Backend architectures for modern cartography

Markus Jobst, Georg Gartner

Research Group Cartography, Department of Geodesy and Geoinformation, Vienna University of Technology

Abstract. This paper explores the offers and needs of service-oriented architectures (SOA) for modern cartography. Beside the overall concept of SOA its benefit for map production should be highlighted. This backend point of view on the IT architectural framework simultaneously delivers specific insight to the possible technical impact of SOA for an interactive map. It will show how minimum standards for the existence and performance influence the pragmatic dimension of a front-end map application and therefore will lead to the topic of quality of services.

Keywords. Map Production, Service-Oriented Architectures, Modern Cartography, Pragmatics, Backend Impact

1. Pragmatics of modern cartography

Cartography may often be related with drawing images of topographic features. Modern cartography still uses the visual communication channel, but also adds a lot of features, which help to explore, understand and use a map. The modern map production lane uses a high digitized and partly automated procedure and ends up in various products, spanning from paper maps, map clothes to any kind of digital devices. The Internet and development of the World Wide Web pushed the access to and usage of maps further [Peterson 2008]. Nowadays maps are accessible for everyone. Provided that an infrastructure like the "mobile Web" and appropriate devices exist, ubiquitous cartography becomes enabled: map applications are available everywhere and anytime. This kind of map distribution facilitates greater access to spatial information, increased interactivity design within and for maps, real-time information processing and intense integration of multimedia levels for geospatial information transmission. Furthermore mobile devices with GPS sensors support navigational tasks (mobile pedestrian navigation systems) as well as data recording. This data collection by volunteers lead to an interesting initiative, which produces a worldwide crowdsourced dataset (OpenStreetMap) that is accessible for everyone. The main requirement for the wide success of OpenStreetMap is definitely the efficient backend architecture, which allows realtime mapping and produc-

tive geospatial data handling in a way that not only experts can produce geospatial datasets [Coleman 2011].

All technological developments lead to a user understanding that a map can be ubiquitously used and even modified. This understanding means that a client can rely on map application quality, which covers data quality, performance, availability and even situation awareness. An evaluation of a map application by the user seems to be driven by a fulfilment of the pragmatic dimension in terms of geospatial communication. A review of search trends in Google shows that people are no longer searching for static maps but are increasingly looking for interactive services. This could be an indicator that efficient map applications in terms of information transmission are needed. The design, handling or functionality of a map application are core elements for the user judgement, but as soon as the map application is pragmatically used, parameters concerning availability, response time or validity become important. These parameters are created within the backend. Therefore modern map makers have to deal with IT architectures and their underlying network technologies.

2. The backend and its influence

The given examples show the importance of IT infrastructures and applications for modern cartography. Most of these examples focus on the front-end, which is the user interface at the client side. It enables all the access to functionality and information transfer to the user. The impact on the users´ imagery of the transmitted information is heavily triggered by all perceivable effects of the user interface. Perceivable effects are the meeting point of the front-end and the backend. A lot of errors within the delivery processes of the backend could be perceivable in the front-end, which may lead to misunderstanding, misinterpretation and distortion of information. Therefore the backend plays an important role for the pragmatic dimension of a map/map application and its acceptance by the user.

The backend consists of several tiers that have to be considered. Most of us know that the Internet works with different network layers, its protocols and unique addressing. This network tier with its bandwidth and connectivity is just one of a considerable series. Other important tiers in the backend concern the data-, service-, e-commerce- and client tier.

Impact of the network tier: The network tier covers all aspects of interconnected components of the computer network. In a very brief description the computer network is a telecommunications network – between data processing nodes – that is used for data communication purposes. The data processing nodes and endpoints are computers that are accomplished with

various network hardware components. Two nodes are defined as "networked" if operations of one node are able to evoke operations on the other node (requests) and understands its responses [Tanenbaum 2006]. There exist several characteristics within the network tier that have an impact on the front-end, like media that are used for signal transmission, communication protocols that are used for network traffic organisation, the scale and topology of the network and its benefits and organisational scopes.

Examples how the network tier impacts the front-end base upon the network performance, -security [Simmonds et al 2004] and -resilience. Network performance refers to the quality definition of a telecommunication product from the viewpoint of a customer. If we adapt this general statement to map applications, then the quality of a network's performance has to be designed according to the need of the map application. For the pragmatic use within modern cartography (e.g. mesh-ups) this requirement means that the network performance supports a fluent exchange with information within the network or at least allows performance enhancing caching mechanisms. The network performance is one aspect that stimulates and enables on-demand interaction with the map. For the measurement of network performance various methods, like the ones for circuit-switched networks (1) or asynchronous transfer mode networks (2), can be used. This means that either (1) the number of rejected requests under heavy traffic loads or (2) data throughput, connect time or stability is measured.

