

The Semantic Grid: A Future e-Science Infrastructure

David De Roure, Nicholas R. Jennings and Nigel R. Shadbolt¹

Dept of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK

{dder,nrj,nrs}@ecs.soton.ac.uk

Abstract

e-Science offers a promising vision of how computer and communication technology can support and enhance the scientific process. It does this by enabling scientists to generate, analyse, share and discuss their insights, experiments and results in an effective manner. The underlying computer infrastructure that provides these facilities is commonly referred to as the Grid. At this time, there are a number of grid applications being developed and there is a whole raft of computer technologies that provide fragments of the necessary functionality. However there is currently a major gap between these endeavours and the vision of e-Science in which there is a high degree of easy-to-use and seamless automation and in which there are flexible collaborations and computations on a global scale. To bridge this practice–aspiration divide, this paper presents a research agenda whose aim is to move from the current state of the art in e-Science infrastructure, to the future infrastructure that is needed to support the full richness of the e-Science vision. Here the future e-Science research infrastructure is termed the *Semantic Grid* (Semantic Grid to Grid is meant to connote a similar relationship to the one that exists between the Semantic Web and the Web). In particular, we present a conceptual architecture for the Semantic Grid. This architecture adopts a service-oriented perspective in which distinct stakeholders in the scientific process, represented as software agents, provide services to one another, under various service level agreements, in various forms of marketplace. We then focus predominantly on the issues concerned with the way that knowledge is acquired and used in such environments since we believe this is the key differentiator between current grid endeavours and those envisioned for the Semantic Grid.

1. Introduction

Scientific research and development has always involved large numbers of people, with different types and levels of expertise, working in a variety of roles, both separately and together, making use of and extending the body of knowledge. In recent years, however, there have been a number of important changes in the nature and the process of research. In particular, there is an increased emphasis on collaboration between large teams, an increased use of advanced information processing techniques, and an increased need to share results and observations between participants who are not physically co-located. When taken together, these trends mean that researchers are increasingly relying on computer and communication technologies as an intrinsic part of their everyday research activity. At present, the key communication technologies are predominantly email and the Web. Together these have shown a glimpse of what is possible; however to more fully support the e-Scientist, the next generation of technology will need to be much richer, more flexible and much easier to use. Against this background, this paper focuses on the

¹ The authors are listed alphabetically.

requirements, the design and implementation issues, and the research challenges associated with developing a computing infrastructure to support future e-Science.

The computing infrastructure for e-Science is commonly referred to as *the Grid* [Foster98] and this is, therefore, the term we will use here. This terminology is chosen to connote the idea of a ‘power grid’: namely that e-Scientists can plug into the e-Science computing infrastructure like plugging into a power grid. An important point to note however is that the term ‘grid’ is sometimes used synonymously with a networked, high performance computing infrastructure. While this aspect is certainly an important enabling technology for future e-Science, it is only a part of a much larger picture that also includes information handling and support for knowledge processing within the e-scientific process. It is this broader view of the e-Science infrastructure that we adopt in this document and we refer to this as the *Semantic Grid* [DeRoure2001]. Our view is that as the Grid is to the Web, so the Semantic Grid is to the Semantic Web [BernersLee99, BernersLee01]. Thus the Semantic Grid is characterised as an open system in which users, software components and computational resources (all owned by different stakeholders) come and go on a continual basis. There should be a high degree of automation that supports flexible collaborations and computation on a global scale. Moreover, this environment should be personalised to the individual participants and should offer seamless interactions with both software components and other relevant users².

The grid metaphor intuitively gives rise to the view of the e-Science infrastructure as a set of services that are provided by particular individuals or institutions for consumption by others. Given this, and coupled with the fact that many research and standards activities are embracing a similar view (e.g., [WebServices01]), we adopt a *service-oriented view* of the Grid throughout this document (see section 2 for a more detailed justification of this choice). This view is based upon the notion of various *entities* (represented as software agents) providing *services* to one another under various forms of *contract* (or service level agreement) in various forms of *marketplace*.

Given the above view of the scope of e-Science, it has become popular to characterise the computing infrastructure as consisting of three conceptual layers³:

- **Data/computation**

This layer deals with the way that computational resources are allocated, scheduled and executed and the way in which data is shipped between the various processing resources. It is characterised as being able to deal with large volumes of data, providing fast networks and presenting diverse resources as a single metacomputer. The data/computation layer builds on the physical ‘grid fabric’, i.e. the underlying network and computer infrastructure,

² Our view of the Semantic Grid has many elements in common with the notion of a ‘collaboratory’ [Cerf93]: a centre without walls, in which researchers can perform their research without regard to geographical location - interacting with colleagues, accessing instrumentation, sharing data and computational resource, and accessing information in digital libraries. We extend this view to accommodate ‘information appliances’ in the laboratory setting, which might, for example, include electronic logbooks and other portable devices.

³ The three layer grid vision is attributed to Keith G. Jeffery of CLRC, who introduced it in a paper for the UK Research Councils Strategic Review in 1999.

which may also interconnect scientific equipment. Here data is understood as uninterpreted bits and bytes.

- **Information**

This layer deals with the way that information is represented, stored, accessed, shared and maintained. Here information is understood as data equipped with meaning. For example the characterisation of an integer as representing the temperature of a reaction process, the recognition that a string is the name of an individual.

- **Knowledge**

This layer is concerned with the way that knowledge is acquired, used, retrieved, published and maintained to assist e-Scientists to achieve their particular goals and objectives. Here knowledge is understood as information applied to achieve a goal, solve a problem or enact a decision. In the Business Intelligence literature knowledge is often defined as actionable information. For example, the recognition by a plant operator that in the current context a reaction temperature demands shutdown of the process.

There are a number of observations and remarks that need to be made about this layered structure. Firstly, all grids that have or will be built have some element of all three layers in them. The degree to which the various layers are important and utilised in a given application will be domain dependent - thus, in some cases, the processing of huge volumes of data will be the dominant concern, while in others the knowledge services that are available will be the overriding issue. Secondly, this layering is a conceptual view on the system that is useful in the analysis and design phases of development. However, the strict layering may not be carried forward to the implementation for reasons of efficiency. Thirdly, the service-oriented view applies at all the layers. Thus there are services, producers, consumers and contracts at the computational layer, at the information layer and at the knowledge layer (figure 1).

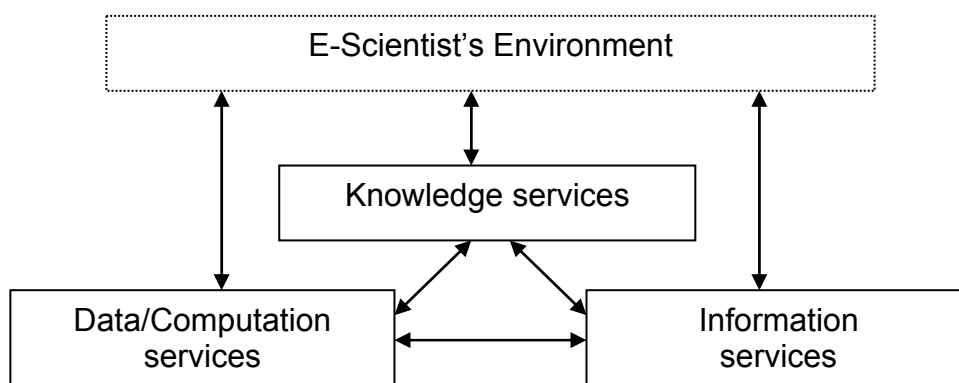


Figure 1: Three layered architecture viewed as services

Although this view is widely accepted, to date most research and development work in this area has concentrated on the data/computation layer and on the information layer. While there are still many open problems concerned with managing massively distributed computations in an efficient manner and in accessing and sharing information from heterogeneous sources (see the companion paper [DeRoure02] for

more details), we believe the full potential of grid computing can only be realised by fully exploiting the functionality and capabilities provided by knowledge layer services. This is because it is at this layer that the reasoning necessary for seamlessly automating a significant range of the actions and interactions takes place. Thus this is the area we focus on most in this paper.

The remainder of this paper is structured in the following manner. Section 2 provides a motivating scenario of our vision for the Semantic Grid. Section 3 provides a justification of the service-oriented view for the Semantic Grid. Section 4 concentrates on knowledge services. Section 5 concludes by presenting the main research challenges that need to be addressed to make the Semantic Grid a reality.

2. A Semantic Grid Scenario

To help clarify our vision of the Semantic Grid, we present a motivating scenario that captures what we believe are the key characteristics and requirements of future e-Science environments. We believe this is more instructive than trying to produce an all embracing definition.

This scenario is derived from talking with e-Scientists across several domains including the physical sciences. It is not intended to be domain-specific (since this would be too narrow) and at the same time it cannot be completely generic (since this would not be detailed enough to serve as a basis for grounding our discussion). Thus it falls somewhere in between. Nor is the scenario science fiction – these practices exist today, but on a restricted scale and with a limited degree of automation. The scenario itself (figure 2) fits with the description of grid applications as “coordinated resource sharing and problem solving among dynamic collections of individuals” [Foster01].

The sample arrives for analysis with an ID number. The technician logs it into the database and the information about the sample appears (it had been entered remotely when the sample was taken). The appropriate settings are confirmed and the sample is placed with the others going to the analyser (a piece of laboratory equipment). The analyser runs automatically and the output of the analysis is stored together with a record of the parameters and laboratory conditions at the time of analysis.

The analysis is automatically brought to the attention of the company scientist who routinely inspects analysis results such as these. The scientist reviews the results from their remote office and decides the sample needs further investigation. They request a booking to use the High Resolution Analyser and the system presents configurations for previous runs on similar samples; given this previous experience the scientist selects appropriate parameters. Prior to the booking, the sample is taken to the analyser and the equipment recognizes the sample identification. The sample is placed in the equipment which configures appropriately, the door is locked and the experiment is monitored by the technician by live video then left to run overnight; the video is also recorded, along with live data from the equipment. The scientist is sent a URL to the results.

Later the scientist looks at the results and, intrigued, decides to replay the analyser run, navigating the video and associated information. They then press the “query” button and the system summarises previous related analyses reported internally and externally, and recommends other scientists who have published work in this area. The scientist finds that their results appear to be unique.

The scientist requests an agenda item at the next research videoconference and publishes the experimental information for access by their colleagues (only) in preparation for the meeting. The meeting decides to make the analysis available for the wider community to look at, so the scientist then logs the analysis and associated metadata into an international database and provides some covering information. Its provenance is recorded. The availability of the new information prompts other automatic processing and a number of databases are updated; some processing of this new information occurs.

Various scientists who had expressed interest in samples or analyses fitting this description are notified automatically. One of them decides to run a simulation to see if they can model the sample, using remote resources and visualizing the result locally. The simulation involves the use of a problem solving environment (PSE) within which to assemble a range of components to explore the issues and questions that arise for the scientist. The parameters and results of the simulations are made available via the public database. Another scientist adds annotation to the published information.

