DATA INTEGRATION IN A MODULAR AND PARALLEL GRID-COMPUTING WORKFLOW

S. Werder [a]

[a] Leibniz Universität Hannover, Institute of Cartography and Geoinformatics, Appelstr. 9a, 30167 Hannover, Germany - stefan.werder@ikg.uni-hannover.de

Commission IV, WG IV/5

KEY WORDS: GIS, Integration, Spatial Infrastructures, Distributed, Processing, Services, Performance, Vector

ABSTRACT:

In the past decades a wide range of complex processes have been developed to solve specific geospatial data integration problems. As a drawback these complex processes are often not sufficiently transferable and interoperable. We propose modularisation of the whole data integration process into reusable, exchangeable, and multi-purpose web services to overcome these drawbacks. Both a high-level split of the process into subsequent modules such as pre-processing and feature matching is discussed as well as another fine-granular split within these modules. Thereby complex integration problems can be addressed by chaining selected services as part of a geo-processing workflow. Parallelization is needed for processing massive amounts of data or complex algorithms. In this paper the two concepts of task and data parallelization are compared and examples for their usage are given. The presented work provides vector data integration within grid-computing workflows of the German Spatial Data Infrastructure Grid (SDI-Grid) project.

1. INTRODUCTION

The object of data integration is to merge information from different data sources. It has to cope with data heterogeneity originating from differences in data models, acquisition time, resolution, and quality.

Manifold aspects of data integration have been tackled by researchers in the past decades. One example for that statement is the progress in data integration of road networks, of which some research activities are mentioned in the following. Lynch and Saalfeld (1985) describe an interactive system for semi-automatic integration of road maps from the United States Geological Survey and maps from the Bureau of the Census. Automatic matching of roads from cadastral and car navigation datasets incorporating statistical investigations and concepts from information theory has been carried out by Walter and Fritsch (1999). For the same two dataset types Volz (2006) used an iterative matching approach which starts at seed nodes and constantly grows the matched road network. Zhang and Meng (2007) also use an iterative approach but additionally consider unsymmetrical buffer sizes. Networks with different level of detail are matched by a rough search for candidate matches followed by a more detailed search in the work of Mustière and Devogele (2008).

The above listed research on data integration of road networks is far from being complete. Nevertheless, it is sufficient to draw two simple but important conclusions. Firstly, all researchers bring in new ideas, concepts, and approaches for the same task, even if they use similar datasets. Secondly, some parts of their research overlap, which poses a great opportunity for reusing or sharing findings and actual implementations.

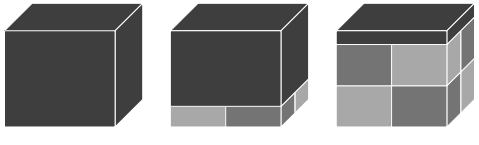
However, the achieved results and progress are often neither easily transferable nor interoperable. Transferability may be restricted by processes that are tailored to specific types of datasets. Also hard-coded values of algorithm parameters complicate the application of data integration software on comparable tasks. Interoperability may be restricted due to missing interfaces and not publicly available user documentation. Another issue is the dependency on specific frameworks and proprietary software packages or libraries.

In table 1 the software frameworks used in the aforementioned data integration research activities are listed. These can be grouped into three categories, namely proprietary, commercial and open source. When sorted by publication date, the first two research works use proprietary respectively no frameworks. This may be due to the fact, that at this time none or only little available frameworks existed for geo-computation. Zhang and Meng (2007) used ArcGIS, which is a commercial framework. Both JUMP and GeOxygene are open source frameworks.

Framework	Research work
None/Proprietary	Lynch and Saalfeld (1985), Walter and Fritsch (1999)
JUMP	Volz (2006)
ArcGIS	Zhang and Meng (2007)
GeOxygene	Mustière and Devogele (2008)

Table 1. Frameworks used in presented research work

We propose modularisation of the whole data integration process into reusable, exchangeable, and multi-purpose web services to overcome these drawbacks. In figure 1 the software architecture of web services (c) is compared to other commonly used architectures.



