MAPPING SPATIAL DATA INFRASTRUCTURES TO A GRID ENVIRONMENT FOR OPTIMISED PROCESSING OF LARGE AMOUNTS OF SPATIAL DATA

Andreas Krüger[a,*], Thomas H. Kolbe[a]

[a]Technische Universität Berlin, Geodesy and Geoinformation Science, Strasse des 17. Juni 135, D-10623 Berlin – (krueger, kolbe)@igg.tu-berlin.de

Special Session SS-3 on Spatial Data Infrastructure and Spatial Information Grid

KEY WORDS: Spatial Data Infrastructure, Distributed Processing, Grid Computing, Service Chains, Performance, Gridification

ABSTRACT:

Within the scope of processing large amounts of data and complex calculations the use of grid computing is a good choice for achieving high performance, because processing of extensive spatial data can be very computationally intensive. At the moment there exist spatial data infrastructures (SDI) as framework with the use of web services, which are standardised by the *Open Geospatial Consortium* (OGC). The architecture of a SDI is focused on creation and distributed processing service chains in geospatial context. In scientific computing grid environments were developed to provide numerous distributed rich equipped resources to share distributed computing power. Thus grid environments can handle the needs of distributed processing of large amounts of spatial data with very complex calculations. SDI and grid environments in fact build two different networks, which permit the creation and distributed processing of service chains. The development process of both networks already extends over several years and results in numerous sophisticated components of its architecture. So an integration of SDI and grid technologies is beneficial by combination of distributed processing geospatial service chains with numerous distributed rich equipped resources of a grid environment.This paper discusses different approaches to bring both worlds – the SDI and the grid environment – together and “gridify” OGC compliant service chains while maintaining the usability for spatial data infrastructure service user and getting an optimised data and information flow in the gridified service chain. This includes a view on gridification of single services, which have to be parallelised and mapped to the grid environment in an adequate way to the gridification approach of the whole service chain. With these approaches this paper presents a formalisation of services and its instances, which interact between the two networks. In addition some additional requirements of service orchestration are described, to understand how an appropriate composition of gridified web services can be realised at runtime by a workflow engine. The given approaches will be illustrated by an example of a realistic OGC compliant web service chain for the simulation of environmental noise dispersion.

1. INTRODUCTION

The processing of spatial data is becoming increasingly relevant in numerous economic sectors and fields of research. There models of continuous wide-area territories consist of large amounts of spatial data. In this context modern concepts of parallel computing are needed to reach acceptable computing time for processing such data models.

In context of processing spatial data the OGC provides a range of standards for a web service architecture which can handle spatial data in a specific way. With these specifications several SDIs are already realised as one kind of network, which provides building and distributed processing service chains on geospatial datasets. In scientific computing another kind of network - grid environments - can execute computational-intensive calculations on very large amounts of data by using web services. There a grid environment provides a distribution of calculations and datasets on numerous rich equipped grid nodes with possibility of distributed high-speed data transfer. The development process of both networks - SDI and grid environment - already extends over several years and results in numerous sophisticated components of its architecture. An integration of both networks can combine the distributed processing by using geospatial service chains with numerous distributed rich equipped resources of a grid environment.

The first part of this paper introduces the concepts behind both types of networks, on the one hand SDI and grid environment on the other hand. Furthermore a scenario – the Noise Dispersion Simulation – will be given as practical example of a SDI service chain to demonstrate the following concepts. Then the term gridification will be explained, followed by the main part, where three different options of using a grid environment by an existing SDI service chain are presented. As conclusion some thoughts about orchestration of gridified services will be presented as demonstration of a future idea - an automatic creation of gridified service chains and its execution in a grid environment by a workflow engine.

2. SPATIAL DATA INFRASTRUCTURES

The function of a SDI is the hosting of geographic data and attributes, sufficient documentation (metadata), a means to discover, visualize and evaluate the data (catalogue and web mapping) and some methods to provide access to the geographic data (Nebert, 2004). For a short time SDIs are extended by the ability to process geographic data. Actual realisations of SDIs are based on a number of web service standards by the OGC, which follow the Service Oriented Architecture (SOA). More on SOA can be found in (Erl, 2005).

