A Virtual Organisation deployed on a Service Orientated Architecture for Distributed Data Mining applications

Thomas Jackson, Mark Jessop, Martyn Fletcher, Jim Austin
Advanced Computer Architectures Group,
Department of Computer Science,
University of York, UK
{tom.jackson, mark.jessop, fletcher, austin}@cs.york.ac.uk

Abstract. Industrial and scientific research activity increasingly involves the geographically distributed utilisation of multiple tools, services and distributed data. Grid and Service Orientated Architecture concepts are being widely investigated as a means to deploy Virtual Organisations to support the needs for distributed collaboration. A generic Distributed Tool, Service and Data Architecture is described together with its application to the aero-engine domain through the BROADEN project. Two fundamental issues for the design of the VO have been addressed: how to maximise the potential of Grid computing to address the complex data mining challenges in the condition monitoring application; and how to maximise the potential of a SOA to build and deploy a flexible and efficient collaborative workbench that integrates the required tools and services.

1 Introduction

The Grid and Service Orientated Architecture (SOA) paradigm offer significant opportunities for the development and deployment of Virtual Organisations (VO) which can support complex scientific and engineering interactions. This paper describes the development of a VO architecture to support a condition health monitoring application in the aero-engine domain. This VO provides an integrated and Distributed Tool, Service and Data Architecture built on Grid technologies [1] which is driven by the operating requirement to derive commercial and scientific benefit from distributed data assets. The architecture has been developed to support the requirements of the BROADEN project (Business Resource Optimisation for

Please use the following format when citing this chapter:

Jackson, T., Jessop, M., Fletcher, M., Austin, J., 2007, in IFIP International Federation for Information Processing, Volume 239, Grid-Based Problem Solving Environments, eds. Gaffney, P. W., Pool, J.C.T., (Boston: Springer), pp. 155-170.

Aftermarket and Design on Engineering Networks) project. BROADEN is a follow-on to the DAME (Distributed Aircraft Maintenance Environment) Grid pilot project [2].

The basis of the VO in this application context is to provide a geographically distributed set of users with access to tools and distributed data to carry out collaborative fault diagnosis. The characteristics of this specific VO are typical of many collaborative VO domains:

- Geographically distributed data – potentially in vast amounts.
- Geographically distributed users - not necessarily distributed to the same places as the data.
- Diverse system stakeholders with different roles and access rights.
- Legacy tools which are standalone but may need to interoperate with other tools.
- Disparate distributed diagnostic tools and services which must be brought together in a configurable workflow system.
- Security and Quality of Service as essential prerequisites

1.1. The Condition Monitoring Application Domain

Modern aero engines operate in highly demanding operational environments with extremely high reliability. However, Data Systems & Solutions LLC and Rolls-Royce plc have shown that the adoption of advanced engine condition monitoring and diagnosis technology can reduce costs and flight delays through enhanced maintenance planning [3]. Such aspects are increasingly important to aircraft and engine suppliers where business models are based on Fleet Hour Agreements (FHA) and Total Care Packages (TCP). Rolls-Royce has collaborated with Oxford University in the development of an advanced on-wing monitoring system called QUICK [4]. QUICK performs analysis of data derived from continuous monitoring of broadband engine vibration for individual engines. Known conditions and situations can be determined automatically by QUICK and its associated Ground Support System (GSS). Less well-known conditions (e.g. very early manifestations of problems) require the assistance of remote experts (Maintenance Analysts and Domain Experts) to interpret and analyse the data. The remote expert may want to consider and review the current data, search and review historical data in detail and run various tools including simulations and signal processing tools in order to evaluate the situation. Without a supporting diagnostic infrastructure, the process can be problematic because the data, services and experts are usually geographically dispersed and advanced technologies are required to manage, search and use the massive distributed data sets. Each aircraft flight can produce up to 1 Gigabyte of data per engine, which, when scaled to the fleet level, represents a collection rate of the order of Terabytes of data per year. The storage of this data also requires vast data repositories that may be distributed across many geographic and operational boundaries. The aero-engine scenario is also typical of many other domains, for example, many areas of scientific research, healthcare, etc.

2 The VO Requirements

The diagnostic focus of the BROADEN and DAME projects has required investigation into two fundamental issues for the design of the VO:

- How to maximise the potential of Grid computing to address the complex data mining challenges in the condition monitoring application;
- How to maximise the potential of SOA to build and deploy a flexible and efficient collaborative workbench that integrates the required tool and services.

This paper reports on both of these aspects; describing a highly distributed pattern matching architecture for Grid deployment and the use of the emerging Enterprise Service Bus (ESB) framework as a means to integrate and deploy a dynamic workbench for the VO collaboration. The requirements to address both of the above issues will be explored in more detail, and will be followed by a detailed description of the solutions being developed.

