This is the Author Accepted version of the paper Candela, L., Akal, F., Avancini, H. et al. DILIGENT: integrating digital library and Grid technologies for a new Earth observation research infrastructure. Int J Digit Libr 7, 59–80 (2007). https://doi.org/10.1007/s00799-007-0023-8

Leonardo Candela · Fuat Akal · Henri Avancini · Donatella Castelli · Luigi Fusco · Veronica Guidetti · Christoph Langguth · Andrea Manzi · Pasquale Pagano · Heiko Schuldt · Manuele Simi · Michael Springmann · Laura Voicu

DILIGENT: Integrating Digital Library and Grid Technologies for a new Earth Observation Research Infrastructure

Abstract This paper introduces Diligent, a digital library infrastructure built by integrating digital library and Grid technologies and resources. This infrastructure allows different communities to dynamically build specialised digital libraries capable to support the entire e-Science knowledge production and consumption life-cycle by using shared computing, storage, content, and application resources. The paper presents some of the main software services that implement the Diligent system. Moreover, it exemplifies the provided features by presenting how the Diligent infrastructure is being exploited in supporting the activity of user communities working in the Earth Science Environmental sector.

Keywords Digital Libraries · Grid · Earth Science

1 Introduction

The way in which what is commonly named "Digital Libraries" (DLs) is perceived has evolved a lot over the last fifteen years. This is also true for the expectations on Digital Libraries. DLs are now moving far beyond any connotation of the term "library" and are rapidly evolving to become general systems dedicated to cover the whole spectrum of the knowledge management task [25, 26]. DLs are now envisioned as systems that are at the centre of any intellectual activity and have no logical, conceptual, physical, temporal, or personal barriers on information [1]. In particular, they are shifting from *content-centric* systems in charge of simply organising and providing access to particular collections of data to *person-centric* systems aiming at providing facilities for communication, collaboration and any kind of interaction among scientists, researchers, and the general audience interested in topics of pertinence to the knowledge the DL has been set up for.

This vision poses new requirements on DL systems. They must deal with many forms of data ranging from digital counterparts of traditional documents, such as PDF files, to complex and multimedia objects combining text, images, audio and/or videos, sensor data, structured and semi-structured data residing in various kinds of information sources. They must act as an integrated working environment providing a seamless and personalised access to a variety of information sources as well as to all the facilities and instruments deemed as relevant for fulfilling the user requirements. These facilities may range from "standard" Digital Library facilities, like search and browse, to co-operative working tools and community specific services that require high computing and storage capabilities. The new DL systems must also be able to easily adapt themselves to the evolving user requirements both in terms of the data the user has access to and the processes the user is entitled to perform, i.e., they must be able to appropriately combine data and services in user specific processes.

Diligent is a DL infrastructure that aims at supporting the realisation of such new DLs. It integrates Digital Library and Grid technologies in a framework where the implementation of the services demanded from the new DL systems becomes possible. Via this infrastructure, members of *dynamic virtual e-Science organ-*

This work is partially funded by the European Commission in the context of the Diligent project, under the 2nd call of FP6 IST priority.

Henri Avancini · Leonardo Candela · Donatella Castelli · Andrea Manzi · Pasquale Pagano · Manuele Simi
Institute of Information Science and Technologies (ISTI) – Italian National Research Council (CNR), Via G. Moruzzi, 1 – 56124 Pisa – Italy E-mail: {avancini, candela, castelli, manzi, pagano, simi}@isti.cnr.it

Fuat Akal · Christoph Langguth · Heiko Schuldt · Michael Springmann · Laura Voicu
University of Basel - Department of Computer Science, Bernoullistrasse 16 – 4056 Basel – Switzerland E-mail: {fuat.akal, christoph.langguth, heiko.schuldt, michael.springmann, laura.voicu}@unibas.ch

Luigi Fusco · Veronica Guidetti
European Space Agency (ESA) – European Space Research Institute (ESRIN), Via G. Galilei, 64 – 00044 Frascati (RM) – Italy E-mail: {luigi.fusco, veronica.guidetti}@esa.int

isations[1] have access to shared knowledge and are enabled to collaborate in a secure, co-ordinated, dynamic and cost effective way. In particular, these organisations are supported in creating their own DLs by selecting collections and services or, even at a lower level, computing and storage resources from a pool of shared resources and by aggregating them.

This paper introduces DILIGENT and exemplifies its use as a powerful instrument for supporting typical activities of the Earth Observation domain. In particular, it focusses on some of the basic services that mostly contribute to the realisation of the framework required by this specific application domain; namely those services that manage the DLs as aggregation of DILIGENT native and domain specific content and application resources, and those services that support the generation and the storage of the large products created in this domain. As it will be shown in the rest of the paper, these services, together with the specific tools already used by the Earth Observation community, provide a new DL framework that offers a large number of new possibilities for improving the information for these communities and for enhancing the quality of their work.

The remainder of this paper is organised as follows: section 2 illustrates emerging requirements related to the use of DLs within the Earth Observation community. Section 3 introduces DILIGENT by focusing on its architecture and on some of its key services responsible for delivering *virtual* Digital Libraries. Section 4 presents the DILIGENT infrastructure that has been set up to support the need for a concrete Earth Observation application scenario. Section 5 surveys related work. Finally, Section 6 concludes.

2 DL in the Earth Observation Sector

Novel digital libraries can play a very important role in the emerging Earth Observation (EO) sector. Concrete examples of possible usages come from the European Space Agency which is nowadays facing new requirements and experimenting new solutions in terms of content and application management.

New DLs can indeed better support typical EO activities by providing facilities for storing, managing and accessing multi-type information, for making community specific applications and high computing-storage capabilities available, and for responding to proper on-demand person-centric aggregation and interoperability of data.

The building of *periodical environmental reports*, for example, is a typical EO activity where new DLs can play a very important role. These complex information objects, mostly built as aggregation of other information objects, require a lot of existing information, usually coming from different heterogeneous sources. This information has to be properly discovered and uniformly accessed. The so collected information has often to be coherently integrated with pertinent information generated on-demand through procedures that often need to access and process huge amount of data. These environmental reports have a relevant social role, therefore they have to be constantly maintained up to date (see Figure 1). This is why in the rest of this paper we will refer to them as *living documents*.

The updating of these environmental reports may, for example, concern the re-generation of the summary maps used to show the distribution of environmental features and characteristics, such as chlorophyll or vegetation indexes. To display the most recent information the maps need to be re-built at each report versioning. Re-building maps may involve a lot of expertise from the human-side and also a lot of facilities from the underlying document management system, i.e., it must be possible to instantiate and run ad-hoc procedures and algorithms dealing with whatever amount of data locally available or either dispersed among different sites. To this end EO experts need facilities for building ad-hoc compound services that generate, manipulate and modify data. This requires to manage all the available resources (data sources, computational resources, etc.), to manage their current status in order that all these resources be used as effectively and efficiently as possible.

Another common activity in the EO domain is the assessment and planning of responses to environmental accidents. The handling of this activity requires the management of complex information that is often generated through specific user-defined processes. The following is a typical example of a process concerned with the acquisition, analysis, and creation of information related to the analysis of the environmental status: *A user acquiring first radar imagery of oil spills decides to retrieve a complementary optical imagery and overlay it to the initial one. This user then to overlays the resulting imagery with tracks of major tanker routes to highlight any correlation and checks it against a WWF coastal map showing coastlines of maximum biodiversity or a mosaic of Mediterranean chlorophyll distribution. Finally, s/he applies wave and wind meteorological information layers to model the behaviour and impact of these spills, either retrospectively or in near-real time.*

Such a complex data access and fusion procedure can be requested to routinely run, depending on the availability of specific computing and storage capabilities, in order that always updated information is available.

From a technical point of view, supporting cases like the above one requires dedicated processes which, once defined, are repeatedly invoked to update the information they provide. In order to support business processes the DL must provide access to all the different heteroge-

[1] Dynamic groups of individuals, institutions and resources, possibly remotely distributed, which are virtually grouped together to achieve a common goal.

neous types of the mentioned information objects, enable the execution and orchestration of the complex fusion and simulation procedures, and provide the storage for the (possibly very large) intermediate results. Moreover, to promote the optimal usage of the resources constituting the infrastructure, the DL must be capable to acquire the optimal quantity of available resources on-demand without reserving them in advance, e.g., deploy a new service instance when the load of the existing ones overcome a certain threshold, undeploy a service instance when this is underused.

Therefore key features required by DLs serving the EO domain are:

- management of very large and distributed virtual organisations;
- seamless access and handling of distributed and heterogeneous data and services;
- on-demand processing of huge amounts of information, thereby making efficient use of the available resources;
- storage of derived data as well as of dependencies between documents and/or data items;
- support for the definition of workflow processes and their scalable and reliable execution;
- traceability of operations performed.

3 The DILIGENT Solution

DILIGENT [12] is an ongoing EU-IST funded project that combines Grid [16–19] and Digital Library [3,20,21,25] technologies in order to provide an advanced test-bed infrastructure for supporting the creation and maintenance of multiple, dynamic DLs activated on the same set of shared resources. DILIGENT is designed as a Service Oriented Architecture [37,36]. This means that both DL-related and internal system functionality is provided by independent services which can be individually deployed, easily and seamlessly combined, etc. As for the Grid application framework, the DILIGENT system exploits both the gLite [15] and the WSRF frameworks [5]. In addition to the project infrastructure, DILIGENT is able to exploit the computing and storage resources made available by the Enabling Grid for E-sciencE (EGEE), the largest European Grid Infrastructure ever been built, which has been set up by the EU funded EGEE project [14]. The access to this physical infrastructure makes DILIGENT capable to exploit an enormous potential in terms of storage and computational resources that can be used in delivering the innovative user-centric DL community-specific services.

This section first briefly introduces the DILIGENT system from both the functional and architectural points of view. Then it presents in more detail those services that mostly contribute to the satisfaction of the EO requirements illustrated in the previous section.

3.1 Overview

From an abstract point of view, a DILIGENT infrastructure acts as a *DL broker*, i.e., a *service* acting as a bridge between DL demands and available resources. The clients of the broker are providers and consumers of DL resources. The providers are the individuals and the organisations that decide to make their resources available, according to certain access and use policies, under the supervision of the infrastructure. The consumers are the user communities that want to build their own DLs. By using the instruments provided by these DLs, the consumers can also produce new content that can, in turn, be made available to other consumers through the DILIGENT infrastructure itself. The same individual can thus act as a consumer and a producer with respect to the infrastructure.

The broker manages resources of different types: *collections* (i.e., sets of information objects searchable and accessible through a single "access point"), *services* (i.e., software tools implementing a specific functionality and whose descriptions, interfaces and bindings are publicly available), *hosting nodes* (i.e., networked entities that offer computing and storage capabilities and supply environments for hosting collections and services), and *EGEE resources* (i.e., computing elements and storage elements).

