Grids for the GiG and Real Time Simulations

Geoffrey Fox, Alex Ho, Shrideep Pallickara, Marlon Pierce, Wenjun Wu

***Abstract*— We study the current architecture of the Grid and Web services and that of the Global Information Grid (GiG) with the Network Centric Operations and Warfare (NCOW) from the Department of Defense. We compare the GiG core enterprise services with those being developed for Grids (the Open Grid Services Architecture) and Web Services (so called WS-* specifications), identifying both similarities and differences. We discuss both modeling and simulation with HLA (High Level Architecture) and broad defense NCOW applications. We illustrate this analysis with an Open Geospatial Community (OGC) compatible set of Geographical Information System Grid services. We illustrate the use of Grids to efficiently support real-time simulation by an application of grids to audio-video conferencing.**

***Index Terms*— Grids, HLA, NCOW, Web Services**

I. Introduction

We look at the current requirements and architectures of the Global Information Grid (GiG) and in particular Department of Defense's NCOW Network Centric Operations and Warfare (NCOW) initiative. In Section II we review the GiG and NCOW and in Section III the analogous situation for the e-science and e-business Web Service and Grid activities [1]. In Section IV, we bring these together and suggest a strategy that allows NCOW to take good advantage of Web Service and Grid technologies that are expected to underlie future large scale commodity systems and be supported by all major vendors. Other discussions of this can be found in [2] and [3] and the latter paper is discussed in Sec. IVB with comments on simulation in IVC. Section V reviews some security considerations. Section VI describes some of our experiences in integrating grid services with high performance messaging for real-time applications to conferencing and GIS (Geographical Information System) services. We will use the term GiG (or NCOW) when referring to DoD's use of the term "Grid" and the unadorned Grid will refer to its use in industry and science (research) and the Global Grid Forum or GGF [4]. This broad meaning of a "Grid" is explained in our separate technology review [1] as Internet-scale distributed services.

Manuscript received August 6 2005. This work was supported in part by NASA and a contract from Ball Aerospace to Indiana University.

Alex Ho (alexho@anabas.com) and Geoffrey Fox (gcf@indiana.edu) are from Anabas Inc. Geoffrey Fox, Shrideep Pallickara (spallick@indiana.edu), Marlon Pierce(mpierce@cs.indiana.edu), and Wenjun Wu (wewu@indiana.edu) are from the Community Grids Laboratory at Indiana University, Bloomington, IN 47404 USA.

II. GiG and NCOW

Here we briefly define the Global Information Grid (GiG) and the Network Centric Operations and Warfare Concept NCOW for which there are many Internet accessible references including [5] and [6]. Important relevant organizations include the Association for Enterprise Integration with NetCentric services activities [7]; the Industry oriented Network Centric Operations Industry Consortium [8] and a broad based forum called the World Wide Consortium for the Grid [9].

The term GiG refers to the physical realization of the information system that supports essentially all aspects of DoD's operations. NCOW describes how the GiG is used to satisfy DoD's requirements while the NCOW RM (Reference Model) is a future-looking description of the NCOW that aids planning. The GiG architecture is consistent with the principles of general DoD studies such as the 1998 C4ISR (Command, Control, Communications, Computer Intelligence, Surveillance, Reconnaissance) architecture document [10]. The NCOW RM adopts a service oriented approach and has a lengthy description of many of the needed services. It does not however relate them in detail to the services defined by the Grid and web service communities.

The NCOW RM defines the Enterprise Information Environment for the GiG including the specification of nine Core Enterprise Services (CES) to which one will add those associated with Communities of Interest (CoI); the latter would often be called domain specific services in the Grid community. DoD also defines a set of Policy related services or Environmental Control Services ECS. The CES are:

1. *Enterprise Services Management* (ESM) : including life-cycle management
2. *Information Assurance(IA)/Security*: supporting confidentiality, integrity and availability
3. *Messaging*: in synchronous or asynchronous fashion
4. *Discovery*: searching data and services
5. *Mediation*: including translation, aggregation, integration, correlation, fusion, brokering publication, and other transformations for services and data
6. *Collaboration*: provides and controls sharing with emphasis on synchronous real-time services
7. *User Assistance*: includes automated and manual methods of optimizing the user GiG experience. NCOW services have capability interfaces which correspond to portal interfaces for the Grid.
8. *Storage*: retention, organization and disposition of all forms of data. Also a sophisticated data strategy.

9. *Application*: provisioning, operations and maintenance of applications.