2.1. The Data Mining and Pattern Matching Problem

A central challenge of the project has been to develop a data mining and pattern matching architecture within a grid framework that can scale to the terabytes of distributed data inherent within the application domain. A fundamental aim of this work has been to devise a solution that separates the distributed nature of the problem from the searching/pattern matching problem. There has also been a requirement to keep the solution generic so that the solutions can be mapped into other grid applications requiring distributed search. To this end the following broad objectives for the system were identified:

- **Scalable**. The system should be designed to operate on large data sets and utilise many remote resources efficiently.
- **Flexible**. It must be possible to add and remove resources and data assets from the system dynamically.
- **Robust**. The system should have high availability and maintain operational capability where one or more resources fail. As a consequence of this there should not be a central point of failure.
- **Transparent**. The distribution should be hidden from the end user.
- **Efficient**. The architecture should minimise the amount of data that is moved across the network infrastructure in achieving the data mining objectives.
- **Parallel**. Where searching at different locations provides concurrent operations, the architecture must support parallel execution.
- **Storage Format Independent**. It should be independent of the underlying database technology used to store the data repositories, and should operate transparently across a heterogeneous, distributed data archive facility.

The BROADEN system requires an architecture that can address these functional requirements in addition to meeting the operational demands imposed by the data management issues. A central premise of the BROADEN system is that it must operate on geographically distributed data. The scenario is that every time an aircraft lands, vibration and performance data is downloaded to the system from the QUICK monitoring units fitted to each engine. In future deployed systems, the volume of data downloadable from the QUICK system may be up to 1 GB per engine per flight. Given the large volume of data and the rate at which it is produced, there is a need to consider how the data is managed.

In order to illustrate the requirements of the aero-engine application, consider the following scenario. Heathrow, with its two runways, is authorized to handle a maximum of 36 landings per hour. Let us assume that on average half of the aircraft landing at Heathrow have four engines and the remaining half have two engines. In future, if each engine downloads around 1 GB of data per flight, the system at Heathrow must be capable of dealing with a typical throughput of around 100 GB of raw engine data per hour, all of which must be processed and stored. The data storage requirement alone for an operational day is, therefore, around 1 TB, with subsequent processing generating yet more data.

With such a vast amount of data being produced a centralised repository may place unacceptable demands on data bandwidth and will not provide a scaleable solution. Small subsets of the data could be moved for processing and analysis purposes but this would be a solution only in some limited cases, as most services will need the ability to use full data sets. Therefore, it is desirable to store data in distributed nodes and to distribute the services which act on the data actually with the data. The architecture should permit the use of distributed nodes to store data (and services), assuming other issues such as security, etc. are satisfied.

2.2. The Workbench Requirements

One of the challenges, to date, of building and deploying VO's based on SOA, is that of systems integration. Toolkits are available, such as the Globus toolkit [5], for developing and deploying web-services. However, these provide little support for the integration of the end-to-end services at the VO level within workbench or similar collaborative working environments. Within the DAME project a major part of the project effort was expended on the design and integration of the system portal that exposed tools and services to the diverse end-users. Although the final demonstrator was effective, it was a bespoke solution and not easily scaled to the needs of industrial deployment. Some of the challenges that have to be addressed within the BROADEN project lie with the nature of the tools and services that need to be deployed within the diagnostic process. A major issue has been the requirement to allow users to add tools to the system with the minimum of change to the tools. However, the characteristics of the tools being deployed in the domain means that they are often not easily integrated within a VO. For example, in many cases:

- Tools are designed to operate standalone - this is typical of legacy tools;
- Tools not designed to interoperate with other tools - again this is typical of legacy tools;

- Tools have thick desktop clients that are not efficiently deployed as a web-service

Similar diverse issues are found when analysing the nature of the web-services. The services issues can be summarised as:

- Services may be centralised - this is typical in legacy systems;
- Service may be distributed - particularly located near the data sources;
- Services may be autonomous e.g. data input services triggered by data arrival;
- Services will need to be composed as links in a workflow

The VO architecture should accommodate all the above issues, but do so in a way that supports flexibility and reconfiguration. The architecture should allow all the above tool types to interact with one another as necessary, with a minimum of change both to the tools and the system architecture, whilst still addressing the need for virtualisation and supporting many geographically dispersed users. In the following sections the data mining and pattern matching Grid solution is described, followed by an analysis of the use of ESB as the integration framework for the VO.

3 The Pattern-Matching SOA Architecture

This section introduces a service orientated architecture for deploying pattern matching and/or search functionality against data distributed over many geographically separate locations. This is based around three primary web service enabled components:

- A data-management system for handling data arriving from an aircraft or other data asset;
- A distributed query system;
- A virtual data archiving service.

The discussion will focus on a description of the architecture for the distributed query service and the virtual data archiving service. At the heart of the distributed query process are a range of web services that encompass the Pattern Match Controller (PMC), the Pattern Match Service (PMS) and the Storage Request Broker (SRB).

The distributed concepts within BROADEN dictate that the PMC and PMS services are hosted remotely and replicated at the diverse data repositories (e.g. at each potential airport where data is downloaded and stored). These services and the data repository form a ‘Data Node’ within the architecture. For the purpose of this architecture definition, each node will be treated as a single resource. In practise, it is likely that each node will utilise many resources, for example, high performance clusters, tape archives, desktop PCs and laptops. The key assumption made is that the communication bandwidth available between resources at a single node is significantly higher than that available between the distributed nodes. The role of each service will be outlined in the following sections.

3.1. The Pattern Match Controller and Search Process

The Pattern Match Controller (PMC) is the front-end service for distributed search operations. The PMC controls the search process at each node and provides all communication between nodes. An instance of the PMC resides at each node.