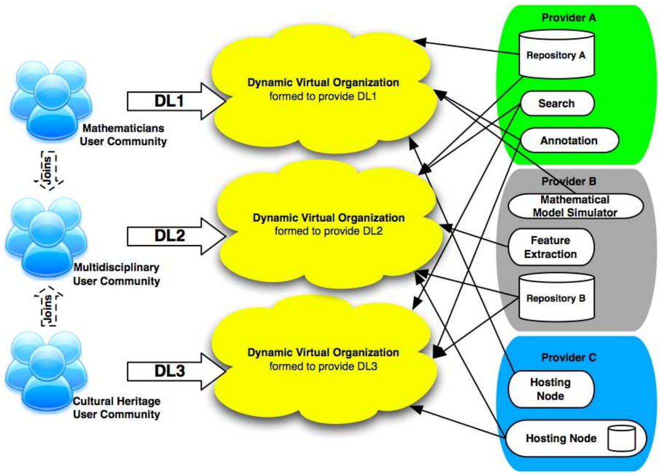

Fig. 2 The DILIGENT Brokerage Scenario

In order to support the controlled sharing of resources among providers and consumers, the DILIGENT infrastructure relies on a mechanism originally introduced in the Grid research area [19] to create *virtual organisations* (VOs). This mechanism models sets of users and resources defining clearly and carefully what is shared, who is allowed to share, and the conditions under which sharing occurs, usually based on an authentication framework. VOs have a limited lifetime and they are dynamically created to satisfy transient specific needs by allo-

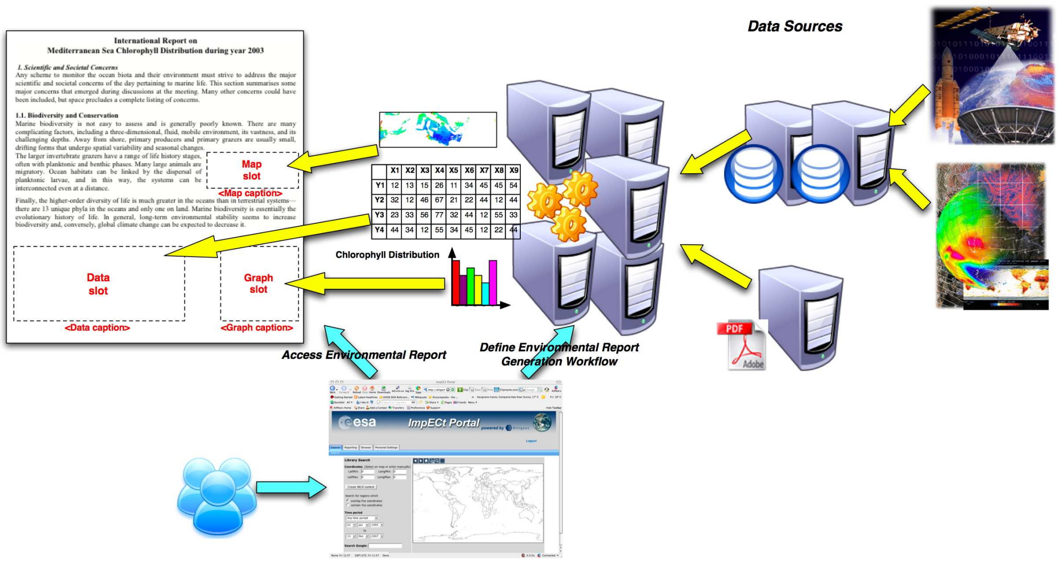

Fig. 1 Periodical Environmental Report Generation

cating and providing resources on-demand. Through the VO mechanism, the DILIGENT infrastructure glues together the users and the resources of a DL (see Figure 2).

The infrastructure offers appropriate tools that enable providers to register and describe their resources. According to which type of resources have been submitted, the infrastructure automatically extracts additional properties that are used to enrich the explicit description.

The infrastructure takes care of the management of the registered resources by supporting their discovery, monitoring, reservation, and by implementing the functionality to provide the required controlled sharing and quality of service.

A user community can create one or more DLs by specifying for each of them the set of the required features. These requirements specify conditions on: (*i*) the information space (e.g., the set of collections to be included or to be harvested, the subject of the required content, the type of the information objects to be handled), (*ii*) the services for supporting the work of the users (e.g., search and browse type), (*iii*) the quality of service (e.g., availability, performance, security), and (*iv*) many other aspects, like the maximum cost, lifetime, etc. The DL broker satisfies those requirements by selecting, and in many cases also deploying, a number of resources among those accessible to the community, gluing them appropriately to build a DL Application and, finally, making this DL accessible through a portal. The composition of a DL is dynamic since the infrastructure continuously monitors the status of the DL resources and, if necessary, changes them in order to offer the best quality of service. By relying on the shared resources many DLs, serving different communities, can be created and modified *on-the-fly*, without considerable investments by and changes in the organisations that set them up.

The DILIGENT system provides the necessary functionality to support the DL brokerage behaviour previously described. In addition, it implements a set of configurable services providing native DL functionality such as information domain mediation, content management, annotation, search, etc., that form the core of any DL. The next section introduces the different functional areas that provide the DILIGENT expected functionality and presents the overall system architecture.

3.2 The DILIGENT Architecture

The logical architecture of the DILIGENT system is depicted in Figure 3. It is based on a service-oriented architecture (SOA) [37,36].

The DILIGENT services rely on an *application framework*, i.e., a set of software libraries and subsystems supporting the operations of the system components. This framework consists of (*i*) the *gLite* Grid middleware [15] which supports access to the EGEE resources, and (*ii*) the WSRF specification [5] implementation released by the Globus project [24]. This application framework enables DILIGENT services to act as Grid Services and access shared resources via the inherent mechanisms of the Grid. The resources are represented by the computing elements and storage elements provided by the EGEE project and by the Grid services provided by DILIGENT itself.

From a conceptual point of view, the services implementing DILIGENT functionality are divided in three

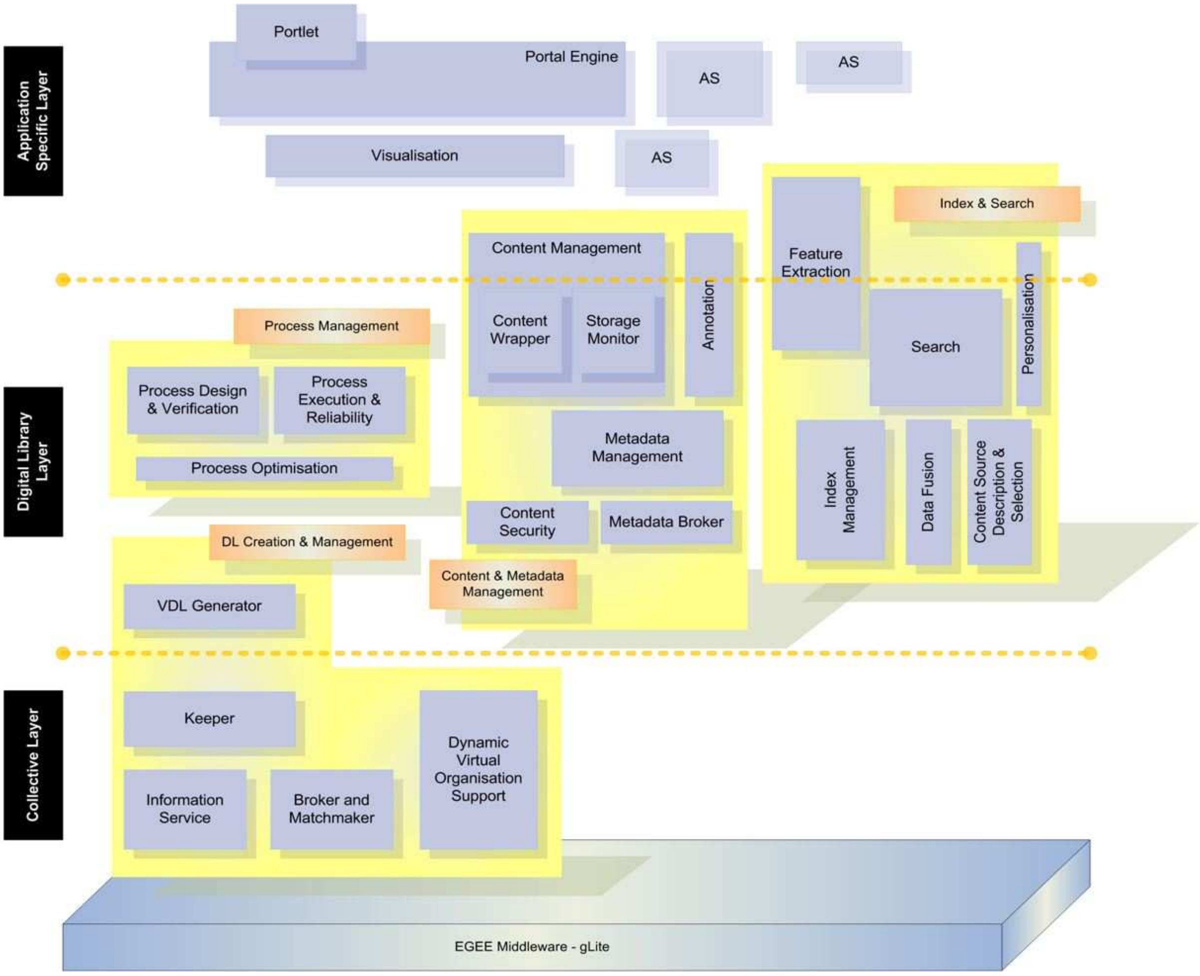

Fig. 3 The DILIGENT Logical Architecture

main layers which sit on top of the underlying EGEE gLite middleware.

The top-most *Application-specific* layer encompasses all the tools and interfaces needed by users to interact with the DL in a particular application domain. The keystone component of this layer is represented by the *Portal Engine*. The user interfaces of each functionality are additional components that can be added to the portal engine instances. Thus by appropriately instantiating and configuring the portal engine, i.e., deciding which user interface components have to be hosted by the engine, the DL user interface is customised with the tools needed to operate and perform the tasks the DL has been set up for. This layer also includes a set of application specific (AS) components, i.e., components implementing the user interfaces to application specific services. This set of components will grow in number during the DILIGENT lifetime because of the brokerage model and of the publishing of new services capable to fulfil specific needs. From a technological point of view, DILIGENT uses an open-source portlet-hosting engine (GridSphere) that is capable to host DILIGENT service user interfaces thanks to its relying on JSR168[2], JavaServer Faces[3] and WSRP[4] standards.

The *Digital Library* layer provides services delivering all the functionality which is independent of any particular DL and/or application domain. Firstly, this layer includes functionality for storing objects and DL content (*Content & Metadata Management*). Secondly, it provides access to DL content (*Index & Search*). Thirdly, it allows for the definition and execution of processes and workflows as they are needed for complex DL applications, e.g., the generation and updating of living documents (*Process Management*). Hereafter a description of the main services implementing the described functionality is provided. *Content Management* and *Metadata Management* provide, respectively, all the functionality required for storing information objects and their associated metadata. Moreover, since DL objects can be organised into collections, the definition and management

[2] http://www.jcp.org/en/jsr/detail?id=168
[3] http://java.sun.com/javaee/javaserverfaces/
[4] http://www.oasis-open.org/committees/download.php/10539/wsrp-primer-1.0.html

of these collections is also supported by the Content Management. The Content Management service is also equipped with replication management facilities. These facilities automatically create and distribute replicas of DL information objects, thereby significantly increasing their availability. *Content Security*, finally, takes care of applying watermark and encryption techniques in order to protect the DL information objects from unauthorised accesses. The latter aspect is particularly relevant since the storage facilities provided by the Content Management service make use of Grid storage facilities, therefore it might occurs that physical files are stored on third party devices that are not under the direct control of the DILIGENT infrastructure. More details on storage and content management are provided in Section 3.4. The access to DL content (Index & Search in Figure 3) is orchestrated by the *Search* service that exploits the capabilities provided by (*i*) the *Index Management*, i.e., the component in charge of building and maintaining indexes of various types on the information domain, e.g., full-text and forward indexes; (*ii*) the *Feature Extraction* that collects a number of feature extraction components specialised for different kinds of media; (*iii*) the *Content Source Description & Selection*, i.e., the component supporting the discovery of the collections where to search in for a given cross collection query. The collections are selected on the basis of the similarities between their content description and the query characteristics; (*iv*) the *Data Fusion*, i.e., the component in charge of merging the result sets coming from querying the involved collections; and (*v*) the *Personalization*, i.e., a service in charge of customising the search results according to the user characteristics. These services are designed to co-operate by relying on the *Process Management* facilities. In particular, the timely reply to a complex search requires that a search service produces an execution plan involving all the needed services modelled by means of a workflow which will be executed in a Peer-to-Peer fashion by the Process Execution service. In short, process[5] management allows for the combination of existing services into more complex workflows which, once defined, can be repeatedly invoked (as occurs, for instance, in updating living documents). More details on process management are given in Section 3.5. In addition to the above mentioned services and functionality, it is important to highlight that the Digital Library layer contains the *VDL Generator*, i.e., the service providing DILIGENT user communities with the facilities for defining the DLs they are interested in. As described in Section 3.1, the DL definition thus obtained is a high level description expressed in terms of the expected DL functionality and content. The VDL Generator, by co-operating with the Collective Layer services, is in charge of identifying the pool of resources needed to fulfil the expressed requirements. The list of the needed resources will then be communicated to the Collective Layer services, namely the Keeper, that will physically aggregate and deploy these resources to form the DL.