III. Grid and Web Services

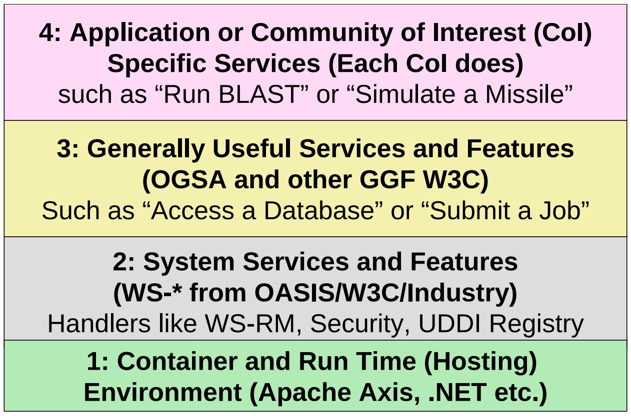

Fig. 1 The Grid and Web Service Institutional Hierarchy

Web Service-based SOA systems [11] are built on XML-based service description languages (WSDL) and message formats (SOAP). See [12] for a review of Web Service concepts. In Fig. 1, we illustrate the current institutional hierarchy of Grid services. We call it "institutional" as the four blocks of service define groups of services or service support of increasing specialization as we move up the figure. In an SOA, we are building services, their interactions (namely messages) and the support for the two fundamental concepts of messages and services. At the bottom level we have what is usually called the hosting environment, which forms the virtual machine on which we are building the "distributed service operating system" contained in the next layer. For services constructed from Java, Apache Axis is the usual container and it provides the message processing needed by the multiple services in the container.

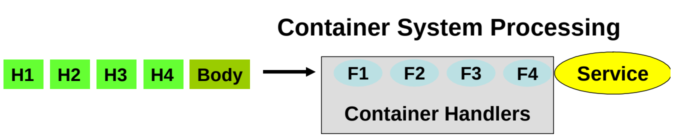

Fig. 2 Message Processing in a Container

As shown in Fig. 2, a SOAP message contains multiple headers and a body. The headers are processed by handlers controlled by the container and these capabilities are included in the second level of Fig. 1. They include operations such as security, service addressing, routing, reliability and possibly aspects of state and meta-data. One can consider handlers as the core system services for Web services. The handlers will in general modify the SOAP envelop contents, which are then formatted by the container so that it can be processed by the appropriate service instance. This instance could be implemented as a Java method corresponding to the WSDL-specified XML in the body. .NET provides similar capabilities for Microsoft environments. Note that the SOAP messages can be transported by any mechanism for which a binding can be defined. This includes the normal HTTP transport but also be message-oriented middleware to give environments like those in modern enterprise software environments such as those using MQSeries or Java Messaging Service.

The capabilities described above are needed in all aspects of Web and Grid service implementations and are also needed by both the GiG and NCOW. Several major international activities aim at setting the standards for the service and handler interfaces. The core level 1 and 2 specifications of Fig.1 are often called the WS-*. In Table 2, we list the broad areas covered by this process which involves multiple standards agencies (OASIS, W3C, GGF, DMTF) and companies such as IBM and Microsoft working inside and outside the community bodies and in different combinations. There are over 60 WS-* proposal specifications – mostly initiated in the last 3 years – with a coverage indicated in Table 1. Column 1 lists our classification of the area and a very incomplete sample of the proposed specifications for each area are given in column 2. See appendices in [1] for more information.

WS-* Specification Area	Examples
1: Core Service Model	XML, WSDL, SOAP
2: Service Internet	WS-Addressing, WS-MessageDelivery; Reliable Messaging WSRM; Efficient Messaging MOTM
3: Notification	WS-Notification, WS-Eventing
4:Workflow/Transactions	BPEL, WS-Coordination
5: Security	WS-Security, WS-Trust, SAML etc. See Sec. V
6: Service Discovery	UDDI, WS-Discovery
7:System Metadata/State	WSRF, WS-Context WS-MetadataExchange,
8: Management	WSDM, WS-Management, WS-Transfer
9: Policy and Agreements	WS-Policy, WS-Agreement
10: Portals and User Interfaces	WSRP (Remote Portlets)

Table 1: The ten areas covered by the core WS-* Specifications

Note that all handler specifications are given in one or more areas of table 1. However this table also includes system services which are broad in scope and so fit in level 2 but are not processed in handlers. Areas 4 and 10 of Table 1 correspond to workflow and user interfaces – core capabilities but not associated with handlers. This is clarified in Fig. 3 which has a functional Web service hierarchy with pervasive system services like security (termed A in figure) separated from workflow (in E for manipulating and linking services) and portals in boxes F and G.

There are many ways of classifying services, and any classification has grey areas, so it is often possible to move a service between adjacent classifications. For example, security combines handlers for individual messages with sophisticated individual services that support authentication and authorization. Meta-data exists throughout any system, and in Table 1 (area 7) and Fig. 3(A), we use the term "system metadata," which is envisaged as the equivalent of Windows registries or UNIX environments. Application metadata is equally critical and normally implemented as a queryable database resource. This would be in Fig. 1 set up as a level 4 domain specific service using a level 3 generic grid database mechanism like OGSA-DAI [13]. As a further illustration of the uncertainties in rigid classifications, Table 1 has separate entries for service discovery and metadata. In fact service

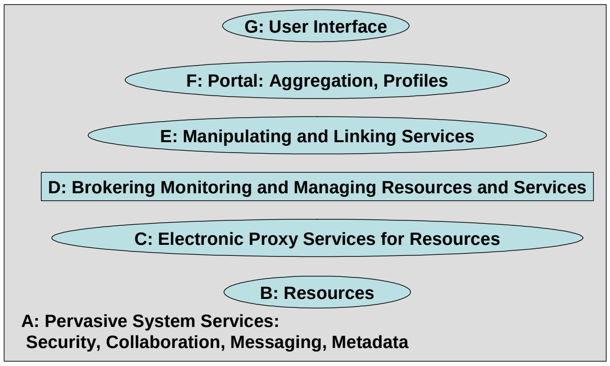

Fig. 3 The Grid and Web Service Functional Hierarchy

discovery is essentially a query to system metadata and included in grey pervasive system services in Fig. 3.

