Intelligent Information Integration: From Infrastructure through Consistency Management to Information Visualisation

Michael Schroeder

Department of Computing, City University, London, UK, msch@soi.city.ac.uk

Abstract. Information agents collect information from various locations, prepare the information and visualise the result to the user. This information integration process needs to address three different aspects: First, the infrastructure to facilitate information integration; second, data preparation and consistency management; and third, information visualisation. For these three aspects, we present current trends such as Grid computing, the Semantic Web and visual datamining and we outline how these technologies apply to geovisualisation.

1 Introduction

Information agents are computational software systems that have access to multiple, heterogeneous, and geographically distributed information sources [WG96]. They perform active searches for relevant information in non-local domains on behalf of their users or other agents. Information from multiple autonomous sources is retrieved, analysed, manipulated and integrated, and visualised. To build information agents three different aspects of the information integration process need to be addressed:

- what is the infrastructure to facilitate information integration,
- how is data prepared and consistency achieved,
- how is the integrated information visualised to the user?

In this context, three important developments occurred recently:

- Regarding the agents infrastructure, there are now large efforts going on in many countries to build Grids, which enable resource sharing and collaboration over wide and open networks.
- The actual integration of information is always a problem as it is difficult to establish a common meaning of the data in an open system. To this end, the Semantic Web effort aims to develop standards and protocols, which cater for the interpretation of data with respect to global and local ontologies. Thus, it establishes the basis for a meaningful integration of distributed data sources.

- Regarding presentation of the agent's integrated data, the area of visual datamining is gaining momentum. In contrast to traditional datamining, visual datamining is a human-centered analysis process.

We will briefly review these three developments and then relate them to geovisualization.

2 Infrastructure: Middleware and the Grid

In a report dating back to the late 1940s, the British Government concluded that the demand for computing power in the UK could be satisfied by two or three computers. They turned out to be wrong. Nowadays computers are ubiquitous and in many areas of scientific computing the demand for computing power is nearly infinite. Although computing power increases by a factor of ten every five years, theoretical results suggest that single processors have their limits [Fos95]. Networking is one way forward beyond these limits. In fact, in the top 500 list of fastest computers [MSD02], which solve a matrix factorisation problem, a machine with 9152 Pentiums ranks among the top and achieves nearly peta-flops. The Internet can add a new dimension to such integrated massively parallel processors: Countless comparatively small computing resources, such as PCs, have the potential to create a vast computing power, if connected. An example of such an approach is distributed.net, which connects some 100,000 computers over the Internet to solve brute-force a decryption challenge [DCT99]. Such a wide area network of PCs is however only one instance of a much broader vision: The transformation of the capability and modalities of scientific research by providing transparent, intuitive, timely, effective and efficient access to distributed, heterogeneous and dynamic resources. These resources include computational facilities, applications, visualisation, data and experimental facilities, integrated and accessible as a single resource over the Internet - the Grid [FK98].

2.1 Requirements for a Grid infrastructure

Various Grid platforms as discussed below have been developed. Common to all of them is the need to provide the following functionality:

- Resource management and task monitoring: The Grid bundles resources such as CPU time, memory, bandwidth, storage, and other more specialised components. Their access and usage needs to be managed - transparent to the user. This includes scheduling of resources and implementation of specific access policies. From the user's perspective, the monitoring of tasks is important. A Grid's task monitoring component needs to be able to feed back the current status of the task and of the resources the task uses.
- Communication infrastructure: The communication services required by applications and infrastructure are very diverse. Generally, two paradigms can be distinguished: peer-to-peer and client-server. The former category covers

point-to-point communication as well as broadcasts, and multicasts. The latter is typically served by remote procedure calls such as Corba, RMI, RPC, DCOM, and WebServices. An important aspect of a communication service is the quality of service it can provide. Is it reliable or not? How fast or slow is it? For multicasts the question of simultaneous and instantaneous message delivery arises and for remote procedure calls whether calls can be guaranteed to be executed exactly, at least or maximally once. Overall, the principle is that there needs to be a trade-off between speed and reliability. And depending on the application at hand, the best communication service needs to be chosen.