Finally, the *Collective Layer* encompasses all the functionality that is needed for delivering the DL as a service oriented application, namely management of all resources and book-keeping of their state. In order to deliver and operate DLs implemented according to the previously described brokerage scenario, the following services, in addition to the VDL Generator, have been envisaged. The *Information Service* (IS) is the service playing the role of a registry supporting resource discovery and monitoring. The *Keeper* and the *Broker and Matchmaker* (BMM) are the services in charge of the creation and maintenance of the DLs. In particular, the former service receives the list of resources needed to implement the expected DL and by interacting with the IS identifies which resources, among those currently available, can be reused and which have to be deployed. For this second set of resources it identifies, by interacting with the BMM, the pool of available hosting nodes and deploys the needed software packages on them. The Keeper co-operates with the Dynamic Virtual Organisation Support to create the trusted environment needed to appropriately operate the pool of resources constituting the DL, i.e., it creates a VO containing the DL resources and defines the rules regulating their usage. The Keeper is also in charge of the maintenance of the resources forming the DL, i.e., it monitors their status by interacting with the IS and takes the appropriate actions (e.g., deploys a new resource instance and adds it to those accessible by the DL) to ensure that all the needed resources are available. Finally, the *Dynamic Virtual Organisation Support* (DVOS) is in charge of managing users and groups as well as authentication and authorisation issues. The latter two aspects are particularly relevant with respect to the brokering scenario previously envisaged. In fact, resource providers will make their resources available to the DILIGENT infrastructure if and only if these resources will be used according to the specified usage policies. To support this scenario, the DVOS completely relies on the Grid VO mechanism [19].

In the following sections we focus on those services that are foundational to operate the DLs envisaged for EO, as described in Section 2. Section 3.3 introduces the service that provides the complete list of the resources available for building and maintaining the EO DLs plus the service in charge of physically deploying and maintaining each of these DLs. Section 3.4 describes the technologies implemented by DILIGENT for managing the information objects. These technologies provide seamless access to and management facilities of the heterogeneous content used in the EO DLs. Finally, Section 3.5 introduces the DILIGENT services dedicated to the design and execution of the compound services. These services provide the EO community with a mechanism

[5] The terms “process”, “workflow” and “compound service” (CS) will be used as synonyms throughout the paper where no confusion arises.

for combining the services and data sources the DL is equipped with in unpredictable workflows implementing user defined processes.

3.3 Resource Management

In a distributed infrastructure where the applications result from the aggregation and composition of services, a set of facilities dedicated to the discovery of services and, in general, to their management is needed. This need is even more urgent in DILIGENT where the presence of the Grid promotes a dynamic development process – hence the dynamic deployment of novel resources. In what follows we detail the DILIGENT services dedicated to fulfil such a need, i.e., the *Information Service* and the *Keeper*.

3.3.1 The Information Service

The Information Service (IS) is the service in charge of supporting the discovery and monitoring of the distributed resources forming the infrastructure. An ever *updated* picture of the whole set of available resources is maintained; by relying on such a picture, single services and DLs can implement self-tuning resource usage and workload balancing algorithms aimed at maximising the use of available resources. In particular, the functionality offered by the IS service includes:

- gathering, storing, and supplying information about the resources and services constituting the infrastructure;
- monitoring these resources and services ensuring the appropriate level of freshness in order to provide reliable information about the current status of the infrastructure.

For instance, in the case of the EO example given in Section 2, the DILIGENT IS must provide the other services belonging to the infrastructure with the complete list of collections and services the community makes available to DILIGENT. By relying on such information the services can expose an interface where a user can select the appropriate data sources and services that once combined in a workflow produce the environmental report satisfying its application needs. Moreover, by providing updated information about their status, the IS supports other services in discovering the most appropriate service instance to use in order to perform a certain task. The service executing a workflow dedicated to create a living document can, for instance, discover and use the less loaded service instances and thus produce the document by using the available resources in an optimal way.

The Information Service gathers and supplies information following an approach inspired by the well-known Grid Monitoring Architecture (GMA) [48] proposed by GGF[6]. The GMA models an information infrastructure as composed by a set of *producers* (that provide information), *consumers* (that request for information) and *registries* or collectors (that mediate the communication between producers and consumers).

In the DILIGENT scenario, the GMA architecture has been implemented as depicted in Figure 4. In particular, the design goals were to have high scalability, availability, and reliability because of (*i*) the potential huge dimension in terms of node, services and resources forming the infrastructure, (*ii*) the high dynamism of the infrastructure where new resources are added and removed continuously, and (*iii*) the central role of the IS in operating the infrastructure. In this scenario, producers and consumers are supported in interacting with the IS via a lightweight component distributed on each hosting node of the infrastructure, called *IS-Client*. This component provides three main kinds of functionality supporting: (*i*) the publication of the information (*IS-IP library*); (*ii*) the information access and discovery via query and subscription/notification mechanisms (*IS-C*); and (*iii*) the local storage and maintenance of useful and constantly updated information (*IS-Cache*). The IS-Client implements an efficient access to the information in the distributed infrastructure and supports its publication while hiding any detail about the routing process that identifies the appropriate collectors where such information is to be published in/retrieved from. Collectors aggregate the produced information. They are distinguished in two types of components, *IS-Registry* and *IS-IC*. The former act as classic registries and are in charge of maintaining the list of available services and their static information while the latter maintain all the dynamic information and are based on a highly distributed architecture.

From an operational point of view it is important to note that each time a service of the infrastructure is deployed, it is registered on the IS-Registry and then it starts producing its dynamic information via the local IS-IP library. In parallel, the IS-Cache takes care of maintaining the set of minimal information needed by the locally hosted services both for publishing and for querying. The IS-Registry is in charge of maintaining the "picture" of the whole infrastructure in line with the actual status by continuously monitoring the service instances.

From a technical point of view the Information Service relies on the WS-* standards and specifications, namely the WSRF framework, WS-Addressing, WS-Security, and WS-Notification. In particular, it exploits the implementations of these specifications provided by the Globus project of which the Aggregator Framework[7] is the foundational software framework the IS is based on. To prove the appropriateness of the solution proposed w.r.t. the design goals, the performance measures observed during the first months of IS operation are provided hereafter. The average publishing rate of informa-

[6] Global Grid Forum, `http://www.ggf.org`

[7] `http://www.globus.org/toolkit/docs/4.0/info/aggregator/`

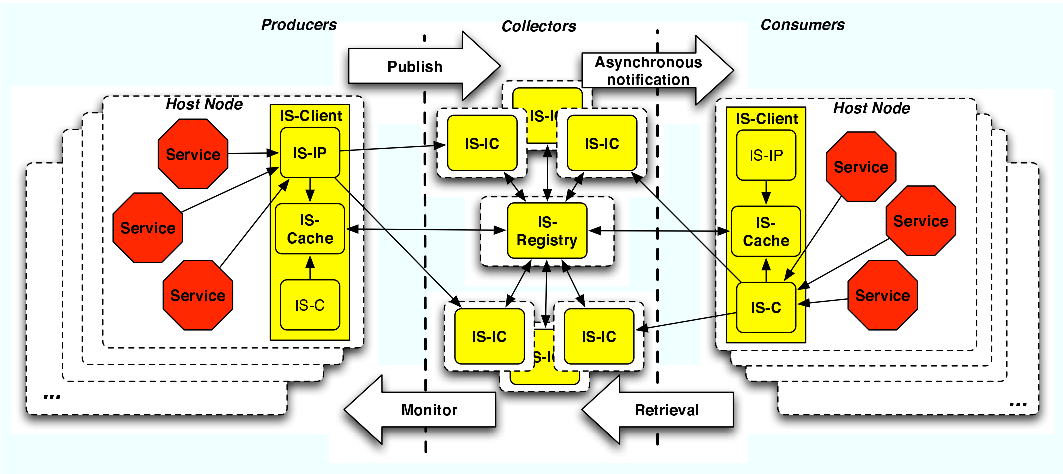

Fig. 4 The Information Service Logical Architecture

tion is about 198,000 profiles and about 96,000 resource property documents per day while the retrieval rate is about 6800 IS entries per day[8].

3.3.2 The Keeper Service

In the DILIGENT context the Keeper plays the role of the DL manager. It is the service in charge of creating the resources constituting the virtual DL and authorising the users to use them via the creation of an appropriate virtual organisation. In order to provide this facility, the Keeper relies on the concepts of *hosting node* and *software package*. The former is a special DILIGENT resource capable of hosting the services. The latter is a collection of software bundles the Keeper is aware of and that, once deployed, provides the functionality of a service. In the EO application described in Section 2, for instance, a software package is the bunch of software that implements the service providing the map overlay functionality while a hosting node is a server where this software can be deployed.

The logical architecture of the Keeper is depicted in Figure 5. It consists of three main components: the *DL Management*, the *Hosting Node Manager* (HNM), and the *Packages Repository*.

The DL Management is in charge of identifying the set of software packages needed to implement the DL it has to manage and the set of hosting nodes where these software components are to be deployed. By interacting with the HNM of these nodes the DL Management directs the deployment. In addition, it co-ordinates and disseminates the *operational context* that transforms this set of distributed resources into a single application. In the DILIGENT terminology, this context is named *DL Map*. This map specifies the DL resource locations and their configurations. Any other dynamic information about a resource (e.g., its status) is maintained and disseminated by the Information Service.

Once the DL is up and running, the Keeper is also in charge of guaranteeing the quality of the overall set of DL functionality at any time by dynamically re-allocating resources and checking periodically their status. In order to support this functionality it accesses and investigates the state of services and resources and un-deploys and/or relocates them in an appropriate way using the information disseminated by the Information Service.

The HNM is the minimal mandatory software that must be installed on each hosting node to support dynamic deployment. Node management involves the following tasks:

- collaborate with the DL Management in order to deploy new services;
- publish the hosting node configuration and status in the Information Service;
- exchange data with the DL Management;
- maintain and expose the hosting node configuration to the hosted services.

Finally, the Packages Repository is the component in charge of storing the software packages and of making them accessible to the HNMs when they need to deploy one of them.

From a technical point of view the resources this service deals with are Web Service instances and related software components. All the software that is dynamically instantiated must be registered in the DILIGENT infrastructure using the provided functionality and must be compliant with the *package model specification*. If a "piece of software" respects the rules of this specification, it can be (*i*) uploaded in the Packages Repository, (*ii*) handled – i.e., deployed and un-deployed – by the HNM,

[8] This data refers to October 2006. Moreover, it is important to highlight that these numbers refer to standard operation mode of the service; comprehensive stress tests are currently ongoing in order to identify the service capacity and the results of these tests are not presented here because they are out of the paper's scope.