One should note that of the over 60 WS-* specifications, only a fraction have been refined into agreed standards and of these standards only a few have been broadly adopted. The WS-I, or Web Services Interoperability consortium, is the industry group developing consensus on both the accepted standards and how to use them in a set of profiles. Currently < 10% of the WS-* (namely XML, WSDL, SOAP, UDDI and parts of WS-Security) are endorsed by WS-I. There are often competing specifications for a given capability such as the example of two similar specifications WSRM and WS-Reliability for reliable messaging. We can expect that as experience grows, specifications will be updated, merged and often dropped as a stable endorsed set emerges.

Although this process illustrates the immaturity of the field, this is an open broad multi-participant activity. We can expect the resultant set of standards to be highly effective and broadly adopted, characteristics of importance to DoD, but that process that can only occur if a broad but necessarily slow process is adopted. Further, the essentials of the resultant architecture are clear – we are typically debating implementation details. Thus we argue that one can use current Grid and web services for NCOW, as this is futuristic by design.

Quality of service and autonomic self healing are critical characteristics for NCOW. There is a serious attempt to deal with this in the Grid as the Service Internet (Table 1-area 2) addresses this for messages and service management (Table 1-area 8) can be used to build autonomic services. However none of these ideas have been tested with systems involving myriad real-time streams and dynamic ad-hoc networks, which are key characteristics of DoD applications. Further, ways to achieve interoperable high performance real-time messaging are understood for web services (see Sec. VIA [14]) but again there is a paucity of implementations and little experience. We believe that this observation is rather general. Grid and web services have a promising architecture but there is little experience with some key features of DoD applications. One important aspect of the Web service architecture is that services and messages (not network drivers and packets) are the primitives. Quality of service is thus defined at a higher level than conventional network approaches and we need to understand how to make these approaches blend properly.

Grids assume that the WS-* specifications will mature and be well implemented. Then we need to design and build the services at levels 3 and 4. Correspondingly there are major efforts to design the "important general services" at level 3 and the OGSA (Open Grid Service Architecture) of the Global Grid Forum is devoted to this. Note that the community assumes that the first step is to define the open interfaces needed for interoperability for all such common services. The rationale is straightforward as these services cross many application domains (communities of interest) and will involve services from many different developers in business, government and academia. The loose coupling of web services requires no agreement on service implementation but it does require agreed interfaces so that the SOAP messaging can communicate between services from different sources. DoD has a similar motivation both to use commercial services and build interoperability frameworks within their application domain. Thus we expect that in levels 3 and 4 one needs major attention to standards for services and data. The standards in level 3 must be broadly endorsed while those in level 4 of Fig. 1 must be endorsed by the applicable community of interest. OGSA currently divides the services in level 3 into categories, which are perhaps easiest to classify in terms of the GGF areas given in first column of table 2 with some examples of their work in column 2.

GGF Area	Standards Activities
1: Architecture	High Level Resource/Service Naming (level 2 of Fig. 1), Integrated Grid Architecture
2: Applications	Software Interfaces to Grid, Grid Remote Procedure Call, Checkpointing and Recovery, Interoperability to Job Submittal services, Information Retrieval,
3: Compute	Job Submission, Basic Execution Services, Service Level Agreements for Resource use and reservation, Distributed Scheduling
4: Data	Database and File Grid access, Grid FTP, Storage Management, Data replication, Binary data, High-level publish/subscribe, Transaction management
5: Infrastructure	Network measurements, IPv6 and high performance networking, Data transport
6: Management	Resource/Service configuration, deployment and lifetime, Usage records and access, Grid economy model
7: Security	Authorization, P2P and Firewall Issues, Trusted Computing

TABLE 2: ACTIVITIES IN GLOBAL GRID FORUM WORKING GROUPS

The above table is illustrative of the international activity setting level 3 standards. There is much additional work within individual projects and organizations and within other standards organizations. A good example of the latter is the work of the Open Geospatial Consortium on Geographical Information System (GIS) services. This also illustrates that different communities might classify services into different levels. NCOW could view GIS as a universal level 3 service and job submittal (a major focus of GGF) as rather specialized and so level 4. DOD high performance computing facilities such as the High Performance Computing Modernization Program (HPCMP) may consider job submission a higher priority. Clearly the Global Grid Forum views job processing as central and so far has not looked at GIS.

IV. Comparison of NCOW with Grid and Web Services

A. *Comparison of Core Services*

In this section we will compare NCOW with the hierarchical service architecture discussed in Sec. III and specific services including those in Table 1 and Table 2.