The network security covers mechanisms and policies within the backend that prevents and monitors unauthorized access, use or modification. The access to use network components, applications or data is maintained by a network administrator. For the pragmatic dimension the front-end needs to support authentification, network certificate detection or data decryption. These requirements may have some enormous impact on the pragmatic dimension if a user needs to login, accept certificates or needs specific plug-ins in order to decrypt information. In many cases interaction is needed and therefore disturbs map application usage or even information perception.

Network resilience describes an acceptable level of "undisturbed" activity for normal network operations or services. This means that a service delivers responses although threats and challenges, like simple misconfigurations or specific attacks, act on it. In order to enhance network resilience possible challenges and risks have to be identified. Appropriate metrics help to protect the network and its embedded services. Examples for network services are distributed processing, networked storage, communication services like messaging or access to applications and information [Smith 2011].

Impact of the client tier: the client tier covers a client software, the device/computer and even the user who uses the client software. It is a node component within the computer network that may also deliver information to the user. Therefore the client interface and the device could be the front-end. But it also provides a backend with its client software, operating environment and/or embedded servers. The client software sends requests to other nodes within the computer network and decodes its answers. A client software "web browser" connects to web servers and retrieves text and pictorial information (webpages) that need to be interpreted for visualisation. The client is one main part of the client-server structure, where the clients connects to the Web or bind specific services. Servers generally wait for potential clients to initiate acceptable connections. Therefore the client needs to support network and interprocess communication standards in order to be understood and accepted by the networks as well as its servers [Shen 2009].

Impact of the e-commerce tier: E-commerce (electronic commerce) is an industry branch where products or services are sold via electronic systems in computer networks. Modern e-commerce generally uses the Internet for the diverse range of technologies such as the Web, e-mail, mobile devices, social media or smartphones. It draws on mobile commerce, electronic funds transfer, supply chain management, online transaction processing, electronic data interchange or automated data collection systems [Snider 1992, Eisingerich 2008, Graham 2008]. The e-commerce tier facilitates the financing and payment aspects of e-business transactions. The e-commerce tier can be divided to the shared digital business infrastructure, the sophisticated model for operations and value-chains, a model to manage business teams, customers and partnerships as well as consistent policies, regulations and social systems [Zorayda 2003]. The e-commerce tier is another bridge between the front-end and the backend, which allows buying, license and using important information. The more the e-commerce tier is embedded in the real-time map production process, the less it disturbs information transmission. For instance a pop-up that asks for your banking account and validation everytime map tiles are loaded will attract the main attention instead of the map application which should transfer the important information.

Impact of the service tier: within a service-oriented architecture the service layer can be considered as as bridge between the higher conceptual layers (Enterprise and process layers) and the lower ones (component and object layers), which is responsible for individual business operations [Bieberstein 2005]. Services describe the functionality of a server-sided map application, which fetch data, information and operations from other nodes within the network in order to solve requests. E.g. cartographic mesh-ups intensively

use various services to achieve predefined results. It may be obvious that increasing reliability and simplicity of services, in terms of reducing the need of interaction and requiring user knowledge, help to keep the users attendance on the important geospatial information transmission.

Impact of the data tier: The data tier provides access to information. It covers its own application, conceptual- and physical tier, which allows accessing, structuring and persistent storing of information. The data tier in terms of holistic backend description does not define the data transmission layer of the network, but the storage area/management for a network-based map application, which could be distributed data sources throughout the Internet. The persistent existence, metadata as well as accessibility are main aspects which are generally visible (or not!) in the front-end. If a dataset is not available anymore or it cannot be found because metadata are missing, then the map application in the front-end will show nothing.

All the previously listed tiers show that the backend, the technical, logical and organisational structure behind a web-based map front-end, is a complex structure that heavily influences the front-end. The reason to use a tier approach is the aim to reach independency between the tiers. This means that each tier is mainly independent from changes of other tiers. E.g. it is possible to exchange databases within the data tier and services will still work until their needed attributes exist; or the e-commerce tier can be modified and the client will still be able to fetch a network connection if the same access variables exist. This means that a network-based map application is complex in terms of technical realisation and interdependency, but if the tier- and component- approach is considered, the maintenance of system components becomes feasible. The conceptual approach of service-oriented architectures follows a tier- and component- approach and therefore offers some useful characteristics for modern map applications.