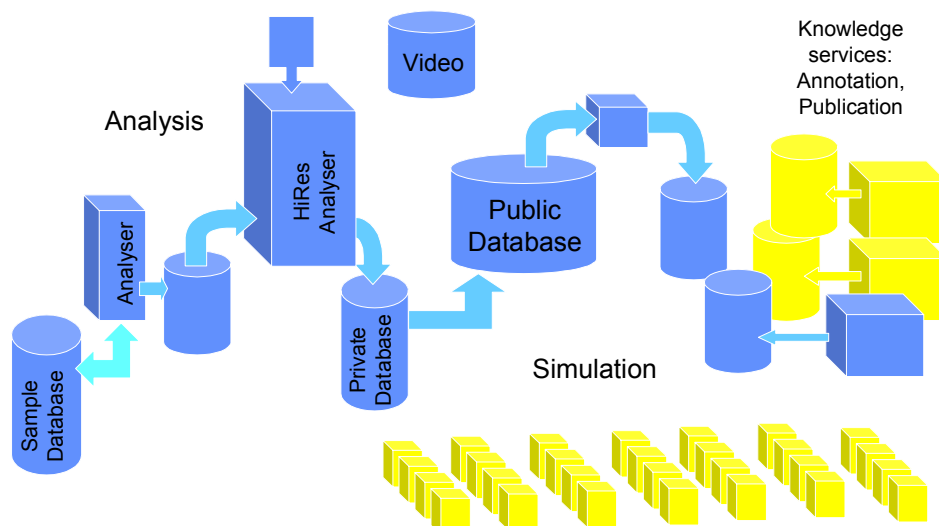


Figure 2: Workflow in the scenario

This scenario draws out a number of underlying assumptions and raises a number of requirements that we believe are broadly applicable to a range of e-Science applications:

- *Storage*. It is important that the system is able to store and process potentially huge volumes of content in a timely and efficient fashion.
- *Ownership*. Different stakeholders need to be able to retain ownership of their own content and processing capabilities, but there is also a need to allow others access under the appropriate terms and conditions.
- *Provenance*. Sufficient information is stored so that it is possible to repeat the experiment, re-use the results, or provide evidence that this information was produced at this time (the latter may involve a third party).
- *Transparency*. Users need to be able to discover, transparently access and process relevant content wherever it may be located in the Grid.
- *Communities*. Users should be able to form, maintain and disband communities of practice with restricted membership criteria and rules of operation.
- *Fusion*. Content needs to be able to be combined from multiple sources in unpredictable ways according to the users' needs; descriptions of the sources and content will be used to combine content meaningfully.
- *Conferencing*. Sometimes it is useful to see the other members of the conference, and sometimes it is useful to see the artefacts and visualisations under discussion.
- *Annotation*. From logging the sample through to publishing the analysis, it is necessary to have annotations that enrich the description of any digital content. This meta-content may apply to data, information or knowledge and depends on agreed interpretations.
- *Workflow*. To support the process enactment and automation, the system needs descriptions of processes. The scenario illustrates workflow both inside and outside the company.
- *Notification*. The arrival of new information prompts notifications to users and initiates automatic processing.
- *Decision support*. The technicians and scientists are provided with relevant information and suggestions for the task at hand.
- *Resource reservation*. There is a need to ease the process of resource reservation. This applies to experimental equipment, collaboration (the conference), and resource scheduling for the simulation.
- *Security*. There are authentication, encryption and privacy requirements, with multiple organisations involved, and a requirement for these to be handled with minimal manual intervention.
- *Reliability*. The systems appear to be reliable but in practice there may be failures and exception handling at various levels, including the workflow.
- *Video*. Both live and stored video have a role, especially where the video is enriched by associated temporal metacontent (in this case to aid navigation).
- *Smart laboratory*. For example, the equipment detects the sample (e.g. by barcode or RFID tag), the scientist may use portable devices for note-taking, and visualisations may be available in the lab.
- *Knowledge*. Knowledge services are an integral part of the e-Science process. Examples include: finding papers, finding people, finding previous

experimental design (these queries may involve inference), annotating the uploaded analysis, and configuring the lab to the person.

- *Growth*. The system should support evolutionary growth as new content and processing techniques become available.
- *Scale*. The scale of the scientific collaboration increases through the scenario, as does the scale of computation, bandwidth, storage and complexity of relationships between information.

3. A Service-Oriented View

This section expands upon the view of the Semantic Grid as a service-oriented architecture in which *entities* provide *services* to one another under various forms of *contract*⁴. Thus, as shown in figure 1, the e-Scientist's environment is composed of data/computation services, information services, and knowledge services. However, before we deal with the specifics of each of these different types of service, it is important to highlight those aspects that are common since this provides the conceptual basis and rationale for what follows. To this end, section 3.1 provides the justification for a *service-oriented* view of the different layers of the Semantic Grid. Section 3.2 then addresses the technical ramifications of this choice and outlines the key technical challenges that need to be overcome to make service-oriented grids a reality. The section concludes (section 3.3) with the e-Science scenario of section 2 expressed in a service-oriented architecture.

3.1 Justification of a Service-Oriented View

Given the set of desiderata and requirements from section 2, a key question in designing and building grid applications is what is the most appropriate conceptual model for the system? The purpose of such a model is to identify the key constituent components (abstractions) and specify how they are related to one another. Such a model is necessary to identify generic grid technologies and to ensure that there can be re-use between different grid applications. Without a conceptual underpinning, grid endeavours will simply be a series of handcrafted and *ad hoc* implementations that represent point solutions.

To this end, an increasingly common way of viewing many large systems (from governments, to businesses, to computer systems) is in terms of the *services* that they provide. Here a service can simply be viewed as an abstract characterization and encapsulation of some content or processing capabilities. For example, potential services in our exemplar scenario could be: the equipment automatically recognising the sample and configuring itself appropriately, the logging of information about a sample in the international database, the setting up of a video to monitor the experiment, the locating of appropriate computational resources to support a run of the High Resolution Analyser, the finding of all scientists who have published work on experiments similar to those uncovered by our e-Scientist, and the analyser raising an alert whenever a particular pattern of results occurs (see section 3.3 for more

⁴ This view pre-dates the work of Foster et al on the Open Services Grid Architecture [Foster02]. While Foster's proposal has many similarities with our view he does not deal with issues associated with developing services through autonomous agents, with the issue of dynamically forming service level agreements, nor with the design of marketplaces in which the agents trade their services.

details). Thus, services can be related to the domain of the Grid, the infrastructure of the computing facility, or the users of the Grid – i.e., at the data/computation layer, at the information layer, or at the knowledge layer (as per figure 1). In all of these cases, however, it is assumed that there may be multiple versions of broadly the same service present in the system.

Services do not exist in a vacuum, rather they exist in a particular institutional context. Thus all services have an owner (or set of owners). The owner is the body (individual or institution) that is responsible for offering the service for consumption by others. The owner sets the terms and conditions under which the service can be accessed. Thus, for example, the owner may decide to make the service universally available and free to all on a first-come, first-served basis. Alternatively, the owner may decide to limit access to particular classes of users, to charge a fee for access and to have priority-based access. All options between these two extremes are also possible. It is assumed that in a given system there will be multiple service owners (each representing a different stakeholder) and that a given service owner may offer multiple services. These services may correspond to genuinely different functionality or they may vary in the way that broadly the same functionality is delivered (e.g., there may be a quick and approximate version of the service and one that is more time consuming and accurate).

In offering a service for consumption by others, the owner is hoping that it will indeed attract consumers for the service. These consumers are the entities that decide to try and invoke the service. The purpose for which this invocation is required is not of concern here: it may be for their own private use, it may be to resell onto others, or it may be to combine with other services.

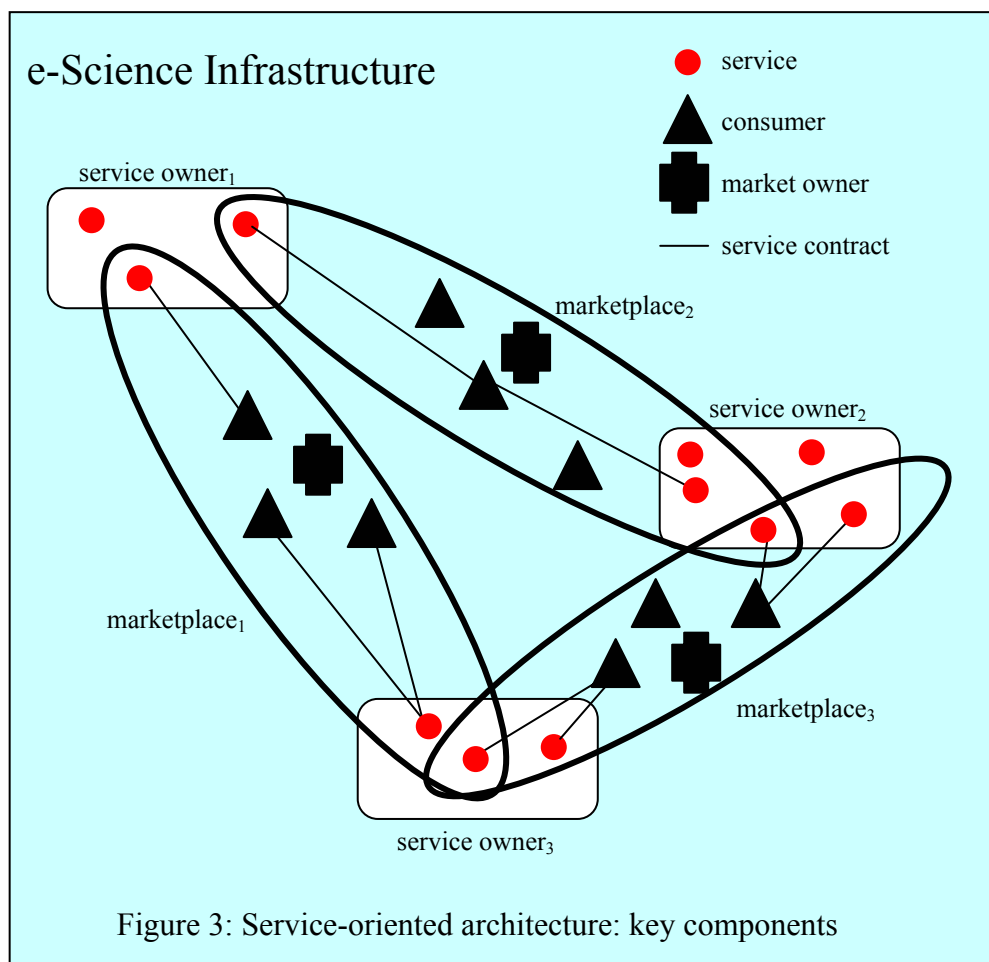
The relationship between service owner and service consumer is codified through a service contract. This contract specifies the terms and conditions under which the owner agrees to provide the service to the consumer. The precise structure of the contract will depend upon the nature of the service and the relationship between the owner and the provider. However examples of relevant attributes include the price for invoking the service, the information the consumer has to provide to the provider, the expected output from the service, an indication about when this output can be expected, and the penalty for failing to deliver according to the contract. Service contracts can either be established by an off-line or an on-line process depending on the prevailing context.