NCOW Service or Feature	WS-* and GGF	Others
A: Broad Principles		
Use Service Oriented Architecture	Core Service Model (WS1), Build Grids on Web Services. Industry best practice	
Grid of Grids	Strategy for legacy subsystems and modular architecture	
B: NCOW Core Services		
CES 1: Enterprise Services Management	WS8, GGF6 Management	CIM
CES 2: Information Assurance(IA)/Security	WS5, GGF7 WS-Security	Grid-Shib, Permis Liberty Alliance etc.
CES 3: Messaging	WS2, WS3	JMS, MQSeries Streams/Sensors
CES 4: Discovery	WS6	
CES 5: Mediation	WS4 workflow	Legacy systems
CES 6: Collaboration	WS and GGF VO	XGSP, Shared Web Service ports
CES 7: User assistance	WS10	Portlets JSR168, Capability Interfaces
CES 8: Storage	GGF4 Data	NCOW Data Strategy
CES 9: Application	GGF2	Best Practice in Grid/Web services
Env. Control Services ECS	WS9	
Resource Infrastructure	GGF5	Ad-hoc networks
C: Key NCOW Capabilities not directly in CES		
System Meta-data	WS7, Semantic Grid, Globus MDS	Annotation
Resource/Service Matching/Scheduling	Distributed Scheduling and SLA's (GGF3)	Extend scheduling to networks and data flow
Geographical Information Systems	Not covered	OGC Standards WFS WMS etc.
Sensors (real-time data)	Not covered	OGC Sensor standards
Data-mining	Not covered	Several Examples

Table 3: Comparison of NCOW Services with Web and Grid Services

Notes to Table 3: WS*n* (*n*=1..10) refer to rows of table 1; GGF*n* (*n*=1..7) refer to rows of table 2. VO refers to Virtual Organization technology which includes Security (WS5, GGF7), Naming (GGF1), Meta-data (WS7). VO supports asynchronous collaboration

Table 3 shows a comparison of "core" services for NCOW in column 1, Grids and Web Services (tables 1 and 2) in column 2 and other sources in column 3. This mapping is as always not without its uncertainties and confusions – partly because although WS-* specifications are all defined precisely (as we only used those that were), the NCOW and GGF areas are often more qualitative as they include futuristic studies. Under NCOW in column 1, we list the nine core services listed in section II together with the ECS (Environmental Control Services). Column 1 of Table 3C has three additional broad areas (resource/service matching and scheduling, meta-data and sensors) that appear important but not clearly part of the ECS and CES. We also added typical important NCOW application (level 4 in Fig. 1) services – namely GIS in Sec. VIB and data-mining.

We first note that in column 1 all the NCOW core services are mentioned once (by definition) while column 2 and 3 contain at least one entry for all of the GGF and WS-* areas. We needed to add the scheduling and meta-data entries in column 1 to make this happen. We emphasize that the NCOW undoubtedly has identified these two extra areas in their full analysis but they are not explicitly in the CES. We suggest that these types of capability are so important that should be classified as "core".

We suggest that table 3 shows rather clearly that the NCOW can be built with an architecture like that discussed in Section III. This suggests one should adapt for DoD some of the strategies used in the Grid and Web services community. This includes critical looks at existing level 2 and 3 (of Fig. 1) specifications and the start of OGSA like activities to define level 3 standards needed for event-based streaming information grids and to satisfy other special DoD needs. We also need to ensure that CoI standards are developed in a coherent fashion in the context of the core level 2 and 3 capabilities and interoperability standards.

We have discussed the Grid of Grids concept at length in [1] and suggest this is one approach to legacy systems. One cannot realistically build a single monolithic grid with all services compliant to the same set of standards. We will always need to federate systems together with different internals. One structures each system as an individual grid and using mediation technology (with at its simplest mapping of the messages in the SOAP infrastructure) federates them together as a Grid of Grids.

B. *Evaluation of Service Oriented Architecture for NCOW*

We can gain further insight by comparing our analysis with that of [3], which provides a very interesting analysis of web services for net-centric military applications. This important paper discusses grids indirectly but only in terms of the Globus capabilities [15], whose latest release corresponds to a Web service framework with the addition of WS-RF to allow services to be associated with resources and their properties. In a nutshell, the weaknesses identified in this paper are important but appear to be addressed either by Grid ideas or are features of particular implementations of web services. In general we suggest that the use of a powerful messaging infrastructure to support web service messaging, systematic use of WS-Policy and its derivatives to provide system wide QoS and deployment of powerful meta-data catalogs address many of the concerns. The particular areas identified in [3] are discussed in the following sub-sections.

1) *Information Architecture and Service Description*

As described in table 1, web services provide a framework for presenting information, and through UDDI and the metadata specifications (WS-Metadata, WSRF, RDF, etc.), they provide frameworks for representing metadata including service information. The specifications for representing data is outside the WS-* of table 1. We would not view this as a limitation of web services but rather a feature. As discussed in tables 2 and 3, the very active data area of the GGF and the OGC are defining frameworks for databases and sensors respectively. The grid architecture suggests that each domain should build on these frameworks to define their own data and metadata syntaxes. DDMS, XMSF and the C4ISR core architecture data model are examples of this in practice within the DoD [10,16,17]. Other interesting DoD domain specific standards include IEEE 1516 (HLA [18]), C2IEDM, XBML [17] for simulations and operational services and the use of XML-based graphics standards. In Sec. 4.1 above, we recommended that the DoD start OGSA-like activities to define the service and data specifications that are needed to support NCOW. We believe this web service data approach shows a good architectural separation and is the appropriate way to build NCOW and so is not a problem with web services but rather an example of their well thought out layered architecture.