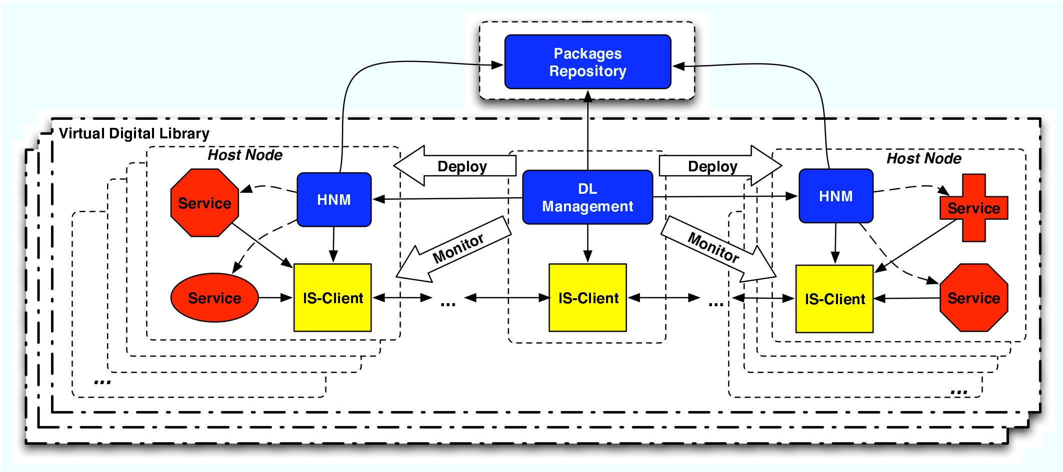

Fig. 5 The Keeper Service Logical Architecture

and (*iii*) dynamically discovered and used by other services. Moreover, since DILIGENT is designed and built according to the SOA paradigm, it is composed by a set of services that can be invoked remotely, i.e., that are accessible via a network interface. A *hosting environment* is needed in order to allow services to operate. In DILIGENT the hosting environment is the Java WS Core, developed by the Globus Project. This service container provides a complete implementation of the WSRF [5] and WS-Notification specifications plus WS-Addressing and WS-Security support based on the Axis Web Services engine developed by the Apache Foundation. Java WS Core also provides a framework that supports both authentication and authorisation mechanisms. In particular, the authorisation features can be extended by each service in order to provide a customised level of authorisation policies.

3.4 Content Management

The Content Management in DILIGENT provides means for persistently storing, physically structuring, and efficiently fetching any kind of content on any Grid enabled Digital Library. Current Grid technology focuses mostly on performance for batch processing. On the other hand, an important challenge with regard to Digital Libraries is that a DL requires also real-time performance especially while searching and browsing content. Therefore, requirements for managing the content of a Digital Library based on the Grid go beyond file-system-like functionality as natively provided by the gLite middleware. Particularly, certain content must be distinguished, content must be interrelated in multiple ways, and content must be described with various application specific properties.

Thus the DILIGENT document model must be suitable for fulfilling the requirements arising in different application scenario as well as take advantage by available storage facilities. The consequence of these requirements is the necessity of Content Management facilities capable to deal with generic *information objects*, i.e., units of information that can be stored or fetched independently of what they actually represent. The Content Management in DILIGENT considers all entities in a DL, e.g., collection, content, metadata, etc. as information objects. For instance, metadata about a content is also an information object which is associated with another information object that represents that content. As a consequence, there is no restriction on the kind of "document" the DILIGENT Content Management can manage, e.g., complex and structured classes of objects can be built via the information object linking mechanism, all metadata format can be stored, metadata and data can be combined and mixed in various ways thanks to the linking mechanism among information objects. The modelling of documents in terms of information objects is an application specific problem. For instance, one could easily build a sample document for the ESA scenario as illustrated in Figure 6. The satellite image in the figure represents the actual content. This content is bound to some metadata which is in form of an XML document. Furthermore, some features of that image, e.g., colour features, can be extracted and associated with that content as another type of metadata. These automatically extracted features can be used for content-based search (e.g., looking for similar images) within collections distributed across the Grid.

To provide this functionality, data management is handled in a layered architecture presented such as that presented in Figure 7.

There are three main layers: (*i*) Content Management, (*ii*) Storage Management and (*iii*) Base Layer.

The Content Management Layer provides the highest level of abstraction and the appropriate interfaces for the tasks of storing, retrieving, and organising information. It consists of four services. The *Collection Management service* (ColM) is used to manage collections of

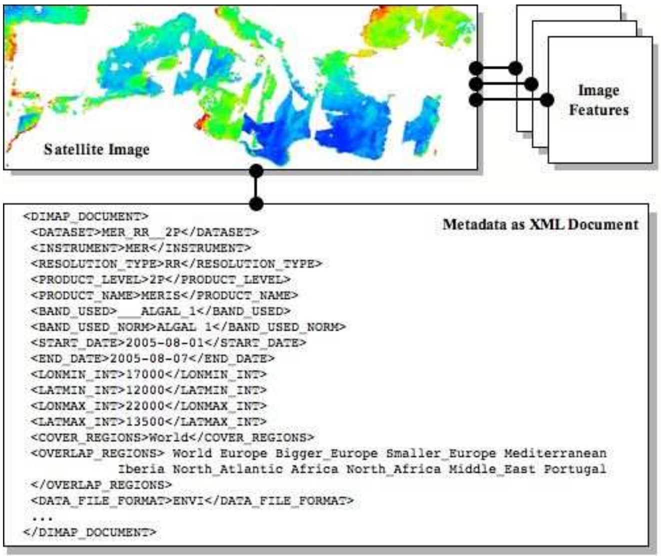

Fig. 6 Document Metadata Relationship

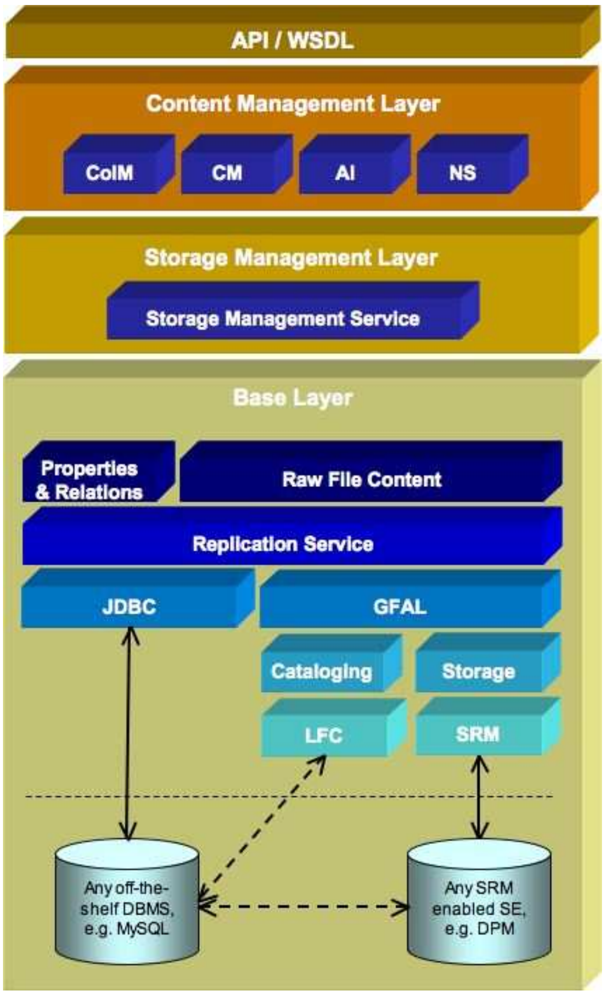

Fig. 7 Data Management Layers

documents in a Digital Library, e.g., the ESA collections. The *Content Management service* (CM) provides high-level operations on documents which are mapped onto generic Storage Management operations. The *Archive Import service* (AI) provides the required operations for accessing and importing available archives[9] which are, for instance, very important for making the ESA collection available via DILIGENT. The *Notification service* (NS) observes any structural and content changes in an information object and triggers the planned task; for instance, when a new satellite image in the ESA scenario is detected a process to create its metadata and update the necessary indexes is triggered. The Content Management Layer therefore provides services to handle semantics like the relationships between documents and collections and documents and metadata.

The Storage Management Layer provides functionality to store information objects. The semantically rich representation supported by the Content Management Layer is mapped to and handled by a common interface for maintaining relationships, properties, and binary content of information objects.

The Base Layer encapsulates the low level details of implementing physical storage and file transfer. It is only available to the Storage Management Service and has to provide the access to all resources of the Grid that are required to store relationships and properties of information objects as well as to distribute and save raw file content.

For maintaining relationships and properties of information objects, any relational database management system (RDBMS) can be used through its JDBC interface, e.g., MySQL or Oracle. For the raw file content, there are two options: (*i*) for relatively small objects, as regarding performance it might be beneficial to store them together with the properties and relationships in the same database; (*ii*) for other files, an SRM (Storage Resource Manager) enabled Storage Element (SE) in the Grid, e.g., DPM (Disk Pool Manager) is used via the GFAL (Grid File Access Layer) interface. Nevertheless, such files will also be managed via the database, i.e., even when storing a document in the Grid, access is only available via the Storage Management Service. The database, in this case, will be used to store a pointer to the physical location of that document as well as its properties and relationships.

For deploying the Storage Management component, one instance of an RDBMS and one instance of an SE must be present at the hosting site. Deploying a database instance on every storage node does not introduce any overhead in terms of installation and maintenance of a

[9] With the term "archive" we refer to any form of *data source* ranging from OAI-PMH compliant data providers to data repositories implemented in a home-made fashion and web pages linking the available documents. The AI service is composed by an extensible set of wrappers that once appropriately configured are capable to extract the available content and metadata and make it "DILIGENT compliant".

new RDBMS. This depends on the fact that gLite storage components typically rely on a back-end database server which is either MySQL or Oracle. Therefore, an RDBMS which is already installed on a storage node can easily be used within the Base Layer for storing information objects. Dashed lines in Figure 7 illustrate that cataloguing and storage components of gLite middleware are also using the same RDBMS. In addition to the Storage Management component, many other services may arbitrarily be deployed at the same hosting nodes. This is in particular beneficial if some form of computation (e.g., feature extraction, filtering, etc.) resides close to the data it operates on.

Another important requirement for DILIGENT is that its services are to be distributed across several sites. That is, data is produced at many places and has to be stored and retrieved efficiently. The Base Layer provides the Replication Service responsible for managing replication. It guarantees transparent access to data stores efficiently, e.g., by routing access request to the nearest or least loaded site which holds the replica of the requested data.

3.5 Service Composition

The requirements coming from the Earth Observation sector which have been identified in Section 2 imply challenges with respect to the (ad-hoc) definition of workflows and the combination of services for the definition, analysis, and processing of data. The need for combining distributed services into user-defined processes demands a sophisticated solution for service composition on top of a Grid environment. An Earth Observation community user should be provided with the means to seamlessly combine several service invocations in a well-defined order and with well-designed execution guarantees so that s/he can establish even more powerful composite services.

In the following sections we discuss the DILIGENT solution for designing composite services and for executing them.

3.5.1 Compound Service Design and Validation

Modern technologies like XML, SOAP[10] and WSDL[11] provide a simple yet powerful means to publish service information and to access services. The platform independent definitions of these technologies further simplify the composition of services to offer new value added services. One way to achieve this goal is to define transactional processes (according to the model of transactional processes over web services [44]). Such processes compose web service calls in an application-specific invocation order together with transactional guarantees. Each activity of a process corresponds to the invocation of a (Web) service. Programming using processes is referred to as "*programming in the large*" or "*mega programming*" [51]. Essentially, processes are again (higher-level) Web services, i.e., they are accessible via SOAP and described by a WSDL document such that other users can easily integrate them into even larger processes.

A process defines the logical dependencies between independent services by specifying an invocation order (control flow), rules for the transfer of data items between different invocations (data flow) as well as transactional semantics, in order to ensure the correct execution of processes in case of concurrency and failures. An infrastructure for transactional processes has to support all these features at runtime. Furthermore, a graphical process modelling tool should support the user in easily specifying these features.