The service owners and service producers interact with one another in a particular environmental context. This environment may be common to all entities in the Grid (meaning that all entities offer their services in an entirely open marketplace). In other cases, however, the environment may be closed and entrance may be controlled (meaning that the entities form a private club).⁵ In what follows, a particular environment will be called a *marketplace* and the entity that establishes and runs the marketplace will be termed the *market owner*. The rationale for allowing individual marketplaces to be defined is that they offer the opportunity to embed interactions in an environment that has its own set of rules (both for membership and ongoing

⁵ This is analogous to the notion of having a virtual private network overlaid on top of the Internet. The Internet corresponds to the open marketplace in which anybody can participate and the virtual private network corresponds to a closed club that can interact under its own rules.

operation) and they allow the entities to make stronger assumptions about the parties with which they interact (e.g., the entities may be more trustworthy or cooperative since they are part of the same club). Such marketplaces may be appropriate, for example, if the nature of the domain means that the services are particularly sensitive or valuable. In such cases, the closed nature of the marketplace will enable the entities to interact more freely because of the rules of membership.

To summarise, the key components of a service-oriented architecture are as follows (figure 3): service owners (rounded rectangles) that offer services (filled circles) to service consumers (filled triangles) under particular contracts (solid links between producers and consumers). Each owner-consumer interaction takes place in a given marketplace (denoted by ovals) whose rules are set by the market owner (filled cross). The market owner may be one of the entities in the marketplace (either a producer or a consumer) or it may be a neutral third party.



Given the central role played by the notion of a service, it is natural to explain the operation of the system in terms of a *service lifecycle* (figure 4). The first step is for a service owner to define a service they wish to make available to others. The reasons for wanting to make a service available may be many and varied – ranging from altruism, through necessity, to commercial benefit. It is envisaged that in a given grid application all three motivations (and many others besides) are likely to be present,

although perhaps to varying degrees that are dictated by the nature of the domain. *Service creation* should be seen as an ongoing activity. Thus new services may come into the environment at any time and existing ones may be removed (service decommissioning) at any time. This means the system is in a state of continual flux and never reaches a steady state. Creation is also an activity that can be automated to a greater or lesser extent. Thus, in some cases, all services may be put together in an entirely manual fashion. In other cases, however, there may be a significant automated component. For example, it may be decided that a number of services should be combined; either to offer a new service (if the services are complementary in nature) or to alter the ownership structure (if the services are similar). In such cases, it may be appropriate to automate the processes of finding appropriate service providers and of getting them to agree to new terms of operation. This dynamic service composition activity is akin to creating a new *virtual organisation*: a number of initially distinct entities can come together, under a set of operating conditions, to form a new entity that offers a new service. This grouping will then stay in place until it is no longer appropriate to remain in this form, whereupon it will disband.

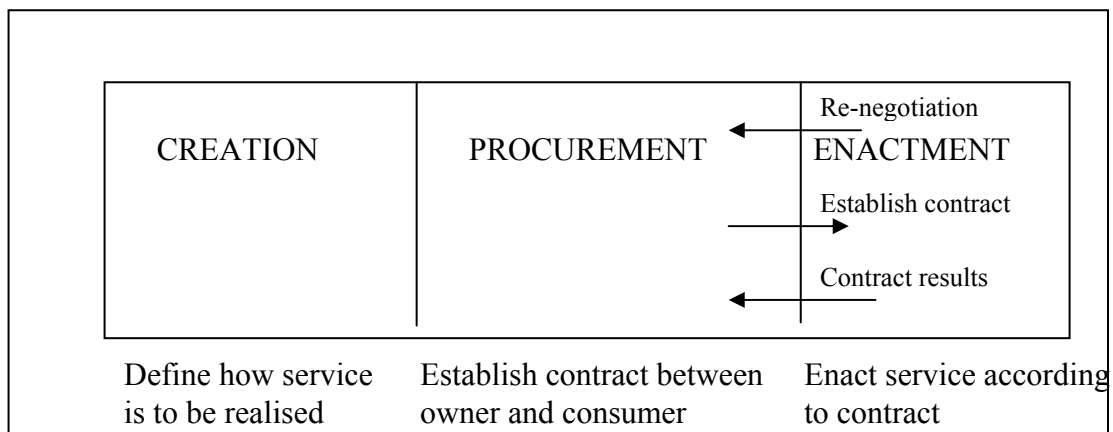


Figure 4: Service lifecycle

The *service creation* process covers three broad types of activity. Firstly, specifying how the service is to be realized by the service owner using an appropriate service description language. These details are not available externally to the service consumer (i.e., they are encapsulated by the service owner). Secondly, specifying the meta-information associated with the service. This indicates the potential ways in which the service can be procured. This meta-information indicates who can access the service and what are the likely contract options for procuring it. Thirdly, making the service available in the appropriate marketplace. This requires appropriate service advertising and registration facilities to be available in the marketplace.

The *service procurement* phase is situated in a particular marketplace and involves a service owner and a service consumer establishing a contract for the enactment of the service according to a particular set of terms and conditions. There are a number of points to note about this process. Firstly, it may fail. That is, for whatever reason, a service owner may be unable or unwilling to provide the service to the consumer. Secondly, in most cases, the service owner and the service consumer will represent different and autonomous stakeholders. Thus the process by which contracts are established will be some form of *negotiation* – since the entities involved need to

come to a mutually acceptable agreement on the matter. If the negotiation is successful (i.e., both parties come to an agreement) then the outcome of the procurement is a contract between the service owner and the service consumer. Thirdly, this negotiation may be carried out off-line by the respective service owners or it may be carried out at run-time. In the latter case, the negotiation may be automated to a greater or lesser extent – varying from the system merely automatically flagging the fact that a new service contract needs to be established to automating the entire negotiation process⁶.

The final stage of the service lifecycle is *service enactment*. Thus, after having established a service contract, the service owner has to undertake the necessary actions in order to fulfil its obligations as specified in the contract. After these actions have been performed, the owner needs to fulfil its reporting obligations to the consumer with respect to the service. This may range from a simple inform indicating that the service has been completed, to reporting back complex content that represents the results of performing the service. The above assumes that the service owner is always able to honour the contracts that it establishes. However, in some cases the owner may not be able to stick to the terms specified in the contract. In such cases, it may have to renegotiate the terms and conditions of the contract; paying any penalties that are due. This enforcement activity is undertaken by the market owner and will be covered by the terms and conditions that the service providers and consumers sign up to when they enter into the marketplace.

Having described the key components of the service-oriented approach, we return to the key system-oriented desiderata noted in section 2. From the above discussion, it can be seen that a service-oriented architecture is well suited to grid applications:

- able to store and process huge volumes of content in a timely fashion;
 - *The service-oriented model offers a uniform means of describing and encapsulating activities at all layers in the Grid. This model then needs to be underpinned by the appropriate processing and communication infrastructure to ensure it can deliver the desired performance.*
- allow different stakeholders to retain ownership of their own content and processing capabilities, but to allow others access under the appropriate terms and conditions;
 - *Each service owner retains control over the services that they make available to others. They determine how the service is realized and set the policy for accessing the service.*
- allow users to discover, transparently access and process relevant content wherever it may be located in the Grid;
 - *The overall system is simply viewed as a number of service marketplaces. Any physical distribution and access problems are masked via the service interface and the service contract. The*

⁶ Automated negotiation technology is now widely used in many e-commerce applications [Guttman98]. It encompasses various forms of auctions (a one-to-many form of negotiation) as well as bi-lateral negotiations. Depending on the negotiation protocol that is in place, the negotiation can be concluded in a single round or it may last for many rounds. Thus negotiation need not be a lengthy process; despite the connotation from human interactions that it may be!

marketplace itself has advertisement and brokering mechanisms to ensure appropriate service owners and consumers are put together.

- allow users to form, maintain and disband communities of practice with restricted membership criteria and rules of operation;
 - *Each community can establish its own marketplace. The marketplace owner defines the conditions that have to be fulfilled before entities can enter, defines the rules of interaction for once the marketplace is operational, and enforces the rules through appropriate monitoring.*
- allow content to be combined from multiple sources in unpredictable ways according to the users' needs;
 - *It is impossible to a priori predict how the users of a system will want to combine the various services contained within it. Thus services must be able to be composed in flexible ways. This is achieved by negotiation of appropriate contracts. This composition can be done on a one-off basis or may represent a more permanent binding into a new service that is offered on an ongoing basis (as in the establishment of a new virtual organisation).*
- support evolutionary growth as new content and processing techniques become available.
 - *Services represent the unit of extension of the system. Thus as new content or processing techniques become available they are simply represented as new services and placed in a marketplace(s). Also new marketplaces can be added as new communities of practice emerge.*

3.2 Key Technical Challenges

The previous section outlined the service-oriented view of the Semantic Grid. Building upon this, this section identifies the key technical challenges that need to be overcome to make such architectures a reality. To this end, table 1 represents the key functionality of the various components of the service-oriented architecture, each of which is then described in more detail in the remainder of this section.

Service Owner	Service Consumer	Marketplace
Service creation	Service discovery	Owner and consumer registration
Service advertisement		Service registration
Service contract creation	Service contract creation	Policy specification
Service delivery	Service result reception	Policy monitoring and enforcement

Table 1: Key functions of the service-oriented architecture components

3.2.1 Service Owners and Consumers as Autonomous Agents

A natural way to conceptualise the service owners and the service consumers are as autonomous agents. Although there is still some debate about exactly what constitutes

agenthood, an increasing number of researchers find the following characterisation useful [Wooldridge97]:

an agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives

There are a number of points about this definition that require further explanation. Agents are [Jennings00]: (i) clearly identifiable problem solving entities with well-defined boundaries and interfaces; (ii) situated (embedded) in a particular environment—they receive inputs related to the state of their environment through sensors and they act on the environment through effectors; (iii) designed to fulfill a specific purpose—they have particular objectives (goals) to achieve; (iv) autonomous—they have control both over their internal state and over their own behaviour⁷; (v) capable of exhibiting flexible problem solving behaviour in pursuit of their design objectives—they need to be both reactive (able to respond in a timely fashion to changes that occur in their environment) and proactive (able to act in anticipation of future goals) .

Thus, each service owner will have one or more agents acting on its behalf. These agents will manage access to the services for which they are responsible and will ensure that the agreed contracts are fulfilled. This latter activity involves the scheduling of local activities according to the available resources and ensuring that the appropriate results from the service are delivered according to the contract in place. Agents will also act on behalf of the service consumers. Depending on the desired degree of automation, this may involve locating appropriate services, agreeing contracts for their provision, and receiving and presenting any received results.

3.2.2 Interacting Agents

Grid applications involve multiple stakeholders interacting with one another in order to procure and deliver services. Underpinning the agents' interactions is the notion that they need to be able to inter-operate in a meaningful way. Such semantic interoperation is difficult to obtain in grids (and all other open systems) because the different agents will typically have their own individual information models. Moreover, the agents may have a different communication language for conveying their own individual terms. Thus, meaningful interaction requires mechanisms by which this basic interoperation can be effected (see section 4.2 for more details).