(a) monolithic (b) plug-in (c) services

Figure 1. System architectures

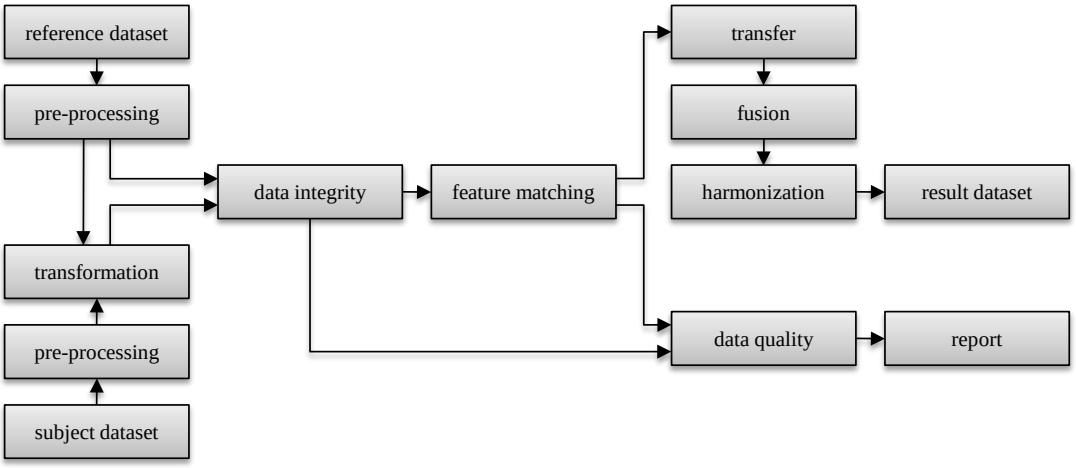


Figure 2. Data integration workflow

In a monolithic software architecture, as shown in figure 1 (a), all application logic is wrapped in a single program. Changes or extensions to this program are only possible if access to the source code is provided. This architecture represents the none or proprietary framework from table 1.

In a plug-in software architecture, as shown in figure 1 (b), additional functionality can be added to an existing program. However, the plug-ins are in most cases limited to usage in a single host software. All other frameworks from table 1 fall into this category.

In figure 1 (c) a software architecture based on loosely coupled services is shown, which is the most flexible of the three software architectures. The program logic resides in small modules respectively web services. Both program and data flow are orchestrated by a lightweight workflow engine, which represents the remaining block of fix program logic in figure 1 (c). Each module can be replaced in favour of a module with comparable output definition. Also the workflow engine itself can be exchanged.

Workflows in data integration are nothing new. However, an implementation as reusable, exchangeable, and multi-purpose web services is new to the authors knowledge. Reusable refers here to services that are limited as little as possible, in order to gain multipurpose data integration tools.

In contrast to the current research state of web services for data integration, research on web service based generalization frameworks has been carried out since the GiMoDig project (Sester et al., 2005). Foerster et al. (2010) summarize the more recent research aspects. Web processing services (WPS) as defined by the Open Geospatial Consortium (2007) play an important role in the developed concept.

In the remainder of this paper, splitting data integration into a modular workflow is discussed in section 2. The actual implementation of selected data integration modules as well as parallelization aspects are presented in section 3. The presented work is part of the German Spatial Data Infrastructure Grid (SDI-Grid) project. Using these data integration modules as part of grid-computing workflow is therefore shown in the last section 4.

2. DATA INTEGRATION WORKFLOW

The term data integration is often used as a synonym to the term conflation. The origin of the term conflation is explained by Lynch and Saalfeld (1985). There conflation of maps is defined as "combining of two digital map files to produce a third map file which is 'better' than each of the component source maps" (Lynch and Saalfeld, 1985). However, we use the term data integration for two reasons. Firstly, data integration is not a technical term and therefore can be understood also by non-experts. Secondly, the term conflation is used in publications for different processes. In some it denotes the whole process as shown in figure 2, whereas in others it refers to one of the listed modules.

In the following, both a high-level split of the data integration process into subsequent modules as well as another fine-granular split within these modules is discussed. The modules as well as the workflow composition are also compared to their definition in related publications. Unfortunately, few publications refer to the whole workflow instead of presenting innovations of some modules. Yuan and Tao (1999) use the ESRI MapObjects and the respective Active-X framework to build one component offering process logic for data integration of polygons. Davis (2003) describes the open source Java Conflation Suite (JCS), which has been developed by the company Vivid Solutions.