Each PMC:

- Has a catalogue of all other nodes in the system, to ensure that search requests can be sent to all nodes.
- Has access to a local Pattern Matching Service (PMS), which it can instruct to carry out searches.
- Manages results for all 'active' searches at its own node.

Pattern match searches can be initiated either from an end-user or from automatic workflows in the task brokerage system. In both cases, a client service (an example application providing a GUI is detailed in [6]) communicates, via web service protocols, with its nearest PMC service to request a search. The query takes the form of a request to match a region of interest against the stored fleet data at each node. The PMC node that receives the request becomes the 'master' node for that unique search task. All other nodes in the system are referred to as 'slaves' for that search.

A search scenario is shown in Figure 1. Numbered arrows show communication between services. The exact order of some communication is dependent upon when individual pattern matching services complete their search.

1. A client delivers a search request to a PMC. This becomes the master PMC for this search, and returns a unique identifier for this search that the client can use in later communication.
2. The master PMC passes the search request to all available slave PMCs.
3. All PMCs (including the master) pass the search request to their local Pattern Matching Service. At each node searching commences.
4. At each node, as the Pattern Matching Service completes its search and passes the result to its local PMC. Pattern Matching Services then clean up, discarding the search results. At the master node, the result is merged into the overall result set.
5. Slave PMCs pass the results to the master PMC. The slave PMCs can now clean up and discard the search results. The master PMC merges the new results into the overall result set as they arrive.
6. The client makes a request for the results. The master PMC returns the complete results with additional information informing the client that the search is complete.

The client can request the current search results at any time. The master PMC returns the current result set with additional information such as how many nodes have completed their search task. Typically, a client will request results at regular intervals until the search is finished.

One of the stated aims was to build a generic framework for diagnostics. An API has been designed to support this objective; it does not contain any domain specific data types or structures. The API only specifies how components should interact with the pattern match architecture. This means that application specific components can be built and configured as required by particular implementations to work with the PMC. The PMC receives domain specific data as part of a search request, but does not need to understand it, just how to deliver it to the appropriate services. The PMC is a re-usable component appropriate for distributing search requests regardless of the problem domain.

Each node can behave as a completely stand-alone pattern matching entity, capable of matching against data stored at its location. As new nodes are added, each PMC maintains a list (or catalogue) of all the nodes within the system. For each node, the IP address of the PMC at that node is the only information required. If a node fails, its entry is kept in each node list maintained by each PMC. This allows search results to reflect the fact that the entire dataset is not being searched against. If a node is to be permanently removed, a de-registration process is invoked.

These architectural features address the requirements for scalable, efficient and flexible functionality. Any number of nodes can be supported, each operates independently, and searches are carried out asynchronously. The PMC passes search requests to and from other services, without requiring any understanding of the nature of the search request. Any data that can be contained within the search request structure can be supplied as a search request. Achieving, appropriate results depends upon providing PMS implementations that can understand and process the request.

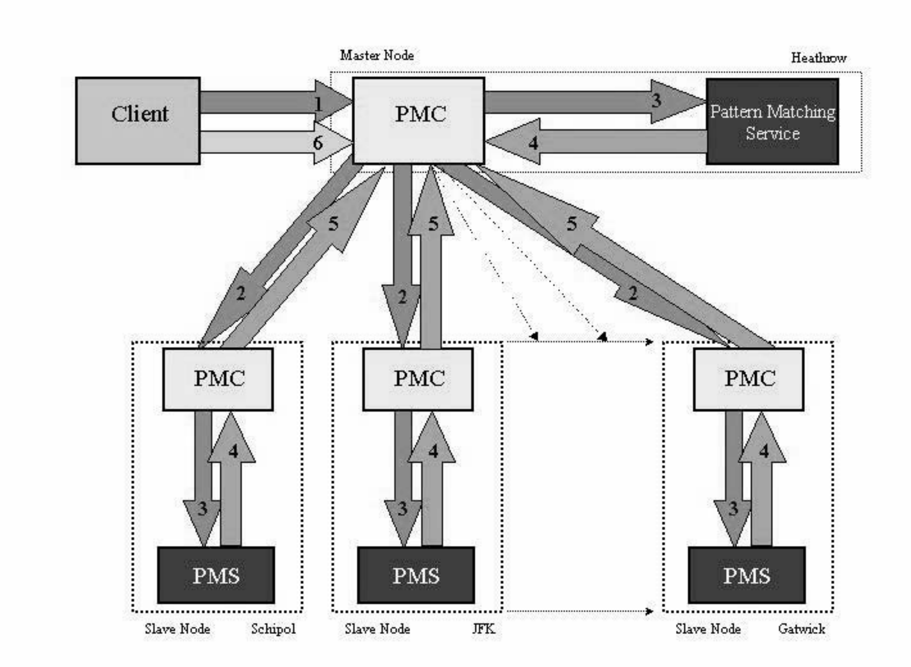

Fig.1. The communication required to complete a typical search process

3.2. Pattern Match Service

The Pattern Matching Service is responsible for performing the search across the data held at a node. This means that the pattern-matching process is handled entirely independently to the search requesting mechanism. This provides for flexibility in the architecture, and does not constrain the query process to any particular pattern-matching algorithm; the PMS can deploy any available pattern matching and/or indexing algorithms as appropriate. It receives search requests from its 'local' PMC and upon completion of the search sends the results back to the PMC. At each node the Pattern Matching Service can access the data stored there without consideration of data stored at other nodes. This feature addresses the requirement for a scalable and concurrent system by permitting the query process to be carried out as an inherently parallel operation across all nodes.