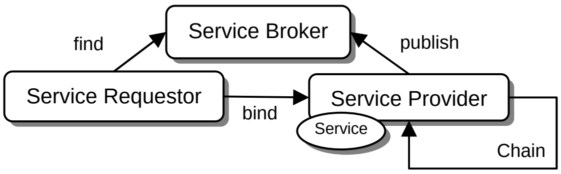


Figure 1. Publish/find/bind Pattern (Percivall, 2003)

The architecture supports dynamic binding of services by publish/find/bind pattern (Percivall, 2003), where a service will be published to a service broker. A requestor can find metadata of the requested service at the service broker and then the requestor can bind the service at the location described in the received metadata (shown in Figure 1).

There a set of special OGC compliant web services (OWS) will be specified, e.g.:

- For calculations on very extensive spatial data the OGC specifies a standardised interface *Web Processing Services* (WPS) to offer any sort of GIS functionality, including access to calculations and/or computation models (Schut, 2005).
- *Web Coverage Services* (WCS) are specified to describe and deliver multidimensional coverage data (Whiteside et al., 2006).
- *Web Feature Services* (WFS) can provide data access functionality and operations on geographic features (Vretanos, 2005).
- *Catalogue Services* (CS-W) handle the discovery and retrieval of spatial data. Also a CS-W stores and provides metadata of services (Nebert et al., 2007).

In general an OWS is stateless. At the moment most OWS use proprietary security concepts. A web service description will be stored in capabilities. Actual the web service description can be accessed by a *GetCapabilities* and a *describeX* operation (X stands for the provided functionality), but a Web Service Description Language (WSDL) support is planned for all new standards. The discovery of services will be realised by CS-W. For messaging a combination of HTTP GET key/value pairs with different formats, like ASCII, binaries and OGC XML formats, e.g. GML, are used. (Hobona et al., 2007).
By composing chains of these services very complex sequences of calculations on spatial data can be realised.

3. THE GRID

In (Treadwell, 2007) a definition of a grid is given by the Open Grid Forum (OGF): “A grid is a system that is concerned with the integration, virtualization, and management of services and resources in a distributed, heterogeneous environment that supports collections of users and resources (virtual organizations) across traditional administrative and organizational domains (real organizations)”.

For developing a distributed system in a grid environment the Open Grid Services Architecture (OGSA) describes an architecture, based on Grid services for each resource (storage and computational units, programs, etc.). In this context a Grid service is defined as a Web service that provides a set of well-defined interfaces and that follows specific conventions. The interfaces address discovery, dynamic service creation, lifetime management, notification and manageability. Furthermore grid services address authorisation and concurrency control. (Foster et al., 2002).

Just like the SDI the OGSA grid technologies are based on the Service Oriented Architecture. So the binding of grid services is also be realised by a publish/find/bind concept. However OGSA and SDI differ in some points. OGSA requires stateful web services for realising introspection, monitoring and discovery. The service description will be realised by WSDL descriptions. In most grid environments the Simple Object Access Protocol (SOAP) will be used as messaging format. The service discovery will be realised by the Monitoring and Discovery Service (MDS) and Universal Description Discovery and Integration (UDDI). A large difference between SDI and OGSA is the security concept. While SDIs use work-in-progress security concepts, most grid environments are based on the Grid Security Infrastructure (GSI), which uses asymmetric encryption per credentials (certificates which contains a public key) and a private key for decryption. (Foster et al., 2004 / Hobona et al., 2007).

4. SCENARIO NOISE DISPERSION SIMULATION

The scenario Noise Dispersion Simulation shall be introduced at this place to give an example of a typical complex scenario which can be implemented using SDI. In the European Union all member states are obliged to provide maps showing the noise immission periodical all five years. There a data collection by measurements at all buildings and places is impracticable. Thus these maps will be generated by simulations of the noise dispersion with inclusion of noise emissions by industry and road, rail, air, and shipping traffic. The generation of such maps needs a 3D model of buildings and a Digital Terrain Model (DTM) as input for the simulation of noise immission at buildings. Additional an enrichment of these input data will be needed, with noise emission levels, traffic frequencies and some other noise related data.