The data integration workflow starts with one reference and one subject dataset. If several datasets exist, these can be integrated piecewise. In comparison to the subject dataset, the reference dataset is of higher quality, which can be characterized by superior accuracy, reliability or actuality. However, the subject dataset contains information that is not present in the reference dataset, which can be characterized by additional features or more detailed attribute information.

Data integration starts with *pre-processing*, which maximizes the similarity of the input datasets. It ensures that "the data sets have a same data format, the same map projection, the same coordinate system" (Yuan and Tao, 1999). If the datasets differ significantly in scale, generalization increases the comparability (Sester et al., 1998). All these processing steps can be split into single services, in order to support fine granularity and to increase reusability, e.g. a service changing the format of a dataset can be useful in most geo-processing workflows. All of

the mentioned pre-processing steps are optional if the datasets already comply to the defined requirements. This applies also to other modules in the workflow, which are only incorporated in a specific workflow if their functionality is explicitly needed.

Transformation of the subject dataset is necessary if its features show systematic differences in position and or orientation in comparison to the reference dataset. This task is named "map alignment" by Yuan and Tao (1999) and "dataset alignment" in JCS (Davis, 2003). Commonly used are affine and bilinear interpolated transformations. The latter is also known as rubber-sheeting. Transformation parameters can be calculated based on manually defined control points or by simple automatic feature matching algorithms using only strong feature matches, e.g. only 4-way road intersections in both datasets.

Checking *data integrity* ensures the mutual integrity of both datasets. In this module features are checked for their validity, with e.g. constraints on attribute value ranges or topology. The single module in figure 2 itself can be split into four more fine-granular modules, as shown in figure 3. Data enrichment calculates values based on attributes (e.g. naming scheme), geometry (e.g. area), and topology (e.g. disjoint relation). These derived values are then compared to user defined reference value ranges. If features fail, the subsequent module either separates the invalid features from the valid ones with a filter or applies automatic corrections. Also a report can be generated, which documents values and integrity state of examined features. The data integrity task is called data quality assurance by Davis (2003). Yuan and Tao (1999) didn't include this process in their workflow.

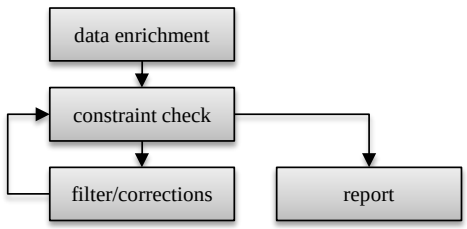


Figure 3. Data integrity (fine-granular split)

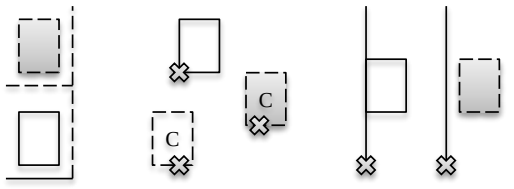
Feature matching is the most important workflow module. According to Yuan and Tao (1999) it's task is to "recognize the spatial feature correspondence of two data sets by criteria such as nearest distance, or topological similarity and associated attributes". The methods and algorithms used differ for the three geometry types point, line and area. Commonly used measures are based on distance, topology and semantic similarity between a candidate match pair. Thereby matches can occur that refer one feature from the subject dataset to exactly one feature of the reference dataset, which is called a 1:1 match. But also 1:n or n:m matches can occur. Most of the road network integration research that we referred to in the introduction, starts with matching nodes of two networks, followed by edge matching, e.g. as described by Stigmar (2005). The logic therefore can be also split from a high-level module into several fine-granular modules.

Two other important aspects of the data integration workflow can be shown exemplarily for the feature matching process. Firstly, data integration often incorporates iterative tasks, which either start with a small subset being increased in each step, or by narrowing down a coarse grained solution to a more detailed or strict one. Secondly, "due to the complexity and the inconsistency of spatial feature representations and technology limitations, human intervention is required [...] to deal with uncertainty or mismatching" (Yuan and Tao, 1999). Although user interaction is only incorporated by Yuan and Tao (1999) in a post-processing task after feature matching, it is useful also for other data integration tasks.