3. SOA offers

A service-oriented architecture (SOA) is a conceptual model for a network of distributed systems, applications and data, which uses standardized interfaces between the tiers [Erl 2004].

In the technical perspective the SOA principles make use of a strict disjunction of the types “user-interface”, “application”, “interfaces”, “system tiers” and “data” [Bell 2008]. For the conceptual design these components are put to different layers according to their types, which enables independency. In fact the technical perspective of SOA uses well-defined standards for all interfaces between the tiers. Only the definition of a wide spectrum of standards allows the loose coupling of the tiers.

The loose coupling means that the interfaces are standardized, well described and independent from proprietary (black-box) structures. Anyway a proprietary application can be embedded as soon as it supports these interface standards. The same applies for the data components. For this reason the value adding composition of SOA components [Goncalves 2011] is supported, which creates one of the most important economic aspects of SOA.

In general the basic SOA concept is made up of eight main design principles originating from IT Web service development and business process management. In the following some of these principles will be described and it will be tempted to show their importance for the frontend/map application of a service oriented map.

The service loose coupling describes that a component can be used with little or no knowledge of its definitions by other components. The usage of a flexible file format such as XML for the description of a service or data and its schema facilitates a loose coupling of interfaces. It enables subscribers to extract clear definitions of how to use a service and its data. A subscriber could publish the collection of statements, which are used and created with a service. This would allow a responsible data- or service provider to test whether their subscribers need adaptations of their interfaces if modifications are made on data or service. Any reducing of information that has to be passed into a service as key data could enhance loose coupling.

In terms of the front-end (map application) loose coupling also enables independency from the service content. E.g. a service that provides the content of a webshop will change this information according to sortiment changes. If loose coupling is enabled then these changes do not have any effect on the interface because it stays the same and the sortiment will be modified automatically based on the new delivery of the service. If a front-end is based on the content of the webshop, then every change of the sortiment causes an adaptation of the front-end.

The principle design idea of SOA counts on reusable services. A service that is created once can be used often. Those services embed a “solution logic” that is independent of specific business processes or technology. In many cases these Web-enabled services are designed to be platform-, software framework- and business model independent.

Apart from the technical realization of service reusability, the embedding of this design principle performs a top-down analysis process, which results in a complete set of usable services. In fact this gathering of candidate services require increased resources. In the end a significant amount of resources can be saved because of eliminating the full development procedure. One important assumption to fully apply service reusability is the given cultural

basis: do service developers reuse others services? Do map makers allow existing services to be incorporated as possible solution although further design adaptations are needed? If the cultural basis does exist, then the given emphasis on service reuse leads to the issues of reliability. A reused service as part of new services provides its functionality to multiple service customers. If its quality of service is not reliable then all incorporating services will lack of reliability.

A geospatial example could be a web map service that is used as layer in other web map services. The resulting service outcome relies on the availability and delivery of the embedded web map service. Because temporary malfunctions of these services may occur, caching technologies try to overcome this disaffection. This method tries to establish service autonomy for short time periods, in which the cache of a web map service delivers the content. Service reusability highlights the main requirements for a pragmatically useful SOA, which can be subsumed to the notion "quality of service".

Service discoverability is a key component of the “search-find-bind” procedure, which defines how resources can be found and thus describes the general sequence of service invocation. Service metadata becomes published to a central register, which allows searching and finding for services. Therefore service discoverability increases service reuse and decreases functional redundancy of services. The development of services with the same functionality becomes obvious. Indirectly, discoverability makes services more interoperable, because specific functionalities can be found and used.

An effective realisation of this design principle requires standardized metadata in consistent and meaningful manner. Therefore service developers have to be enforced to use and apply existing metadata standards whenever possible. Core definitions of metadata are the mandatory part and may be extended with individual optional supplementations.

Following the principle of service abstraction, service discoverability calls for a minimum set of metadata attributes in order to make service discoverable. This inversely dependency has to be considered within the architectural concept of service infrastructure creation. This direct relation leads to a careful separation of the general service metadata and its external functionality description from the internal, not documented service procedures, which should not be embedded within the service metadata.

The existence of metadata for services and data allow for an automatic binding process of these resources into a front-end. E.g. information coming from observation sensors could directly be embedded into a map.

The principle of service composability encourages the reuse of services in various and multiple solutions as well as defines that services themselves may be made up of composed services. Therefore any recomposition does not depend on the size and complexity of a service composition. Composability is one main aspect of SOA, which promotes new service solutions by reusing existing ones and therefore support a value adding chain by technological independency.