Up to now National Mapping Agencies can deliver 2D geobase data, i.e. 2D digital landscape and cadastre models, 2,5D raster data DTM, and 2,5D laser scan data Digital Surface Models (DSM). Some heterogeneous data of noise emission, traffic frequencies and so on are available at different other organisations. So a possible scenario of Noise Dispersion Simulation needs some preliminary calculations for data delivery, 3D model generation, and data enrichment, realised by different OWS. Based on these OWS the Noise Dispersion Simulation can be realised by another WPS. The resulting maps can be delivered by a Web Map Service (WMS). Figure 2 shows a service chain, where a WCS provides raster data, two WFS geobase and noise emission data, and two WPS realise the 3D generation and noise dispersion simulation.

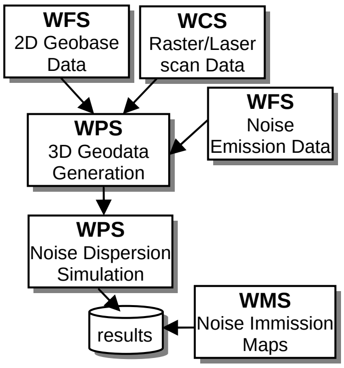


Figure 2. Service chain for Noise Dispersion and Immission Simulation

5. GRIDIFICATION OF SDI

5.1 Goals

In general gridification is the casting of existing applications and services into the framework of a grid environment. As described above the architecture of SDI and grid environments differ in some points. In geospatial context the maintaining of the SDI look and feel for users is an important aspect. Users of a SDI want to create the SDI service chain in a proven way, but they want to get the advantages of using the grid architecture. Thus in focus of a spatial data infrastructure three goals of gridification can be identified:

1. A possibility to combine one gridified OWS with other (possibly non-spatial) grid services and other gridified OWS is needed.
2. An access from or to non-gridified OWS should be possible.
3. The gridification of SDI service chains should be carried out in such a way, that a maximised performance between connected services can be reached by adequate parallelisation techniques which may address from a single service up to the entire service chain.

To reach these goals the compatibility of both networks must be realised. Thereby both a view on gridification of single services and a view on mapping complex connected service chains to a grid environment are needed.

5.2 Partitioning of Spatial Data

Several methods of partitioning exist for processing data models, whereby three of them can be identified as appropriate to processing large amounts of object-oriented spatial data:

1. Processing of data can be parallelised by object types, e.g. buildings, transportation areas, water bodies, etc.
2. Different operations can be executed on the whole dataset, distributed on different computation nodes.
3. Data tiling is an applicable partitioning method, where the whole dataset of a territory will be divided into tiles, which can be parallel processed.

In the given scenario of Noise Dispersion Simulation the data tiling is a good choice, because every service in the realised service chain will work on continuous spatial data of wide areas. Also each service has object-oriented data models as input and output. The choice for a partitioning method should be deliberate, because if the same method in several connected services of the whole service chain can be used, an equal type of gridification between these services can be tried. In this case this type of gridification can be continued through entire parts of a service chain. A problem for realising parallelised processing by data tiling is the adjustment calculation that needs calculation of overlapping border sizes for all tiles. This aspect produces additional overhead.

5.3 Grid-enabling OpenGIS Web Services

To execute an OWS or an instance of it in a grid environment some adaptations with respect to the grid architecture have to be realised:

- Required interfaces of the OGSA, e.g. *GridService*, etc., must be added to the OWS, to make it possible deploy the service to the grid environment.
- As most OWS are stateless web services, these services need to be augmented by a lifecycle management and a state for each connection.
- Since up to now WSDL is rarely used for service description of OWS, the support of a WSDL description has to be added.
- OWS need an adaptation to the grid security concept of asymmetric encryption per credentials and private key for decryption.
- The support of SOAP as message format is required.
- However the interfaces and functionality of the OWS should be maintained.