The *data quality* module is neither represented in the workflow of Yuan and Tao (1999) nor in the JCS workflow of Davis (2003). Nevertheless it is an important side-product, because all required input information is already available at this point of the workflow. The International Organization for Standard-ization (2002a) defines quality is defined as "totality of characteristics of a product that bear on its ability to satisfy stated and implied needs". Due to high acquisition costs, availability of geospatial data over the internet and usage of data in domains that were not envisioned upon data capture, geospatial datasets are used nowadays by a wider audience and in manifold disciplines. Judging the fitness for use of a specific dataset for a specific application therefore lies normally in the responsibility of the data user, and not of the data producer.

Some of the data quality elements described in the ISO 19113 norm (ISO, 2002b) can be determined based on the results of the data integrity and feature matching modules. *Completeness* measures can be derived from feature matching results. Excess data, i.e. features in the subject dataset that have no correspondence in the reference dataset, are part of the commission set. Vice versa, absent data, i.e. features existing only in the reference dataset, are part of the omission set. For the quality element *logical consistency* adherence to the rules of the conceptual schema and adherence to values of the value domains as well as topological consistency (ISO, 2002b) can be (partly) checked by the data integrity module. Absolute or external accuracy, which is a measure for *positional accuracy* (ISO, 2002b), can be checked for a subject dataset against a reference dataset by evaluating the results from the feature matching module. For measuring temporal accuracy an extension to 4D is required, which is not in the scope of this paper. *Thematic accuracy* includes classification correctness, which can be measured by "comparison of the classes assigned to features or their attributes to [...] e.g. reference dataset" (ISO, 2002b). Therefore this measure is also a side-product of the feature matching module.

The results of feature matching provide the base for the following modules which are also shown in figure 2. Although both Yuan and Tao (1999) as well as Davis (2003) combine the last step of their workflows into one module, we choose to separate it into the three modules *transfer*, *fusion* and *harmonization*. The main reason for the separation into these information aggregation methods is the need for classifying data integration research outcomes by their result or respectively by the selected methods used to reach this goal.



(a) transfer (b) fusion (c) harmonization

Figure 4. Methods for information aggregation

Transfer copies features that are only present in the subject dataset, but not in the reference dataset, into the latter. An example for that constellation are newly constructed buildings or transport areas. This example is shown in figure 4 (a). There features with a dotted line or border originate from the subject dataset and are copied into the reference dataset, which is characterized by solid lines or borders. Another example is the data integration of a road network from one data source with buildings from another.

Fusion combines the information of features being present in both reference and subject dataset. The combination can be based on a geometric mean of the features' geometries or any other weighting of them. Thereby one of the geometries may be preferred due to higher accuracy or newer acquisition date. Fusion also covers information aggregation based on attributes of corresponding features. In figure 4 (b) the cross denotes a fix origin. The building with the solid line is from the reference dataset, whereas the building with the dashed line and the attribute value "C" is from the subject dataset. The grey filled building on the right shows the fusion result. The result is characterized by an average geometry and the attribute value "C" originating from the subject dataset. Another approach for weighting geometries is discussed by Butenuth et al. (2007).

Harmonization enforces or restores constraints that have to be satisfied at all times. Constraints can be defined for individual features, but also between features of the same or different feature types. For more information on constraints see Werder (2009). The constraints are enforced in a way that an optimal solution for the whole dataset is achieved, e.g. by relying on optimization techniques or adjustment theory. The example in figure 4 (c) shows at the left side a building that touches the road, which would lead to an invalid dataset. Therefore in the result dataset the building is shifted away from the road.

Based on the introduced three modules, the step "discrepancy correction or information transferring" in the workflow of Yuan and Tao (1999) considers only the fusion aspect. The step "geometry alignment and/or information transfer" in the workflow of Davis (2003) considers only the fusion as well as the transfer aspect.

3. MODULES AND PARALLELIZATION

In this chapter the modules that have been actually developed as part of the SDI-Grid project are presented. The input datasets for data integration are both road networks. Whereas in a first prototype a cadastral dataset from a German National Mapping Agency was used as reference dataset, it was replaced with freely available geographic data from the OpenStreetMap project (openstreetmap.org) in the final prototype. The subject road dataset is of lower geometric accuracy but includes several attributes being relevant for computing detailed noise propagation simulations. The attributes of the road lines in the subject dataset provide information about e.g. average number of vehicles in the time slots day, evening and night, total percentage of trucks, and road surface material.