2) Document-Centric Nature of Web Services

[3] notes that services exchange self describing messages specified in XML and that this implies messages can be considered documents and so that one has a document centric architecture where each message can in general be totally different and large in size. The self-describing message architecture of web services leads to the loose coupling of the services as if needed each message can be processed independently without reference to a complex and fragile context. Further, although the messages are independent and could each be produced with different principles, this is not recommended best practice. As emphasized in the core architecture discussions [12], one can assert system wide strategies with WS-Policy and its growing number of derivatives including special policy languages for security and reliable messaging. Thus each message is created by a service and this service should use a policy (valid over an appropriate domain with in general many services with a common policy) to decide on how to create a coherent set of messages that will realize the desired system QoS. The different policy statements can be kept in metadata catalogs that define their scope and content. One does of course need to set up administrative services to define these policies which could be part of the NCOW Environmental Control Services sector in table 3.

We have discussed in [1] that SOAP in its simplest “angle bracket” representation can be an inefficient messaging system. However by binding SOAP to a sophisticated messaging infrastructure like NaradaBrokering and by choosing appropriate Infoset-preserving representations, one can get very high performance secure fault tolerant messaging. We deduce that web service messaging could lead to problems, but good choice of architecture and implementation can lead to messaging that is more powerful than that in previous distributed systems.

3) Time-critical Events

Web services do not have any special support for time critical events and this needs to be added for both simulation (Sec IVC) and mission critical operations. We have developed sophisticated time-sensitive support for a service oriented audio-video conferencing environment [19] described in Sec. VIC with the infrastructure having 1-2 ms overhead with NTP used to create a global consistent time. This shows that one can straightforwardly meet the roughly 30ms timing constraints needed in this application. Other time critical applications need sophisticated scheduling capabilities discussed in Sec IVA. This difficult area needs to be addressed but we expect the grid/web service architecture to have no special problems.

4) Life Cycle Support for Services

The WS-I agreed Web service specifications have no identified support for life cycle management, and currently this is left to each implementation. The WS-RF framework [15] does however include a resource lifetime specification that can be used to support life cycle management. The management specifications, especially WSDM, also address this area. Currently there is controversy in the web service community as to how much should be universally specified and how much should be left to the individual developer. This a healthy debate, and we expect it to continue and that eventually more areas will develop detailed specifications. Thus life cycle support will quite likely get specified and broadly agreed but we are not there yet. DoD needs to identify its needs, study existing approaches such as WS-RF/WSDM and adopt a flexible strategy that can align itself with industry best practice.

5) Reliable Messages

There are weaknesses in the current reliable messaging specifications especially in the lack of support for multicast destinations. The new WS-RM Policy specification [1] can provide coherence to the use of these specifications. Achieving more efficient multicast messaging (currently the source must immediately send a separate message to each recipient) is consistent with the web service architecture but not its current implementation. Systems like NaradaBrokering (Sec. VIA) support efficient multicast algorithms, and one can modify WS-Addressing to allow virtual topic-based addressing and to enable the Addressing handlers on SOAP intermediaries to implement the optimal publish-subscribe routing algorithms. Software routing through brokers or intermediaries in SOAP is in its infancy and we can expect such capability to appear in the future.

6) Security

We discuss security in Sec. V and in [1] but note here that we do not agree that there are any special problems with the interaction of web service security with the other specifications like notification. This is agreed to be a critical area where much more work and experience is needed.

7) Scalability

[3] suggests that there will be difficulties in scaling web services especially in areas of QoS including reliable messaging. We disagree with the premise of this paper on the disadvantages of the message-based architecture. As discussed above, one uses system wide policy to produce a coherent QoS architecture and use the self describing messages to enable loose coupling and hence scaling of the system.

C. Modeling and Simulation: HLA and RTI

Modeling and simulation is of critical importance to the

DoD and the Defense Modeling and Simulation Office DMSO [20] has developed a very sophisticated distributed object-based framework termed HLA or High Level Architecture [18]. This framework is supported by the RTI Runtime Infrastructure middleware. HLA defines a model (architecture) for simulation built from distributed objects (DO). As in conventional in DO systems there are well defined HLA interfaces but unlike for web services the "wire" representation of the communication between objects is left unspecified. RTI implements these interfaces with a publish-subscribe communication infrastructure. HLA is built from objects and interactions (roughly transient objects) organized into federates. Federates could be as small as a single object or as large as a complete simulation. HLA does not address the structure of federates but rather the mutual interaction of a set of federates that form a federation; there can be many simultaneous federations but we are interested in defining a simulation specified by the federates in a single federation; RTI controls the information transfer between the objects, various management tasks such as creating a federate and supports various models for time and space that ensure a faithful simulation respecting geometry and temporal ordering. We can highlight 6 areas of the HLA Specification that RTI must support: Declaration Management, Data Distribution Management, Time Management, Object Management, Ownership Management and Federation Management

Now let us consider how we can use Grid and Web services to implement the HLA specification. The HLA model is consistent with the "Grid of Grids" concept for we can view each federate or each federation as its own grid and then our Grid RTI will link the separate federate grids into a federation grid. Suppose we imagine taking each HLA interface and define the corresponding WSDL; this is straightforward but what does it imply with respect to the analysis above. The final 3 RTI areas (object, ownership and federation management) are classic level 3 services in the classification of Fig. 1; they are generally useful services in the management of HLA distributed objects. Ownership management for example supports the movement of objects and their attributes between federates while object and federation management support functions such as the creation and update of the managed entities. We wonder if supporting such object operations for the Grid could make use of WS-RF [1.15] whose goals include support for distributed object concepts with resources and their properties.

