In Proc. of the 20th Int. Workshop: Intelligent Computing in Engineering, 2013

Planning Inner-City-Railway-Tracks: Dynamic Integration of Geospatial Web Services in a Collaborative Multi-Scale Geometry Modelling Environment

H. Steuer, A. Donaubauer, T.H. Kolbe
Chair of Geoinformatics, Technische Universität München, Germany
M. Flurl, R.-P. Mundani, E. Rank
Chair for Computation in Engineering, Technische Universität München, Germany
steuer@tum.de

Abstract. Planning inner-city-railway-tracks is an interdisciplinary and highly complex task that involves plenty of different stakeholders. Currently, the different planners work more or less separated in an asynchronous manner. To facilitate a collaborative planning process between these different stakeholders we developed a collaboration platform. Clearly, the integration of geographical information and geoprocessing results into the planning process respectively the different modelling tools, improves this process in a significant way. In this paper, we will show how to describe the needed geographical information in a suitable way and how to integrate these pieces of information into the different planning tools via the collaboration platform in a unified, dynamic, and generic way.

1. Introduction

Naturally, the highly complex process of planning inner-city-railway-tracks involves plenty of different stakeholders from different domains such as civil engineers, subsoil, environment, and fire safety experts. These different adepts often use their own specific geometric modelling respectively planning tools thereby creating their own specific data in its own proprietary data structure. Thus, the different stakeholders work more or less separated in a not very contiguous manner. To combine, evaluate, and revise their specific work, the created data is exchanged between the different adepts in a file-based fashion. Normally, the results of this evaluation and revision process are discussed in personal meetings or teleconferences resulting in an iterative process. This kind of collaboration obviously extends the duration of the planning process in a significant and undesired way.

Clearly, in the process of planning inner-city-carriageways geographic information is of major importance. For instance, to plan the concrete track course one needs information about the soil conditions, the sub terrestrial buildings, the sewage water system, etc. Thanks to spatial data infrastructure initiatives like INSPIRE[1] these pieces of information are or at least will be available in the near future provided by certain web services. But even though these pieces of information are available in principle, at the moment, the sheer integration of this information into the common modelling respectively planning tools is not facilitated in a convenient way. Additionally, there are at least three further problems concerning these web services: Different services are hosted by different dispersed servers. Different services provide the needed data, but they provide this data in diverging scales. Furthermore, standardised web service interfaces have limitations

1Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE) compared to the requirements of a synchronous collaborative planning scenario (see Breunig, et al., 2011).

In order to overcome the problems described above we developed a collaboration platform. This platform facilitates a synchronous modelling and planning process between modelling engineers and Geographic Information System (GIS) experts. The basis for the synchronous modelling process is a so-called procedural geometry model, which describes the geometry by its several construction steps in contrary to an explicit representation. This procedural model supports the concept of different Levels of detail (LoDs). Different LoDs enable different planners to work in the abstraction level of their specific task, thereby using the geometry modelling tools they are accustomed to. Additionally, we developed a component structure that allows the integration of different data sources into the different planning tools via the collaboration platform. To integrate GIS data into the planning process respectively the planning tools, we investigated the description of collections of Geo Web Services in so-called Geospatial Web Service Context Documents.

In this paper, we will describe the principle idea of Geospatial Web Service Context Documents. Secondly, we will illustrate the integration of web services using these documents and our component structure into the several planning tools. Thereby, we will show that this integration process is facilitated in a convenient, i.e. a unified, dynamic, and generic way.