Once semantic inter-operation has been achieved, the agents can engage in various forms of interaction. These interactions can vary from simple information interchanges, to requests for particular actions to be performed and on to cooperation, coordination and negotiation in order to arrange interdependent activities. In all of these cases, however, there are two points that qualitatively differentiate agent

⁷ Having control over their own behaviour is one of the characteristics that distinguishes agents from objects. Although objects encapsulate state and behaviour (more accurately behaviour realisation), they fail to encapsulate behaviour activation or action choice. Thus, any object can invoke any publicly accessible method on any other object at any time. Once the method is invoked, the corresponding actions are performed. In this sense, objects are totally obedient to one another and do not have autonomy over their choice of action.

interactions from those that occur in other computational models. Firstly, agent-oriented interactions are conceptualised as taking place at the *knowledge level* [Newell82]. That is, they are conceived in terms of which goals should be followed, at what time, and by whom. Secondly, as agents are flexible problem solvers, operating in an environment over which they have only partial control and observability, interactions need to be handled in a similarly flexible manner. Thus, agents need the computational apparatus to make *run-time* decisions about the nature and scope of their interactions and to initiate (and respond to) interactions that were not foreseen at design time (cf. the hard-wired engineering of such interactions in extant approaches).

The subsequent discussion details what would be involved if all these interactions were to be automated and performed at run-time. This is clearly the most technically challenging scenario and there are a number of points that need to be made. Firstly, while such automation is technically feasible, in a limited form, using today's technology, this is an area that requires more research to reach the desired degree of sophistication and maturity. Secondly, in some cases, the service owners and consumers may not wish to automate all of these activities since they may wish to retain a degree of human control over these decisions. Thirdly, some contracts and relationships may be set up at design time rather than being established at run-time. This can occur when there are well-known links and dependencies between particular services, owners and consumers.

The nature of the interactions between the agents can be broadly divided into two main camps. Firstly, those that are associated with making service contracts. This will typically be achieved through some form of automated negotiation since the agents are autonomous [Jennings01]. When designing these negotiations, three main issues need to be considered:

- *The Negotiation Protocol*: the set of rules that govern the interaction. This covers the permissible types of participants (e.g. the negotiators and any relevant third parties), the negotiation states (e.g. accepting bids, negotiation closed), the events that cause negotiation states to change (e.g. no more bidders, bid accepted) and the valid actions of the participants in particular states (e.g. which messages can be sent by whom, to whom, at what stage).
- *The Negotiation Object*: the range of issues over which agreement must be reached. At one extreme, the object may contain a single issue (such as price), while on the other hand it may cover hundreds of issues (related to price, quality, timings, penalties, terms and conditions, etc.). Orthogonal to the agreement structure, and determined by the negotiation protocol, is the issue of the types of operation that can be performed on agreements. In the simplest case, the structure and the contents of the agreement are fixed and participants can either accept or reject it (i.e. a take it or leave it offer). At the next level, participants have the flexibility to change the values of the issues in the negotiation object (i.e. they can make counter-proposals to ensure the agreement better fits their negotiation objectives). Finally, participants might be allowed to dynamically alter (by adding or removing issues) the structure of the negotiation object (e.g. a car salesman may offer one year's free insurance in order to clinch the deal).

- *The Agent's Decision Making Models*: the decision-making apparatus the participants so as to act in line with the negotiation protocol in order to achieve their objectives. The sophistication of the model, as well as the range of decisions that have to be made, are influenced by the protocol in place, by the nature of the negotiation object, and by the range of operations that can be performed on it. It can vary from the very simple, to the very complex.

In designing any automated negotiation system the first thing that needs to be established is the protocol to be used (this is called the *mechanism design problem*). In this context, this will be determined by the market owner. Here the main consideration is the nature of the negotiation. If it is a one-to-many negotiation (i.e., one buyer and many sellers or one seller and many buyers) then the protocol will typically be a form of auction. Although there are thousands of different permutations of auction, four main ones are typically used. These are: English, Dutch, Vickrey, and First-Price Sealed Bid. In an English auction, the auctioneer begins with the lowest acceptable price and bidders are free to raise their bids successively until there are no more offers to raise the bid. The winning bidder is the one with the highest bid. The Dutch auction is the converse of the English one; the auctioneer calls for an initial high price, which is then lowered progressively until there is an offer from a bidder to claim the item. In the first-priced sealed bid, each bidder submits their offer for the item independently without any knowledge of the other bids. The highest bidder gets the item and they pay a price equal to their bid amount. Finally, a Vickrey auction is similar to a first-price sealed bid auction, but the item is awarded to the highest bidder at a price equal to the second highest bid. More complex forms of auctions exist to deal with the cases in which there are multiple buyers and sellers that wish to trade (these are called double auctions) and with cases in which agents wish to purchase multiple interrelated goods at the same time (these are called combinatorial auctions). If it is a one-to-one negotiation (one buyer and one seller) then a form of heuristic model is needed (e.g. [Faratin99; Kraus01]). These models vary depending upon the nature of the negotiation protocol and, in general, are less well developed than those for auctions.

Having determined the protocol, the next step is to determine the nature of the contract that needs to be established. This will typically vary from application to application and again it is something that is set by the market owner. Given these two, the final step is to determine the agent's reasoning model. This can vary from the very simple (bidding truthfully) to the very complex (involving reasoning about the likely number and nature of the other bidders).

The second main type of interaction is when a number of agents decide to come together to form a new virtual organisation. This involves determining the participants of the coalition and determining their various roles and responsibilities in this new organisational structure. Again this is typically an activity that will involve negotiation between the participants since they need to come to a mutually acceptable agreement about the division of labour and responsibilities. Here there are a number of techniques and algorithms that can be employed to address the coalition formation process [Sandholm00; Shehory98] although this area requires more research to deal with the envisaged scale of grid applications.

3.2.3 Marketplace Structures

Marketplaces should be able to be established by any agent(s) in the system (including a service owner, a service consumer or a neutral third party). The entity which establishes the marketplace is here termed the market owner. The owner is responsible for setting up, advertising, controlling and disbanding the marketplace. In order to establish a marketplace, the owner needs a representation scheme for describing the various entities that are allowed to participate in the marketplace (terms of entry), a means of describing how the various allowable entities are allowed to interact with one another in the context of the marketplace, and what monitoring mechanisms (if any) are to be put in place to ensure the marketplace's rules are adhered to.

3.3 A Service-Oriented View of the Scenario

The first marketplace is that connected with the scientist's own lab. This marketplace has agents to represent the humans involved in the experiment, thus there is a *scientist agent* (SA) and a *technician agent* (TA). These are responsible for interacting with the scientist and the technician, respectively, and then for enacting their instructions in the Grid. These agents can be viewed as the computational proxies of the humans they represent – endowed with their personalised information about their owner's preferences and objectives. These personal agents need to interact with other (artificial) agents in the marketplace in order to achieve their objectives. These other agents include an *analyser agent* (AA) (that is responsible for managing access to the analyser itself), the *analyser database agent* (ADA) (that is responsible for managing access to the database containing information about the analyser), and the *high resolution analyser agent* (HRAA) (that is responsible for managing access to the high resolution analyser). There is also an *interest notification agent* (INA) (that is responsible for recording which scientists in the lab are interested in which types of results and for notifying them when appropriate results are generated) and an *experimental results agent* (ERA) (that can discover similar analyses of data or when similar experimental configurations have been used in the past). The services provided by these agents are summarised in table 2.

Agent	Services Offered	Services Consumed By
Scientist Agent (SA)	resultAlert reportAlert	Scientist Scientist
Technician Agent (TA)	MonitorAnalysis	Technician
Analyser Agent (AA)	configureParameters runSample	ADA ADA
Analyser Database Agent (ADA)	logSample setAnalysisConfiguration bookSlot recordAnalysis	Technician Technician TA AA
High Resolution Analyser Agent (HRAA)	bookSlot configureParameters runAnalysis videoAnalysis monitorAnalysis reportResults replayExperiment suggestRelatedConfigurations	SA Scientist Scientist Scientist, Technician Technician SA Scientist Scientist
Interest Notification Agent (INA)	registerInterest notifyInterestedParties findInterestedParties	Scientists, Technicians ADA Scientist

Experimental Results Agent (ERA)	FindSimilarExperiments	HRAA
-------------------------------------	------------------------	------

Table 2: Services in the scientist's lab marketplace

The operation of this marketplace is as follows. The technician uses the `logSample` service to record data about the sample when it arrives and the `setAnalysisConfiguration` service to set the appropriate parameters for the forthcoming experiment. The technician then instructs the TA to book a slot on the analyser using the `bookSlot` service. At the appropriate time, the ADA informs the AA of the settings that it should adopt (via the `configureParameters` service) and that it should now run the experiment (via the `runSample` service). As part of the contract for the `runSample` service, the AA informs the ADA of the results of the experiment and these are logged along with the appropriate experimental settings (using the `recordAnalysis` service). Upon receipt of these results, the ADA informs the INA of them. The INA then disseminates the results (via the `notifyInterestedParties` service) to scientists who have registered an interest in results of that kind (achieved via the `registerInterest` service).

When interesting results are received, the SA alerts the scientist (via the `resultAlert` service). The scientist then examines the results and decides that they are of interest and that further analysis is need. The scientist then instructs the SA to make a booking on the High Resolution Analyser (via the `bookSlot` service). When the booking is made, the HRAA volunteers information to the scientist about the configurations of similar experiments that have previously been run (via the `suggestRelatedConfigurations` service). Using this information, the scientist sets the appropriate configurations (via the `configureParameters` service). At the appropriate time, the experiment is started (via the `runAnalysis` service). As part of the contract for this service, the experiment is videoed (via the `videoAnalysis` service), monitoring information is sent to the technician (via the `monitorAnalysis` service) and a report is prepared and sent to the SA (via the `reportResults` service). In preparing this report, the HRAA interacts with the ERA to discover if related experiments and results have already been undertaken (achieved via the `findSimilarExperiments` service).

The scientist is alerted to the report by the SA (via the `reportAlert` service). The scientist decides the results may be interesting and decides to replay some of the key segments of the video (via the `replayExperiment` service). The scientist decides the results are indeed interesting and so asks for relevant publications and details of scientists who have published on this topic. This latter activity is likely to be provided through an external marketplace that provides this service for the wider community (see table 4). In such a marketplace, there may be multiple *Paper Repository Agents* that offer the same broad service (`findRelatedPapers` and `findRelatedAuthors`) but to varying degrees of quality, coverage, and timeliness.

Armed with all this information, the scientist decides that the results should be discussed within the wider organisation context. This involves interacting in the Scientist's Organisation Marketplace. The agents involved in this marketplace are the *research meeting convener agent* (RMCA) (responsible for organising research meetings) and the various *scientist agents* that represent the relevant scientists. The

services provided by these agents are given in table 3. The RMCA is responsible for determining when research meetings should take place, this is achieved via the `arrangeMeeting` service through interaction with the SAs of the scientists involved. The scientist requests a slot to discuss the latest experimental findings (via the `setAgenda` service) and provides the appropriate data for discussion to the RMCA that disseminates it to the SA's of the relevant participants (via the `disseminateInformation` service). As a consequence of the meeting, it is decided that the results are appropriate for dissemination into the scientific community at large.

Agent	Services Offered	Service Consumed By
Research Meeting Convener Agent (RMCA)	<code>arrangeMeeting</code> <code>setAgenda</code> <code>disseminateInformation</code>	SAs Scientist SAs
Scientist Agent (SA)	<code>arrangeMeeting</code> <code>receiveInformation</code>	RMCA RMCA

Table 3: Services in the scientist's organisation marketplace

The general scientific community is represented by a series of distinct marketplaces that are each responsible for different aspects of the scientific process. As decided upon at the organisation's meeting, the sample data is logged in the appropriate international database (using the `logSample` service). This database has an attached notification service at which individual scientists can register their interests in particular types of data (via the `registerInterests` service). Scientists will then be informed, via their SA, when new relevant data is posted (via the `disseminateInformation` service).