- Data storage and movement: Most Grid-aware applications will be data intensive. Thus, remote access to data is an important aspect of a Grid infrastructure. Besides access to databases, this issue includes the provision of a distributed file system. An application running in one location may need data stored in a file in another location. Thus infrastructure is required, which provides e.g. a uniform global name space that allows applications to refer to files, a host of I/O operations, and the ability to optimise performance by caching and even migrating files and data.
- Security and authorisation: In open systems, security is a prime concern, which needs to be considered from two opposing positions: protection of the user's interests and protection of the infrastructure provided to the user. For the user it is important that the Grid-infrastructure handles data confidentially and that no information is exposed. For transmission and storage these concerns can typically addressed by encryption, but ultimately the host will execute a user's program using his or her data, thus making it accessible. On the other hand, malicious users are a threat to the host. To deal with this problem, the Grid-infrastructure needs to authenticate users and implement access policies and resource accounting to limit the power of a user's task and thus protect the hostt.
- Development and execution tools: To support the user and make the Grid-infrastructure easily deployable, development and execution tools are needed. Such tools should provide formal, portable programming paradigms and languages, that express parallelism and support software synthesis and re-use. To ease the migration to a Grid-solution the automated porting of legacy code is desirable and the provision of standard Grid-enabled services for common tasks.

A number of Grid platforms have been developed (e.g. LSF, the Load Sharing Facility, www.platform.com; Seamless Thinking Aid, starsv1.koma.jaeri.go.jp/en; Legion, legion.virginia.edu/overview.hml; and TeraGrid, www.teragrid.org, which connects multiple Grid systems). We will discuss two of them below. All of these efforts need to be seen in the context of the Open Grid Services Architecture, which will be a standard for Grid infrastructure.

2.2 Globus

The Globus toolkit (www.globus.org) provides a number of components implementing the requirements discussed above. Regarding resource management and task monitoring, Globus provides the Globus Resource Allocation Manager for allocation of computational resources and for monitoring and control of computation on those resources. Furthermore, there is a module, which manages resource reservation and allocation, and another one, which facilitates distributed access to structure and state information of the system. Executables are handled by a component, which supports construction, caching, and location of executables. As communication infrastructure, Globus implements an extended version of the File Transfer Protocol, GridFTP. The extensions include the use of security protocols, partial file access, and management of parallelism for high-speed transfers. A component for global access to secondary storage uses among others GridFTP for remote access to data via sequential and parallel interfaces. Regarding security and authentication, Globus uses for example the Secure Socket Layer (SSL) for encrypted data transfer and certificates according to the X.509 standard.

2.3 Unicore

Unicore (UNiform Interface to COmputing REsources, www.unicore.de) provides transparent online access to resources of supercomputer centers creating a seamless high-performance computing portal. Unicore supports the remote submission, compilation, and running of applications and their input/output data.

A Unicore client enables the user to create, submit and control jobs from any workstation or PC on the Internet. The client connects to a Unicore gateway which authenticates both client and user, before contacting the Unicore servers, which in turn manage the submitted Unicore jobs. They incarnate abstract tasks destined for local hosts into batch jobs and run them on the native batch subsystem. Tasks to be run at a remote site are transferred to a peer Unicore gateway. All necessary data transfers and synchronizations are performed by the servers. They also retain status information and job output, passing it to the client upon user request.

Unicore addresses security on all levels: user authentication is performed using X.509 certificates. A public-key infrastructure has been established for the German HPC centers that enforces rigorous control of certificates. User authorization is handled by the participating sites using their proven mechanisms. Unicore sites completely retain their autonomy to authorize users and to allocate resources to them. To transfer jobs, control information and application data, SSL is used to guarantee data integrity and confidentiality. Job representations are signed with the originating user's private key to prevent third parties from tampering with the job contents.

3 Consistent Information Integration: The Semantic Web

The Grid provides the infrastructure to manipulate and integrate distributed information. The next level of complexity is then how to guarantee a common understanding of the distributed data and how to achieve consistency once it is integrated. A common understanding of the data's meaning can only be agreed upon if a common ontology - a concept hierarchy, which describes the domain at hand - is being referred to. But in a distributed, open environment this is unlikely to happen. The semantic web [BLHL01] (www.semanticweb.org) aims to address this problem. The staring point for the Semantic Web effort is the problem that most of the information online available is intended for humans and not machines. Web pages contain extensive rendering information besides the content, which is a character sequence as any other to a machine. To capture e.g. authorship of a document, a string such as `by Michael Schroeder` would be part of the document. But to a machine this string appears no different than any other. To tackle this problem XML, the eXtensible Markup Language, was introduced. It allows one to tag contents, thus introducing a first level of semantic description to the pure contents. To guarantee the consistency within one XML document, there is document type definition (DTD) or an XML Schema, which defines the grammar for the document. Thus, given an XML document, a name could be enriched by the information that it is an author, e.g. `<author>Michael Schroeder</author>`.

However, XML is only a first step, as another web page might refer to the same author using different XML tags: `<creator>Michael Schroeder</creator>`, thus making it difficult to automatically determine the equivalence of the two names. This shortcoming of XML is aimed at by the Semantic Web by using local and global ontologies to specify the schemas and meta data of the contents. Before we can give an example of such a global ontology, we need to resolve how to represent this information.