2. Related Work

2.1 Collaborative Geometric Modelling

For a long time, collaborative work in the field of geometric modelling has been in the focus of scientific research. Bidarra, van der Berg, and Bronsvoort (2002) presented the Alibre Design 2D/3D CAD software. In the client server structure called webSPIFF a central server hosts the geometric model, while engineers using a webSPIFF client can work synchronously on this model. Though this approach facilitates synchronous work using the webSPIFF clients, it doesn't support the usage of common CAD software tools, such as Siemens NX, Autodesk Inventor. Li, Gao, and Wang (2007) developed an approach for synchronous modelling using neutral modelling commands. Vendor specific modelling commands are translated into neutral commands, sent to a central collaboration server that forwards these neutral commands to the other clients. These clients then translate the neutral commands back into their own vendor specific commands. This approach is similar to ours but does not support the concept of different levels of detail, a major improvement in our concept. Additionally, since the context of our approach is the planning process of inner-city carriage way, we extend the normally used modelling functions by tunnel specific modelling commands. the Sun, et al. (1996) provided a consistency model for real-time cooperative editing systems, thereby providing a convergence scheme for causality and intention preservation. Tang, Chou, and Dong (2007) researched and classified conflict situations in the field of collaborative modelling und suggested strategies to solve these situations. Fan and Sun (2012) presented a dependency-based automatic locking scheme to facilitate real-time collaborative programming work. Cai, et al. (2012) researched a flexible concurrency structure in the field of synchronous geometric modelling. Thereby, a collaborative feature dependency graph allows locking strategies to ensure the model consistency. The principal idea of this locking scheme seems very natural but is not satisfying in our context, since a modification of a low level element may lock the whole model or at least large parts of it. Borrmann (2008) provided the CoCoS platform in order to incorporate simulation data in a collaborative planning process.

Bormann, Ji, and Jubierre (2012) created a multi-scale geometry model, thereby providing possibilities to work in different levels of detail, while the system automatically preserves the consistency of the model.

2.2 Collaborative Use of Geospatial Information

With the upcoming of OGC (Open Geospatial Consortium) web service standards fine-grained access to geographic information has been introduced to GIS. Instead of retrieving files with possible large geographic extent and potentially containing many undesired features (in geoinformatics the term feature denotes an abstraction of real-world phenomena; see ISO 19109:2006) users can access exactly the desired features in a given geographic region by specifying spatial, logical or temporal filters. In addition to accessing geospatial data, users may also access data processing functions through OGC web service interfaces. In this paper, we will address the following OGC web services: the Web Feature Service[2] (WFS), Web Map Service[3] (WMS), and Web Processing Service[4] (WPS).

WFS enables access to GML (Geography Markup Language) data (ISO 19136:2007) whose main geometry object types are boundary representations: points, lineStrings, surfaces, and solids. If a WFS additionally allows the optional write-access to its features it is usually called WFS-T (Transactional WFS).

WMS are designed not to access geographic data themselves but a rendered picture in formats such as PNG or JPEG. The standard allows for an optional operation to access further information about particular features shown on a map. WMS do not allow write-access on the underlying data.

WPS implementations provide arbitrarily complex geospatial processes and can in principle work on any type of data. Since geo processing libraries usually do not support parametric geometry models as used in common CAD tools an integration of these different geometry models remains a challenge.

Collaborative use of geodata has been focus of scientific work in the past. Neis, Singler, and Zipf (2010) present an approach to collaborative mapping for emergency management while Klimke and Döllner (2010b) introduce an approach to collaborative work within 3D city models. Both approaches singularly rely on the OGC standard WFS-T (transactional Web Feature Service) for collaborative work on a single resource of vector data (in contrast to raster data). For collaborative work with geodata from different web services the Web Map Context[5] (WMC) standard has been introduced as early as 2003. Despite it being limited to rendered maps (in particular WMS) it is applied in approaches such as presented by Smolders, et al. (2011) to integrate several results of a single analysis into one Web Map Context Document.

Since 2004 the OGC works on a standard for a Context Document which shall reference all OGC web service standards (besides WMS and WFS also OGC Web Processing Services (WPS)) but as of now has not published a final document. In the following section we show our approach to a

[2] OpenGIS Web Feature Service 2.0 Interface Standard, Version 2.0.0 OGC Document 09-025r1 and ISO/DIS 19142. http://www.opengeospatial.org/standards/wfs
[3] OpenGIS Web Map Server Implementation Specification, Version 1.3.0 OGC Document 06-042. http://www.opengeospatial.org/standards/wms
[4] OpenGIS Web Processing Service, Version 1.0.0 OGC Document 05-007r7. http://www.opengeospatial.org/standards/wps
[5] Web Map Context Documents, Version 1.1.0 OGC Document 05-005 http://www.opengeospatial.org/standards/wmc

Geospatial Context Document which is based on the work of the OGC but supports several additional features.