Concerning parallelization, the two concepts of *task parallelism* and *data parallelism* are implemented in the modules of the prototype and in the workflow respectively. The term task parallelism stands for several independent subtasks that operate on the same or different datasets. The term data parallelism denotes subtasks that each process a part of the whole dataset. More information about the two concepts can be found in the paper of Werder and Krüger (2009).

The first high-level module in the data integration workflow shown in figure 2 is *pre-processing*. In the following figure 5 the process is shown with its fine-granular services. For the sake of clarity not all actually implemented services are shown in the figure. The workflow implements task parallelism, because both datasets are processed at the same time. The first service creates an additional attribute holding unique ids for each feature. The second services repair inconsistencies concerning the noding in the road networks with different logic for each dataset. The third service repairs over- and undershoots in the network. So far the services of the reference and subject dataset run independently from each other. The fourth service in figure 5 identifies (road) lines that are present in one dataset, but not in the other. For this computation the results from the previous over- and undershoot services have to be joined. Therefore the standalone line services can start with the computation not before the previous services succeeded.

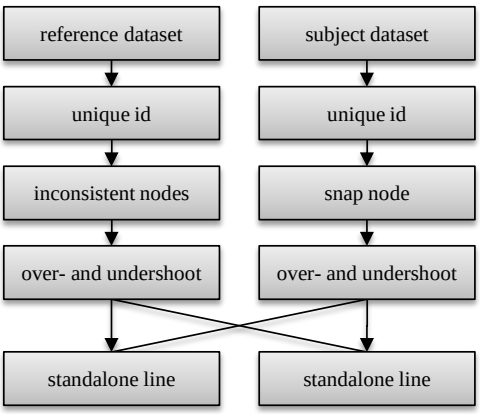


Figure 5. Pre-processing of road networks

The implemented services are based on algorithms from the Java Topology Suite (tsusiatsoftware.net), the OpenJUMP project (openjump.org), GeoTools (geotools.org) and the Road Matcher (www.vividsolutions.com) developed by the Canadian company Vivid Solutions. Existing concepts were used when possible, but also additional logic was added to provide the necessary algorithms for data integration tasks.

For example, the logic for affine and bilinear interpolated *transformations*, which represent the transformation modules in figure 2, were based on the source code of both the OpenJUMP project as well as the Java Topology Suite. Nevertheless, the provided source code had to be cleaned and some additional wrapper logic had to be provided in order to encapsulate the functionality in the two service.

The *data integrity* module is still work in progress, but it is already able to process constraints on single features as well as constraints on feature classes and relations. Its results will be published as a dissertation.

The *feature matching* module is based on the Road Matcher from Vivid Solutions, which is also used in the work of Stigmar (2005). Because only the attribute information from the subject dataset has to be transferred to the reference dataset, only the fusion module is needed as part of the data integration workflow. This confirms the statement from section 2 that only

modules that are explicitly needed are finally part of a specific workflow.

So far only independent subtasks operating on different datasets have been presented in this section, such as the tasks shown in figure 5. The concept of task parallelism however also includes tasks operating on the same dataset. In order to discuss this issue, an extension of the workflow is shortly discussed in the following.

In order to improve feature matching results, we extended the workflow in our project by a module performing shape recognition. This module provides beside others information about parallel lines and circles in road datasets and is executed prior to the feature matching module. The identification of circles uses a variation of the Random Sample Consensus algorithm, which was first published by Fischler and Bolles (1981). This algorithm is robust, but also uses a high number of iterations which make it more time intensive than other modules. Therefore the module is implemented using the individual CPUs of a multi-core processor at the same time. Each CPU is calculating a fraction of the road lines, e.g. 25 percent if four CPUs are used. The individual results are then combined into a single report.

Task parallelism performs best for simple and fast algorithms, such as the presented creation of unique ids. Also complex algorithms perform well, if only the CPU limits the speed. If however input and output operations, such as reading and writing data, dominate the execution time, then data parallelism should definitely be considered. The module transformation is a good example for data parallelism. Massive datasets covering e.g. complete countries can be split into several partitions, which are then each processed by a different computer. Finally, the results are collected and merged again into a single dataset.

The developed *partitioning* service offers different strategies for dividing features into corresponding partitions, which are shown in figure 6. The strategy type (a) collects all features which bounding box lies inside the partition. This leads to duplicate features if individual bounding boxes are covered by more than one partition, e.g. some features are both in partition 1 and 2. In contrast, strategy type (b) creates no duplicates, because it collects all features which centroids lie inside a partition. Strategy type (c) simply cuts the feature geometry at partition borders, which is suitable especially for lines.