The gridification of each single service has consequences on performance of a whole service chain while executing in the grid environment. So two points are important while gridification of a single OWS:

- In order to maximise the performance of processing large amounts of spatial data the distribution of data and services within the grid has to be optimised.
- A harmonised interconnection of the gridified web services is needed to reach a high level of parallelism.

Some approaches of grid-enabling single OWS already exist. (Di et al., 2003) introduces a solution where an interface of an OWS will be realised outside of the grid environment. Inside the Grid environment a server for each web service is running, which contains the full functionality of this web service, including the data management and capabilities. Thus the interface represents a facade to the server as a grid service with its functionality. An OWS can use the grid-enabled service by accessing the interface at the SDI network.

In (Shu et al., 2006) another concept for grid-enabled OWS was introduced. This concept encapsulates OWS into a grid layer, where the adaptation to the needs of the grid environment will be realised by implementation of additional *portTypes* (interfaces). The access to OWS interfaces will be done by internal invocation of such grid services. A cascading concept enables access to non-grid-enabled OWS.

After the view on requirements for executing a single OWS in the grid environment the focus will be set on gridification of whole service chains in the next chapters.

5.4 Using a Grid Environment for SDI Service Chains

Now we will examine, how the two networks, the SDI and the grid environment, can be combined to reach high performance, on the one hand in one service and on the other hand in the whole service chain. In SDI service chains often occurrences of complex calculations on large amounts of spatial data can be identified. The use of a grid environment should distribute, parallelise, and thus speedup these complex calculations. In the following, we propose three different approaches how to map SDI structures to a grid environment.

As **first option** the grid environment can be used only for executing calculations of a number of single OWS, shown in Figure 3. There an OWS will be mapped onto grid service instances, which will be encapsulated by additional OGSA interfaces and functionalities. These grid-enabled instances will be deployed to several nodes of the grid environment. Calculations run on different grid nodes and different data portions. After processing, the results will be merged by the parent OWS in the SDI. Thus a horizontal data and information flow between both networks will be realised. With this option the complete vertical data and information flow between two connected OWS happens in a merged way in the SDI. The grid is only employed to carry out computations, but no flow of data and information is facilitated between grid services.

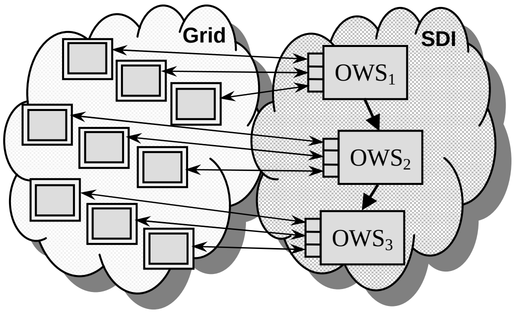


Figure 3. Using the grid only for web service calculations

For user of a SDI environment this solution brings some advantages:

- For a SDI user the handling of gridified service chains is same as the handling of non-gridified service chains.
- The orchestration of services is still conducted in the SDI framework.
- This solution enables the composition of gridified and non-gridified services, because each grid-enabled service instance can connect to other grid or grid-enabled services and the parent OWS can connect to other non-gridified services.

The partial utilisation of the grid functionalities – no distributed data and information flow can be realised – and an impossible continued type of using the grid environment through entire parts of a service chain can be named as negative aspects.

The **second option** of using the grid environment for a SDI service chain is a complete realisation as chains of pure grid services, illustrated in Figure 4. Each OWS will be implemented as a grid service, which will be deployed to several grid nodes working on different portions of the whole dataset. The connection between two services and the complete data and information flow will be realised in the grid environment, which creates a grid service chain without interconnection to the SDI network. In this solution a distributed processing can be done by redundant execution of such a service chain on different grid nodes, where each chain works on different portions of the geodata. Only a vertical data and information flow happens in the grid environment. A horizontal flow between both networks doesn't exist.