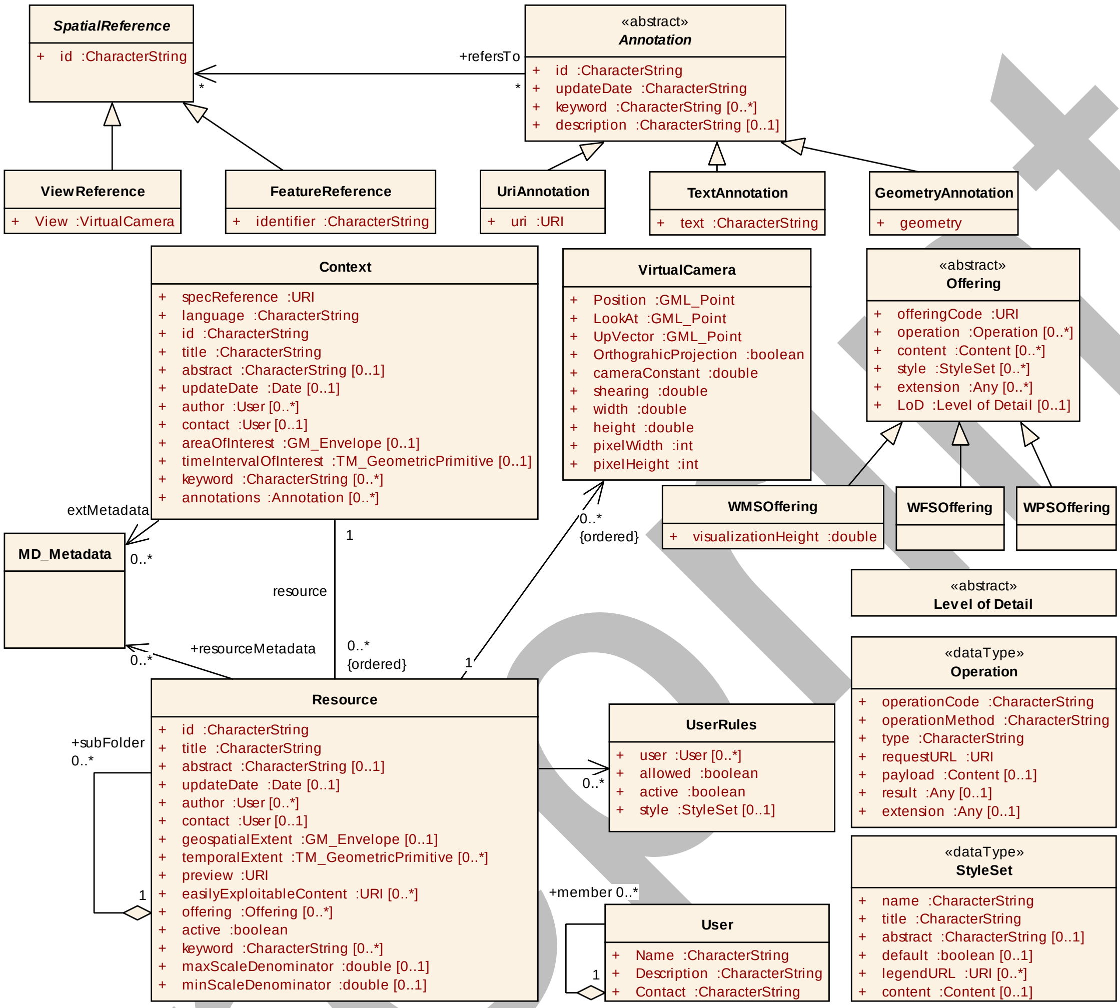

Figure 1: UML class diagram model of the Geospatial Web Service Context Document

3. Geospatial Web Service Context Document.

Due to their nature as web services WMS and WFS offer themselves as tools for collaborative work with geographic information. Each instance of these web services can potentially provide large amounts of data, many different layers (e.g. an official WMS in our test-region offers following layers: areal image, development plan, wrap-around of development plans, and a hybrid representation of aerial image and development plan) or features (our WFS instance offers buildings, their walls, roofs, and ground surfaces as well as a digital terrain model and geometry of the sewage system) for large geographical extents. These web services offer an operation which delivers metadata about the underlying layers respectively features and there exists an OGC specification for Catalogue Services which are helpful for discovery of web services. The metadata always describes these layers/feature classes as a whole. In order to access only relevant subsets filter expressions have to be encoded in the request to the web service. In order to make collaborative use of information about which data is relevant, these requests have to be communicated by the stakeholders. Since the communication of multiple separate requests to single web services can quickly become unmanageable additional mechanisms have to be used.

Our solution to communicate relevant geographic information (and in future geospatial processes) is an XML-encoded Geospatial Web Service Context Document which at its core bundles sets of requests to web services (classes Resource and Offering in figure 1). One such Geospatial Document contains all relevant information for analysis of one specific geospatial problem including additional information as we describe below.

We chose not to use the approach which has been introduced by the OGC Web Map Context Document: describing the parameters of the request (e.g. spatial extent, layers and styles) in an XML document. Using this approach implies that any client has to be equipped with a capability to rebuild the original request using these parameters. It has the disadvantage that the context document definition has to be extended for each additional web service type and with it the clients capabilities to handle this document.