Besides XML, there is RDF, the resource description framework, which allows one to capture meta data. RDF is based on triples of a subject, predicate, and object. A triple (s, p, o) expresses that a resource s has a property p with value o. Therefore p is a binary relationship. However, RDF can express relationships of any arity (number of parameters) by simply splitting them into more than one triple. An object can also be a value, enabling triples to be chained, and in fact, any RDF statement can itself be an object or attribute - this is called reification and permits nesting. RDF Schema are to RDF what XML Schema are to XML: they permit definition of a vocabulary. Essentially RDF Schema provides a basic type system for RDF such as $Class$, $subClassOf$ and $subPropertyOf$. RDF Schema are themselves valid RDF expressions. To continue the authorship example above, there is one bit missing: A global ontology we can refer to. One example of such an effort is Dublin Core (www.dublincore.org), which defines standards for meta data. Dublin Core defines e.g. tags for the creator of a document.

Let us now use Dublin Core as a point of reference to describe authorship of a document building on XML and RDF:

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
```

```
 4   <rdf:Description about="http://www.soi.city.ac.uk/~msch/">
 5     <dc:creator>Michael Schroeder</dc:creator>
 6     <dc:title>Information Agents</dc:title>
 7     <dc:description> Informations agents need to provide solutions for
 8                      information integration infrastructure,
 9                      consistency management, and information
10                      visualisation. </dc:description>
11     <dc:date>2002-10-10</dc:date>
12     <dc:format>text/html</dc:format>
13     <dc:language>en</dc:language>
14     <dc:publisher>ICA</dc:publisher>
15   </rdf:Description>
16 </rdf:RDF>
```

Line 2 states that this XML document will contain RDF triples using the syntax referred to in the given URL. Line 3 imports the meta data tags defined in Dublin Core. Furthermore line 2 and 3 abbreviate the corresponding pointers, which are used from line 4 to 16, as `rdf` and `dc`. Lines 5 to 14 contain the actual predicates and their values and among others the tag `dc:creator` followed by the author's name. Because this tag refers to a global ontology, which is accessible to others, it can be used to create a joint understanding of the data across applications. In this particular example, all applications using Dublin Core - as e.g. a couple of search engines - will be able to answer a query for documents authored by Michael Schroeder properly.

However, a global ontology may not be appropriate for all domains, therefore global and local ontologies can be used. In domains, where cohesion is necessary and fruitful, organisations will develop standard ontologies (examples are www.dmtf.org for the telecommunication industry, www.bpmi.org for business processes, www.papinet.org for the paper supply chain, and www.hr-xml.org for human resources). This is also an active area in GIS [FED$^+$00,FEAC02,DEHS00]. In [FEAC02], the authors develop the idea of ontology-driven GIS. They show how to integrate geographical information systems using ontologies, which are e.g. based on existing approaches such as WordNet [Fel00]. They argue then that such ontologies can be mapped to interfaces, which connect the software components of the GIS. Different from the semantic web, such an approach is not really open, as it does not provide a global ontology, which is online accessbile. It rather hard-codes an ontology into the glue, which integrates different components. A similar approach is followed by the OpenGIS consortium [Ope], which specifies ontologies for geographic objects using interface definition languages such as Corba IDL or Microsoft's COM.

A final step is the development of ontology mark-up languages, which enable reasoning. DAML+OIL is such an effort, which defines a language to express relationships between classes and caters for reasoning about these relationship. Another effort, which focuses specifically on the use of rules to specify integrity constraints, deduction, and reactive behaviour is RuleML [BTW01]. RuleML (www.ruleml.org) aims to standardise a rule mark-up language, which facilitates the interchange of rules. Such rules could be used to specify e.g. the semantic integrity of data.

4 Information Visualisation and Visual Datamining

With the data consistently integrated, we can turn to the next challenge: how to turn it into knowledge? To this end, the idea of visual datamining is gaining momentum. While datamining focuses on algorithms to analyse the data, visual datamining emphasises that the task is a human-centered process. In GIS, this idea of integrating traditional datamining technqiues with interactive, visual exploration is actively pursued [AAS$^+$01,AA99b,GHRW01,GPG02,MWH$^+$99]. The basis of such an approach are the three distinct areas of information visualisation [War00,Spe00], human-computer-interaction, and datamining.