Several considerations have to be kept in mind: with a growing number of service compositions, some services will get highly reused because they fulfil basic needs. Hence these compositions will become dependent on the selected services they share with all the other compositions. It is foreseeable that such service compositions will lead to critical functionality. In order to avoid these critical situations alternate standby services will be needed, which means that a calculated number of redundant services need to be allowed and the composition architecture has to be precisely analysed and designed. A solution of “how and when to use standby services” could be embedded in a service contract.

The quality of provided components covers standardization and appropriate communication of data/information quality on one hand and service quality on the other. These requirements for quality describe central needs for SOA-based map applications.

4. Needs for SOA-based maps

The central needs for SOA-based map applications from the viewpoint of the front-end (the map as user interface) are quality agreements for all system components/nodes. The conceptual idea of service-oriented architecture supports ongoing changes and thus a "living system". This means that components/information sources can be added, modified or removed. Like in the World Wide Web the whole system is changing. We know examples of webpages showing broken links to images, which have been removed from their source, or blogs that show up new information every day. This fact can also be adapted to the service-oriented map application. In addition to missing information layers, the quality of embedded services will decide on success of the application: How fast is the response? Are there peak times, when the service produces errors? Are there offline times? Is the quality of the content homogeneous enough for the mesh-up?

In the role of the mapmaker, who uses SOA and depends on data and service quality, a certain degree of confidence into the nodes is needed. There are two dimensions that have to be considered before a source can be added to a map product: data quality and service quality.

Data quality is directly related to the recording, generation and processing of geospatial data. The specific aim and its aggregation from other sources may have impact on its quality. Descriptions of data processing as well as the initial aim of the data product are important hints, beside a lot of others like dataset consistency, -validity, etc., if a dataset could be used as source information. All these describing elements will give no indication if the information is homogeneous with other data sources. In general it needs further processing in order to be homogeneous within the map application [Hopfstock 2012, Schmidt et al 2012].

Service quality helps to reach the main principles of SOA. It takes the usage of appropriate standards, schema consistency and precise documentation as a given, in order to reach the aim of automatization (as computer-to-computer communication). But there is one aspect that is not controlled within the establishing process of services (like standards, documentation) - the quality of a service underlies IT infrastructure, its architecture, security and operation management. In the end this aspect informs about reliance of a service. This value can be directly used in the resulting reliance description of a SOA-based map application which has to use the worst embedded source in terms of reliance.

The quality of service convention is needed for SOA-based map applications in order to support the pragmatic dimension. Herein three positions have to be considered: availability, performance and capacity.

Availability assures that a service is available for the time of its being. This means that it would have a described "uptime". In terms of organizational structures the uptime of a service could be described by the time of supervision, e.g. Monday to Friday, 9 a.m. to 5 p.m.. In fact it does not make sense for a SOA to embed services that will be switched off for the weekend, especially within international usage. Therefore the uptime of a service is meant to be 7days/week and 24 hours. A planned and unplanned downtime ratio could be defined and communicated within metadata. This allows for detailed planning and definition of reliance in terms of availability.

Performance describes how long a service needs to send a response. Obviously a reference frame and amount of data has to be specified so that this time span makes sense. Still there will be enough variance for the interpretation, e.g. the measures timespan counts from receiving the first byte up to sending the first byte. This example interpretation would remove network traffic from the value. Another possible interpretation could be to measure the first byte sent by an client up to receiving the last byte of a response at the client (information transfer complete).

Capacity indicates how many parallel requests can be completed without generating errors or influencing performance. Depending on the amount of parallel requests, a service may need more time to response a request (queuing) or it will abort a request. Both situations are the result of reaching a maximum capacity of the service.

In the role of a map user data- and service quality are observable in the front-end. For many users service quality is the more serious matter than data quality. The reason is simple: a service should be available on demand. It should deliver valid results within a meaningful timeframe. Therefore service quality directly supports the pragmatic dimension of a map application and is the most important aspect for a successful information transmission.

5. Resume and future aspects for a SOA based cartography

This contribution focuses on the backend of modern cartography and how it influences information transmission through the front-end. The backend is either used for digital map production or SOA-based map applications. In both cases the impact on the front-end is enormous and leads to disaffection, misunderstanding or frustration. Some influence on information transmission is indispensable. Modern cartography builds on technological backend structures. There is a direct dependency of modern (SOA-based) map applications to these structures - without the backend the digital map will not work.

