GraSSML: Accessible Smart Schematic Diagrams for All

Z. BEN FREDJ
Oxford Brookes University
Department of Computing
Wheatley Campus, OX33 1HX
+44 (0)1865 485603
zbenfredj@brookes.ac.uk

D.A. DUCE
Oxford Brookes University
Department of Computing
Wheatley Campus, OX33 1HX
+44 (0)1865 484528
daduce@brookes.ac.uk

ABSTRACT
Graphical representations are a powerful way of conveying information. Their use has made life much easier for most sighted users, but people with disabilities or users who work in environments where visual representations are inappropriate cannot access information contained in graphics, unless alternative descriptions are included.

We describe an approach called Graphical Structure Semantic Markup Languages (GraSSML) which aims at defining high-level diagram description languages which capture the structure and the semantics of a diagram and enable the generation of accessible and "smart" presentations in different modalities such as speech, text, graphic, etc. The structure and the semantics of the diagram are made available at the creation stage. This offers new possibilities for allowing Web Graphics to become "smart".

Categories and Subject Descriptors
H.5.3 [**Group Organization Interfaces**] Web-based interaction, I.3.6 [**Methodology and Techniques**]: Languages, K.4.2 [**Social Issues**] Assistive technologies for persons with disabilities.

General Terms
Design, Experimentation, Human Factors, Languages, Theory, Legal Aspects.

Keywords
Accessibility, Smart Diagrams, Structure, Semantics, XML, SVG, Semantic Web, RDF, OWL, SPARQL.

1. INTRODUCTION
Diagrams are a fundamental component in the exposition of scientific research and are unavoidable in professional life. Despite their importance, not much has been done to provide accessibility support for information of this kind. To allow full access to the web it is also important that people with disabilities can create accessible web content containing accessible web graphics. It is important to understand that accessibility is not only for people with obvious disabilities but also for people who simply access information and learn in different ways.

The emergence of SVG has changed the way 2D graphics are created on the web. SVG offers many advantages and has introduced accessibility features that raster formats do not offer [1]. When using SVG, information about the diagram is available to the browser in terms of the objects it is composed of. Nevertheless, a number of limitations remain as it only captures diagrams at a low level of abstraction. It is more a "final form" presentation, which involves some drawbacks in the direct creation, modification and access to complex, highly structured, diagrams.

The next section describes the problem in more detail and presents the limitations of SVG and of some of the approaches previously taken to resolve the problem of graphic accessibility. Then we present our approach to the problem which generates presentations from high level descriptions. The following section discusses the current state of the GraSSML development and system architecture. The final section contains conclusions and thoughts on future work.

2. PROBLEM AND RELATED WORK
Many workers have explored different methods to make graphics accessible to blind or visually impaired people by representing graphics through an auditory interface, tactile drawings, text description, etc. Although these approaches partially address the accessibility problem of graphics, they present some limitations.

The approaches taken by the "Blind Information System" (BIS) [2] and "Graphical User Interfaces for Blind People" (GUIB) [3] projects depend on human intervention by a moderator, not necessarily the author of the graphic. In both cases the resulting description of the picture depends on the analysis and indexing of a third party. It is an important responsibility for the moderator who decides what information to convey and thus indirectly imposes a view when the picture is being read (e.g. inadvertently omitting important information).

The TeDUB project [4] (Technical Drawings Understanding for the Blind) explored the possibility of a semi-automatic analysis of diagrams. Their approach for the presentation and navigation of graphical information offers many advantages for blind users. But the (semi-) automatic analysis of the diagram information might produce wrong results and might need active human intervention and time.

The W3C Recommendation for 2D graphics is SVG. For the remainder of this paper we focus on approaches that use SVG as the presentation format. There are profiles of SVG adapted to the needs of mobile devices and technologies. SVG presents many advantages and provides many accessibility benefits [1]. However, some important issues still remain to be addressed.

SVG does not capture diagrams at a high level of abstraction. It is a "final form" presentation, which involves some drawbacks in the direct creation of complex, highly structured, diagrams. It is difficult at this level to handle the resizing and positioning of different shapes in complex diagrams. A simple modification such

W4A at WWW2006, 23rd-26th May 2006, Edinburgh, UK

as changing the alignment from vertical to horizontal can be awkward. SVG does not allow flexible readjustment of layout in response to viewer requirements and the viewing environment, such as different screen formats (PDA, mobile phones etc.).

The issue is that the intentions of the author are not totally captured. The structure of the SVG may reflect the sequence of operations used to create the diagram, rather than the intrinsic object structure within the diagram itself. This is likely to be the case for diagrams created with a general-purpose drawing tool.

Although SVG stores structural information about graphical shapes as an integral part of the image and allows metadata to be attached to primitives, there is little real scope for generating alternative presentations from the description at this level. An additional issue is that an SVG document contains the semantics of the diagram only implicitly. The "alternative equivalents", which allow the author to include a text description for each logical component and a text title to explain the component's role in the diagram, could become tedious for the creation of complex diagrams. If the metadata added by the author are not accurate enough, the semantics of the diagram could differ from the description obtained from the metadata.