3.3. Example Query Process

To provide an example of the functionality a PMS might perform we describe the search process in the BROADEN system; matching vibration patterns against the historical flight data. The aim is to find vibration patterns that are 'similar' to the anomaly just observed (the 'query' pattern, q). The problem can be stated as finding a set of time series sub-sequences R .

$$R = \{x \in X | d(x, q) \leq \tau\}$$

τ is the maximum distance (using an appropriate metric) that a vibration pattern can be from the query and be considered a match. This is an instance of the 'query by content' or 'query by example' problem, a significant area of data mining research [7, 8, 9]. Where τ is varied to return a set of a fixed number (k) of matches in R , this is referred to as a k-nearest neighbour (k-NN) search.

Within the BROADEN system this search algorithm is implemented using the AURA [10] technology, which provides high performance neural network based pattern matching. AURA is pivotal to the requirements for meeting search requests within tight operational time constraints. It is a highly parallelised and massively scalable search engine that can implement the k-nearest neighbour algorithm extremely efficiently on massive datasets. However, as already stated, the PMS architecture is algorithm independent and the AURA pattern match engine can easily be replaced or accompanied by other pattern matching algorithms as required for the relevant data set. Simple API's allow any algorithm to be made available to the PMS if deployed as a standard web-service.

3.4. PMS Data Transfer

Passing large volumes of data (that may not be required) around the system is likely to increase search times and/or bandwidth requirements. To reduce the volume of network traffic generated, the actual data for each match is not passed between services. Details for each match include an identifier that specifies the location of the data for that result. If a client wishes to examine a result it must fetch the data, which could be located at any node in the system. Clients should be able to access/retrieve result data through a single service, without concern for which node the data is located at, i.e. treating the dataset as a single entity at a single location.

A data management system is required that will provide a single logical view to the distributed dataset and allow clients to access data from the same service, regardless of location. This data management service is responsible for 'looking up' the location of the data and fetching it.

3.5. Storage Resource Broker (SRB)

The Storage Request Broker [11] was selected as the storage mechanism to provide a virtual file indexing system at each node within the system. SRB is a tool for managing distributed storage resources, ranging from large disc arrays to tape backup systems. Files are indexed via SRB and can then be referenced by logical file handles that require no knowledge of where the file physically exists. A Meta-data Catalogue (MCAT) is maintained which maps logical handles to physical file locations. Additional system specified meta-data can be added to each record. This means that the catalogue can be queried in a data centric way, e.g. engine serial number and flight date/number, or via some characteristic of the data, rather than in a location centric fashion. When data is requested from storage, it is delivered via parallel IP streams, to maximise network through-put, using protocols provided by SRB. SRB can operate in heterogeneous environments and in many different configurations from completely stand-alone, such as one disc resource, one SRB server and one MCAT, to completely distributed with many resources, many SRB servers and completely federated MCATs.

The BROADEN system currently utilises a single MCAT but the architecture described here could be deployed with an MCAT at each node. In this configuration a user could query their local SRB system and yet work with files that are hosted remotely in a number of diverse database/file systems. The arrival of aircraft vibration data is simulated at each node; new data is stored directly into SRB on a local resource. Each location stores engine data under a logical file structure based on node location, engine type, engine serial number and flight information. Data may be made visible to any user or client regardless of their location. Pattern Matching Services access local data using SRB, querying the SRB MCAT to locate all data at that node.

The client application can access SRB directly to fetch the data for individual matches as required. This means that the client application does not need to know, or be concerned with where the data is actually located. SRB's logical view allows PMCs and pattern match services to operate on a distributed data set only accessing local data, while other BROADEN services can treat the data as if held at a central repository.

4 The Generic Distributed Tool, Service and Data Architecture

The discussion has presented an architecture that permits data mining and pattern matching queries to be carried out on highly distributed data, and on data that could potentially be managed in a highly heterogeneous range of database technologies. The architecture is generic in that the services are not constrained to any particular search task or any set of search algorithms. Considerable efforts have been expended to make the architecture robust and scalable, and to separate out the mechanisms for requesting and managing the results of queries from the process of pattern matching through the diverse data repositories. It has been demonstrated and deployed in a real-world application domain and found to be highly effective in managing the problems associated with searching distributed data assets. The following sections describe how this architecture is deployed at the heart of the VO system built around an ESB enabled workbench/portal.