Declaration, Data Distribution and Time management are more interesting here as they correspond to lower level system services. Declaration management essentially specifies a rich publish/subscribe model for object and interaction attributes that is the data exchange model in HLA. It seems likely that one could reproduce this with WS-Eventing/Notification although the rich functionality will need additional filters and services. For example, our implementation [14] of WS-Eventing uses its extensibility to support not only the default XPATH but also topic and regular expression based subscriptions. Data distribution management in RTI supports a publish/subscribe model that includes general geometric constraints; the latter could be added as a special filter to WS-Eventing/Notification. Some sophisticated HLA features such as publishers collaborating for a single declaration and the flow control to throttle delivery can be implemented in particular grid middleware like NaradaBrokering but are outside the current web service specifications. Time management supports the different types of simulation (time-stepped, optimistic or conservative event driven) but it is not present in the current web service specifications and one would have to build this as a set of services. This general support of simulations would be a valuable addition to the level 3 Grid/web service specifications. It should be linked to Grid workflow (managing all federates in a federation) of which simulation can be considered a special case. We should try not only to link typical HLA simulations but also tightly coupled parallel simulations and the impressive agent-based simulations like Transims [21] which are being extended to all the critical infrastructures [22]. Again middleware like NaradaBrokering has significant support for timing (see Sec. IVB(3)) and could be important in implementing such Grid simulation support.

We find the lack of overlap between HLA interfaces and table 3 significant. There was only synergies in the publish-subscribe notification area. This tells us that by adding grids to HLA, we are going to achieve greater capability as we will allow any simulation access to the rich range of grid and web services available for Grid systems. Equally important we suspect many RTI implementations must include many of the capabilities in table 3; we assume these typically use proprietary inconsistent solutions. If one agrees to build HLA on top of Grid and Web service technology, we will immediately generate infrastructure with substantially greater interoperability.

Above we gave a general analysis of the relationship between Web/Grid technologies and those specified for HLA in IEEE 1516. A complete integration of these technologies would be a major but possible activity. Fox and Furmanski built an early system like this with a Java/CORBA implementation (JWORB) of DMSO RTI 1.3 and applied it through the DoD HPCMP PET program [23] as WebHLA [24,25]. JWORB did not fully implement all the complex time management schemes in HLA and was not deployed in production. Here we describe several interesting projects tackling parts of the problem using today's web and grid services.

The major XMSF project [17] is developing Web service interfaces for federates and using these to provide important services – especially visualization – and web service messaging with high performance extensions [26,27]. One should be able to link to XML enabled visualization systems such as ICE from Army Research Laboratory [28]. In this approach one is not redoing HLA but rather making its external links interoperable to Grids and Web Services. The Singapore group has integrated the Globus system [29] with

HLA in a very interesting way using proxies [30]. They are able to support "federates-on-demand" with Globus supplying the link between distributed federates with some created dynamically in a Grid computer pool. Note RTI is wrapped in the communication but remains essentially unchanged in its interaction with the simulation objects. A European project has pursued a similar goal of using Grid technology to migrate federates [31].

V. DOD Security Requirements for Grid Web Services

As discussed in Section II, Web Services are messaging-centric, with modular, extensible SOAP message envelops used for message packaging. This has some interesting implications for the GiG. First, XML Encryption allows different parts of the message to be encrypted differently. Thus, a single message may contain parts that are readable by different groups. Thus secure SOAP messages may be directly mapped to access controls and user roles. The same message may be sent to different recipients with, for example, "public", "itar", "restricted", and similar subsections. Second, secure SOAP messages are self-contained and self-descriptive. This allows secure messages to be stored persistently and independently of particular network connections. Advanced qualities of service, such as replay, message archive, and reliable messaging systems make use of this capability.

The largest drawback to Web Service security is its current lack of performance and lack of implementations that use SOAP intermediaries. For recent reviews, see [32] and [33]. Most secure Web Service Grids focus on remote-procedure style request-responses to invoke remote applications and binary-channel data transfers, so the use of intermediaries and optimized message performance is not critical: application runtimes or binary data transfer operations dominate the execution time. We may also assume that the actual SOAP message traffic in these systems is relatively small. SOAP-based information systems are very different. RPC-style Grids use application-centric messages to accomplish tasks, but in information systems, a large percentage of messages are human-centric (i.e. are to be rendered in human-comprehensible form, rather than machine comprehensible form). Information systems have significantly higher amounts of message traffic, and much of this traffic is time-critical. It is therefore essential that ways of increasing performance for message processing and transport. Efficient XML Infoset-based representations [34] can greatly improve performance and should be investigated.