In DILIGENT, all these key features are provided by the CS Design and Verification service, by means of a graphical modelling tool, which allows for the easy composition of services into process definitions while controlling their creation, editing, viewing, and storing. In DILIGENT, we are using the BPEL[12] standard as a starting point for the process specification language. Our modelling tool generates BPEL-compliant process definitions, which we have enriched with transactional properties. An application developer can then export new/revised processes from the modelling tool to the DILIGENT system, where they can be executed. Such a modelling tool has been implemented in Java and runs as an applet, integrated into a portlet running on the DILIGENT portal host.

Figure 8 shows a screen shot of the applet. The modelling tool offers process design in a graphical box and arrow approach for the control flow. The rounded boxes in Figure 8 indicate the activities of the process (service invocations). The concept of a process whiteboard, i.e., the in-process variable area, has been introduced in order to model the data flow within the process. This means that the data flow is defined as mappings from the process instance data space (the "Whiteboard") to the service request parameters, and back from the response values to the whiteboard after execution.

On the right hand side, the figure depicts the whiteboard of the current process (tool box "Whiteboard"), which contains the global variables of a process instance. During process execution, the whiteboard of a process instance is filled with the process arguments (tool box "Process Input Arguments"). The value of Whiteboard parameters might be modified, during process execution, by service invocations. Finally, the parameters returned from a service invocation are fed back to the whiteboard

[10] Simple Object Access Protocol, `http://www.w3.org/TR/soap/`
[11] Web Service Description Language, `http://www.w3.org/TR/wsdl`
[12] Business Process Execution Language for Web Services, `http://www.ibm.com/developerworks/library/specification/ws-bpel/`

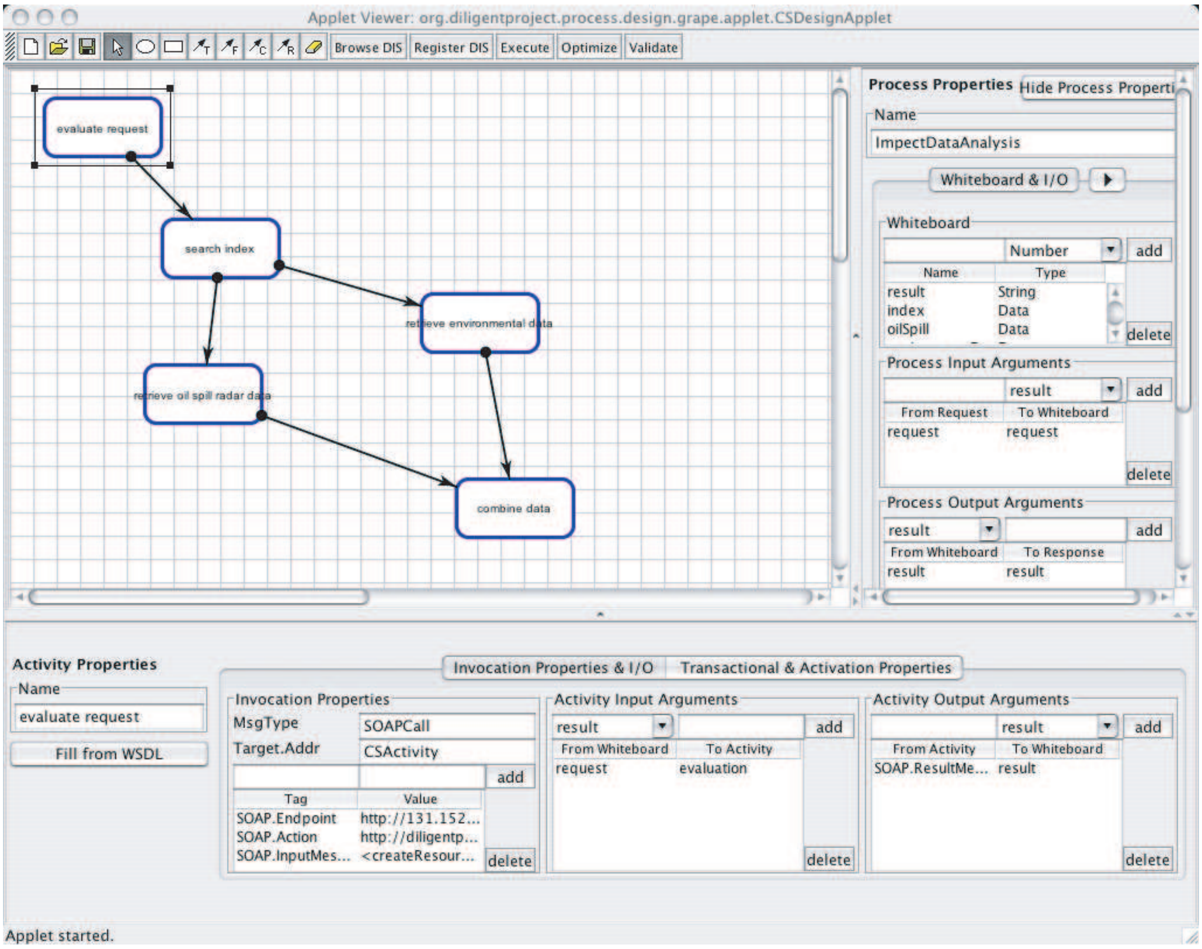

Fig. 8 A screen shot of the modelling tool for process design.

and the result of the process invocation is assembled from its contents (tool box "Process Output Arguments").

A service invocation retrieves its input parameters from the whiteboard (see tool box "Activity Input Arguments" at the bottom), and maps output parameters back to the whiteboard (tool box "Activity Output Arguments"). Further, arrows define the control flow of a process.

Failure handling strategies at the application level are provided by compensating activities and alternative execution paths. Therefore, each activity is described by a set of transactional properties like rollback, compensation, retriability, and failure probabilities. Given a transactional process description, the modelling tool is able to check the correctness of the description exploiting formal criteria.

Each activity of a process corresponds to an invocation of a service (which corresponds, in BPEL terminology to an `<invoke>` activity). This service can either be a basic (web) service or a process itself. Furthermore, because of their BPEL-compliance, activities can also consist in (*i*) assignments (of the type BPEL `<assign>`), i.e., constructs which can be used to update the values of the process variables with new data; (*ii*) "wait" activities (of the type BPEL `<wait>`), which suspend the operation for a given period of time or until certain time has passed; and (*iii*) "empty" activities (of the type BPEL `<empty>`), which correspond to a no-operation used for synchronisation purposes. Structured BPEL activities are also supported, such as (*i*) scope (of the type BPEL `<scope>`), which defines a block of activities; (*ii*) sequence (of the type BPEL `<sequence>`), which implies the execution of a set of activities one after the other; (*iii*) flow (of the type BPEL `<flow>`), which implies the execution of one or more activities concurrently, i.e., with a parallel processing; and (*iv*) "switch" activities (of the type BPEL `<switch>`), which means that the switch chooses the set of activities to execute in accordance to a condition.

Two different orders are defined on the set of activities: the partial precedence order and the preference order. The former specifies the regular order in which activities are executed, i.e., the order in which the associated services are invoked. An activity can only be executed when all its precedent activities have successfully finished and when the conditions on its execution are fulfilled. Since the partial precedence order is a partial order, intra-process parallelization can be realised by parallel branches (fork/join). Instead, the preference order specifies alternative executions that can be chosen when an execution path fails.

Web services are called by a SOAP component of the runtime engine and this fact is reflected in the process specification. The address, operation name, and parameters of the SOAP call are input parameters for the activity. In order to assist the user during the design of SOAP calls, the modelling tool allows the designer to choose a service among the list of the registered ones. When a registered service is selected, the modelling tool parses the service WSDL description and makes its available operations selectable with a click. When the designer has selected the desired operation, the SOAP parameters are automatically set and s/he only needs to map input and output parameters from and to the whiteboard respectively.

At the end of the process design task, the modelling tool outputs the process description and new/modified versions of a process can be registered and stored in the Diligent Information Service. The tool also enables the user to browse the Diligent Information Service to look for existing processes. Further, it provides filters for the narrowing of the search, in order to ease the user's search for already existing processes. Before saving a process the Diligent Information Service, the tool invokes the process validation task that applies syntactical and semantic correctness checks. Optionally, this validation task can check any transactional or additional properties that the process specification may have. The result of the validation process is a "validation signature" confirming that the process is correctly defined. Processes without the validation's signature or whose signature is corrupted will be rejected by the runtime engine.

By completely following the SOA paradigm, Diligent processes can encompass not only Diligent services but also external services, independently of the location where these services are hosted and of the application implementing them (for instance, services defined by other DL systems). In addition, Diligent processes can be also used from outside Diligent because they are services as well (i.e., they have a service interface and can be called accordingly), they

3.5.2 Compound Service Execution

After a process has been designed and stored in the Diligent Information Service, users or other services within the Diligent infrastructure may start its execution. This is done by sending a *Start CS* message to (any running instance of) the service responsible for the execution of compound services. The process is then executed in a decentralised manner as described below.

Consider the sample process depicted in Figure 9, which represents an excerpt of the Earth Observation application described in section 2. Different shapes repre-

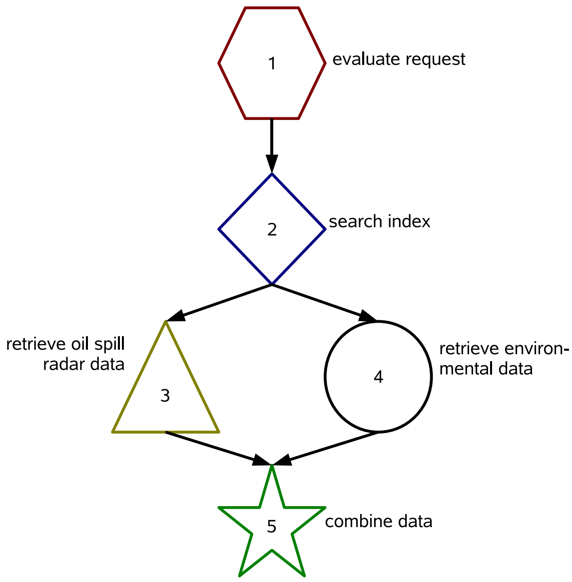

Fig. 9 Sample distributed process

sent different types of services. It should be stressed that the activities specified in the compound service specification refer only to the *type* of the service, not to actual running instances identified by concrete endpoint references. The actual nodes where the process is executed are determined at runtime using information about the deployment of the different services and the current status of the available nodes.

Of course, the orchestration and co-ordination of the execution of a process must be handled by some process execution engine; however, in order to provide true decentralised execution, the orchestration of the process execution should not be done by a single instance of such an engine – otherwise, this would create a single point of failure and a potential hot spot –, rather it should be distributed, ideally closely along the path that the process execution takes. This implies that the nodes hosting the services to be executed are equipped with the execution engine, so that each activity invoked in a workflow can be handled by the local engine. Therefore, on each node of the Diligent infrastructure a copy of the process execution engine has been deployed.[13]

The functionality for co-ordinating compound service execution is provided by a WSRF service named *CSEngine*. The CSEngine is built on top of the existing distributed process execution engine *OSIRIS* [45,46], which has been modified and enhanced to support WSRF service calls and to use the information service provided by the Diligent infrastructure. The actual execution of a process is a chain of interaction between the CSEn-

[13] It is absolutely not required that is each node the execution engine be installed, since web service calls can of course be done remotely; however it is advisable to have services and execution engine at the same host.

gine services on the involved nodes; each CSEngine in turn invokes the target services locally. Figure 10 gives an overview of the most important components of the CSEngine service and the interactions taking place during the execution of a process (in this case, the execution of activity 3 of the process depicted in Figure 9). In the following, the individual components and their relationships are described in more detail.