From a human-computer-interaction point of view it is important that the visual datamining process supports operations such as projections, filtering and selection, linking and brushing, zooming, details on demand, overviews, and visual querying. Besides supporting the human's interaction in the datamining process, visual datamining deploys information visualisation techniques, which can be broadly classified as geometric, icon-based, pixel-oriented, and hierarchical [KA01,Kei01,Kei02]. Here we briefly review some of these techniques and put them into context.

4.1 Geometric techniques

Two very general geometric techniques are scatter plots and parallel coordinates [ID90]. In their most basic form, scatterplots plots depict objects with associated x and y-value at the corresponding position of a coordinate system (see Fig. 1). The basic 2D approach can be extended to 3D, but suffers then from well-known problems of 3D such as occlusion and difference in perception of depth in comparison to height and width. It is also not obvious how to visualise high-dimensional data with a scatterplot. One approach is to apply dimension reduction, which creates however difficulties interpreting the plot as the data was transformed and information lost; another approach simply plots all of the variables against each other, creating a quadratic number of 2D scatterplots. While no information is lost, the interpretation is nonetheless difficult, as many different plots need to be mentally linked. Furthermore, it is difficult to label objects in large scatterplots and the Euclidean space that the scatterplots use, may not be appropriate for data that originates from a space with a different topology. However, scatterplots are simple, can give a good overview and depict the basic structure and are therefore used a lot. Another technique that is fairly general, simple, and therefore wide-spread are parallel coordinates (see Fig. 1). Parallel coordinates also use a coordinate system as basis. In contrast to scatterplots, they can display high-dimensional data, by associating an object's attributes with values on the x-axis. The corresponding value of the object's attributes are then plotted along the y-axis creating a graph representing the

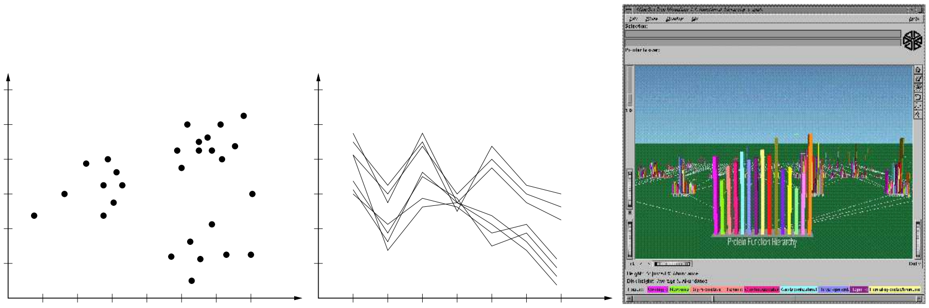

Fig. 1. A scatter plot, parallel coordinates, and an information landscape tree created with MineSet [htt]. The tree uses the plain and information at nodes makes use of the third dimension.

object. Even if many graphs are plotted at the same time a general trend can still be seen. However, a big problem is that often the order of attributes is arbitrary, although it is highly important for the ability to interpret the parallel coordinates. Different orderings can lead to more or less “overwriting” of graphs. Therefore, parallel coordinates should only be used if it is possible to order the attributes.

Two less prevailing information visualisation techniques are information landscapes and pro-section views. In information landscapes (see Fig. 1) two dimensions are used for spatial layout, while the third dimension is used for data display [Bra96]. Therefore they are not truly 3D and do not suffer from all the problems 3D views create. However, they do exhibit some of the virtues of 3D. It is possible to seamlessly zoom from an overview to a detailed view of the data. Pro-section views [FB94,Spe00] are related to scatterplots. They address the problem of displaying high-dimensional data with a scatterplot. The idea is to reduce dimensionality by applying projections and sections to the data (hence the name).

4.2 Icon-based techniques

Icon-based techniques aim to preserve all the information by mapping attributes to different visual features of an icon representing the object as a whole. Two prominent members of this class are starplots [Fie79] and Chernoff faces [Che73]. Star-plots (see Fig. 2) represent the value of an attribute through the length of lines radiating from the icon’s center. The lines for all the attributes are distributed at an even angle around the center, thus creating a star shape. Often the tips of the star’s beams are connected in order to create a closed shape. Similar to parallel coordinates, star plots succeed in displaying high-dimensional data without any dimension reduction. But they also suffer from the same problem: The order of attributes has an impact on the resulting overall shape and therefore on how the data is perceived. Furthermore, starplots are difficult to compare to each other as it is difficult to quantify the differences. This applies also to Chernoff faces, which map attribute values to up to 18 facial features such as lips, nose, ears, etc. of a stylised face. The idea behind this mapping is that human cognition is especially capable to recognise faces. However, it is not clear whether