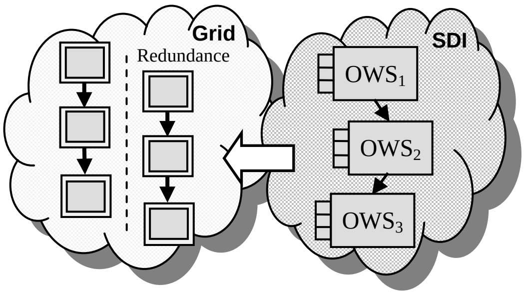


Figure 4: Realisation as Grid Service Chain

For processing of the service chain, some advantages can be identified:

- This solution allows a realisation of an equal use of the grid environment between several services, because in all cases the whole service chain, not only single services will be gridified.
- Since the processing of the service chain runs completely within the grid environment, a full use of the grid functionalities is possible.

However for users of the SDI network there are some disadvantages in this solution. It is impossible to connect a gridified with a non-gridified OWS, since a lack of an interface to the SDI network by each grid service. Also a service chain will be mapped to the grid environment in a non-transparent way, because the realisation of the service chain in the grid environment can differ from a non-gridified service chain design in the SDI.

The **third option** of using a grid environment for a SDI service chain is shown in Figure 5. It will maintain a SDI service chain while using the grid environment for calculation and distributed data transfer between connected services. Each OWS will be mapped onto identical service instances, which will be encapsulated by additional OGSA interfaces to grid service instances. These grid-enabled instances will be deployed to different grid nodes within the grid environment. So the processing of service calculations runs in a distributed way on grid nodes. The intermediate results will not be merged by the parent OWS in the SDI network, but the grid service instances connect to matching instances of the next OWS, which is gridified in the same way. Thus the data and information flow between gridified services will be realised distributed between matching service instances in the grid environment. Additional by each parent OWS should be able to merge the distributed dataset, to connect to other OWS which are non-gridified or gridified by another method over a non-distributed SDI connection. This solution realises both, a horizontal data and information flow between both networks and a vertical flow in the grid environment (possibly also in the SDI).

This solution enables a flexible gridification option with a lot of benefits:

- A continued type of using the grid environment through entire parts of a service chain is realisable, since services with equal type of partitioning can be matched by direct connection of the grid service instances.
- An interconnection between gridified and non-gridified services is realisable, where OWS chains exist in the SDI framework and grid or gridified service chains exist in the grid environment.
- The grid functionalities can be full used, since the processing and data transfer can be completely realised in the grid environment.
- The orchestration of services can be done in the SDI, so the specification of service chains doesn't change for users from the SDI domain.

This solution has a high complexity, which needs a possibility to continue one type of gridification between entire parts of a service chain to exploit all benefits.

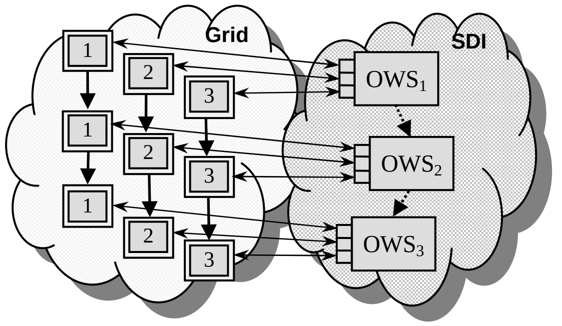


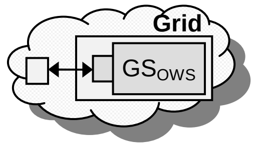
Figure 5. Using the grid for calculations and data transfer

The decision which option is the best for using a grid environment depends on the kinds of parallelisation of the composed services. If an equal type of partitioning between connected services is applicable, the third solution will most likely bring the highest performance, because this solution realises a distributed processing of calculations and a distributed data and information flow between services. But if there isn't an applicable equal type of partitioning the first option is the best choice with an independent gridification of several services. The second solution can be used if there is no requirement of maintaining an interconnection to the SDI network.