Instead we allow for lean clients by encoding the complete requests as HTTP-POST. If a client needs to access the individual parameters of the requests it still can parse the information from the request. By including complete POST-requests in our Context Document we allow for client applications which only need to be able to send POST requests and handle the results. Notably if a client application can handle GML-geometry as a result of a WFS-request it should also be able to handle GML-geometry as a result of a WPS-request without further knowledge of the architecture of the underlying web processing service. Nevertheless a GIS-client with full geo web service client functionality is needed for the initial generation of the Context Document.

We base our Context Document on the draft of the OGC Web Context Document which we find lacking in several aspects we describe in the following.

3.1 User Group Specific Rules

Experts from different fields have different requirements on a specific geospatial problem. A subsoil expert e.g. needs different geospatial data for the evaluation of a location of a planned escape shaft than a civil engineer. In order to cope with these different needs we include a mechanism to define user rules in our Geospatial Web Service Context Document. That way we can encode any relevant information for one specific problem in one document while allowing for specialised views on this problem. In the most simplistic case the authors of the document can make specific layers or features visible only for specific user groups (see class UserRules in fig.1).

For a more fine-grained addressing of user specific demands we intend to add the concept of level of detail (see class LevelOfDetail in fig. 1). The concept of different level of details (LoD) has become popular by the introduction of the CityGML[6] standard and is well suited in the context of planning inner-city-railway-tracks: depending on the planning task e.g. every nut, bolt, and screw of a pipe installation has to be known or just an approximate geometry of the pipes.

As the concept of different levels of detail (LoD) is fundamental to the overall collaboration platform and therefore must be implemented even in lean clients, it is explicitly modeled in the con-

[6] OGC City Geography Markup Language (CityGML) En-coding Standard, Version 2.0.0 OGC Document 12-019 http://www.opengeospatial.org/standards/citygml

text document schema. Normally, different LoDs result in different requests to Geo Web Services. Therefore the class “LevelofDetail” is associated with the class “Offering” in the schema. Research is still needed in order to cope with different concepts for expressing LoD which will result in different specializations of the abstract class “LevelOfDetail”. In addition to the discrete LoD levels as defined for the procedural model of the track (see section 6) and for CityGML our research in this area also investigates more fine-grained models such as defined by (Stadler and Kolbe, 2007) which allows for expressing the LoD as a quantitative measure which could be modelled as feature-level metadata.