The issues relating to the difficulties experienced with integrating services into a SOA to facilitate a Virtual Organisation were described earlier. The lack of tool support for integration was identified. This issue is now being addressed within the developers' community and one of the emerging frameworks for building SOA's is the Enterprise Service Bus [12] concept. The purpose of an ESB is to provide a communications 'bus', based on XML web-service standards, that facilitates application and process integration by providing distributed processing, intelligent routing, security, and dynamic data transformation. All of these services are essential elements of an end-to-end SOA, and by providing them in a standardised way as part of an infrastructure it avoids the overhead of system developers having to implement these in a bespoke manner. The other advantages of an ESB approach, which make it amenable to adoption in the BROADEN VO, are that it is distributed and that it is message based, to provide loose coupling between services. An instance of an ESB may be used within the architecture as a simple messaging and translation mechanism between the tools of the system. Tools are able to register with the ESB providing information about their type of service, their functions and the respective data formats. Translation specifications are provided to the ESB in a standard format and the ESB provides translation facilities on a tool pair basis and even on a function pair basis, if necessary.

The ESB keeps a registry of all connected tools, and routes messages between tools and translator components. Tools, once registered, might become unavailable, move to different locations or change their data formats without informing the ESB. Therefore a method of continuous verification and notification of each service will also be implemented.

Workflow, for use in the Global Workflow Manager, can be expressed in BPEL4WS and in order to make the update of workflows more user-friendly, a GUI will be included in the Workbench.

All of these properties have led to its adoption as the proposed integration framework for the BROADEN Virtual Organisation. This VO provides support to a diverse range of distributed end-users through access to a workbench. This workbench is being developed as a browser-based portal (for ease of deployment) that sits on top of an ESB architecture which orchestrates and integrates the underlying diagnostic tools and services.

Figure 2 provides an overview of the generic architecture that is being developed around the ESB. The elements shown are:

- The Graphics and Visualisation Suites which contain a set of tools at a particular user location.
- The Enterprise Service Bus to enable tool interactions.
- The distributed nodes which encapsulate the data and high performance services (see figure 3).
- The global workflow manager which provides service orchestration on demand from the individual users or as an automatic activity in an event driven mode.

Figure 3 shows the generic overview of the simplified architecture:

- The Pattern Match Controller (PMC) is a distributed component, which manages the distribution of service requests and collection and aggregation of processed results.
- The Processing Services act on local data and provide high performance search, signal extraction facilities, etc.
- Distributed Data Management provides access to local and distributed data through Storage Request Broker abstraction mechanisms.
- The Local Workflow Manager Management provides automatic and requested workflows to be enacted with a single distributed node. A local workflow may also be part of a global workflow controlled by the global workflow manager.
- The Local Resource Broker manages selection of local resources in keeping with specified Service Level Agreements (SLAs).
- Data Loading Services populate the local data stores in each node. These services may also perform other tasks such as data integrity checking, error correction, etc. on input data.
- Databases / Data stores are provided for each type of data resident in the node.

The capabilities of PMC and ESB are complementary; one provides flexibility at the cost of performance, the other reduces the flexibility but gives a gain in performance. An analysis is underway to measure these issues within the BROADEN application. We see the eventual possibility of migrating the functional aspects of ESB that are determined as essential into the PMC, once these have become clear through trial deployment.

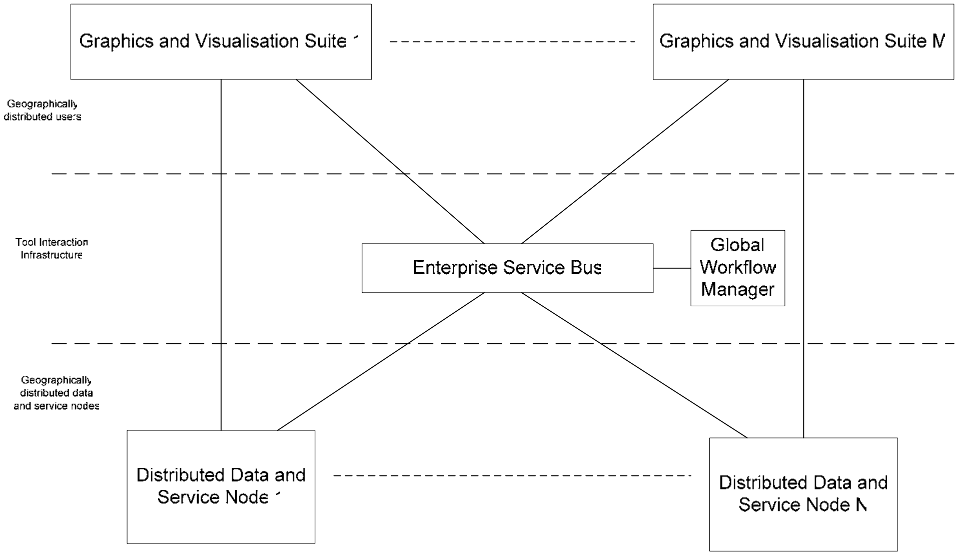
Graphics and Visualisation Suite 1
Graphics and Visualisation Suite M
Geographically distributed users
Tool Interaction Infrastructure
Enterprise Service Bus
Global Workflow Manager
Geographically distributed data and service nodes
Distributed Data and Service Node 1
Distributed Data and Service Node N