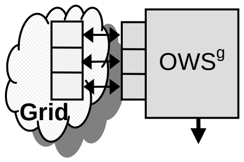
5.5 Categories of gridified Services

The three options of mapping SDI functionalities to a grid environment require the following categories for gridified services:

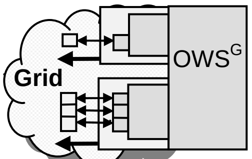
- First we can realise a gridification by grid-enabling the OWS. An OWS is realised as a grid service without connection to the SDI. A parallelisation can be realised by parallel execution of multiple instances of such a grid service.



- Next an encapsulated gridification can be identified as category for a gridified OWS. In this solution the gridified OWS consists of a SDI and a grid part. The OWS functionality will be distributed and executed by grid instances of the service. The parent SDI part collects the calculation results and transfers the whole result in the SDI world to the next service.



- Third a transcendent gridification of an OWS can be realised, where the gridified OWS also consists of a SDI and a grid part. The part residing in the SDI doesn't recollect the distributed results. The grid instances will realise the data and information flow in the grid environment by a direct connection to grid instances of another service which need the results of this part of the dataset.



5.6 Notation for gridified Services

In this section a suggestion for a notation of the above-named three different types of gridified OWS will be introduced, to formalise this distinction and understand the given figures. The first type, the gridification by grid-enabling the OWS and redundance, forms a real grid service which encloses the OWS functionality. So for such a service we suggest a denotation by the abbreviation $\mathbf{GS_{OWS}}$, where "GS" stands for "Grid Service" subscripted by the enclosed type of OWS, e.g. GS_{WCS}, GS_{WFS}, etc. The second type of gridified OWS, the encapsulated gridification, forms an OWS which encapsulates the use of the grid environment as grid service instances. Thus we decide to use the abbreviation $\mathbf{OWS^g}$. There "OWS" is a placeholder for the concrete gridified OWS superscripted by the lower case letter "g", since such a gridified service only makes limited and local use of the grid functionalities, e.g. WCS^g, WFS^g, etc. A transcendent gridified service will be denoted as $\mathbf{OWS^G}$. Such services make use of the full grid functionalities, so the concrete OWS can be superscripted by the upper case letter "G", e.g. WCS^G, WFS^G, etc. These types of services are called "transcendent", because such a service works overarching in both networks while using all features of the both worlds, the SDI world and the grid environmental world.

5.7 Gridification of Noise Dispersion Simulation Scenario

Now let's take an exemplary look at the gridification of the whole OWS chain in the Noise Dispersion Scenario. Let's assume that there is a service chain with a possibility to apply one type of partitioning through the entire service chain - the data tiling. In this case the computation of calculations will be realised at each OWS by grid service instances in an equal way at several grid nodes. Each grid service instance processes the calculations of the OWS functionality on a different data tile, e.g. a 3D geodata generation on a tile of 1x1km. Another gridified OWS provides spatial data of equal data tiles, e.g. 2D geoobjects and so on.

What happens if the encapsulated gridification will be used? The results of the computation will be merged after the finished processing at the parent part residing in the SDI and the data and information flow will be done in a merged way between the parent OWS facades at the SDI. Figure 6 illustrates this solution. For realising high performance this solution brings two problems. First bottlenecks cannot be avoided at the one centralised connection between two OWS in the SDI network. Second the unnecessary split and join operations, to create and merge tiles of the whole dataset at each OWS will bring additional overhead.