3.2 Annotations

According to Brush (2002) annotations are a very natural way to collaborate on a document. Klimke and Döllner (2010a) demonstrated that the concept of annotations can not only be applied to textual documents but also to 3D scenes. It includes textual annotations on spatial data as well as geometry annotations in form of geo-referenced objects. We included their data model with minor adjustments in our Geospatial Web Service Context Document to enable users to enrich their communication (see classes SpatialReference, Annotation and their subclasses in fig. 1).

3.3 Camera Views

A specific view of a scene can contain a lot of information since the visibility of structures or lines-of-sight are view dependent (Klimke and Döllner, 2010a). In a collaborative platform communication of view parameters is a desirable feature (see class VirtualCamera in fig. 1). By basing the parameters of our Virtual Camera object on GML-geometry we may cater for different spatial reference systems. The drawback of this approach is that the client has to be able to interpret these spatial reference systems (the same goes for any geometry). In our project, we solve this by using only spatial references systems which are easy to transform into local reference system used by CAD-software, but we are aware that this cannot be a universally valid approach.

In this section, we showed how to describe sets of requests to Geo Web Services providing spatial data in a Geospatial Web Service Context Document enriched with additional information like annotations, user rules and level of details. In the following section we present our concept of integrating this document into our collaboration platform using CAD modelling tool as clients.

4 The Collaboration Platform

4.1 Synchronous Geometric Modelling

Synchronous geometric modelling has been focused in scientific research at least during the last three decades. In our approach, to facilitate synchronous geometric modelling we first studied possibilities to describe geometry in an abstract way by storing the construction steps that make up the complete geometry model, in contrary to explicit geometry models, where the concrete geometry data is stored, such as vertices, edges, etc. Based on the results, we developed a so-called procedural geometry model (Borrmann, Ji and Jubierre, 2012). A major improvement compared to previous research approaches is the integration of different levels of details (LoDs) into this procedural model. In a further step we integrated this procedural model into a collaboration platform (Flurl et al., 2012). This platform then provides the possibility for different planners to work synchronously on this geometric model. Thereby, using the modelling tools they are accustomed to, the different stakeholders can work in the abstraction level (LoD) of their specialised tasks, e.g. the planning of the concrete track, the modelling of the interior or exterior of the tunnel tube, or the detailed planning of an escape shaft.

4.2 Integration of Web Services

To use available web services in a convenient way, we investigated possibilities to integrate them into the different planning tools via the collaboration platform in a unified, dynamic, and generic way. Thereby, unified means the integration mechanism, i.e. loading, visualization, and scaling should be the same for all types of web services. A dynamic integration means it should be possible to load web services at the beginning of a collaborative session as well as during a session. Since one doesn't know all services and all types of services at a certain point in time, one also should be able to integrate new types of web services during runtime, when they at least fulfil some criteria about their content.

As a basis for a unified integration process we developed a component structure as depicted in figure 2. This structure is a concrete application of the composite design pattern. In our case, the main idea of this structure is to encapsulate all kind of relevant data sources into Parts, which implement the common Component interface. This interface itself defines common functionality to load, rescale, and visualise a specific Part. Additionally, an Assembly class is defined, that also implements the Component interface: Assemblies aggregate different Parts into one Assembly. Since an assembly and all these different Parts must implement the Component interface, they all must provide functionality to load, visualise, and rescale their related geometry data. Here, it should be mentioned, that in the concrete modelling tools, each Part is the basis for exactly one working document.