Fig.2. Overview of the Generic Tool, Service and Data Architecture

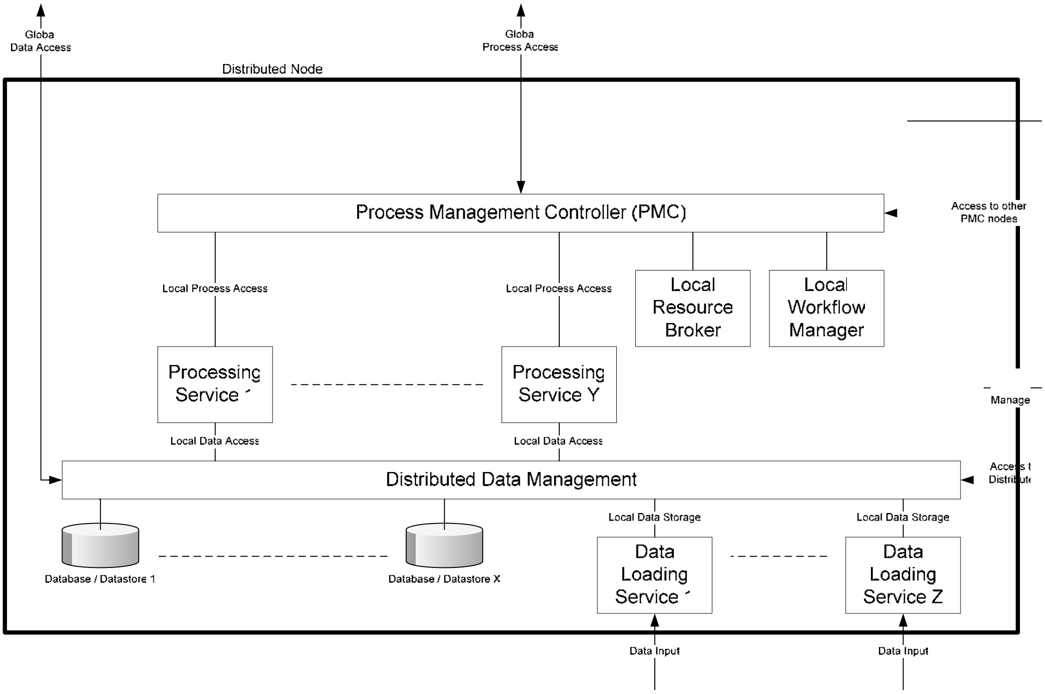

Fig.3. Overview of a Generic Distributed Service and Data Node

Figure 4 shows there are a number of tools that communicate with each other, ranging from visualisation services, data analysis, data management and workflow orchestration. The visualisation tools provided are:

- Case Based Reasoning Viewer [13] including Performance Curve Test (PCT) Viewers.
- Signal Data Explorer (SDE) is a visualisation tool used to view vibration spectral and extracted time series data [6, 14] and performance data. It also acts as the front end to the high performance pattern matching services.
- QUICK Data Viewer is a visualisation tool which allows a user to view spectral data, extracted features and events occurred [4]

The data repository at each node is responsible for storing all raw engine data along with any extracted and AURA encoded data. Each node is populated with:

- The Pattern Match Service (PMS) which is based on Advanced Uncertain Reasoning Architecture technology [10] and provides high performance pattern matching for use with the SDE or in workflows.
- The QUICK Feature Provider service which provides features and events to a client such as the QUICK Data Viewer.
- The XTO (eXtract Tracked Order) service which extract specific time series from the raw spectral vibration data.
- The Data Orchestrator is the Data Loading Service and is responsible for "cleaning" and storing all raw engine data.

Also included in the BROADEN architecture is a centralised CBR Engine and Knowledge Base [13].

The development carried out to date suggests that ESB provides an appropriate framework in which to deploy and develop the VO for the application domain, in that all of the above diverse services can be readily integrated into the architecture. The generic architecture has been developed to be application neutral. It will be implemented for the BROADEN application during the course of the project. This will allow us to assess the strengths and weaknesses of PMC and ESB for the task.
Future work will include:

- The testing and demonstration of the generic architecture in the BROADEN application domain.
- The testing and demonstration of the generic architecture in other application domains.
- The introduction of fault tolerance as appropriate (inter and intra node).
- The exploration of fault tolerance techniques within the ESB.

5 Conclusions

The paper has described the development of a VO to support distributed diagnostics for condition health monitoring applications. For this VO domain, as for many other application domains, the issues of distributed data management are central. A distributed SOA architecture has been presented that is being deployed within an industrial pilot study within the BROADEN project. The architecture is generic and will support a diverse range of data analysis methods for VO's. As an example, the methods are being redeployed within the context of a UK wide e-Science project to develop a collaborative working environment and data arching system for neural spike train data (the CARMEN project). Lessons learnt from the DAME project in regard to the development of a workbench to support the VO have been carried forward and have motivated the adoption of the Enterprise Service Bus as an integration framework. This is contributing to the development of an architecture that is flexible, configurable and based on common standards that will encourage reuse.

This work builds on the tools and services developed during the DAME project. A generic architecture has been developed, which integrates:

- Geographically distributed nodes containing data repositories and services.
- Geographically distributed users.
- Legacy Tools.
- Purpose designed tools.

It is a Grid based architecture used to manage the vast, distributed, data repositories of aero-engine health-monitoring data. In this paper we have focused on the use of a generic architecture to enable the general concept to be in other application areas and with varying degrees of legacy systems and designs. The middleware elements are generic in nature and can be widely deployed in any application domain requiring distributed tools and services and data repositories.