	Services Offered	Services Consumed By
International Sample Database Agent (ISDA)	<code>logSample</code> <code>registerInterests</code> <code>disseminateInformation</code>	Scientist Scientist SAs
Paper Repository Agent (PRA)	<code>findRelatedPapers</code> <code>findRelatedAuthors</code>	SAs SAs
Scientist Agent (SA)	<code>receiveRelevantData</code> <code>arrangeSimulation</code>	Scientist Scientist
Simulation Provider Agent (SPA)	<code>offerSimulationResource</code> <code>utiliseSimulationResource</code>	SA SA
Problem Solving Environment Agent (PSEA)	<code>whatSimulationTools</code> <code>simulationSettingInfo</code> <code>analyseResults</code>	Scientist Scientist Scientist

Table 4: Services in the general scientific community marketplace

One of the scientists who receives notification of the new results believes that they should be investigated further by undertaking a new round of simulations. The scientist instructs the SA to arrange for particular simulations to be arranged. The SA enters a marketplace where providers of processing capabilities offer their resources (via the `offerSimulationResource` service). The SA will arrange for the appropriate amount of resource to be made available at the desired time such that the simulations can be run. Once these contracts have been established, the SA will invoke the

simulation (via the `utiliseSimulationResource` service). During the course of these simulations, the scientist will make use of the Problem Solving Environment Agent (PSEA) to assist in the tasks of determining what simulation tools to exploit (via the `whatSimulationTools` service), setting the simulation parameters appropriately for these tools (via the `simulationSettingInfo` service), and analysing the results (via the `analyseResults` service).

This then characterises our scenario as an active marketplace of agents offering and consuming services. As already indicated, we do not expect that this complete set of interactions will be dealt with seamlessly by computational agents in the near future. However, it provides a level of abstraction and defines capabilities that we claim it is important to aspire to if the full potential of the Semantic Grid is to be realised.

4. The Knowledge Layer

The aim of the knowledge layer is to act as an infrastructure to support the management and application of scientific knowledge to achieve particular types of goal and objective. In order to achieve this, it builds upon the services offered by the data/computation and information layers (see [DeRoure02b] for more details of the services and technologies at these layers).

The first thing to reiterate with respect to this layer is the problem of the sheer scale of content we are dealing with. We recognise that the amount of data that the data grid is managing is likely to be huge. By the time that data is equipped with meaning and turned into information we can expect order of magnitude reductions in the amount. However what remains will certainly be enough to present us with the problem of *infosmog* – the condition of having too much information to be able to take effective action or apply it in an appropriate fashion to a specific problem. Once information is delivered that is destined for a particular purpose, we are in the realm of the knowledge grid. Thus at this level we are fundamentally concerned with abstracted and annotated content and with the management of scientific knowledge.

We can see this process of scientific knowledge management in terms of a life cycle of knowledge-oriented activity that ranges over knowledge acquisition and modelling, knowledge retrieval and reuse, knowledge publishing and knowledge maintenance (section 4.1). Next we discuss the fundamental role that ontologies will play in providing the underpinning semantics for the knowledge layer (section 4.2). Section 4.3 then considers the knowledge services aspects of our scenario. Finally, we review the research issues associated with our requirements for a knowledge grid (section 4.4).

4.1 The Knowledge Lifecycle

The knowledge lifecycle can be regarded as a set of challenges as well as a sequence of stages. Each stage has variously been seen as a bottleneck. The effort of acquiring knowledge was one bottleneck recognised early [Hayes-Roth83]. But so too are; modelling, retrieval, reuse, publication and maintenance. In this section we examine

the nature of the challenges at each stage in the knowledge lifecycle and review the various methods and techniques at our disposal.

Although we often suffer from a deluge of data and too much information, all too often what we have is still insufficient or too poorly specified to address our problems, goals and objectives. In short, we have insufficient knowledge. *Knowledge acquisition* sets the challenge of getting hold of the information that is around, and turning it into knowledge by making it *usable*. This might involve, for instance, making tacit knowledge explicit, identifying gaps in the knowledge already held, acquiring and integrating knowledge from multiple sources (e.g. different experts, or distributed sources on the Web), or acquiring knowledge from unstructured media (e.g. natural language or diagrams).

A range of techniques and methods has been developed over the years to facilitate knowledge acquisition. Much of this work has been carried out in the context of attempts to build knowledge-based or expert systems. Techniques include varieties of interview, different forms of observation of expert problem solving, methods of building conceptual maps with experts, various forms of document and text analysis, and a range of machine learning methods [Shadbolt95]. Each of these techniques has found to be suited to the elicitation of different forms of knowledge and to have different consequences in terms of the effort required to capture and model the knowledge [Hoffman95; Shadbolt99]. Specific software tools have also been developed to support these various techniques [Milton99] and increasingly these are now web enabled [Shaw98].

However, the process of explicit knowledge acquisition from human experts remains a costly and resource intensive exercise. Hence, the increasing interest in methods that can (semi-) automatically elicit and acquire knowledge that is often implicit or else distributed on the web [Crow01]. A variety of information extraction tools and methods are being applied to the huge body of textual documents that are now available [Ciravegna01]. Examples include programs to extract information about protein function from various scientific papers, abstracts and databases that are increasingly available on-line. Another style of automated acquisition consists of systems that observe user behaviour and infer knowledge from that behaviour. Examples include recommender systems that might look at the papers downloaded by a researcher and then detect themes by analysing the papers using methods such as term frequency analysis [Middleton01]. The recommender system then searches other literature sources and suggests papers that might be relevant or else of interest to the user.

Methods that can engage in the sort of background knowledge acquisition described above are still in their infancy but with the proven success of pattern directed methods in areas such as data mining they are likely to assume a greater prominence in our attempts to overcome the knowledge acquisition bottleneck.

Knowledge modelling bridges the gap between the acquisition of knowledge and its use. Knowledge models must be able *both* to act as straightforward placeholders for the acquired knowledge, *and* to represent the knowledge so that it can be used for problem-solving. Knowledge representation technologies have a long history in Artificial Intelligence. There are numerous languages and approaches that cater for

different knowledge types; structural forms of knowledge, procedurally oriented representations, rule based characterisations and methods to model uncertainty, and probabilistic representations [Brachman83].

Most large applications require a range of knowledge representation formats. CommonKADS [Schreiber00], one of the most comprehensive methodologies for the development of knowledge intensive systems, uses a range of modelling methods and notations- including logic and structured objects. It also factors out knowledge into various types and identifies recurrent patterns of inference and knowledge type that denote characteristic problem solvers. These patterns are similar to design patterns in software engineering [Gamma95] and attempt to propose a set of components out of which problem solving architectures can be composed. One of the major constituents of the models built in CommonKADS are domain ontologies which we discuss in the next section.

Recently with the explosion of content on the web there has arisen the recognition of the importance of metadata. Any kind of content can be “enriched” by the addition of annotations about what the content is about [Motta00]. Such semantic metadata is an important additional element in our modelling activity. It may indicate the origin of content, its provenance, value or longevity. It may associate other resources with the content such as the rationale as to why the content is in the form it is and so on.

Certainly given the sheer amount of content available in a grid context it is crucial to have some technical support for metadata “enrichment”. To this end a number of systems are now under development that aim to take given metadata structures and help annotate, tag or associate content with that metadata [Motta02, Handschuh02].

In any modelling exercise it is important to recognise that the modelling reflects a set of interests and perspectives. These may be made more or less explicit but they are always present. It is also important to recognise that models may be more or less formal and aspire to various degrees of precision and accuracy. The model is, of course, not the object or process, rather it is an artefact built with a particular set of goals and intentions in mind.

Once knowledge has been acquired and modelled, it needs to be stored or hosted somewhere so that it can be retrieved efficiently. In this context, there are two related problems to do with *knowledge retrieval*. First, there is the issue of finding knowledge again once it has been stored. And second, there is the problem of retrieving the subset of content that is relevant to a particular problem. This will set particular problems for a knowledge retrieval system where content alters rapidly and regularly.

Technologies for information retrieval exist in many forms [Sparck-Jones97]. They include methods that attempt to encode structural representations about the content to be retrieved such as explicit attributes and values. Varieties of matching algorithm can be applied to retrieve cases that are similar to an example or else a partial set of attributes presented to the system. Such explicit Case Based Reasoning [Lenz98] and Query engines have been widely adopted. They suffer from the problem of content encoding – the ease with which new content and examples can be represented in the required structural format. There are also perennial issues about the best measures of similarity to use in these systems.

Other retrieval methods are based on statistical encoding of the objects to be retrieved. These might be as vectors representing the frequency of the terms in a document or other piece of content. Retrieval is a matter of matching a query of an example piece of content against these stored representations and generating closest matches [Croft00].

Search engines such as Google that are manifestly capable of scaling and also demonstrate good retrieval performance rely on concepts such as relevance ranking. Here given any set of terms to search Google looks at the interconnected nature of content and the frequency of its being accessed to help determine in part the rank of how good a match to the material sought it is likely to be.

In the general field of content retrieval there is no one dominant paradigm - it can occur at the fine grained level at which point it is a form of information extraction, or else at the level of complete documents or even work flows or data logs that might encode entire experimental configurations and subsequent runs.

One of the most serious impediments to the cost-effective use of knowledge is that too often knowledge components have to be constructed afresh. There is little *knowledge reuse*. This arises partly because knowledge tends to require different representations depending on the problem-solving that it is intended to do. We need to understand how to find patterns in knowledge, to allow for its storage so that it can be reused when circumstances permit. This would save a good deal of effort in reacquiring and restructuring the knowledge that had already been used in a different context.

We have already alluded to the form of reuse embodied in methodologies such as CommonKADS. Here a problem-solving template for monitoring might be used in one domain and its general structure reused elsewhere. The actual ontology of components or processes might be another candidate for reuse. Complete problem solving runs or other results might offer the chance to reuse previously solved problems in areas that are similar. Workflows themselves might be reused. Technical support in the area of reuse tends to be focused on the type of product being reused. At one end of the spectrum we have reuse of ontologies in tools such as Protégé [Schreiber00b], at the other there are tools to facilitate the reuse of complete problem solving architectures [Motta99, Fensel99, Crubézy02]. Obstacles to reuse include the very real possibility that it is sometimes easier to reconstruct the knowledge fragment than hunt for it. Even when it is found it is often necessary to modify it to suit the current context. Some knowledge is so difficult to model in a reusable fashion that an explicit decision is made to reacquire when needed.

Having acquired knowledge, modelled and stored it, the issue then arises as to how to get that knowledge to the people who subsequently need it. The challenge of *knowledge publishing* or disseminating can be described as getting the right knowledge, in the right form, to the right person or system, at the right time. Different users and systems will require knowledge to be presented and visualised in different ways. The quality of such presentation is not merely a matter of preference. It may radically affect the utility of the knowledge. Getting presentation right involves understanding the different perspectives of people with different agendas and systems with different requirements. An understanding of knowledge content will help to

ensure that important related pieces of knowledge get published at the appropriate time.