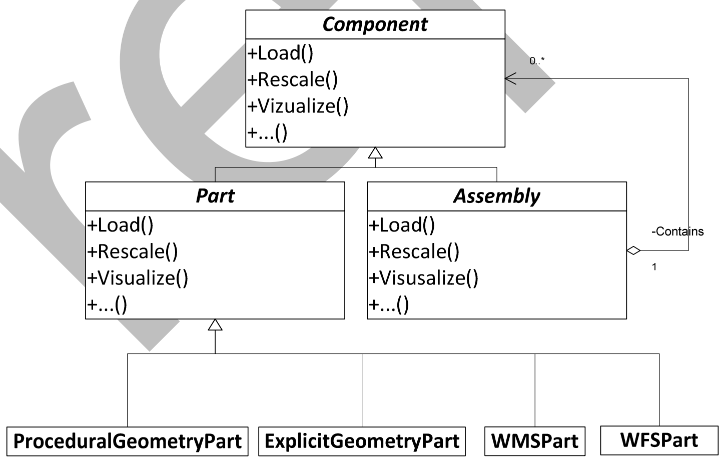

Figure 2: UML class diagram of the component structure

In concrete, we encapsulate procedural geometry data into a Procedural Geometry Part, explicit geometry into an Explicit Geometry Part, and WMS or WFS services into WMS respectively WFS Parts. As mentioned above, different parts can be grouped into one assembly. Thus in particular, different WMS and WFS Parts can be grouped into one single assembly. Since a Geospatial Web Service Context Document describes one or more different web services, we encapsulate a Geospatial Web Service Context Document into one Assembly containing the related Parts respectively services. In the following, we will refer to such an assembly as a web Service Assembly.

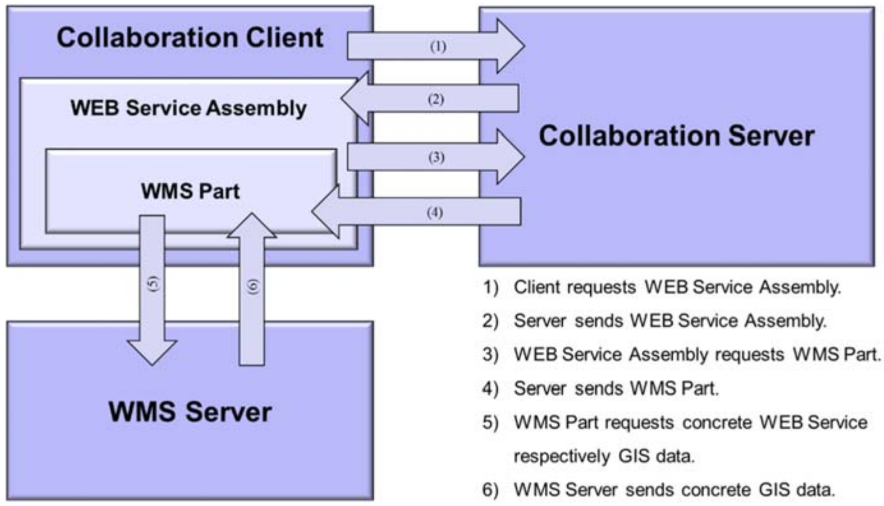

Figure 3: Integrating a web service assembly

We now want to illustrate the integration of a web service assembly containing one single WMS Service in detail (figure 3).

At first, the client requests a web Service Assembly from the collaboration platform. When the client receives it (2), the client calls its load functionality. The Assembly then requests all the Parts described respectively contained in itself, in our case only one single WMS Part (3). After the WMS Part has been received from the server, the Assembly calls the WMS Parts load functionality (4), thereby providing the information contained in Geospatial Web Service Context Document regarding the server location and the contained geometry data formats. The WMS Part utilises this information and loads the real data provided by the concrete service (5). Finally, the Part informs the Assembly that the loading process is finished (6), whereupon the Assembly calls the visualise functionality of the Part. Finally the WMS Part visualises itself in a new working document. One important fact about this workflow is that the collaboration server is not responsible for loading and sending a large amount of geometry data. This avoids the loading of GIS data to become a bottleneck for the collaboration server that primarily has to orchestrate the collaborative workflow and not the simple streaming of geometry data.

As indicated above, different Parts can be grouped into one Assembly. This facilitates loading different Parts in one single step and creating one single Assembly document displaying these different Parts: this way it is possible to visualise a map service containing surface data and a procedural model containing the principle track course in one and the same document. In principle, there are two main problems in adding the geometry contained in different Parts in one and the same Assembly document: first, different Parts normally encapsulate different data formats. Secondly, this data refers to different coordinate systems and different scales.