VI. Activities in the Community Grids Laboratory

In this section, we review some relevant research activities at the Community Grids Laboratory.

A. *NaradaBrokering*

NaradaBrokering (NB) [14] is a topic-based publish/subscribe system that supports numerous features including messaging for Grid and web services. All communications (ranging from SOAP+XML messages to audio and video streams) are treated as messages that can be routed by NB. NB thus provides a general purpose messaging substrate that can be used to provide higher level quality of service.

- The brokers are distributed and organized hierarchically for efficient, globally scalable message routing.
- NB supports structured topic names and content-based subscription.
- NB links can support several different connection and transport protocols, including TCP/IP, parallel TCP/IP, UDP, SSL, multicast, HTTP, etc.
- NB implements numerous qualities of service for messages, including reliable message delivery, message archiving and replay, message fragmentation/coalescence, and end-to-end security.
- Web Service support for SOAP message routing, WS-Reliable Messaging, and WS-Eventing are available today and will be augmented with other web service messaging handlers.

Geographical information and collaboration services are two examples using NaradaBrokering discussed below.

B. *Geographical Information System Grid Services*

We are implementing [35] Web Service versions of several Geographical Information System services specified by the Open Geospatial Consortium. These include a) the Web Feature Service, which can be used to store and retrieve archived geospatial data; b) the Web Map Service, which can be used to render features into human-comprehensible maps; c) Information Services, which can be used by clients to find service instances with appropriate capabilities; and d) Sensor Enablement Services (based on SensorML), which can be used to transport and filter real-time data.

Recent work has focused on a) using NaradaBrokering to filter Global Positioning System data in real-time; b) creating higher performance versions of feature and map services, which don't rely upon HTTP for SOAP transport; and c) streaming video versions of the map server, which can be integrated into collaboration systems such as GlobalMMCS.

C. *Collaboration Grids: GlobalMMCS and XGSP*

In our lab research in collaboration technologies, we design and build the generic service infrastructure that can be used to bridge between different audio/video collaboration systems [19]. XGSP is an XML-based general-purpose session management layer that supports the collaboration CES in table 3. GlobalMMCS implements XGSP to support numerous services (frame buffer capturing, Polycom bridging, Access Grid bridging, etc) to deliver A/V to various clients. Specialized GlobalMMCS clients are in development that can be used to capture and annotate images from video streams using shared white boards.

VII. Conclusion

We have related current Grid and Web services to DoD's general NCOW and HLA simulation areas looking explicitly at core services. We see requirements in the real-time arena that have not been studied in detail from the Web services point of view. Nevertheless we see important advantages for DoD to adopt Grid and Web services and some clear action items in defining important services synergistic with those already considered by industry and research.

References

[1] Geoffrey Fox, Alex Ho, Marlon Pierce *Collection of Grid Resources for DoD Applications*, Internal Reports June 2005,. http://grids.ucs.indiana.edu/ptliupages/publications/gig/. Major Grid references include *The Grid 2: Blueprint for a new Computing Infrastructure*, edited by Ian Foster and Carl Kesselman, Morgan Kaufmann 2004 and *Grid Computing: Making the Global Infrastructure a Reality* edited by F. Berman, G. Fox and A. Hey, Wiley, March 2003.

[2] Yun-Tung Lau , Service-Oriented Architecture and the C4ISR Framework, http://www.stsc.hill.af.mil/crosstalk/2004/09/0409Lau.html

[3] Ken Birman, Robert Hillman, Stefan Pleisch, *Building network-centric military applications over service oriented architectures* SPIE Conference DEFENSE TRANSFORMATION AND NETWORK-CENTRIC SYSTEMS Orlando Florida 31 March 2005.http://www.cs.cornell.edu/projects/quicksilver/public_pdfs/GIGonWS_final.pdf

[4] Global Grid Forum http://www.gridforum.org

[5] Rick Hayes-Roth *Comments on NCOW RM 1.01*, October 19 2004 http://www.w2cog.org/documents/RHR_comments_re_NCOW_RM_1.01.doc

[6] Bill Levitt, NCOW RM Development Group *Update on Target Technical View - Emerging Net-Centric Standards - NCOW Reference Model v1.1* The Open Group Conference January 25, 2005, San Francisco http://www.opengroup.org/gesforum/uploads/40/6574/NCOW_TTV_V1.1_Open_Group.ppt.

[7] AFEI (Association for Enterprise Integration) NetCentric Enterprise Services Workshops http://www.afei.org/news/NCES_Workshops.cfm.

[8] NCOIC Network Centric Operations Industry Consortium http://www.ncoic.org/

[9] W2COG World Wide Consortium for the Grid http://www.w2cog.org/

[10] Command, Control, Communications, Computer Intelligence Surveillance Reconnaissance (C4ISR) Core Architecture Data Model Version 2.0 http://www.fas.org/irp/program/core/fnlrprt.pdf

[11] Booth, D. et al.. "Web Service Architecture." W3C Working Group Note, 11 February 2004. http://www.w3c.org/TR/ws-arch

[12] Sanjiva Weerawarana, Francisco Curbera, Frank Leymann, Tony Storey, Donald F. Ferguson, *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*, Prentice Hall March 22, 2005, ISBN: 0-13-148874-0.