We could show that the pragmatic dimension for geospatial information transmission with modern technologies depends for a large extend on backend architectures. Even the basic technical structures, like the network, do have their impact on the user interface. In order to reach independency - in a way that maintenance and component updates become feasible - a tier approach can be established. This tier approach is build on standardized interfaces and services - the basic concept of SOA. The SOA concept follows simple rules of the World Wide Web, which are adapted to other domains like maps in the Internet. This conceptual approach generally brings in independency between the different tiers by following specific design principles, but it does not remove the impact of the backend characteristics on the front-end/user interface. We could highlight that the main sources that may influence information transmission are data quality and service quality. Whereas data quality can be described prior, service quality calls for experience in IT architectures and predictions of service usage. Service quality is

the result of specific IT architecture construction and load (processing, network, DB) removal methodologies.

In the end the map application user does not care about the backend structures. Within a SOA-based map application the user will never be able to explore IT architectures and he/she may not have the knowledge to do so. But the impact of backend architectures will be observable in the front-end. The map user is affected by slow answers of the system, availability of information layers or random failures in situations when the information is needed.

An evaluation of future trends shows that SOA-based cartography will increase in the coming years. Spatial data infrastructures, which are the sources for geospatial information, are established throughout the world and provide easy and standardized access to these sources. Additionally recordings of real-time sensors, which are published with standardized interfaces, increase the number of information sources. Mapmakers learn how to deal with real-time information and how to combine different sources in a homogeneous way. Complementary we have to learn how to deal with a large amount of geospatial data services and their changing behaviour. How can we evaluate the right services for our map application? What are the quality enhancing methodologies for map production in SOA? How to achieve sustainability? How to approach the pragmatic dimension of geospatial communication? A lot of questions go along with SOA-based map applications and the fulfilment of the pragmatic dimension in terms of communication in future.

References

Bell M. (2008) Introduction to Service-Oriented Modeling. Service-Oriented Modeling: Service Analysis, Design and Architecture; Wiley&Sons, p.3., ISBN 978-0-470-14111-3.

Bieberstein N., Bose S., Fiammante M., Jones K., Shah R. (2005) Service-oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap. Published by: FT Press, 2005. ISBN 0-13-187002-5, ISBN 978-0-13-187002-4

Coleman D.J. et al (2011) Volunteered Geographic Information: the nature and motivation of produsers, International Journal of Spatial Data Infrastructures Research, Special Issue GSDI-11

Eisingerich A. B.; Kretschmer T. (2008) In E-Commerce, More is More, Harvard Business Review. 86 (March): 20–21

Erl T. (2004) Service-Oriented Architecture. Concepts, Technology and Design, Prentice Hall PTR, Upper Saddle River, 2004, ISBN 0-13-185858-0

Gonçalves V, Ballon P (2011) Adding value to the network: Mobile operators' experiments with Software-as-a-Service and Platform-as-a-Service models. Telematics and Informatics, 28(1), 12-21.

Graham M. (2008). "Warped Geographies of Development: The Internet and Theories of Economic Development" (PDF). Geography Compass 2 (3): 771. doi:10.1111/j.1749-8198.2008.00093.x

Hopfstock A. (2012) A pragmatic approach to the production of a European Reference Map, in Markus Jobst (ed.) Service-Oriented Mapping 2012, Jobstmedia, Vienna, ISBN 978-3-9502039-2-3.

Peterson M. P. (2008) Trends in Internet and Ubiquitous Cartography, Cartographic Perspectives, Number 61, 2008

Schmidt M., Jörg W. (2012) basemap.at – Creating a Harmonised Web Basemap for Austria, in Markus Jobst (ed.) Service-Oriented Mapping 2012, Jobstmedia, Vienna, ISBN 978-3-9502039-2-3.

Shen X., Yu H., Buford J., Akon M. (2009) Handbook of Peer-to-Peer Networking (1st ed.). New York: Springer. p. 118. ISBN 0-387-09750-3

Simmonds A; Sandilands P; van Ekert L (2004). "An Ontology for Network Security Attacks". Lecture Notes in Computer Science. Lecture Notes in Computer Science 3285: 317–323. ISBN 978-3-540-23659-7

Smith, P. (2011) Network resilience: a systematic approach, IEEE Communications Magazine, vol.49, no.7, pp.88-97, July 2011

Snider, J.H.; Ziporyn T. (1992). Future Shop: How New Technologies Will Change the Way We Shop and What We Buy. St. Martin's Press. ISBN 978-0-312-06359-7

Tanenbaum A. S. , Computer Networks, Fourth Edition, Pearson Education 2006 (ISBN 0-13-349945-6)

Zorayda R. A. (2003) e-Commerce and e-Business, UNDP-APDIP, 47 pages, ISBN 978-81-312-0076-6 (S.20)