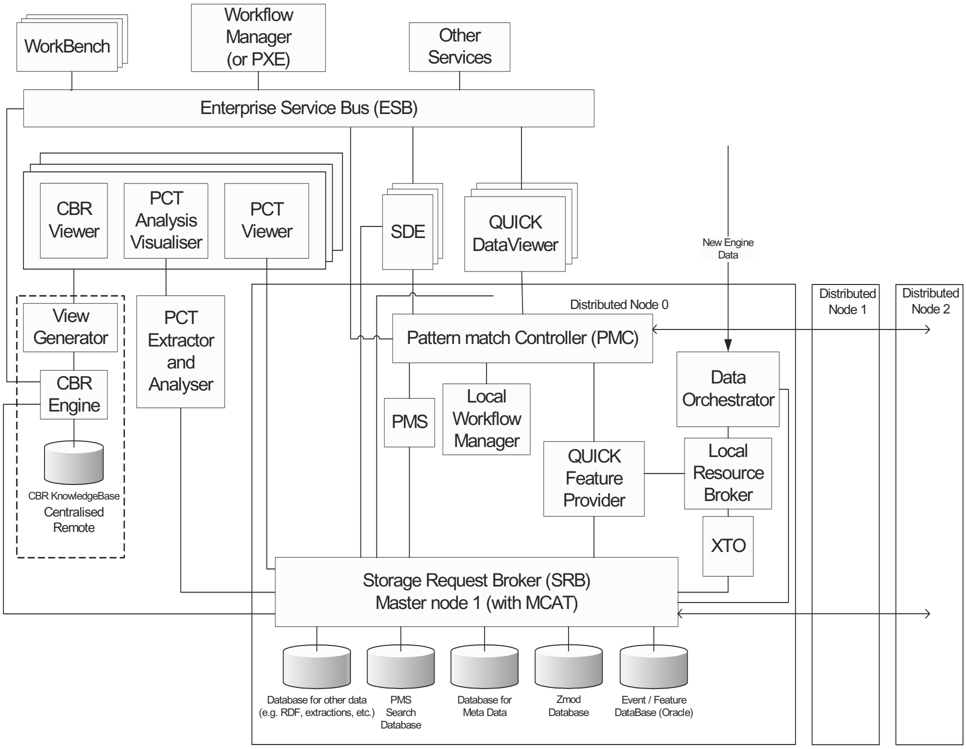

Fig.4. Overview of BROADEN Tool, Service and Data Architecture

6 Acknowledgements

The work reported in this paper was developed and undertaken as part of the BROADEN (Business Resource Optimisation for Aftermarket and Design on Engineering Networks) project, a follow-on project to the DAME project. The BROADEN project is funded via the UK Department of Trade and Industry under its Technology Innovation Programme and the work described was undertaken by teams at the Universities of York, Leeds and Sheffield with industrial partners Rolls-Royce, EDS, Oxford BioSignals and Cybula Ltd.

7 References

[1] I. Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", Int. J. Supercomputer Applications, vol. 15, no. 3, 2001.

[2] J. Austin et al., "Predictive maintenance: Distributed aircraft engine diagnostics," in The Grid, 2nd ed, I. Foster and C. Kesselman, Eds. San Mateo, CA: Morgan Kaufmann, 2003, Ch. 5.

[3] Data Systems and Solutions Core Control^{TM} technology. www.ds-s.com/corecontrol.asp

[4] A. Nairac, N. Townsend, R. Carr, S. King, P. Cowley, and L. Tarassenko, "A system for the analysis of jet engine vibration data," Integrated Computer-Aided Eng., vol. 6, pp. 53–65, 1999.

[5] The Globus Toolkit, www.globus.org

[6] Laing, B., Austin, J., A Grid Enabled Visual Tool for Time Series Pattern Match, In: Proceedings of the UK e-Science All Hands Meeting 2004, Nottingham, UK.

[7] R. Agrawal, C. Faloutos, and A. Swami, "Efficient Similarity Search in Sequence Databases", in Proc. 4th Int. Conf. Foundations of Data Organization and Algorithms (FODO), 1993, pp. 69-84.

[8] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality Reduction for Fast Similarity Search in Large Time-Series Databases", Knowl. Inf. Syst., vol. 3, no. 3, pp. 263-286, 2001.

[9] E. Keogh and S. Kasetty, "On the Need for Time-Series Data Mining Benchmarks: A Survey and Empirical Demonstration", in Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2002, pp102-111.

[10] The AURA and AURA-G web site, Advanced Computer Architectures Group, University of York, UK. http://www.cs.york.ac.uk/arch/NeuralNetworks/AURA/aura.html.

[11] SDSC Storage Request Broker [Online]. Available: http://www.sdsc.edu/srb/

[12] David A. Chappell. "Enterprise Service Bus – Theory in Practice". O'Reilly. ISBN 0-596-00675-6.

[13] M. Ong, X. Ren, G. Allan, V. Kadirkamanathan, H. A. Thompson, P. J. Fleming (2004). "Decision support system on the Grid". Proc Int'l Conference on Knowledge-Based Intelligent Information & Engineering Systems, KES 2004.