Technologies to help publish content in fast and flexible are now starting to appear. One such is the distributed link service (DLS). This is a method for associating hyperlinks with content in such a way that the link is held separate from the content and not represented in the content itself. This means that different link structures or *link bases* can be associated with the same content. This allows very different hypertext structures to be associated with the same content and supports very different styles of publishing and subsequently navigating content [Carr98]. More recently DLS systems have been built that generate links that can be switched in and out depending on the ontology or conceptualisation in play at the time [Carr01]. Ontologies can also act as filters on portals. By looking at the metadata associated with content the portal can elect to show various content in different ways to different users [<http://www.ontportal.org.uk/>].

Given accumulated fragments of knowledge, methods now exist to thread this information together and generate connected text to explain or present the fragments [Bontcheva01, Bontcheva01b]. Some publication models seek to generate extended narratives from harvested web content [Sanghee02]. Publishing services extend to concepts such as the Open Archives Initiative [Harnad01] and ePrints [Hitchcock00]. In these models individuals deposit their papers with associated metadata. The ePrints system for example can then offer the basis for additional services running on a significant publication base. For example, it currently runs on the Los Alamos Physics Archive consisting of some 100,000 documents and offers citation and automatic cross-indexing services [<http://opcit.eprints.org/>].

Problems with publication include the fact that it has to be timely and it should not overwhelm the recipient with detail nor content that is not of interest. Related to these last two issues we find technologies under development to carry out summarisation [Knight00] of texts and subject content identification [Landauer97, Landauer98].

Finally, having acquired and modelled the knowledge, and having managed to retrieve and disseminate it appropriately, the last challenge is to keep the knowledge content current – *knowledge maintenance*. This may involve the regular updating of content as knowledge changes. Some content has considerable longevity, while other knowledge dates quickly. If knowledge is to remain useful over a period of time, it is essential to know which parts of the knowledge base must be updated or else discarded and when. Other problems involved in maintenance include verifying and validating the content, and certifying its safety.

Historically, the difficulty and expense of maintaining large software systems has been underestimated. Where that information and knowledge content is to be maintained in a distributed fashion the problem would appear to be even more acute. Whether it is a repository full of documents or databases full of experimental data the problem of curation needs to be addressed early in the system design process. Moreover, it needs to be tackled early in the knowledge life cycle. When content is acquired and modelled metadata regarding its provenance, quality and value ought to be captured too. Otherwise one has little evidence about what it is important to maintain and what are the likely consequences if it is changed or removed.

Technologies have been developed to look at the effects of refining and maintaining knowledge bases [Carbonara99]. These attempt to implement a range of checking algorithms to see if altering the knowledge base leads to cyclic reasoning behaviour or else disables or enables new classes of inference or behaviours. A different type of maintenance relates to the domain descriptions or conceptualisations themselves. Again it is important that at the point at which the ontology is designed careful thought is given to those parts of the conceptualisation that are likely to remain stable as opposed to areas where it is recognised that change and modification is likely to happen. Once built an ontology is typically populated with instances to produce the knowledge bases over which processing occurs. Populating ontologies with instances is a constant process of maintenance and whenever it is carried out there can be much post processing to eliminate for example duplicate instances from the knowledge base [Alani02].

As with so many aspects of the knowledge life cycle, effective maintenance will also depend on socio-technical issues having to do with whether there are clear owners and stakeholders whose primary function is content and knowledge management.

We have already indicated that if the knowledge intensive activities described above are to be delivered effectively in the Semantic Grid context then a crucial step is to establish a basic level of semantic interoperation [section 3.2.2). This requires the development of a shared vocabulary, description or conceptualisation for the particular domain of interest. It is to this ontological engineering that we now turn.

4.2 Ontologies and the Knowledge Layer

The concept of an *ontology* is necessary to capture the expressive power that is needed for modelling and reasoning with knowledge. Generally speaking, an ontology determines the extension of terms and the relationships between them. However, in the context of knowledge and web engineering, an ontology is simply a published, more or less agreed, conceptualization of an area of content. The ontology may describe objects, processes, resources, capabilities or whatever.

Recently a number of languages have appeared that attempt to take concepts from the knowledge representation languages of AI and extend the expressive capability of those of the Web (e.g., RDF and RDF Schema). Examples include SHOE [Luke00], DAML [Hendler00], and OIL [vanHarmelen00]. Most recently there has been an attempt to integrate the best features of these languages in a hybrid called DAML+OIL. As well as incorporating constructs to help model ontologies DAML+OIL is being equipped with a logical language to express rule-based generalizations.

However the development of the Semantic Grid is not simply about producing machine-readable languages to facilitate the interchange and integration of heterogeneous information. It is also about the elaboration, enrichment and annotation of that content. To this end, the list below is indicative of how rich annotation can become. Moreover it is important to recognize that enrichment or meta-tagging can be applied at any conceptual level in the three tier grid of figure 1. This yields the idea of meta-data, meta-information and meta-knowledge.

- Domain ontologies: Conceptualisations of the important objects, properties and relations between those objects. Examples would include an agreed set of annotations for medical images, an agreed set of annotations for climate information, and a controlled set of vocabulary for describing significant features of engineering design.
- Task ontologies: Conceptualisations of tasks and processes, their interrelationships and properties. Examples would include an agreed set of descriptors for the stages of a synthetic chemistry process, an agreed protocol for describing the dependencies between optimisation methods, and a set of descriptions for characterizing the enrichment or annotation process when describing a complex medical image.
- Quality ontologies: Conceptualisations of the attributes that knowledge assets possess and their interrelationships. Examples would include annotations that would relate to the expected error rates in a piece of medical imaging, the extent to which the quality of a result from a field geologist depended on their experience and qualifications, and whether results from particular scientific instruments were likely to be superseded by more accurate devices.
- Value ontologies: Conceptualisations of those attributes that are relevant to establishing the value of content. Examples would include the cost of obtaining particular physics data, the scarcity of a piece of data from the fossil record, and how widely known a particular metabolic pathway was.
- Personalisation ontologies: Conceptualisations of features that are important to establishing a user model or perspective. Examples would include a description of the prior familiarity that a scientist had with particular information resources, the amount of detail that the user was interested in, and the extent to which the user's current e-Science activities might suggest other content of interest.
- Argumentation ontologies – A wide range of annotations can relate to the *reasons why* content was acquired, why it was modelled in the way it was, and who supports or dissents from it. This is particularly powerful when extended to the concept of associating discussion threads with content. Examples are the integration of authoring and reviewing processes in on-line documents. Such environments allow structured discussions of the evolution and development of an idea, paper or concept. The structured discussion is another annotation that can be held in perpetuity. This means that the reason for a position in a paper or a design choice is linked to the object of discussion itself.

The benefits of an ontology include improving communication between systems whether machines, users or organizations. They aim to establish an agreed and perhaps normative model. They endeavour to be consistent and unambiguous, and to integrate a range of perspectives. Another benefit that arises from adopting an ontology is inter-operability and this is why they figure large in the vision for the Semantic Web [BernersLee01]. An ontology can act as an interlingua, it can promote

reuse of content, ensure a clear specification of what content or a service is about, and increase the chance that content and services can be successfully integrated.

A number of ontologies are emerging as a consequence of commercial imperatives where vertical marketplaces need to share common descriptions. Examples include the Common Business Library (CBL), Commerce XML (cXML), ecl@ss, the Open Applications Group Integration Specification (OAGIS), Open Catalog Format (OCF), the Open Financial Exchange (OFX), Real Estate Transaction Markup Language (RETML), RosettaNet, UN/SPSC (see www.diffuse.org), and UCEC.

We can see examples of ontologies built and deployed in a range of traditional knowledge intensive applications ranging from chemical processing [Lopez99] through to engineering plant construction [Mizoguchi00]. Moreover, there are a number of large-scale ontology initiatives underway in specific scientific communities. One such is in the area of genetics where a great deal of effort has been invested in producing common terminology and definitions to allow scientists to manage their knowledge [<http://www.geneontology.org/>]. This effort provides a glimpse of how ontologies will play a critical role in sustaining the e-Scientist.

This work can also be exploited to facilitate the sharing, reuse, composition, mapping, and succinct characterizations of (web) services. In this vein, [McIlraith01] exploit a web service markup that provides an agent-independent *declarative API* that is aimed at capturing the data and metadata associated with a service together with specifications of its properties and capabilities, the interface for its execution, and the prerequisites and consequences of its use. A key ingredient of this DAML work is that the markup of web content exploits ontologies. They have used DAML for *semantic markup* of Web Services. This provides a means for agents to populate their local knowledge bases so that they can reason about web services to perform automatic web service discovery, execution, composition and interoperation.

It can be seen that ontologies clearly provide a basis for the communication, integration and sharing of content. But they can also offer other benefits. An ontology can be used for improving search accuracy by removing ambiguities and spotting related terms, or by associating the information retrieved from a page with other information. They can act as the backbone for accessing information from a community web portal [Staab00]. Moreover Internet reasoning systems are beginning to emerge that exploit ontologies to extract and generate annotations from the existing web [Decker99].

Given the developments outlined in this section, a general process that might drive the emergence of the knowledge grid would comprise:

- The development, construction and maintenance of application (specific and more general areas of science and engineering) and community (sets of collaborating scientists) based ontologies.
- The large scale annotation and enrichment of scientific data, information and knowledge in terms of these ontologies
- The exploitation of this enriched content by knowledge technologies.

There is a great deal of activity in the whole area of ontological engineering at the moment. In particular, the World Wide Web Consortium (W3C) has a working group developing a language to describe ontologies on the web; this Web Ontology language, which is known as OWL, is based on DAML+OIL. The development and deployment of ontologies is a major topic in the web services world and is set to assume an important role in grid computing.

4.3 Knowledge Layer Aspects of the Scenario

Let us now consider our scenario in terms of the opportunities it offers for knowledge services (see table 5). We will describe the knowledge layer aspects in terms of the agent-based service oriented analysis developed in section 3.3. Important components of this conceptualization were the software proxies for human agents such as the *scientist agent* (SA) and the *technician agent* (TA). These software agents will interact with their human counterparts to elicit preferences, priorities and objectives. The software proxies will then realise these elicited items on the Grid. This calls for knowledge acquisition services. As indicated in section 4.1, a range of methods could be used. Structured interview methods invoke templates of expected and anticipated information. Scaling and sorting methods enable humans to rank their preferences according to relevant attributes that can either be explicitly elicited or pre-enumerated. The laddering method enables users to construct or select from ontologies. Knowledge capture methods need not be explicit – a range of pattern detection and induction methods exist that can construct, for example, preferences from past usage.

One of the most pervasive knowledge services in our scenario is the partial or fully automated annotation of scientific data. Before it can be used as knowledge, we need to equip the data with meaning. Thus agents require capabilities that can take data streaming from instruments and annotate it with meaning and context. Example annotations include the experimental context of the data (where, when, what, why, which, how). Annotation may include links to other previously gathered information or its contribution and relevance to upcoming and planned work. Such knowledge services will certainly be one of the main functions required by the Analyser Agent and Analyser Database Agent (ADA). In the case of the High Resolution Analyser Agent (HRAA) we have the additional requirement to enrich a range of media types with annotations. In the original scenario this included video of the actual experimental runs.

These acquisition and annotation services along with many others will be underpinned by ontology services that maintain agreed vocabularies and conceptualizations of the scientific domain. These are the names and relations that hold between the objects and processes of interest to us. Ontology services will also manage the mapping between ontologies that will be required by agents with differing interests and perspectives.