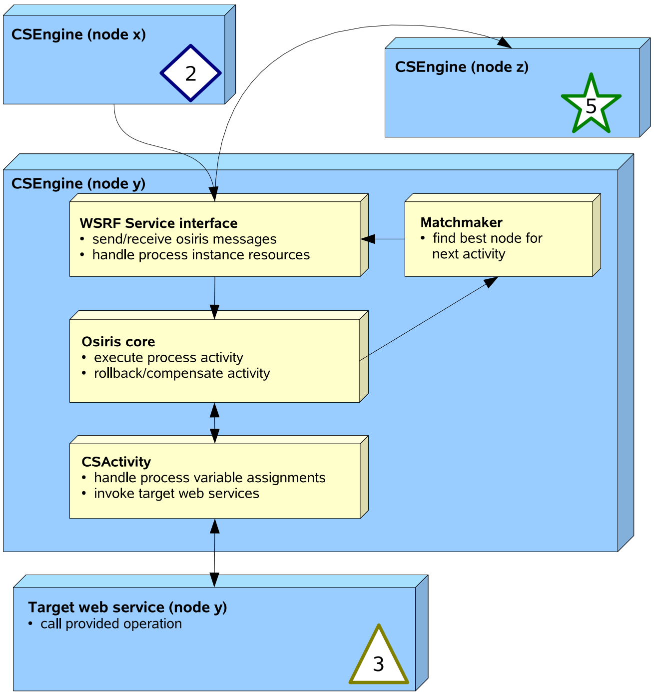

Fig. 10 Architecture of the CSEngine Service

WSRF Service Interface The main functionality of this component is to provide the SOAP interface to the CSEngine, i.e. it provides web service operations which hand on the received data to the OSIRIS core engine. It also contains methods for invoking these operations on other CSEngine nodes, i.e., for forwarding process execution messages to other nodes. Finally, this component maintains, as a WS-Resource, the information about process state for each compound service execution the node has been involved in. This is done by saving the (input and output) messages related to the process in the WS-Resource. These backup states may be required when an already executed part of the process has to be rolled back or compensated.

OSIRIS Core This component handles the orchestration of the execution of activities. It receives the message from the WSRF service interface and by using the process definitions (replicated from the DILIGENT Information Service), it determines which step of which process this message corresponds to. Moreover, it enriches the message with all required additional information (e.g., other static parameters about the activity, as defined in the process specification) and forwards the message to the relevant component – represented as the CSActivity component in DILIGENT. On return, the process definition is checked for the successor activities of the current node, the corresponding OSIRIS messages are generated and ultimately forwarded to the Matchmaker.

CSActivity The CSActivity is the component which actually executes the given compound service activity. At the moment, two types of activities are implemented, which correspond to the BPEL <invoke> and <assign> activities; the former invokes a target web service (typically residing on the same node where the CSEngine is), the latter is used for evaluating and performing process variable assignments.

Matchmaker The matchmaker receives an OSIRIS message containing, among others, information about the next activity to be executed in the compound service. The matchmaker is in charge of determining which node is the best one for handling this message by using the message itself (in particular, the information about the type of activity and the type of the target web service) as described below. The resulting target node is added to the message, and the message is handed back to the WSRF Service Interface that forwards it to the CSEngine on the target node.

As mentioned before, the most important aspect of the process execution in DILIGENT is its distribution: even the co-ordination and orchestration of the process execution is distributed among multiple instances of the *CSEngine* service. At runtime, ideally the only remote service calls should take place between the involved CSEngine instances, while the invocations of the actual target services for a process activity are handled locally at the respective target node.

The decision where the next activity of a process is to be executed is made dynamically at runtime by the Matchmaker component using information from the DILIGENT IS. On startup, the component gathers a list of all the services running on the local node. It then queries the IS for process specifications containing invocations of either of these services, because the CSEngine may become involved in the execution of these processes. Finally, a list of the potential successor services (i.e., the *next* activities in the respective processes) is created from this information. The IS is then queried for deployment information about these services, and for QoS-related information about the corresponding nodes – a significant example is the current load of the node. With all this information available, the routing decision can be made using the local knowledge only. However, the requested information may change over time (e.g., new process specifications might be added, service instances might be deployed or un-deployed, the state of the nodes might change, etc.). Therefore, in addition to the initial queries, corresponding subscriptions to notifications about status changes are set up. This allows for timely updates of the state information, while avoiding the overhead which would be introduced by constant polling.

The matchmaker is able to find the best successor nodes thanks to its local knowledge which includes information on the current node and the successor service type. However, seen from a global perspective (i.e., knowing the complete process specification), this decision may be considered sub-optimal – e.g., one might be willing to have more control over the allocation of activities in parallel flows of the process, taking into account data proximity issues. Therefore, in some cases pre-planning of the allocation in terms of the involved nodes is important as well; the invocation of the process is then accompanied by hints for the preferred execution node for each activity, which are considered at runtime. This type of pre-planning is most useful for short-running processes because the state of the infrastructure is expected to change more and more over time. In DILIGENT this type of process execution is particularly useful for search processes.

4 The ImpECt Scenario

The DILIGENT test-bed infrastructure has been used for experimentations in the setting of a typical application scenario from the environmental sector. The setup of this scenario, named IMPECT (Implementation of Environmental Conventions) has been driven by ESA, especially in terms of the selection of the registered information sources and tools, the definition of the virtual organisation and the definition of an appropriate DL serving the needs of this scenario. The DILIGENT DL serves this application scenario in supporting the creation of technical and periodical environmental reports related to international and regional conventions on marine pollution.

The underlying Grid infrastructure used to support IMPECT relies on three gLite based infrastructures: the DILIGENT test-bed infrastructure, the ESRIN EO Grid infrastructure, and the EGEE pre-production infrastructure. The former infrastructure consists of five sites located in Pisa (Italy), Rome (Italy), Athens (Greece), Basel (Switzerland), and Darmstadt (Germany). In total, these sites provide storage and computational capabilities of 41 processors, 38.72 GB RAM, and 3.28 TB disk space. The second infrastructure is formed by computing and storage resources aggregated for serving the Earth Observation community. Controlled sharing mechanisms from the underlying Grid make resources available in the DILIGENT infrastructure by simply registering them within the DILIGENT Information Service and granting access to a recognised DILIGENT user. This infrastructure provides a total of 90 processors, 160 GB RAM, and 30 TB disk space. The last one is an infrastructure built for the purposes of the EGEE project. The main goal of this project is to build the largest European Grid Infrastructure, as demonstrated by the following numbers: over 30,000 CPUs with about 5 Petabytes (5 million Gigabytes) of storage. Clearly, having access to such a powerful set of resources, changes the way scientific research takes place.

The data sources added to the DILIGENT infrastructure to serve IMPECT consist of "external" content providers, such as the NASA CEOS IDN initiative[14], the European Environment Agency[15], and Medspiration[16], plus a pool of three community specific data sources, all placed at the ESA's European Space Research Institute (ESRIN), namely:

- EO ESA web portal[17] documents and data. This is a collection of heterogeneous information objects collected from a large number of providers. For instance, it contains classic documents like research studies and meteorological papers, EO products provided by ESA, DLR, NASA, and others, images like satellite images of the Black Sea showing swirling blooms of phytoplankton colouring the surface waters blue and green, various maps presenting data on different studies like oil pollution, burn scars, cloud cover, etc.
- EO Grid on demand system[18] EO products. Examples of such products are the Chlorophyll-1 measure corresponding to the algal pigment index 1 expressed as a chlorophyll concentration in mg.m-3, the mosaic made up of true colour images using four out of 15 MERIS spectral bands taken from Envisat with data combined from the selected separate orbital segments with the intention of minimising cloud cover as much as possible by using the corresponding data flags, and vegetation indexes measuring the amount and vigour of vegetation at the surface.
- EO catalogue together with relevant databases and archives from past Mediterranean-oriented scientific activities, such as those routinely generated within the ESA - European Commission Global Monitoring for Environment and Security (GMES[19] services). This information includes, among others, data to estimate the amount of oil spills and other coastal pollution accidents over specific geographic regions (see MARCOAST project[20]).

The DILIGENT Content Management services are in charge of harvesting the data and their descriptions, i.e.,

14 http://idn.ceos.org
15 http://www.eea.eu.int/
16 http://www.medspiration.org/products/
17 http://www.eoportal.org
18 http://giserver.esrin.esa.int/
19 http://www.gmes.info
20 http://www.gmes-marcoast.com

metadata, from all the above mentioned different providers and to deliver or reference the content within the system. The import of such information requires standard interfaces for describing the content to be harvested. Several solutions have been investigated at ESA (see Section 5.3 for reference) to solve this problem. Specifically, ImpECt enhances the metadata descriptions of geospatial information by using standard service interfaces as promoted by OGC[21] and data descriptions as proposed by ISO[22] standards. Further, ImpECt is expected to include, among other data sources, an ESRIN hosted GeoNetwork node[23] that will fully conform to the ISO standard rules concerning metadata exposure and harvesting.

The services and tools registered in the Diligent infrastructure comprise those accessible through the EO ESA web portal, as well as atmospheric profile generation, volcano activity monitoring, and mosaics generators provided by the EO Grid on-demand system.

The infrastructure also provides specific services supporting the generation of reports. These have been developed from scratch as part of the Diligent project activities. They can exploit the Diligent service composition facilities in order to set up and handle user defined distributed workflows of available services. Workflows may involve any of the EO services, e.g., vegetation index or mosaics generators, web map and web features services, data analysis and visualisation services, numerical models services, hazards planning services. These EO and general services, whether used separately or aggregated and chained within automated workflows, allow to interactively maintain complex environmental reports fully realising the notion of living documents of basic importance in the environmental application context.

5 Related Work

Due to the broad nature of the Diligent project, related work spans a couple of different areas. In what follows, we focus on the various initiatives set up in different countries aiming at implementing infrastructures supporting e-Science applications(Section 5.1), on the technologies related to the described Diligent features, e.g., dynamic service deployment and service composition (Section 5.2), and finally we present initiatives relevant to the ImpECt scenario (Section 5.3).

5.1 e-Science Initiatives

A series of initiatives have been put in place in several countries in order to support and implement new infrastructures aiming at revolutionising the way through which joint research is conducted. The majority of these initiatives focuses on mechanisms for *opening* the access to heterogeneous information sources, providing search facilities over these sources, or implementing vertical solutions (typically portal based) to serve the specific application needs of a certain community.

The European Commission has recently focussed its attention on the implementation of a European "*e-Infrastructure*". In particular, during a recent workshop held in Brussels[24] the development of an Infrastructure for Digital Repositories has been stressed as one of the key steps towards the realisation of a more general e-Infrastructure serving the European researchers in their activity. The participants to this workshop have recognised Digital Repositories as knowledge resources that are essential for the European Research Area, much in the same way as data transport network, Grid computing resources or data storage are. The Diligent test-bed already covers most of the basic functionality required for such repository infrastructures. However, in order to be become a real production Repository Infrastructure it still needs a real organisational model that regulates the sharing and interoperability policies among a large number of participating organisations.