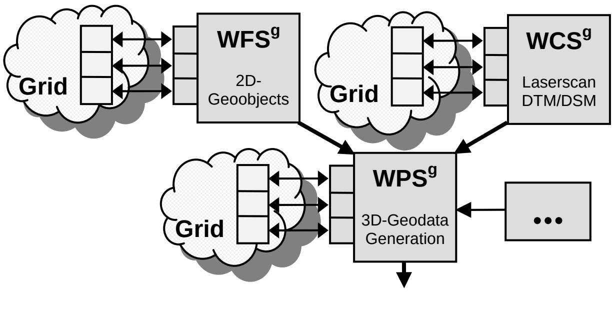


Figure 6. Excerpt of Noise Dispersion Simulation Scenario with encapsulated Gridification

In this scenario the transcendent gridification, shown in Figure 7, is the better solution, because a continuous application of one gridification type between entire parts of a service chain can be realised. Since there is a possibility of using data tiling at all OWS of the Noise Dispersion Scenario, a continuous application of data tiling between the services of this OWS chain can avoid the problems of the encapsulated gridification. Corresponding grid service instances of different services can be created, which will be directly connected for data and information transfer. Thus a distributed data and information flow over the entire service chain can be realised.

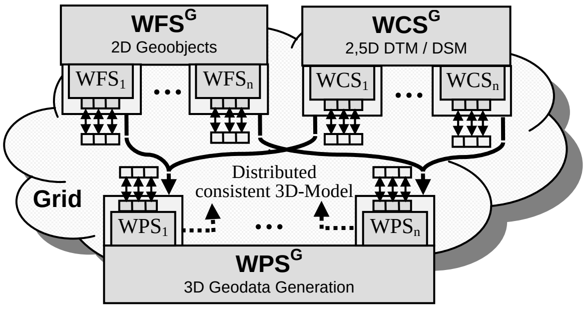


Figure 7. Excerpt of Noise Dispersion Simulation Scenario using transcendent Gridification

The benefits of using the transcendent gridification in such a scenario with a continuous application of one type of partitioning between all services are obvious. Only one split of data or calculations at the beginning and one join of the distributed data or calculations at the end of the whole service chain are required. Thus needless splits and joins can be avoided. Bottlenecks will be reduced by the distributed data and information transfer.

6. ORCHESTRATION OF SERVICES

In this chapter an approach will be given to store required metadata for an adequate orchestration of gridified OWS chains, which can automatically be done by a workflow engine. This concept is shown in Figure 8. Each gridified OWS was realised with a concrete structure of using the grid environment. So it has a gridification kind, a number of distributed grid service instances and each grid service instance works on concrete parts of the whole dataset or all needed calculations. With this information a parallelisation input and output signature can be created, which forms a unique description of the OWS gridification. The connection of two gridified OWS can be matched by these signatures. The parallelisation input signature of a subsequent OWS must be compatible with the output signature of the preceding OWS in order to realise the connection. The service endpoints of the grid service instances within the grid environment must be known to find the corresponding instances from the other OWS. If this information is stored as extension of the capabilities of each OWS, a SOA typical publish-find-bind concept can be enabled over the *getCapabilities* method of the related OWS, shown in Figure 8 for transcendent gridification. This will be done additional to the standard publish-find-bind concept of the service discovery. Each grid service instance will be published at the extended OWS capabilities. Additional managing functionality inside the OWS or extern realised as workflow engine OWS can find matching services and grid service instances by accessing the extended capabilities of the related services and comparing the parallelisation input and output signatures. The grid service instances can bind its counterparts with information about related service endpoints.

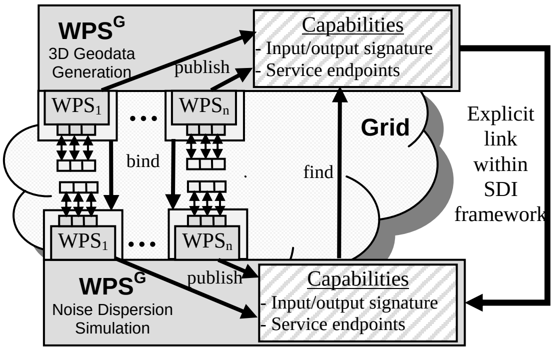


Figure 8. Metadata Storage for Orchestration of transcendently gridified OWS

As a future idea the existing workflows could be automatically generated using the additional information with a workflow engine. First a set of implemented services with annotated information at the capabilities are given to the workflow engine. Then another set of existing workflow structures for building service chains will be stored at the workflow engine. Now a workflow engine can realise a workflow by taking one workflow structure and the related gridified services, identified by the extended capabilities information. The services will be orchestrated to a gridified OWS chain and the chain will be deployed to corresponding grid nodes.