[13] OGSA-DAI Grid and Web database interface http://www.ogsa-dai.org/.

[14] NaradaBrokering Messaging System http://www.naradabrokering.org

[15] Globus Toolkit GT4 http://www.globus.org/toolkit/docs/4.0/.

[16] Department of Defense Discovery Metadata Standard (DDMS) Version 1.2 http://diides.ncr.disa.mil/mdreg/user/DDMS.cfm

[17] Extensible Modeling and Simulation Framework (XMSF) Project http://www.movesinstitute.org/xmsf/xmsf.html .

[18] High Level Architecture HLA https://www.dmso.mil/public/transition/hla/

[19] GlobalMMCS Service oriented Collaboration Environment from Community Grids Laboratory http://www.globalmmcs.org.

[20] Defense Modeling and Simulation Office DMSO https://www.dmso.mil/public/

[21] C. Barrett et al.. TRANSIMS: Transportation Analysis Simulation System. Technical Report LA-UR-00-1725, Los Alamos National Laboratory Unclassified Report, 2001. See http://transims.tsasa.lanl.gov/ and http://www.transims.net/.

[22] S. Eubank et al.,.*Modeling Disease Outbreaks in Realistic Urban Social Networks*, Nature, 429, pp. 180-184, May (2004).

[23] High Performance Computing Modernization Program (HPCMP) http://www.hpcmo.hpc.mil/.

[24] Bernholdt, D., Fox, G., Furmanski, W. Natarajan, B., Ozdemir, H., Ozdemir, Z., and Pulikal, T., ``WebHLA - An Interactive Programming and Training Environment for High Performance Modeling and Simulation", in *Proceedings of the DoD HPC 98 Users Group Conference*, April 30, 1998. http://www.new-npac.org/users/fox/documents/furmpapers/paper20.html

[25] Fox, G., Furmanski, W., and Ozdemir, H., ``Object Web (Java/CORBA) based RTI to Support Metacomputing M&S," in *Proceedings of the International Test and Evaluation Workshop on High Performance Computing*, June 10, 1998. http://www.new-npac.org/users/fox/documents/furmpapers/owrtipaper16.html

[26] D. M. Moen and J. M. Pullen, *Enabling real-time distributed virtual simulation over the internet using host-based overlay multicast*, in Proceedings of the IEEE/ACM Distributed Simulation-Real Time Application Symposium, 2003, pp. 30–36. http://netlab.gmu.edu/XMSF/pubs/ds-rt03_moen-pullen.pdf

[27] J. Mark Pullen, Ryan Brunton, Don Brutzman, David Drake, Michael Hieb, Katherine L. Morse, Andreas Tolk. *Using Web Services to Integrate Heterogeneous Simulations in a Grid Environment*. To appear in Proceedings of the International Conference on Computational Science 2004, Krakow, Poland http://www.vmasc.odu.edu/publications/tolk/DS-GRID04-10.pdf

[28] Jerry A. Clarke and Raju R. Namburu, "A Distributed Computing Environment for Interdisciplinary Applications", Concurrency and Computation: Practice and Experience Vol. 14, Grid Computing environments Special Issue 13-15, pages 1161-1174, 2002.

[29] Globus Project http://www.globus.org.

[30] Yong Xie, Yong Meng Teo, Wentong Cai and Stephen J Turner. "Service Provisioning for HLA-based Distributed Simulation on the Grid", in Procs of the *19th IEEE/ACM/SCS Workshop on Principles of Advanced and Distributed Simulation* (PADS 2005), pp.282-291, Monterey, California, USA, June 2005 http://www.comp.nus.edu.sg/~xieyong/publication/PADS2005_xie.pdf

[31] K. Zajac, M. Bubak, M. Malawski, and P. M. A. Sloot, *Towards a grid management system for HLA-based interactive simulations*, in Proceedings of 7th IEEE Intl Symposium on Distributed Simulation and Real Time Applications (DS-RT 2003), S. Turner and S. Taylor, Eds. Delft, The Netherlands: IEEE Computer Society, October 2003, pp. 4–11.

[32] Marty Humphrey et al., "State and Events for Web Services: A Comparison of Five WS-Resource Framework and WS-Notification Implementations." HPDC 14, July 24-27, 2005. http://www.caip.rutgers.edu/hpdc2005/index.html

[33] Francois Taiani, Matti A. Hiltunen, and Richard D. Schlichting, "The Impact of Web Service Integration on Grid Performance." HPDC 14, July 24-27, 2005. http://www.caip.rutgers.edu/hpdc2005/index.html

[34] Sangyoon Oh, Hasan Bulut, Ahmet Uyar, Wenjun Wu, Geoffrey Fox *Optimized Communication using the SOAP Infoset For Mobile Multimedia Collaboration Applications* Proceedings of the International Symposium on Collaborative Technologies and Systems CTS05 May 2005, St. Louis Missouri, USA. http://grids.ucs.indiana.edu/ptliupages/publications/OptSOAP_CTS05.pdf

[35] GIS Grid Services http://www.crisisgrid.org