In the US the term "*cyberinfrastructure*" has been coined by the National Science Foundation (NSF) to denote new research environments in which advanced computational, collaborative, data acquisition and management services are made available to researchers through high-performance networks [4]. The term is now widely used to embrace a range of environments that are emerging from the changing and innovative practices – often called "*e-Science*" or "*e-research*" – of scientists and scholars in all disciplines. The cyberinfrastructure comprises hardware, software, and a set of supporting services made available to researchers by their home institutions as well as through federations of institutions and national and international disciplinary programs. In this context the San Diego Supercomputing Center (SDSC) is providing a number of services and community specific software among which we mention the Storage Resource Broker [43] (SRB). The SRB is a Grid based data management system that implements a distributed logical file system providing seamless access to distributed storage resources. Widely used content repository systems, like DSpace [47] and Fedora [29] as well as DLs, like the Library of Congress, are presently using the SDSC SRB as a platform for ensuring preservation and long term availability of the access to digital information [39,42]. Another initiative of the SDSC closely connected to this paper is represented by the GEON project[25]. It is a typical example of a portal based solution aiming at providing searching, semantic integration, and visualisation of collections as well as geospatial and 4D data-sets. More-

[21] http://www.opengeospatial.org/
[22] http://www.iso.org/
[23] http://www.fao.org/geonetwork/
[24] http://cordis.europa.eu/ist/rn/ri-cnd/wshop-080606.htm
[25] http://www.geongrid.org/

over, the portal provides a series of tools that support data analysis and model execution by relying on a high-performance computing platform. At a first look this application presents many similarities with the DILIGENT implementation of the ImpECt scenario. The main difference resides in the dynamic nature of the DILIGENT approach where the pool of resources needed to implement the application scenario (the services and the data) as well as the portal providing access to this customised pool of resources are aggregated on-demand. Moreover, it is important to recall that the SDSC is in charge to maintain a persistent archive of the NSDL [27,22] collections thus providing a classic example of usage of Grid facilities to maintain huge amounts of data.

In the UK, a series of initiatives started to investigate and promote the large scale scientific research activity that is carried out through distributed global collaborations enabled by the Internet. Once again, the focus is on knowledge sharing while the infrastructures produced are not dynamic as those DILIGENT promotes. The eBank project[26], for example, brings together chemists, digital librarians and computer scientists in an interdisciplinary collaboration which explores the potential for integrating research data-sets into Digital Libraries by using common technologies, mainly the OAI-PMH [30]. The main goal is to provide an archive of research results, raw and analysed data in order to avoid the "publication bottleneck" and provide a data quality check, i.e., promote *open access*. During the first phase (2003-2004) an OAI-PMH repository was created and a number of data-sets from the chemistry sub-discipline of crystallography were deposited in it, offering a local browse and search interface, tailored to crystallography. The second phase, currently ongoing, aims to identify complex digital object descriptions and apply them to the description of scientific data.

In Germany, in September 2005, with the goal to establish e-Science methods in the german scientific community, six user community specific Grid projects and the D-Grid Integration Project – aiming at integrating all developments from the different community projects into a single platform – started. Among them, the Text-Grid[27] project plans to support text-based disciplines by providing tools for scientific editing, definition of standardised interfaces for publication software, modules for scientific text processing as well as administration of defined and controlled access to such data and tools. As in all the other initiatives that have been cited, also in the German ones the utilisation of Grid facilities is limited to the usage of distributed computing and storage capacity. There is no attempt to create an integrated framework where all the different types of resources needed for creating a DL can be shared. In contrast, DILIGENT promotes an enhanced vision were the services needed to operate are dynamically deployed by relying on available hosting nodes.

As examples of projects that exploit Grid facilities for supporting specific DL functionality we cite Cheshire3 [32] and Digital Library GRID [40].

Cheshire3 is an Information Retrieval system recently developed in a partnership between Berkeley and the University of Liverpool. It operates both in single processor and in Grid distributed computing environments. Recently, a new release of this system capable of processing and indexing also documents stored in the SRB via their inclusion in workflows has been designed.

Digital Library GRID is an on-project at the Old Dominion Digital Library Research Group. Its goal is to build a high performance OAI [31] federated search service. Data are gathered by relying on the OAI-PMH [30] protocol and processed by a testbed that uses 3 Grid nodes to perform the high-latency tasks of harvesting and indexing from 3 data providers. The Grid is also used to transmit these indices and metadata to a small cluster of search engines (3 nodes), where each of the nodes is working on one or more indices it receives from the harvesting nodes.

5.2 DILIGENT Related Technologies

With respect to resource monitoring and discovery in a distributed and dynamic environment as envisaged in the DILIGENT context, the Globus Monitoring & Discovery System (MDS)[28] [11] plays a leading role. In particular, this system allows users to discover which resources are considered part of a Virtual Organization and to monitor those resources. MDS services provide query and subscription interfaces to arbitrarily detailed resource data and a trigger interface that can be configured to take action when pre-configured conditions are met. The DILIGENT Information Service relies on this service and the related technologies in order to provide the set of features needed to operate on the pool of resources constituting DILIGENT.

The dynamic service deployment functionality the Keeper is in charge of has been explored and developed in many different contexts. A recent paper [41] produced in the context of the Globus project surveys related works and presents the dynamic deployment infrastructure that is implemented in the Globus Toolkit Java Web Services container.

With respect to the content management, it is important to separate the storage part from the information object management part where the latter kind of functionality is built on top of the former. This approach is the same we adopted in DILIGENT as well and, as previously stated, is in line with the solution proposed by the Fedora [29] and DSpace [47] repositories in order to exploit third party storage facilities like the SRB [43].

[26] http://www.ukoln.ac.uk/projects/ebank-uk/
[27] http://www.d-grid.de/index.php?id=167&L=1
[28] http://www.globus.org/toolkit/mds/

A concrete experience in implementing a Digital System able to exploit third party storage resources by providing high level content management facilities is reported in [9].

With respect to services composition, a lot of approaches to define specifications and languages exist. The most common one – and the one chosen as a starting point for the service composition chosen in Diligent – is BPEL4WS (Business Process Execution Language for Web Services), which combines ideas from WSFL (Web Services Flow Language) and XLANG (XML Business Process Language). Similarly, the WS-Choreography specification provides an information model that describes the data and the relationships between them. This model is needed to define a choreography that describes the sequence and conditions in which the data are exchanged between two or more participants in order to meet some useful purpose. The Web Service Conversation Language (WSCL) allows the definition of business level conversations supported by a Web service. WSCL specifies the XML documents being exchanged, and the allowed sequencing of these document exchanges. WSCL conversation definitions are themselves XML documents and can therefore be interpreted by Web service infrastructures and development tools.

For the execution of business processes (composite web services), several commercial systems like the IBM WebSphere Choreographer [50] or Microsoft BizTalk [38] exist. These commercial systems follow a centralised approach, where every call to a service provider returns to the process engine. Albeit navigation tasks can be distributed in a cluster, storage of process instances is usually done by using a single, centralised database instance (products like Oracle 10g can also support clustered databases). In addition, various workflow and process management systems have been developed in academia (e.g. WISE [33], MENTOR-lite [23], Kepler [2]). However, except for systems like MENTOR-lite which are able to distribute the task of orchestrating process execution to a set of (predefined) cluster nodes, these systems are also relying on a centralised execution engine, whereas in Diligent even the orchestration and co-ordination of the workflows is distributed at runtime in a peer-to-peer manner.

5.3 Earth Observation Initiatives

Two key EO initiatives are of particular importance with respect to the ImpECt scenario previously presented:

- The management of metadata to describe satellite derived data sets based on the use of emerging standards for managing geospatial information;
- The exploitation of data processing services as offered via the Grid on Demand Earth Observation web portal.

Earth Observation Metadata handling The international community is putting a lot of effort in improving the way to generate metadata descriptions for Earth Observation resources like raw satellite data, processors, services, and dedicated catalogues. Relevant initiatives are the following:

- opensource FAO GeoNetwork[29], which allows geographically referenced thematic information be share among different FAO Units, UN Agencies, NGO's and other institutions like ESA;
- Open Geospatial Consortium[30], which aims at creating open and extensible software application programming interfaces for geographic information systems (GIS) and other mainstream technologies based on internationally accepted standards for geospatial and location based services;
- International Organisation for Standardization[31], active in defining standards to manage metadata for describing geospatial information.

A number of recently held meetings among ESA, JRC, and FAO besides standards bodies representatives demonstrate the willingness to converge to a uniform and standardised way in describing Earth Observation data products, processors, algorithms and catalogues.

Earth Observation Grid on-Demand web portal The ESA Earth Observation Grid on-Demand web-portal (GoD)[32] is a demonstration of distributed architecture based on Grid and Web Service technologies used to manage distributed resources.

The main functionalities offered by this portal are:

- to provide an integrated environment enabling users to develop new algorithms or generate high-level and global products;
- to act as a unique and single access-point to various metadata and data holdings for data discovery, access and sharing;
- to provide the community reference environment for the generation of systematic application products coupled with direct archives and near real-time data access.

In particular, the Grid on-Demand portal allows for autonomous discovery and retrieval of information about data-sets for any area of interest, exchange of large EO data products, and triggering concurrent processes to carry out data processing and analysis on-the-fly.

A typical application, such as the generation of 10-day composite (e.g., chlorophyll or sea surface temperature maps) over the whole Mediterranean area derived from Envisat-MERIS and AATSR data, involves the reading of some 10-20 GB data for generation of a final summary products of some 10-20 MB. The presence of a Grid

[29] http://www.fao.org/geonetwork/
[30] http://www.opengeospatial.org/
[31] http://www.iso.org/
[32] http://eogrid.esrin.esa.int

on-Demand node with the EO facilities performing data acquisition or data archiving realise this application with a great saving of data circulation and network bandwidth consumption.

Access to Grid computing resources is handled transparently by the EO Grid interfaces that are based on Web Service technology (HTTP-S and SOAP/XML) and developed by ESA within the DataGrid project (EC Grant IST-2000-25182E).

It is important to recall that the facilities offered by this portal have been included in the DILIGENT infrastructure, thus making it possible to implement a business model different from the one based on the GoD portal, i.e., DILIGENT implements the *business on demand* approach where the relevant services, data sources and computing resources are dynamically aggregated to fulfil the requirement of a user community. DILIGENT makes the re-use of a data source or of a GoD service possible in contexts different from those the service or the data source has been developed for, thus enhancing the e-Science production process and making it cheaper.

Besides the above mentioned initiatives, the NSF promotes projects like Unidata[49] and Linked Environments for Atmospheric Discovery (LEAD) [13] presenting many commonalities with the DILIGENT project. In particular, the Unidata project implements a portal serving a community of about 160 institutions by providing Earth-related data and a variety of software packages that can be downloaded for free and, once installed, allow the Unidata community to perform certain processing tasks. Differently from DILIGENT, no effort is spent on the integration of these tools into a single application. The LEAD project instead is a web portal application making meteorological data, forecast models, and analysis and visualization tools available. The similarities with DILIGENT resides in (*i*) the usage of a service oriented architecture, (*ii*) the usage of workflows in implementing complex applications orchestrating many web services, and (*iii*) the usage of local and remote computing and storage resources as well as services by relying on grid facilities. The main difference consists in the lack of dynamic deployment in the LEAD project. The resulting LEAD portal when compared with the DILIGENT approach can be considered as a "static" DL serving the mesoscale weather research community.

6 Conclusion

The role nowadays played by Digital Libraries has evolved a lot with respect to that originally assigned to them. They are now conceived as systems providing access to any kind of information, thus becoming a central pillar in the process of producing and consuming knowledge, also known as e-Science.

In this paper we have presented DILIGENT, a test-bed infrastructure supporting e-Science communities in building the DLs needed to support their daily activities. The *business on-demand* model implemented by DILIGENT will make it easy and effective to provide Digital Libraries as technologies for serving the purposes of dynamic and evolving communities. To implement this business model, DILIGENT combines DL and Grid technologies in order to make available a very comprehensive and extensible pool of resources ranging from collections of information objects and general purpose services to community specific services, storage resources, and computing elements. This process promotes and broadens the reuse of resources thus making the whole DL development process cheaper, faster, and effective.