this also applies to the *stylised* faces. In fact, experiments [MER99] indicate that perception of Chernoff faces is a serial process and not pre-attentative. While the faces are intuitive and compact (for up to 18 variables), they suffer from some problems. The display is limited to 18 variables and the facial features cannot be easily compared to each other. How does the size of an ear compare to the angle of an eye brow? This is particularly bad, as the facial properties have very different visual properties: perception of the area an oval covers (the face) is not comparable at all to perception of angles, line width, and curviness. As a result, a different assignment of attributes to facial features will change the perception of the face radically and the mapping of which variable to assign to which feature greatly influences the interpretation [CE98]. Additionally, the values of variables cannot be read from the faces' features, there may be an emotional component when interpreting faces, the faces' symmetry means redundancy of information, and the display of many faces may create a texture, which distracts from the interpretation of the individual faces. Nonetheless, it was found [MER99] that Chernoff faces are useful for trend analysis, but not for decision making. Overall, Chernoff faces are intuitive, but due to the above limitations they are difficult to use effectively.

Two other techniques, which use icons and can represent high-dimensional data are stick figures [PG88] and colour icons [KK94]. The former maps attribute values to angles between "sticks", which represent the attributes. Thus each object is mapped to a concatenation of sticks. A criticism that applies is that sequences of angles may not be optimal for perception. If however, the spatial arrangement of a large number of stick figures is chosen appropriately, then they create a texture and can give a very good overall impression on the data as a whole. So, although they generally suffer from similar problems as Chernoff faces do, they can be used for a different purpose. Colour-icons are in the same spirit mapping attribute values to colour, where the attribute itself has a fixed location in the icon. Again, the same problem arises as with many of the techniques above: the spatial arrangement of attributes is of great importance for the end result, especially, since neighbouring colours influence each other's perception by the user.

4.3 Pixel-oriented

While most geometric techniques tend to work well for a medium number of attributes, icon-based approaches are not suitable for large numbers of attributes. Pixel-oriented techniques complement this picture, as they tend to work well to get an overview over a large set of objects, which possibly have a large set of attributes. Colour maps [Ber81] (see Fig. 3) are the most prominent pixel-oriented technique. Typically, colour maps are tables, whose rows contain the objects and columns the attributes. Each cell is then coloured according to the value of the

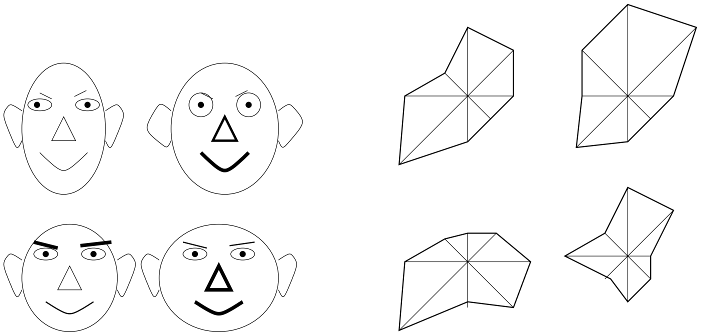

Fig. 2. A set of Chernoff faces and some star plots.

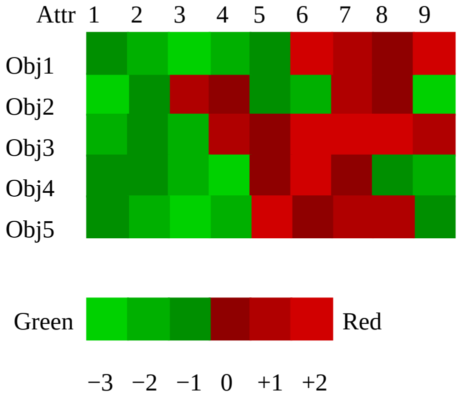

Fig. 3. A color map for 5 entities with 9 attributes.

column's attribute for the row's object. A limiting factor for the value range to be displayed is our perception of colour. While the space of different colours may be huge, it is not straigt-forward how to create linear colour scales with a high perceived resolution [War00].

Using a single pixel on the screen as a cell, colour maps can easily display data sets as large as the medium's resolution caters for. However, similar to the other techniques, colour maps highly depend on the order of columns and rows and of the choice of colour mapping. One approach is to cluster objects and attributes and order them according to their similarity. This will ensure that the neighbourhood of a cell is not too different from itself, so that colours re-enforce themselves creating regions with boundaries instead of seemingly random spots of colour spread all over the display.

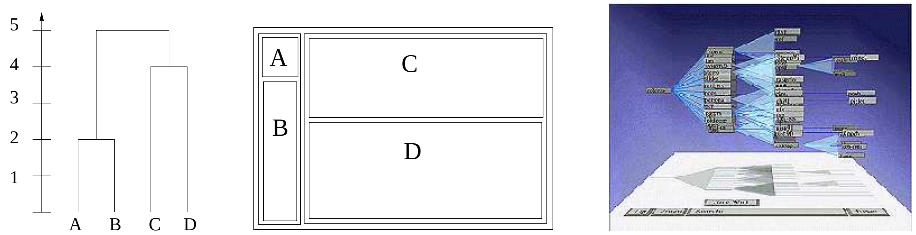

Fig. 4. A dendrogram, a tree map, and a cone tree [JCM93].

4.4 Hierarchical techniques

This type is particularly useful if the given data is already hierarchical by nature. One way of creating hierarchies, even if the data is not hierarchical, is hierarchical clustering, which produces binary trees, whose leafs represent the objects and whose parent nodes correspond to clusters of objects. Hierarchical clustering is e.g. used in [GPG02] to cluster multivariate spatial data.

