificial Intelligence, in Manufacturing Experience With the Contract Net, pp. 285-310, Research Notes in Artificial Intelligence. Los Altos, CA: (Morgan Kaufmann Publishers, 1987)
[7] Rob Allan, Alison Allden, David Boyd, Rob Crouchly: Roadmap for a UK Virtual Research Environment, Report of the JISC JCSR VRE Working Group (2004) http://www.jisc.ac.uk/index.cfm?name=programme_vre
[8] Globus Toolkit, Available at http://www.globus.org/toolkit/
[9] Apache Apollo, WS-RF Project, Available at http://ws.apache.org/wsrf/
[10] GNUTELLA Web: Gnutella Home. http://www.gnutella.com/
[11] JABBER SOFTWARE FOUNDATION Jabber IM http://www.jabber.org/about/overview.html
[12] JXTA http://www.jxta.org/
[13] S. Graham, I. Robinson, and et al. Web Services Resource Framework Primer, Draft 5. Available at http://www.oasis-open.org/committees/wsrf, 2005.
[14] Web Services Distributed Management (WSDM), Available at http://devresource.hp.com/drc/specifications/wsdm/index.jsp
[15] O. Kaykova, O. Kononenko, V. Terziyan, and A. Zharko: Community Formation Scenarios in Peer-to-Peer Web Service Environment, Databases and Applications, Innsbruck, Austria
[16] Semantic Web Services Resource Framework (WSRF-S) report available at www.wsmo.org/TR/d31/v0.1/d31v01_20050808.pdf
[17] Omer Rana, Asif Akram, Rashid Al-Ali, David W.Walker, Gregor von Laszewski, Kaizar Amin: Quality of Services Based Grid Communities. In Web Services and Agent Systems. (Kluwer Academic , 2004)

Related Content