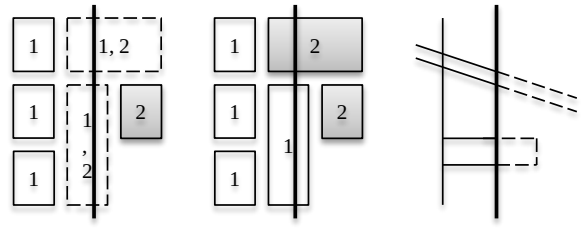


Figure 6. Partitioning types

4. GRID-COMPUTING WORKFLOW

According to the checklist of Foster (2002) grid-computing combines decentralized resources for collaborative utilization using standard protocols and interfaces.

Within the SDI-Grid project a powerful workflow engine has been developed, which facilitates the creation of grid jobs to a high degree (Fleuren and Müller, 2008). The workflow engine is shown in figure 7. It is able to call the commonly used web services defined by the Open Geospatial Consortium, including the Web Processing Service (OGC, 2007), through the so-called OGC proxy. Being based on the Business Process Execution Language (BPEL), the workflow engine is also able to trigger the execution of standard web services outside the grid.

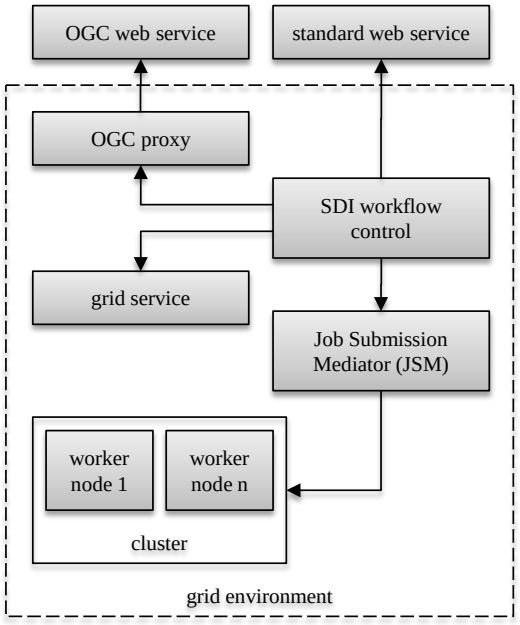


Figure 7. Overview of SDI-Grid workflow engine

The most important innovation is however the Job Submission Mediator (JSM). It enables and simplifies the execution and management of jobs containing legacy applications on the grid cluster. Therefore the JSM is used to execute the presented data integration modules as grid jobs on several worker nodes of the cluster. These jobs are also provided as individual web services to grid clients.

In order to execute the developed data integration modules as grid services three requirements have to be met. Firstly, the modules must be executable in the environment that is provided by the worker nodes. The operating system of the worker nodes is Scientific Linux, which posed no problems because all modules are implemented in the Java programming language. Secondly, templates for job descriptions have to created. These job descriptions control which program is executed, which input files have to be transferred to the cluster via secure grid file transfer, and which output files have to be copied to which destination. The templates contain variables, e.g. ${OUT.PATH _NAME}/ ${OUT.REPORT_NAME}.xml. In order to be able to create a job description for a specific set of input files and a specific job configuration, a service has been implemented that is able to copy a job description template and replace all variables with the respective values, e.g. with the actual path and file name of the output files. Variable substitution is also used for controlling the data integration logic, e.g. for defining the distance tolerance for the determination of standalone lines. This is possible because all developed integration modules can be controlled using XML configuration files.

For more details about the SDI-Grid workflow engine see Fleuren et al. (2010).

5. CONCLUSIONS

In this paper a modular approach to data integration was proposed. Reusable, exchangeable, and multi-purpose web services allow for building complex workflows tailored to the specific needs a data integration problem. Also the important side product data quality was introduced into the workflow. The definitions of the processes transfer, fusion, and harmonization make the classification of research outcomes and selected methods more consistent. By creating fine-granular modules and using parallelization both massive amounts of data and complex algorithms can be handled by data integration. Grid-computing offers the collaborative utilization of resources and therefore access to computation power even for big areas or especially complex algorithms. The workflow engine developed as part of the SDI-Grid project simplifies the execution of legacy applications on the worker nodes of grid clusters.