The clusters are often displayed as dendrograms, i.e. drawings of binary trees with the additional convention that the difference in height between parent and children indicates the similarity of the two children. Consider the example shown in Fig. 4 on the left. Let us assume that objects A, B, C, D have the values $1, 3, 8, 12$, repectively. If we define the similarity between two objects as the absolute difference between the corresponding values, then A and B with a similarity of 2 are a first cluster and C and D are a second cluster with a similarity of 4. If we define the similarity between clusters as the similarity of their nearest neighbours, then the first cluster has a similarity of 5 to the second one. The above hierarchical clustering is depicted in the dendrogram in Fig. 4 on the left. If the clusters can be associated with a value representing the cluster, then tree maps [Sch92,HMM00] (see Fig. 4) are useful visualisations, which convey the value attached to a group of objects effectively. In tree maps, an object or group of objects is represented as a rectangle, whose size reflects the attribute value. The rectangle for a group of objects is filled with the rectangles representing its members. As a result, tree maps are good at conveying aggregated single attribute values of a hierarchical structure. For the example, the tree map shows two main rectangles of an area of $1 + 3 = 4$ and $12 + 8 = 20$. Treemaps have a problem though. Should doubling of the attribute's value lead to an area of double the size or with a base line of doubled length (and thus quadrupled area)? Invariably, no matter which choice is taken, user's may have the opposite expectation.

Finally, there are cone trees [RMC91,JCM93] (see Fig. 4), which can be seen as 3D representation of dendrograms. Like other 3D representations, they can have the benefit of allowing users to easily zoom between overview and detail. But they also suffer from the problem of occlusion and difficulty to navigate and find the desired information.

To summarise, visual datamining builds heavily on information visualisation techniques to turn datamining into a human-centered process. Different information visualization techniques are useful for different types and sizes of data and at different stages of the datamining process. The hierarchical techniques are suitable for tree structures, which can be e.g. the result of hierarchical clustering. High-dimensional data can either be reduced in its dimensionality or directly visualised. For the former, scatterplots are very useful. For the latter, parallel coordinates and stick figures can provide a useful overview over the data even if it is very high dimensional. If the data contains not too many attributes (less than ca. 20), then the icon-based methods (Chernoff faces and star-plot) can be used.

To produce the data suitable for the above techniques a host of algorithms for clustering and dimension reduction are applicable. The data and the analysis algorithms link visual datamining to the previous two sections on the Grid and Semantic Web. The data to be visualised will often come from different sources, which require consistent integration, which can be achieved through global taxonomies as promoted by the Semantic Web. The analysis algorithms transforming and preparing the data for visualisation and the visualisation itself are often computationally very intensive and could use the Grid to run efficiently.

5 Cross-fertilisation

To put it in a nutshell, information agents perform intelligent information integration. To implement such agents, an infrastructure is required that supports gathering and efficent processing of large data sets. This can be achieved by a Grid. The agents need to consistently integrate the data, which can be supported by the Semantic Web, and finally the agents need to present results to the user, which can be done using visual datamining techniques.

How do these trends - the Grid, the Semantic Web, and visual datamining - relate to geovisualization? In [MK01], MacEachren and Kraak pose a number of research challenges to geovisualization. One of three main challenges relates to visualization-computation integration and in particular

> 3. To address the engineering problem of bringing together disparate technologies, each with established tools, systems, data structures and interfaces. Four specific problems identified are: [3.1] to develop computational architectures that support integrating databases with visualization; [3.2] identify the database functions needed to support the real-time interaction demanded by visually facilitated knowledge construction; [3.3] determine the impact that underlying data structures have on the knowledge construction process; and [3.4] develop mechanisms for working discovered objects back into a consistent data model. [MK01]

An example, where the above problems are tackled for a specific system has been implemented by Andrienko [AA99b,AA99a]. Their system integrates the datamining tool Kepler and the geovisualization system Descartes. Kepler analyses economic and demographic data for different European countries. It accesses the appropriate databases and runs learning algorithms to relate the different relations such as gross domestic product or infant mortality. The results of Kepler are then visualised as a map by Descartes. Both systems run independently and act as servers, which cooperate and which are accessed by a user's clients, which in turn are linked, too.