In our structure, the first problem is already solved, since the Part certainly knows its concrete data format and is responsible for visualising itself. To overcome the second problem, an assem-

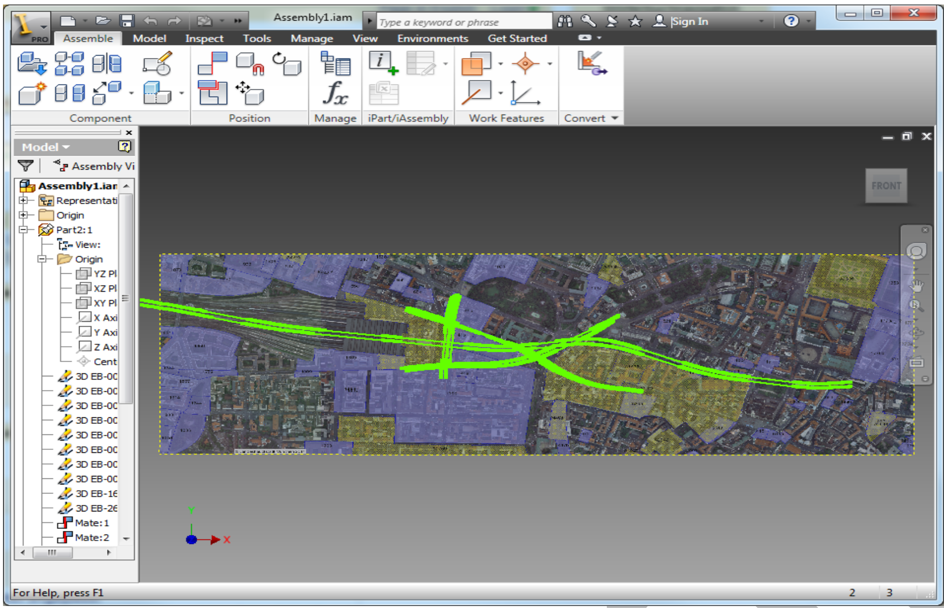

Figure 4: Assembly of a WMS-part showing a hybrid representation of aerial image and development plan and a procedural geometry-part representing a model of several underground railway-tracks.

bly defines a global coordinate and global unit system. To adapt the different Parts to these systems, the assembly uses the rescale functionality provided by the different Parts. Therefore, it simply hands over its global scaling systems to the respective Parts, when the loading process is finished via the rescale functionality. When the client finally calls the visualise functionality, the Parts can adapt themselves to the common assembly document using these global systems.

Above we have explained that the integration process of different data sources is unified using the common component structure. Furthermore, we have seen that the loading process of a specific Part is separated from the modelling process. Thus, it is possible to load additional Parts during runtime using the provided Component functionality, allowing for a not only unified, but also dynamic integration process.

Finally, since the loading, rescaling, and visualisation process is encapsulated via the Component interface, it is possible to integrate services respectively data sources to the modelling process that are a priori unknown to the client. The one and only thing that must be guaranteed by the collaboration platform is, that a requested Part implements the functionality defined by the Component interface in a meaningful way. So the loading process is not only unified and dynamic, but also generic.

5. Conclusion

In this paper, we showed how collaborative work with geographical information can be facilitated using our concept of Geospatial Web Service Context Document. In comparison to other approaches we do not only bundle requests to several web services but additionally include information which and how data shall be presented to specific user groups, and we allow for annotations and encoding of 3D camera views to enhance the collaborative workflow.

Furthermore, we demonstrated how this information can be integrated into the different planning tools via the collaboration platform in a unified, dynamic, and generic way. The basis for this integration is a component structure, which facilitates the incorporation of different data sources, in our case web services and procedural geometry models, in a convenient way.

In figure 4 you see a result of our approach: the successful combination of a WMS Part containing a hybrid representation of aerial image and development plan and underground railway tracks embodied in a procedural geometry Part.