Agent Requirements	Knowledge Technology Services
Scientist Agent (SA)	Knowledge Acquisition of Scientist Profile Ontology Service
Technician Agent (TA)	Knowledge Acquisition of Technician Profile Ontology Service Knowledge Based Scheduling Service to book analyser
Analyser Agent	Annotation and enrichment of instrument streams

(AA)	Ontology Service
Analysers Database Agent (ADA)	Annotation and enrichment of databases Ontology Service
High Resolution Analyser Agent (HRAA)	Annotation and enrichment of media Ontology Service Language Generation Services Internet Reasoning Services
Interest Notification Agent (INA)	Knowledge Publication Services Language Generation Services Knowledge Personalisation Services Ontology Service
Experimental Results Agent (ERA)	Language Generation Services Result Clustering and Taxonomy Formation Knowledge and Data Mining Service Ontology Service
Research Meeting Convener Agent (RMCA)	Constraint Based Scheduling Service Knowledge Personalisation Service Ontology Service
International Sample Database Agent (ISDA)	Result Clustering and Taxonomy Formation Knowledge and Data Mining Services Ontology Service
Paper Repository Agent (PRA)	Annotation and enrichment of papers Ontology Service Dynamic Link Service Discussion and Argumentation Service
Problem Solving Environment Agent (PSEA)	Knowledge Based Configuration of PSE Components Knowledge Based Parameter Setting and Input Selection Ontology Service

Table 5: Example knowledge services in the scenario

Personalisation services will also be invoked by a number of the agents in the scenario. These might interact with the annotation and ontology services already described so as to customize the generic annotations with personal markup – the fact that certain types of data are of special interest to a particular individual. Personal annotations might reflect genuine differences of terminology or perspective – particular signal types often have local vocabulary to describe them. Ensuring that certain types of content are noted as being of particular interest to particular individuals brings us on to services that notify and push content in the direction of interested parties. The Interest Notification Agent (INA) and the Research Meeting Convener Agent (RMCA) could both be involved in the publication of content either customized to individual or group interests. Portal technology can support the construction of dynamic content to assist the presentation of experimental results.

Agents such as the High Resolution Analyser (HRAA) and Experimental Results Analyser (ERA) have interests in classifying or grouping certain information and annotation types together. Examples might include all signals collected in a particular context, or sets of signals collected and sampled across contexts. This in turn provides a basis for knowledge discovery and the mining of patterns in the content. Should such patterns arise these might be further classified against existing pattern types held in international databases – in our scenario this is managed in marketplaces by agents such as the International Sample Database Agent (ISDA).

At this point agents are invoked whose job it is to locate other systems or agents that might have an interest in the results. Negotiating the conditions under which the results can be released, determining the quality of results, might all be undertaken by agents that are engaged to provide result brokering and result update services.

Raw results are unlikely to be especially interesting so that the generation of natural language summaries of results will be important for many of the agents in our scenario. Results that are published this way will also want to be linked and threaded to existing papers in the field and made available in ways that discussion groups can usefully comment on. Link services are one sort of knowledge technology that will be ubiquitous here – this is the dynamic linking of content in documents in such a way that multiple markups and hyperlink annotations can be simultaneously maintained. Issue tracking and design rationale methods allow multiple discussion threads to be constructed and followed through documents. In our scenario the Paper Repository Agent (PRA) will not only retrieve relevant papers but mark them up and thread them in ways that reflect the personal interests and conceptualizations (ontologies) of individuals or research groups.

The use of Problem Solving Environment Agents (PSEAs) in our simulation of experimentally derived results presents us with classic opportunities for knowledge intensive configuration and processing. Once again these results may be released to communities of varying size with their own interests and viewpoints.

Ultimately it will be up to application designers to determine if the knowledge services described in this scenario are invoked separately or else as part of the inherent competences of the agents described earlier. Whatever the design decisions, it is clear that knowledge services will play a fundamental role in realizing the potential of the Semantic Grid for the e-Scientist.

4.4 Research Issues

The following is a list of the key research issues that remain for exploiting knowledge services in the Semantic Grid. In many cases there are already small-scale exemplars for most of these services; consequently many of the issues relate to the problems of scale and distribution

- Languages and infrastructures are needed to describe, advertise and locate knowledge services. We need the means to invoke and communicate the results of such services. This is the sort of work that is currently underway in the Semantic Web effort of DAML-S [ref]. However, it is far from clear how this work will interface with that of the agent based computing, web services and grid communities.
- Methods are required to build large-scale ontologies and tools deployed to provide a range of ontology services.
- Annotation services are required that will run over large corpora of local and distributed data. In some cases, for example, the annotation and cleaning of physics data, this process will be iterative and will need to be near real time as well as supporting fully automatic and mixed initiative modes. These annotation tools are required to work with mixed media.
- Knowledge capture tools are needed that can be added as plugins to a wide variety of applications and which draw down on ontology services. This will

include a clearer understanding of profiling individual and group e-Science perspectives and interests.

- Dynamic linking, visualization, navigation and browsing of content from many perspectives over large content sets
- Retrieval methods based on explicit annotations.
- Construction of repositories of solution cases with sufficient annotation to promote reuse as opposed to discovering the solution again because the cost of finding the reusable solution is too high.
- Deployment of routine natural language processing as Internet services. Capabilities urgently required include: tagging and markup of documents, discovering different linguistic forms of ontological elements, and providing language generation and summarization methods for routine scientific reporting
- Deployment of Internet based reasoning services – whether as particular domain PSEs or more generic problem solvers such as scheduling and planning systems.
- Provision of knowledge discovery services with standard input/output APIs to ontologically mapped data
- Understanding how to embed knowledge services in ubiquitous and pervasive devices

5. Conclusions

This paper has outlined our vision of the Semantic Grid as a future e-Science infrastructure in which there is a high degree of easy-to-use and seamless automation and in which there are flexible collaborations and computations on a global scale. We have argued that this infrastructure should be conceptualised and implemented as a service-oriented architecture in which agents interact with one another in various types of information marketplace. Moreover, we have highlighted the importance of knowledge services in this vision and have outlined the key research challenges that need to be addressed at this level.

In order to make the Semantic Grid a reality, a number of research challenges need to be addressed. These include (in no particular order):

- *Smart Laboratories.* We believe that for e-Science to be successful and for the Grid to be effectively exploited much more attention needs to be focused on how laboratories need to be instrumented and augmented. For example, infrastructure that allows a range of equipment to advertise its presence, be linked together, annotate and markup content it is receiving or producing.
- *Service-Oriented Architectures.* Research the provision and implementation of grid facilities in terms of service oriented architectures. Also research into service description languages as a way of describing and integrating the Grid's problem solving elements.
- *Agent Based Approaches.* Research the use of agent based architectures and interaction languages to enable e-Science marketplaces to be developed, enacted and maintained.
- *Trust and Provenance.* Further research is needed to understand the processes, methods and techniques for establishing computational trust and determining the

provenance and quality of content in Grid systems. This extends to the issue of digital rights management in making content available.

- *Metadata and Annotation.* Whilst the basic metadata infrastructure already exists in the shape of RDF, metadata issues have not been fully addressed in current grid deployments. It is relatively straightforward to deploy some of the technology in this area, and this should be promoted. RDF, for example, is already encoding metadata and annotations as shared vocabularies or ontologies. However, there is still a need for extensive work in the area of tools and methods to support the design and deployment of e-Science ontologies. Annotation tools and methods need to be developed so that emerging metadata and ontologies can be applied to the large amount of content that will be present in Grid applications.
- *Knowledge Technologies.* In addition to the requirement for the research in metadata and annotation, there is a need for a range of other knowledge technologies to be developed and customised for use in e-Science contexts. These include knowledge capture tools and methods, dynamic content linking, annotation based search, annotated reuse repositories, natural language processing methods (for content tagging, mark-up, generation and summarisation), data mining, machine learning and internet reasoning services. These technologies will need shared ontologies and service description languages if they are to be integrated into the e-Science workflow. These technologies will also need to be incorporated into the pervasive devices and smart laboratory contexts that will emerge in e-Science.
- *Integrated Media.* Research into incorporating a wide range of media into the e-Science infrastructure. This will include video, audio, and a wide range of imaging methods. Research is also needed into the association of metadata and annotation with these various media forms.
- *Content Presentation.* Research is required into methods and techniques that allow content to be visualised in ways consistent with the e-Science collaborative effort. This will also involve customising content in ways that reflect localised context and should allow for personalisation and adaptation.
- *e-Science Workflow and Collaboration.* Much more needs to be done to understand the workflow of current and future e-Science collaborations. Users should be able to form, maintain and disband communities of practice with restricted membership criteria and rules of operation. Currently most studies focus on the e-Science infrastructure *behind* the socket on the wall. However this infrastructure will not be used unless it fits in with the working environment of the e-Scientists. This process has not been studied explicitly and there is a pressing need to gather and understand these requirements. There is a need to collect real requirements from users, to collect use cases and to engage in some evaluative and comparative work. There is also a need to more fully understand the process of collaboration in e-Science.
- *Pervasive e-Science.* Currently most references and discussions about grids imply that their primary task is to enable global access to huge amounts of computational power. Generically, however, we believe grids should be thought of as the means of providing seamless and transparent access from and to a diverse set of networked resources. These resources can range from PDAs to supercomputers and from sensor's and smart laboratories to satellite feeds.
- *e-Anything.* Many of the issues, technologies and solutions developed in the context of e-Science can be exploited in other domains where groups of diverse stakeholders need to come together electronically and interact in flexible ways.

Thus it is important that relationships are established and exploitation routes are explored with domains such as e-Business, e-Commerce, e-Education, and e-Entertainment.

7 References

- [Alani02] Alani, H., Dasmahapatra, S., Gibbins, N., Glaser, H., Harris, S., Kalfoglou, Y., O'Hara, K., and Shadbolt, N. Managing Reference: Ensuring Referential Integrity of Ontologies for the Semantic Web. 14th International Conference on Knowledge Engineering and Knowledge Management, Spain, October, 2002.
- [BernersLee01] Berners-Lee, T., Hendler, J. and Lassila, O. "The Semantic Web", Scientific American, May 2001.
- [BernersLee99] Berners-Lee, T. with Fischetti, M. "Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor", Harper, San Francisco, 1999.
- [Brachman83] Brachman, R.J. & Levesque, H.J. (1983). *Readings in Knowledge Representation*. San Mateo, California: Morgan Kaufmann Publishers.
- [Bontcheva01] Bontcheva, K. Tailoring the Content of Dynamically Generated Explanations. M. Bauer, P.J. Gmytrasiewicz, J. Vassileva (eds). *User Modelling 2001: 8th International Conference, UM2001*, Lecture Notes in Artificial Intelligence 2109, Springer Verlag, 2001.
- [Bontcheva01b] Bontcheva, K., Wilks, Y. Dealing with Dependencies between Content Planning and Surface Realisation in a Pipeline Generation Architecture. In *Proceedings of International Joint Conference in Artificial Intelligence (IJCAI'01)*, August 7-10. Seattle, 2001.
- [Carr98] Carr, L., De Roure, D., Davis, H., Hall, W., Implementing an Open Link Service for the World Wide Web. *World Wide Web Journal*. 1(2), 61-71. Baltzer. 1998
- [Carr01] Carr, L., Hall, W., Bechhofer, S. and Goble, C. (2001) Conceptual Linking: Oncology-based Open Hypermedia. *Proceedings of the Tenth International World Wide Web Conference, Hong Kong, May 1-5* p.334-342.
- [Cerf93] Cerf, V. G., et al., "National Collaboratories: Applying Information Technologies for Scientific Research", National Academy Press: Washington, D.C., 1993.