The Grid: Given the challenges and the above example system, how can the open problems be addressed in a principled and general way? Problem 3.1 and 3.2 allude to the infrastructure required to facilitate information integration including tools and data, which are often physically distributed, as e.g. in the example system. The reference in 3.2 to real-time interaction, which is also present in the example system, also means that computations have to be fast, which is often only possible with bundled computational power. Both concerns - the transparent access to distributed resources and the provision of computational power - are catered for by the Grid. The Grid can therefore form the backbone for such systems. In the above example system, Descartes and Kepler could use the Grid to perform computationally intensive tasks such as rendering of maps and the execution of the datamining algorithms. In their current client-server implementation the servers could become bottlenecks if too many clients access them. In a Grid implementation of such a system, the clients could directly initiate the computations of Descartes and Kepler in locations, which make the best trade-off between e.g. available CPU cycles and proximity to the client to reduce latency due to transfer of large data sets.

Semantic Web: Problem 3.3 and 3.4 of MacEachren and Kraak's challenges touch on the importance of data structures, data models, and consistency in general. In Adrienko's system the integration is hard-coded and mappings between the different systems have to be carefully designed. The general problem of semantic consistency and inter-operability between distributed data sources is the concern of the Semantic Web. The Semantic Web can provide the technology needed to define standardised ontologies, which can be complemented by local ontologies where appropriate. In such a system, data sources can be linked automatically, as the tags of a data entry refer to a common global ontology and thus indicate that a concept used in one source has the same meaning as in another. Such a mechanism contributes to the solution of consistent data models put forward in [MK01], which is a prerequisite for automated integration. This work is particularly interesting as standard ontologies for GIS are currently being developed [FED$^+$00,FEAC02,DEHS00,Ope]

Visual datamining: In [MK01], the authors explicitly refer to the integration of Knowledge Discovery and Datamining and the example system by Andrienko is such an integration of a KDD and a GIS tool to facilitate explorative data analysis and visual datamining. All results in these two areas regarding processes and techniques may be useful for more specific geographical data as well. As identified in [MK01] a specific problem relates to "how to incorporate the location and time components of multi-variate data within visual and analytical methods." This question is not (yet) answered by visual datamining, but it makes an important contribution, to emphasis interaction and visualisation as important parts of datamining, which is a challenge for geovisualization, too.

Conclucion: To summarise, in this article we have reviewed three major enabling technologies for intelligent information integration, namely Grid computing, the Semantic Web, and visual datamining. We have discussed their relation to geovisualization by showing how they address geovisualization challenges put forward in [MK01]: In future geovisualization systems, the Grid could provide the infrastructure for transparent access to distributed data and computational resource, the Semantic Web could be used to achieve automatic, dynamic, and consistent data integration, and visual datamining could be used to visually explore geographic data.

Acknowledgements: I'd like to thanks Jason Dykes, Menno-Jan Kraak, Alan McEachren, Penny Noy, and the anonymous reviewers for their valuable and detailed comments to this paper, which helped to improve it.

References

[AA99a] Gemady Andrienko and Natalia Andrienko. GIS visualization support to the C4.5 classification algorithm of KDD. In *Proceedings of the 19th International Cartographic Conference*, pages 747–55, 1999.

[AA99b] Gemady Andrienko and Natalia Andrienko. Knowledge-based visualization to support spatial data mining. In *Proceedings of Intelligent Data Analysis*, pages 149–60. LNCS 1642, Springer Verlag, 1999.

[AAS$^+$01] N. Andrienko, G. Andrienko, A. Savinov, H. Voss, and D. Wettschereck. Exploratory analysis of spatial data using interactive maps and datamining. *Cartography and Geographic Information Science*, 28(3):151–65, 2001.

[Ber81] Jacques Bertin. *Graphics and Graphic Information Processing.* Walter de Gruyter, 1981.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.