In future work, we will improve our recent concepts and try to integrate new features in our collaboration platform, e.g. simulation respectively simulation data concerning the pedestrian flow in a train station.

References

Bidarra, R., Van De Berg, E. and Bronsvoort, W. F. (2002). A Collaborative Feature Modelling System, In: Journal of Computing and Information in Engineering, Vol. 2, Issue 3(2002), 192 et seq.

Borrmann, A. (2008). Computerunterstützung verteilt-kooperativer Bauplanung durch Integration interaktiver Simulationen und räumlicher Datenbanken,35-56, Aachen: Shaker Verlag.

Borrmann, A., Ji, Y. and Jubierre, J.R. (2012). Multi-scale geometry in civil engineering models: Consistency preservation through procedural representations. In: ICCCBE, 2012, Moscow, Russia.

Breunig, M., Rank, E., Schilcher M., Borrmann A., Hinz, S., Mundani, RP., Ji, Y., Menninghaus, M., Donaubauer, A., Steuer, H., Vögtle T. (2011). Towards Comput-er-Aided Collaborative Subway Track Planning in Multi-Scale 3D City and Building Models. In: Proceedings of 6th 3D Geoinfo, Wuhan.

Brush, A.J. (2002). Annotating Digital Documents for Asynchronous Collaboration, PhD. Dissertation, University of Washington

Cai, X., Li, X., He, F., Han, S. and Chen, X. (2012). Flexible Concurrency Control for Legacy CAD to Construct Collaborative CAD Environment. In: Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol. 6, Issue 3 (2012), pp. 324-339.

Fan, H., Sun, C. (2012). Dependency-based Automatic Locking for Semantic Conflict Prevention in Real-Time Collaborative Programming. In: SAC, 2012, Riva del Garda, Italy.

Flurl, M., Mundani, R.-P., Borrmann, A. and Rank, E. (2012) A collaborative multi-scale planning platform: Concept and implmentation approach. In: ICCCBE, 2012, Moscow, Russia.

Klimke, J., Döllner, J. (2010a). Geospatial Annotations for 3D Environments and their WFS-based Implementation. In: Painho, M., Santos, M.Y., Pundt, H. (Eds.). Geospatial Thinking, Berlin Heidelberg: Springer, pp. 379-397

Klimke, J., Döllner, J. (2010b). Combining Synchronous and Asynchronous Collaboration within 3D City Models. In: Fabrikant, S., Reichenbacher, T., Kreveld, M., Schlieder, C. (Eds.). Geographic Information Science, Berlin Heidelberg: Springer, pp. Vol. 6292, 115-129

Li, M., Gao S. and Wang, C. (2007). Real-Time Collaborative Design With Heterogeneous CAD Systems Based on Neutral modelling Commands. In: Journal of Computing and Information Science in Engineering, Vol. 7 (2007), pp. 113-125.

Neis, P., Singler, P., Zipf, A. (2010). Collaborative mapping and Emergency Routing for Disaster Logistics-Case studies from the Haiti earthquake and the UN portal for Afrika. In: Geospatial Crossroads@ GI-Forum, Vol. 10

Smolders, S., Alegre, C., Gianfranceschi, S., Gilles, M., Resch, B., Everding, T. (2011). GENESIS Employing Web Processing Services and Sensor Web Technology for Environmental Management

Stadler, A., Kolbe, T.H. (2007). Spatio-semantic Coherence in the Integration of 3D City Models. In: Proceedings of the 5th International Symposium on Spatial Data Quality, Enschede.

Sun, C., Yang, Y., Zhang, Y. and Chen, C. (1996). A Consistency Model and Supporting Schemes for Real-time Cooperative Editing Systems. In: 19th Australian Computer Science Conference, 1996, Melbourne, Australia.

Tang, M., Chou, S.-C. and Dong, J.-X. (2007). Conflict classification and solving for collaborative feature modeling. In: ScienceDirect, Vol. 21, Issue 2(2007), pp. 211-219