[14] Martyn Fletcher, Tom Jackson, Mark Jessop, Bojian Liang, and Jim Austin. "The Signal Data Explorer: A High Performance Grid based Signal Search Tool for use in Distributed Diagnostic Applications." CCGrid 2006 – 6th IEEE International Symposium on Cluster Computing and the Grid. 16-19, May 2006, Singapore.

Q&A – Tom Jackson

Questioner: Dennis Gannon

Can the system continuously monitor the data searching for faults and alerting users of possible problems? Is this software accessible? Is it open source?

Tom Jackson

Yes, the system has workflow scripts that automatically scan the data for faults as it arrives at the data repository. If faults are found an alert is sent to a diagnostic engineer.

The PMC middleware stack has been developed within the UK e-Science programme and is open source.

Questioner: Bill Applebe

(Remark: this is very useful for clinical DB's that cannot be copied – only subset queries legal) How is the client side implemented?

Tom Jackson

There is a standardized web-service (SOAP) client side API that can be easily integrated into user application software.

Questioner: Anne Trefethen

1. *How does the evolving OGSA standard affect the infrastructure you are using?*
2. *Security is a major issue for industry use of "grid" – have you solved these issues?*

Tom Jackson

Answer (1): We restrict ourselves to web-service standards for the current implementation, to avoid issues with evolving OGSA standards.

Answer(2): We have addressed them within the context of security measures within the available GT4 toolkits, in that the services can ask for authentication and authorization via X509 certificates.

Questioner: Brian Smith

How do you handle false pattern matches? Are they frequent and how do they impact the acceptance of your approach?

Tom Jackson

False positives need to be avoided as far as possible; they can prove costly within the maintenance process. We are still validating the algorithms to determine how serious or frequent this may be. There is still a relative lack of fault data as the systems do not go live on commercial aircraft until late 2006.

Tuesday AM Panel Discussion

Panel

- Boyana Norris
- Jem Treadwell
- Tom Jackson

Questioner: Brian Smith

Quality of Service (QoS) typically has metrics associated with it. Also interactions back to the requestor after the analyzer discovers what is possible. Do you see these metrics being developed and interaction being supported?

Tom Jackson

Yes, we are working on intelligent QoS systems that permit analysis of operational conditions and dynamic adaptation to fit, where possible, the constraints of the user QoS profile.

Questioner: Brian Smith

Are other areas of aircraft monitoring possible and being investigated?

Tom Jackson

Yes, we have had some talks with other airline company's about the use of the technology for airframe monitoring, as one example.

Questioner: Anne Trefethen

Does the OGSA take quality of service into account?

Tom Jackson

There is a lot of activity within GGF addressing the issues relating to QoS, and it is a high priority issue.

Boyana Norris

In the case of CQoS, continuous close interaction with an ever-expanding number of application domains is a priority and directly affects the design and implementation of the interfaces and middleware component infrastructure.

Jem Treadwell

Yes, QoS is addressed in the Self-Management capability, and the GGF's GRAAP working group is developing the WS-Agreement specification, which can be used to establish service-level agreements. Note that there is no activity on self-management within OGSA at the moment, as we focus on the more basic capabilities.

Questioner: Anne Trefethen

Are we convinced that service architectures of the sort being discussed are going to deliver across a range of applications?

Tom Jackson

The major IT companies, such as IBM and Sun, continue to back the SOA model as the basis for systems integration and deployment. The model is becoming increasingly mature and accepted across a broad range of disciplines. Emerging technologies such as Enterprise Service Bus (ESB) make the SOA approach even more valid as they enhance the integration process.

Jem Treadwell

We (HP) see service-oriented architecture being used by customer and partner organizations in a wide range of markets, both for new applications and in making legacy applications more easily available. As an architectural principle it just makes sense, and its use is not restricted to applications based on Web services.

Questioner: Anne Trefethen

Are the developments of these underlying service architectures sufficiently connected to the applications?

Boyana Norris

As I mentioned in the answer to the previous question, the CQoS effort is considering a broad range of applications in all the stages of the design and implementation of the interfaces and middleware component infrastructure.

Questioner: Brian Ford

As you prepare to commercialize this system, are there particular problems rooted in its grid nature that make the activity more difficult, or even not possible?

Tom Jackson

The system is currently being evaluated for commercial deployment. The academic partners have spin out companies that provide a commercial route for deployment of the software. The fact that it is Grid based does not appear to pose any problems with commercialization at this stage.

Questioner: Brian Ford

How do you establish the effectiveness of this technical approach in commercial environments? How do you establish the value to concerns such as the national health service, oil industry etc? The value-for-money for the organization is not always clear.

Tom Jackson

This is the basis for the current DTI BROADEN project. The project concepts were developed under UK e-Science programme, DAME, but the commercial scalability and deployment issues are being addressed within BROADEN in partnership with Rolls-Royce IT providers, EDS.

Questioner: Mary Thomas

What is the status/progress of speeding up security (GSI) protocols?

Jem Treadwell

While there are a number of groups working on security within GGF (now OGF), I'm not sufficiently involved to comment on what they're doing, or their progress. However, I think this question relates to Globus rather than OGSA or GGF, and we need to be clear that Globus and OGSA are not the same thing: the Globus Toolkit is a grid middleware implementation. [Thanks to Keith Jackson, who supplied a response on behalf of the Globus development team.]