7. CONCLUSION

In this paper, three options for the utilisation of a grid environment for the implementation of SDI service chains were introduced and discussed. OWS instances must be grid-enabled to allow the execution within the OGSA. It was shown, that it’s worthwhile to find (if possible) a continuous application of one type of partitioning through entire parts of a service chain. Then a transcendent gridification can use the full features of the grid environment. For the given gridification options of OWS a classification and notation were suggested to formalise this aspect. Additionally a concept for storing additional metadata about the gridified OWS in the capabilities was given, to enable an automated orchestration and execution of matching gridified OWS to service chains.

The discussed gridification options in this paper should allow a flexible integration of grid technology in the context of OWS chains with respect to different application scenarios, e.g. the described Noise Dispersion Simulation. As a future work at least one of these concepts will be realised in a solution of SDI service chains. This will be done on a set of scenarios, which will integrate SDI and grid technologies for distributed processing of large amounts of spatial data.

Another future research field is the investigation of an appropriate formalisation for a parallelisation input/output signature with all required descriptions of the gridified OWS, which can extend the OWS capabilities. Also the extension of the capabilities by service endpoints of all grid service instances should be realised. Then a workflow engine should be developed, which can handle these extended capabilities realising an automated orchestration and execution of OWS with different types of gridification but corresponding signatures.

REFERENCES

Di, L., Chen, A., Yang, W., Zhao, P., 2003. The Integration of Grid Technology with OGC Web Services (OWS). *NWGISS for NASA EOS Data,* HPDC12 & GGF8, Seattle, USA.

Erl, T., 2005. *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River, USA.

Foster, I., Kesselman, 2004. *The Grid 2:* Blueprint for a new Computing Infrastructure. Elsevier, San Francisco, CA.

Foster, I., Kesselman, C., Nick, J.M., Tuecke, S., 2002. Grid Services for Distributed System Integration. *IEEE Computer Society Press,* Volume 35, Issue 6, pp. 37-46.

Hobona, G., Fairbairn, D., James, P., 2007. *Workflow Enactment of Grid-Enabled Geospatial Web Services*. School of Civil Engineering and Geosciences, Newcastle University, UK.

Nebert, D.D., 2004. *The SDI Cookbook*. GSDI Association. http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf (accessed 16 Apr. 08).

Nebert, D., Whiteside, A., Vretanos, P., 2007. *OpenGIS® Catalogue Services Specification* - Version 2.0.2, OpenGIS® Implementation Specification, Open Geospatial Consortium Inc.

Percivall, G., 2003. *OGC Reference Model* - Version 0.1.3., Open Geospatial Consortium Inc.

Shu, Y., Zhang, J. F., Zhou, X., 2006. A Grid-enabled Architecture for Geospatial Data Sharing. *Proceedings of the APSCC 2006*, IEEE Computer Society Press, pp. 369-375.

Schut, P., 2005. *OpenGIS® Web Processing Service* - Version 1.0.0, OpenGIS® Standard, Open Geospatial Consortium Inc.

Threadwell, J., 2007. *Open Grid Services Architecture Glossary of Terms* - Version 1.6., Open Grid Forum.

Vretanos, P.A., 2005. *Web Feature Service Implementation Specification* - Version 1.1.0., OpenGIS® Implementation Specification, Open Geospatial Consortium Inc.

Whiteside, A., Evans, J.D., 2006. *Web Coverage Service (WCS) Implementation Specification* - Version 1.1.0., OpenGIS® Implementation Specification, Open Geospatial Consortium Inc.

ACKNOWLEDGEMENTS

This research has been carried out within the framework of the GDI-Grid project as part of the German D-Grid initiative. It was funded by the German Federal Ministry of Education and Research (BMBF). We thank the BMBF for supporting our work.