- [Carbonara99] Carbonara, L. and Sleeman, D. (1999) Effective and Efficient Knowledge Base Refinement. *Machine Learning*, Vol 37, pp143-181.
- [Ciravegna01] Ciravegna, F.: "Adaptive Information Extraction from Text by Rule Induction and Generalisation" in Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001), Seattle, August 2001."
- [Croft00] Croft, W. B. Information Retrieval Based on Statistical Language Models. ISMIS 2000: 1-11
- [Crow01] Crow, L. and Shadbolt, N. R. (2001) Extracting Focused Knowledge from the Semantic Web, *International Journal of Human Computer Studies*, Vol. 54 (1) pp 155-184.
- [Crubézy02] Crubézy, M., Lu, W., Motta, E. and Musen, M. A. . Configuring Online Problem-Solving Resources with the Internet Reasoning Service. Conference on Intelligent Information Processing (IIP 2002) of the International Federation for Information Processing World Computer Congress (WCC 2002), Montreal, Canada. Kluwer. 2002.
- [DAML02] DAML Services Coalition (alphabetically Anupriya Ankolenkar, Mark Burstein, Jerry R. Hobbs, Ora Lassila, David L. Martin, Drew McDermott, Sheila A. McIlraith, Srini Narayanan, Massimo Paolucci, Terry R. Payne and Katia Sycara), "DAML-S: Web Service Description for the Semantic Web", in The First International Semantic Web Conference (ISWC), June, 2002, pp 348-363.
- [Decker99] Decker, S., Erdmann, M., Fensel, D. and Studer, R. "Ontobroker: ontology-based access to distributed and semi-structured information" in R. Meersman (ed.) *Semantic Issues in Multimedia Systems: Proceedings of DS-8*, Kluwer Academic, Boston, 1999, 351-369.
- [DeRoure01] De Roure, D., Jennings, N. R. and Shadbolt, N. R., "Research Agenda for the Semantic Grid: A Future e-Science Infrastructure" Technical Report of the National e-Science Centre, UKeS-2002-02, 2001.
- [DeRoure02] De Roure, D., Baker, M., Jennings, N. R., and Shadbolt, N. R., "The Evolution of the Grid" in this volume, 2002.
- [Faratin99] Faratin, P., Sierra, C., and Jennings, N.R., "Negotiation decision functions for autonomous agents" *Int. J. of Robotics and Autonomous Systems* **24** (3-4), 1999, 159-182.
- [Fensel99] Fensel, D., Benjamins, V. R., Motta, E. and Wielinga, B. (1999). UPML: A Framework for knowledge system reuse. In *Proceedings of the International Joint Conference on AI (IJCAI-99)*, Stockholm, Sweden, July 31 - August 5, 1999.

- [Foster98] Foster, I., and Kesselman, C., (eds), “The Grid: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann, July 1998.
- [Foster01] Foster, I., Kesselman, C., and Tuecke, S., “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, *Int. Journal of Supercomputer Applications and High Performance Computing*, 2001.
- [Foster02] Foster, I., Kesselman, C., Nick, J. and Tuecke, S., *The Physiology of the Grid: Open Grid Services Architecture for Distributed Systems Integration*, presented at GGF4, Feb. 2002 <http://www.globus.org/research/papers/ogsa.pdf>.
- [Gamma95] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA, Addison-Wesley.
- [Guttman98] Guttman, R.H., Moukas, A. G., and Maes, P., “Agent-mediated electronic commerce: a survey” *The Knowledge Engineering Review* 13 (2) 1998 147-159.
- [Handschuh02] Handschuh, S., Staab, S. and Ciravegna, F. S-CREAM --- Semi-automatic CREAtion of Metadata. 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2002) Sigüenza, Spain.
- [Harnad01] Harnad, Stevan (2001) The Self-Archiving Initiative. *Nature*, 410 p.1024-1025.
- [Hayes-Roth83] Hayes-Roth, F. Waterman, D. A. and Lenat, D. B. (1983) *Building Expert Systems*. Reading, Mass.: Addison-Wesley.
- [Hendler00] Hendler, J., and McGuinness, D., “The DARPA Agent Markup Language,” *IEEE Intelligent Systems* 15 (6), 2000, 72–73.
- [Hitchcock00] Hitchcock, S., Carr, L., Jiao, Z., Bergmark, D., Hall, W., Lagoze, C. and Harnad, Stevan (2000) Developing services for open eprint archives: globalisation, integration and the impact of links. *Proceedings of the 5th ACM Conference on Digital Libraries, San Antonio, Texas, June 2000*. p.143-151.
- [Hoffman95] Hoffman, R., Shadbolt, N.R., Burton, A.M. and Klein,G. (1995) “Eliciting Knowledge from Experts: A Methodological Analysis” *Organizational Behavior and Decision Processes*, 62 (2) 1995, 129-158. Academic Press.
- [Jennings00] Jennings, N.R., “On agent-based software engineering”, *Artificial Intelligence* 117, 2000, 277-296.

- [Jennings01] Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., and Wooldridge, M. "Automated Negotiation: Prospects, Methods and Challenges" *Int Journal of Group Decision and Negotiation* 10(2) 2001, 199-215.
- [Knight00] Knight, K. and Marcu, D. Statistics-Based Summarization --- Step One: Sentence Compression, *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 2000.
- [Kraus01] S. Kraus "Strategic Negotiation in Multi-agent Environments" MIT Press. 2001.
- [Landauer97] Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240.
- [Landauer98] Landauer, T. K., Foltz, P. W., & Laham, D. (1998) Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259-284.
- [Lenz98] Lenz, M., Bartsch-Spörl, B., Burkhard, H., and Wess, S. (Eds.) *Case-Based Reasoning Technology - From Foundations to Applications*. Lecture Notes in Artificial Intelligence 1400, Springer Verlag, 1998
- [Lopez 99] Lopez, M.F., Gomez-Perex, A. et al., Building a chemical ontology using Methontology and the ontology design environment, *IEEE Intelligent Systems*, Vol.14, No.1, pp.37-46, 1999.
- [Luke00] S. Luke and J. Heflin, "SHOE 1.01. Proposed Specification", www.cs.umd.edu/projects/plus/SHOE/spec1.01.html, 2000 (current 20 Mar. 2001).
- [McIlraith01] McIlraith, S. A., Son, T. C., and Zeng, H., "Semantic Web Services" *IEEE Intelligent Systems*, 16 (2) 2001, 46-53.
- [Middleton01] Middleton, S. E., De Roure, D and Shadbolt, N. R. (2001) *Capturing knowledge of user preferences: ontologies in recommender systems*. *Proceedings of the First International Conference on Knowledge Capture, K-CAP2001*, ACM Press.
- [Milton99] Milton, N., Shadbolt, N., Cottam, H. and Hammersley, M. (1999). Towards a Knowledge Technology for Knowledge Management. *International Journal of Human-Computer Studies*, 51(3), 615-64.
- [Motta99] Motta, E., Fensel, D., Gaspari, M. and Benjamins, R. (1999). Specifications of Knowledge Components for Reuse. *Eleventh International Conference on Software Engineering and Knowledge Engineering (SEKE '99)*. June 1999
- [Motta00] Motta, E., Buckingham Shum, S. and Domingue, J. (2001) *Ontology-Driven Document Enrichment: Principles, Tools and Applications*.

- International Journal of Human Computer Studies, 52(5), pp. 1071-1109, 2000.
- [Motta02] Motta, E., Vargas-Vera, M., Domingue, J., Lanzoni, M., Stutt, A., and Ciravegna, F. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2002) Sigüenza, Spain.
- [Mizoguchi00] Mizoguchi, R., Kozaki, K., Sano, T., and Kitamura, Y.: Construction and Deployment of a Plant Ontology, 12th International Conference on Knowledge Engineering and Knowledge Management, Juan-les-Pins, French Riviera, October, 2000.
- [Newell82] Newell, A., "The Knowledge Level" Artificial Intelligence 18 1982, 87-127.
- [Sandholm00] Sandholm, T., "Agents in Electronic Commerce: Component Technologies for Automated Negotiation and Coalition Formation" Autonomous Agents and Multi-Agent Systems 3(1) 2000, 73-96.
- [Sanghee02] Sanghee, K., Alani, H., Hall, W., Lewis, P., Millard, D., Shadbolt, N., Weal, M. Artequakt: Generating Tailored Biographies with Automatically Annotated Fragments from the Web. In *Proceedings Semantic Authoring, Annotation and Knowledge Markup Workshop in the 15th European Conference on Artificial Intelligence*, Lyon, France. 2002.
- [Schreiber00] Schreiber G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N.R., Van de Velde, W. and Wielinga, B. (2000) *Knowledge Engineering and Management*. MIT Press.
- [Schreiber00b] Schreiber, G., Crubezy, M. & Musen, M. A. A Case Study in Using Protege-2000 as a Tool for CommonKADS. 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, 33-48. 2000. Springer LNAI.
- [Shadbolt99] Shadbolt, N.R., O'Hara, K., Crow, L. (1999) The Experimental Evaluation of Knowledge Acquisition Techniques and Methods: History, Problems and New Directions *International Journal of Human-Computer Studies*, 51(4), 729-755.
- [Shadbolt95] Shadbolt, N.R. and Burton, M. (1995) Knowledge elicitation: a systematic approach, in *Evaluation of human work: A practical ergonomics methodology* edited by J. R. Wilson and E. N. Corlett, Taylor and Francis, London, England, 1995. pp.406-440. ISBN-07484-0084-2.
- [Shaw98] Shaw, M.L.G. & Gaines, B.R. (1998). WebGrid-II: developing hierarchical knowledge structures from flat grids. In *Proceedings of the 11th Knowledge*

Acquisition Workshop (KAW' 98). Banff, Canada, April 18-23, 1998.
Available at <http://repgrid.com/reports/KBS/WG/>.

- [Shehory98] Shehory, O., and S. Kraus, S., “Methods for task allocation via agent coalition formation” *Artificial Intelligence*, 101 (1-2) 1998, 165-200.
- [Sparck-Jones97] Sparck-Jones, K., & Willett, P., *Readings in information retrieval*, Mountain View: Morgan Kaufmann, 1997.
- [Staab00] Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H.-P., Studer, R. and Sure, Y. “Semantic community Web portals” *Proc. of WWW-9*, Amsterdam, 2000.
- [vanHarmelen00] van Harmelen, F., and Horrocks, I., “FAQs on OIL: The Ontology Inference Layer,” *IEEE Intelligent Systems* 15 (6), 2000, 69–72.
- [WebServices01] *Proceedings of W3C Web Services Workshop*, April 11-12, 2001.
<http://www.w3.org/2001/03/wsws-program>
- [Wooldridge97] Wooldridge, M., “Agent-based software engineering”. *IEE Proc on Software Engineering* 144 (1) 1997, 26-37.