In order to demonstrate the feasibility of the DILIGENT approach, we have shown its capability in fulfilling the requirements arising in a concrete environmental e-Science scenario. The use of the DILIGENT infrastructure in supporting the IMPECT scenario has demonstrated its validity as a powerful tool for such a relevant and wide-spread community. Just as current use of Grids enables Earth Observation researchers to send their computations to the data – rather than relocating huge amounts of data to the processing nodes –, so Digital Libraries set up through DILIGENT come to the users providing them with seamless access to the resources they need.

The experimentation carried out in IMPECT is just the starting point of a larger introduction of Grid enabled Digital Libraries within the Earth Observation community. The availability of a powerful infrastructure as DILIGENT is let us envision many other envisioned improvements of the EO work, like the disappearing of problems nowadays related to the management of huge amounts of heterogeneous content.

There are other ongoing experimentations of DILIGENT. Particularly interesting is an experimentation with a virtual research community in the humanities research field, which exploits part of the IMPECT resources for supporting research activities that were inconceivable in the past. This experimentation shows the capability of DILIGENT to provide and manage the sharing of heterogeneous resources while, at the same time, it demonstrates its ability to support the creation of user-centred DLs offering new research tools. Further experimentation aiming to support different virtual organisations in completely different domains is also on schedule. These will exploit the same sources and tools as registered for IMPECT by promoting a cross-fertilisation process.

Acknowledgements This work is partially funded by the European Commission in the context of the DILIGENT project, under the 2nd call of FP6 IST priority. We would like to thank all the DILIGENT partners that have contributed with their suggestions, comments and expertise to this work. Special thanks go to Maria Bruna Baldacci for her valuable help in improving the the readability of the paper.

References

1. DIGITAL LIBRARIES: Future Directions for a European Research Programme. Brainstorming report, DELOS, San Cassiano, Alta Badia, Italy (2001)
2. Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludäscher, B., Mock, S.: Kepler: An Extensible System for Design and Execution of Scientific Workflows. In: M.J. Franklin, B. Moon, A. Ailamaki (eds.) Proc. of the 16th Intl. Conf. on Scientific and Statistical Database Management (SSDBM'04), Santorini Island, Greece, June 21–23 2004 (2004)
3. Arms, W.Y.: Digital Libraries. The MIT Press (2001)
4. Atkins, D.E., Droegemeier, K.K., Feldman, S.I., Garcia-Molina, H., Klein, M.L., Messerschmitt, D.G., Messina, P., Ostriker, J.P., Wright, M.H.: Revolutionizing science and engineering through cyberinfrastructure. Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure (2003)
5. Banks, T.: Web Services Resource Framework (WSRF) - Primer. Committee draft 01, OASIS (2005). `http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-01.pdf`
6. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web Services Architecture. Tech. rep., W3C (2004). URL `http://www.w3.org/TR/ws-arch/`. W3C Working Group Note
7. Bush, V.: As We May Think. The Atlantic Monthly **176**(1), 101–108 (1945)
8. Candela, L., Castelli, D., Pagano, P., Simi, M.: Moving Digital Library Service Systems to the Grid. In: C. Türker, M. Agosti, H.J. Schek (eds.) Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures, no. 3664 in Lecture Notes in Computer Science, pp. 236–259. Springer Verlag (2005)
9. Candela, L., Castelli, D., Pagano, P., Simi, M.: OpenDLibG: Extending OpenDLib by exploiting a gLite Grid Infrastructure. In: 10th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2006 (2006)
10. Castelli, D., Candela, L., Pagano, P., Simi, M.: DILIGENT: A DL Infrastructure for Supporting Joint Research. In: I.C. Society (ed.) 2nd IEEE-CS International Symposium Global Data Interoperability - Challenges and Technologies, pp. 56–69 (2005)
11. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid information services for distributed resource sharing. In: Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10). IEEE Press (2001)
12. DILIGENT: A DIgital Library Infrastructure on Grid ENabled Technology. `http://www.diligentproject.org/`. IST-2003-004260
13. Droegemeier, K.K., Chandrasekar, V., Clark, R., Gannon, D., Graves, S., Joseph, E., Ramamurthy, M., Wilhelmson, R., Brewster, K., DOmenico, B., Leyton, T., Morris, V., Murray, D., Plate, B., Ramachandran, R., Reed, D., Rushing, J., Weber, D., Wilson, A., Xue, M., Yalda, S.: Linked environments for atmospheric discovery (lead): A cyberinfrastructure for mesoscale meteorology research and education. In: 20th Conf. on Interactive Info. Processing Systems for Meteorology, Oceanography, and Hydrology, Seattle, WA, Amer. Meteor. Soc. (2004)
14. EGEE: Enabling Grids for E-sciencE. `http://public.eu-egee.org/`. INFSO 508833
15. EGEE: gLite: Lightweight Middleware for Grid Computing. `http://glite.web.cern.ch/glite/`
16. Foster, I.: What is the Grid? A Three Point Checklist. GRIDtoday **1**(6) (2002)
17. Foster, I., Kesselman, C.: The Grid: Blueprint for a Future Computing Infrastructure. Morgan-Kaufmann (2004)
18. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum (2002)
19. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organization. The International Journal of High Performance Computing Applications **15**(3), 200–222 (2001)
20. Fox, E.A., Akscyn, R.M., Furuta, R., Leggett, J.J.: Digital Libraries. Communications of the ACM **38**(4), 23–28 (1995)
21. Fox, E.A., Marchionini, G.: Toward a Worldwide Digital Library. Communications of the ACM **41**(4), 29–32 (1998)
22. Geisler, G., Giersch, S., McArthur, D., McClelland, M.: Creating Virtual Collections in Digital Libraries: Benefits and Implementation Issues. In: Proceedings of the second ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 210–218. ACM Press (2002). DOI http://doi.acm.org/10.1145/544220.544265
23. Gillmann, M., Weikum, G., Wonner, W.: Workflow Management with Service Quality Guarantees. In: M.J. Franklin, B. Moon, A. Ailamaki (eds.) Proc. of the 2002 ACM SIGMOD Int. Conf. on Management of Data, Madison, Wisconsin, June 3–6, 2002, pp. 228–239. ACM Press (2002)
24. Globus Alliance: The Globus Alliance Website. `http://www.globus.org/`
25. Ioannidis, Y.: Digital libraries at a crossroads. International Journal on Digital Libraries **5**(4), 255–265 (2005)
26. Ioannidis, Y., Maier, D., Abiteboul, S., Buneman, P., Davidson, S., Fox, E., Halevy, A., Knoblock, C., Rabitti, F., Schek, H., Weikum, G.: Digital library information-technology infrastructures. International Journal on Digital Libraries **5**(4), 266–274 (2005)
27. Lagoze, C., Arms, W., Gan, S., Hillmann, D., Ingram, C., Krafft, D., Marisa, R., Phipps, J., Saylor, J., Terrizzi, C., Hoehn, W., Millman, D., Allan, J., Guzman-Lara, S., Kalt, T.: Core services in the architecture of the national science digital library (NSDL). In: Proceedings of the second ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 201–209. ACM Press (2002). DOI http://doi.acm.org/10.1145/544220.544264
28. Lagoze, C., Krafft, D.B., Payette, S., Jesuroga, S.: What Is a Digital Library Anyway? D-Lib Magazine **11**(11) (2005)
29. Lagoze, C., Payette, S., Shin, E., Wilper, C.: Fedora: An Architecture for Complex Objects and their Relationships. Journal of Digital Libraries, Special Issue on Complex Objects (2005)
30. Lagoze, C., Van de Sompel, H.: The OAI Protocol for Metadata Harvesting. `http://www.openarchives.org/OAI/openarchivesprotocol.html`
31. Lagoze, C., Van de Sompel, H.: The open archives initiative: building a low-barrier interoperability framework. In: Proceedings of the first ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 54–62. ACM Press (2001). DOI http://doi.acm.org/10.1145/379437.379449
32. Larson, R.R., Sanderson, R.: Grid-based digital libraries: Cheshire3 and distributed retrieval. In: JCDL '05: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 112–113. ACM Press, New York, NY, USA (2005)
33. Lazcano, A., Schuldt, H., Alonso, G., Schek, H.J.: WISE: Process based E-Commerce. Bulletin of the IEEE Technical Committee on Data Engineering **24**(1), 46–51 (2001)

34. Lesk, M.: Expanding Digital Library Research: Media, Genre, Place and Subjects. In: Proceedings of the International Symposium on Digital Library 1999: ISDL'99, pp. 51–57. Tsukuba, Ibaraki, Japan (1999)
35. Licklider, J.C.R.: Libraries of the Future. MIT Press (1965)
36. Lomow, G., Newcomer, E.: Understanding SOA with Web Services. Independent Technology Guides. Addison Wesley Professional (2005)
37. MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P., Metz, R.: Reference Model for Service Oriented Architecture 1.0. Tech. rep., OASIS (2006). Public Review Draft 1.0
38. Metha, B., Levy, M., Meredith, G., Andrews, T., Beckman, B., Klein, J., Mital, A.: BizTalk Server 2000 Business Process Orchestration. IEEE Data Engineering Bulletin **24**(1), 35–39 (2001)
39. Moore, R.W., Marciano, R.: Building preservation environments. In: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries (JCDL 2005), p. 424 (2005)
40. Old Dominion University Digital Library Research Group: Digital Library GRID Project. `http://128.82.7.230/grid/`
41. Qi, L., Jin, H., Foster, I., Gawor, J.: HAND: Highly Available Dynamic Deployment Infrastructure for Globus Toolkit 4. Submitted for Publication, 2006
42. Rajasekar, A., Moore, R., Berman, F., Schottlaender, B.: From Digital Preservation Lifecycle Management for Multi-media Collections. In: 8th International Conference on Asian Digital Libraries, ICADL 2005, Bangkok, Thailand, December 2005, pp. 380–384. Springer (2005)
43. Rajasekar, A., Wan, M., Moore, R., Schroeder, W., Kremenek, G., Jagatheesan, A., Cowart, C., Zhu, B., Chen, S.Y., Olschanowsky, R.: Storage Resource Broker - Managing Distributed Data in a Grid. Computer Society of India Journal, Special Issue on SAN **33**(4), 42–54 (2003)
44. Schuldt, H., Alonso, G., Beeri, C., Schek, H.J.: Atomicity and Isolation for Transactional Processes. ACM Transactions on Database Systems (TODS) **27**(1), 63–116 (2002)
45. Schuler, C., Schuldt, H., Türker, C., Weber, R., Schek, H.J.: Peer-to-Peer Execution of (Transactional) Processes. International Journal of Cooperative Information Systems (IJCIS) **14**(4), 377–405 (2005)
46. Schuler, C., Weber, R., Schuldt, H., Schek, H.J.: Scalable Peer-to-Peer Process Management - The OSIRIS Approach. In: Proceedings of the 2^{nd} International Conference on Web Services (ICWS'2004), pp. 26–34. IEEE Computer Society, San Diego, CA, USA (2004)
47. Tansley, R., Bass, M., Smith, M.: DSpace as an Open Archival Information System: Current Status and Future Directions. In: T. Koch, I. Sølvberg (eds.) Research and Advanced Technology for Digital Libraries, 7th European Conference, ECDL 2003, Trondheim, Norway, August 17-22, 2003, Proceedings, Lecture Notes in Computer Science, pp. 446–460. Springer-Verlag (2003)
48. Tierney, B., Aydt, R., Gunter, D., Smith, W., Swany, M., Taylor, V., Wolski, R.: A Grid Monitoring Architecture. Tech. Rep. GFD.6, Global Grid Forum Document Series (2002)
49. Unidata: Providing data, tools, and community leadership for enhanced earth-system education and research web page. `http://www.unidata.ucar.edu`
50. IBM WebSphere Application Server Enterprise Process Choreographer. `http://www7b.software.ibm.com/wsdd/zones/was/wpc.html`
51. Wiederhold, G., Wegner, P., Ceri, S.: Towards megaprogramming. Communications of the ACM **35**(11), 89–99 (1992)