[Bra96] T. Bray. Measuring the web. In *In 5th Intl. WWW Conference*, 1996.

[BTW01] Harold Boley, Said Tabet, and Gerd Wagner. Design rationale of ruleml: A markup language for semantic web rules. In *Proc. of SWWS01*, Standford, Ca, USA, July/August 2001.

[CE98] M. C. Chuah and S. G. Eick. Information rich glyphs for software management data. *IEEE Computer Graphics and Applications*, 1998.

[Che73] H. Chernoff. Using faces to represent points in k-dimensional space graphically. *Journal of American Statistical Association*, 68:361–368, 1973.

[DCT99] Inc. Distributed Computing Technologies. http://www.distributed.net, 1999.

[DEHS00] Mark David, Max Egenhofer, Stephen Hirtle, and Barry Smith. UCGIS emerging research theme: Ontological foundations for geographic information science. Technical report, UCGIS, 2000. www.ucgis.org/ermerging/ontology_new.pdf.

[FB94] G. Furnas and A. Buja. Prosections views: Dimensional inference through sections and projections. *Journal of Computational and Graphical Statistics*, 3(4):323–353, 1994.

[FEAC02] F. Fonseca, M. Egenhofer, M. Agouris, and P. Camara. Using ontologies for integrated geographic information systems. *Transactions in GIS*, 6(3):231–57, 2002.

[FED$^+$00] F. Fonseca, M. Egenhofer, C. Davis, , and K. Borges. Ontologies and knowledge sharing in urban GIS. *Computer, Environment and Urban Systems*, 24(3):232–51, 2000.

[Fel00] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 2000.

[Fie79] S. E. Fienberg. Graphical methods in statistics. *American Statisticians*, 33:165–78, 1979.

[FK98] Ian Foster and Carl Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kauffman, 1998.

[Fos95] Ian Foster. *Designing and Building Parallel Programs*. Addison-Wesley, 1995.

[GHRW01] M. Gahegan, M. Harrower, T.-M. Rhyne, and M. Wachowicz. The integration of geographic visualization with databases. *Cartography and Geographic Information Science*, 28(1):29–44, 2001.

[GPG02] D. Guo, D. Peuquet, and M. Gahegan. Opening the black box: Interactive hierarchical clustering for multivariate spatial patterns. In *10th ACM International Symposium on dvances in Geographic Information Systems*, 2002.

[HMM00] I. Herman, G. Melancon, and M. Marshall. Graph visualization and navigation in information visualisation: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[htt] http://www.sgi.com/chembio/resources/mineset. Silicon graphics mineset report.

[ID90] A Inselberg and B. Dimsdale. Parallel co-ordinates. a tool for visualising multi-dimensional geometry. In *Proc. of Visualization'90*, pages 361–70, San Francisco, CA, USA, 1990.

[JCM93] J.G.Robertson, S.K. Card, and J.D. Mackinlay. Information visualization using 3(d) interactive animation. *Communications of the ACM*, 36(4):57–71, 1993.

[KA01] Daniel A. Keim and Mihael Ankerst. Visual data mining and exploration of large databases. Tutorial at ECML/PKDD01, 2001.

[Kei01] Daniel Keim. Visual exploration of large datasets. *Communications of the ACM*, 44(8):38–44, 2001.

[Kei02] Daniel Keim. Information visualization and visual datamining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[KK94] D. Keim and H. Kreigel. VisDB: Database exploration using multidimensional visualisation. In *Proc. of Computer Graphics and Applications*, pages 40–49, 1994.

[MER99] C. Morris, D. Ebert, and P. Rhengans. An experimental analysis of the pre-attentativeness of features in Chernoff faces. In *Proceedings of Applied*

Imagery Pattern Regognition: 3D Visualization for Data Exploration and Decision Making, 1999.

[MK01] Alan MacEachren and Menno-Jan Kraak. Research challenges in geovisualization. *Cartography and Geographic Information Science, Special Issue on Geovisualization*, 28(1):3–12, 2001.

[MSD02] Hans-Werner Meuer, Erich Strohmaier, and Jack Dongarra. Top500 list. High Performance Networking and Computing Conference, 2002. http://www.netlib.org/benchmark/top500.html.

[MWH$^+$99] A. M. MacEachren, M. Wachowicz, D. Haug, R. Edsall, and R. Masters. Cosntructing knowledge from multivariate spatitemporal data: integrating geographic visualization with knowledge discovery in database methods. *Cartography and Geographic Information Science*, 13(4):311–34, 1999.

[Ope] OpenGIS. www.opengis.org.

[PG88] R. Picket and G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proc. of IEEE Conf on systems*, pages 514–19, Piscataway, NJ, USA, 1988. IEEE Press.

[RMC91] G. Robertson, J. Mackinlay, and S. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proc. of ACM SIGCHI conference on Human Factors in Computing Systems '91*, pages 189–94, 1991.

[Sch92] B. Schneiderman. Tree visualization with tree-maps: 2-D space-filling approach. *ACM Transactions on Graphics*, pages 92–99, 1992.

[Spe00] Robert Spence. *Information visualisation.* ACM Press, 2000.

[War00] C. Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann, 2000.

[WG96] Gio Wiederhold and Michael Genesereth. The basis for mediation. Technical report, Standford University, 1996.