ACKNOWLEDGEMENTS

The research described in this paper is part of the GDI-Grid project (Spatial Data Infrastructure Grid) funded by the German Federal Ministry of Education and Research (BMBF). The support is gratefully acknowledged.

REFERENCES

Butenuth, M., v. Gösseln, G., Tiedge, M., Heipke, C., Lipeck, U. and Sester, M., 2007. Integration of Heterogeneous Geospatial Data in a Federated Database. *ISPRS Journal of Photogrammetry & Remote Sensing 62*, pp. 328-346.

Davis, M., 2003. Java Conflation Suite: Technical Report. http://www.vividsolutions.com/JCS/bin/JCS%20Technical%20Report.pdf (accessed 2. Jun. 2010).

Fischler, M. and Bolles, R., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), pp. 381-395.

Fleuren, T., Braune, S., Kunz, C. and Rüttgers, S., 2010. Meilenstein 32: Implementierung des Gesamtsystems (in German). http://www.gdi-grid.de/index.php?id=73 (accessed 2. Jun. 2010).

Fleuren, T. and Müller, P., 2008. BPEL Workflows Combining Standard OGCWeb Services and Grid-enabled OGC Web Services. In: *Proc. 34th Euromicro Conference on Software Engineering and Advanced Applications.*

Foerster, T., Lehto, L., Sarjakoski, T., Sarjakoski, L.T. and Stoter, J., 2010. Map generalization and schema transformation of geospatial data combined in a Web Service context. *Computers, Environment and Urban Systems*, 34(1), pp. 79-88.

Foster, I., 2002. What is the Grid? A Three Point Checklist. *GridToday*, 1(6).

International Organization for Standardization, 2002a. ISO 19101:2002: Geographic information – Reference model. 42 p.

International Organization for Standardization, 2002b. ISO 19113:2002: Geographic information – Quality principles. 29 p.

Lynch, M. and Saalfeld, A., 1985. Conflation: Automated Map Compilation - A Video Game Approach. In: *Proc. Auto-Carto VII*, pp. 343-352.

Mustière, S. and Devogele, T., 2008. Matching Networks with Different Levels of Detail. *GeoInformatica*, 12(4), pp. 435-453.

Open Geospatial Consortium Inc. (OGC), 2007. OpenGIS Web Processing Service, Version: 1.0.0, OGC 05-007r7.

Sester, M., Anders, K.-H. and Walter V., 1998. Linking Objects of Different Spatial Data Sets by Integration and Aggregation. *Geoinformatica*, 2(4), pp. 335-358 .

Sester, M., Sarjakoski, L.T., Harrie, L., Hampe, M., Koivula, T., Sarjakoski, T., Lehto, L., Elias, B., Nivala, A.-M. and Stigmar, H., 2005. Real-time Generalisation and Multiple Representation in the GiMoDig Mobile Service. Public deliverables D4.4.1, D7.1.1, D7.2.1, D7.3.1 and D1.2.31. http://gimodig.fgi.fi/deliverables.php (accessed 2. Jun. 2010).

Stigmar, H., 2005. Matching Route Data and Topographic Data in a Real-Time Environment. In: *Proc. ScanGIS'2005*, pp. 89-107.

Volz, S., 2006. An Iterative Approach for Matching Multiple Representations of Street Data. In: *Proc. of the JOINT ISPRS Workshop on Multiple Representations and Interoperability of Spatial Data*, pp. 101-110.

Walter, V. and Fritsch, D., 1999. Matching Spatial Data Sets: a Statistical Approach. *International Journal of Geographical Information Systems (IJGIS)*, 13(5), pp. 445-473.

Werder, S., 2009. Formalization of Spatial Constraints. In: *Proc. 12th AGILE Conference on GIScience*, 13 p.

Werder, S. and Krüger, A., 2009. Parallelizing Geospatial Tasks in Grid Computing. *GIS.SCIENCE*, 3, pp. 71-76.

Yuan, S. and Tao, C., 1999. Development of Conflation Components. In: *Proc. Geoinformatics '99*, pp. 1-13.

Zhang, M. and Meng, L., 2007. An iterative road-matching approach for the integration of postal data. *Computers, Environment and Urban Systems*, 31(5), pp. 597- 615.