Some research groups have explored the accessibility features of SVG and have attempted to address some of the SVG limitations.

The "Science Access Project" research group has successfully explored many accessibility features of SVG in their ViewPlus project [5] but it also has identified some of its limitations [6]: some SVG documents become less accessible when created without <title> and <desc> elements and some are very badly structured and therefore less informative. The ViewPlus project has a good approach by exploring the information behind the picture but the solutions proposed to overcome SVG limitations need too much effort in adding information and/or reorganizing it. It can be tedious and time consuming. It illustrates the fact that SVG is too low level and not informative enough regarding the structure of the graphical information.

An extension to SVG, called Constraint Scalable Vector Graphics (CSVG) [7], has been proposed. It partially addresses some of the SVG limitations by proposing additional capabilities which allow alternate layouts for the same logical group of components in a diagram. But whilst permitting a more flexible description of figures, CSVG remains very close to SVG and still captures diagrams at a similar low level of abstraction.

The SVG linearizer tool [8] generates a textual linear representation of the content of an SVG file by using a metadata vocabulary describing it. The author has to describe the SVG content using this RDF vocabulary and to add textual descriptions to all elements that constitute primary RDF resources. Then, from this information plus information contained in the SVG file itself, an HTML file is generated. This operation can be tedious and not very efficient in very complex diagrams and adding the RDF annotations is an onerous task for the author. This method is too dependent on the creator's patience and willingness to produce appropriate metadata.

These previously presented approaches that we categorize as "bottom-up approaches" have been looking at the problem upside down. They start with the graphical representation of the diagram. The diagram is analyzed and interpreted by a predefined system and/or a moderator (who is not usually the creator of the diagram). The latter is required to add "metadata" to help understand the information. Some of these approaches involve writing difficult programs in order to "discover" the structure or semantics of the graphics.

In 1997, John A. Gardener [9] gave an overview of the concepts of "smart graphics" (information behind a picture) and intelligent graphics browsers for accessing such information. He highlighted the fact that "Nearly every part of smart graphics technology exists today, but to our knowledge there is no complete package that incorporates everything necessary to author a smart picture, incorporate it into an electronic document, and display it intelligently". Our system GraSSML aims, by means of its family of languages, to be such a package by providing access to this most valuable "information behind the diagram". Our hypothesis in the GraSSML approach is that many of the limitations involved in current approaches can be overcome if the information on the structure and the semantics of the graphics were made "part of the graphics", i.e. take a top-down approach.

3. THE GraSSML APPROACH

Our approach aims at facilitating the creation, modification, access and adaptation of diagrams as well as making the information "behind" the diagram available at the creation stage. The availability of this information is then explored to generate alternative representations improving accessibility of diagrams.

The main idea behind our approach is to reduce a task to a sequence of transformations between inputs and outputs expressed in different "Little Languages" [10]: "The GraSSML family of Languages" (Figure 1).

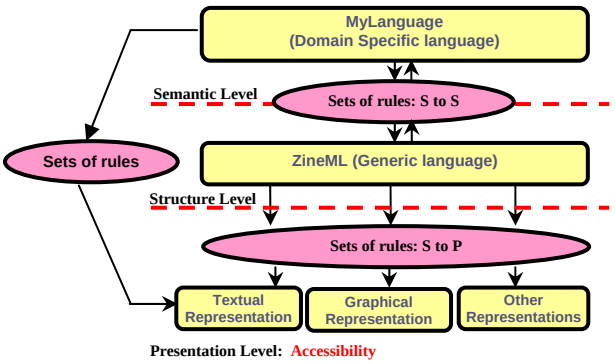

Figure 1: The Levels of GraSSML

An important aspect of this project is concerned with the syntax and semantics of diagrams. In natural language, words can be mapped into a set of meanings whereas in a visual language, geometric objects do not have a unique semantic interpretation. There are a very large variety of diagrammatic notations [11]. There are no universal visual conventions and each person interprets graphical information using his own mental schema and/ or imagination. For a computer life is not so easy. A computer needs a set of formal representational conventions to carry out this interpretation. Each domain has its own notation, syntax and semantic rules. A set of syntactic and semantic rules needs to be defined. As a starting point, we should analyze a wide range of diagrams in a specific domain in order to develop a diagrammatic semantic grammar. Following this specific predefined grammar, the user should be able to define the syntax and semantics of his diagram. Technologies emerging from the Semantic Web Activity will be a base to define such a grammar (RDF / RDFS / OWL).

There are many classification schemes for diagrams. In this work we have used the structural taxonomy proposed by Lohse [12] and have chosen the class of structural diagrams (which includes process and hierarchical diagrams) to study in this project.

4. THE GraSSML SYSTEM

4.1 A Simple Example

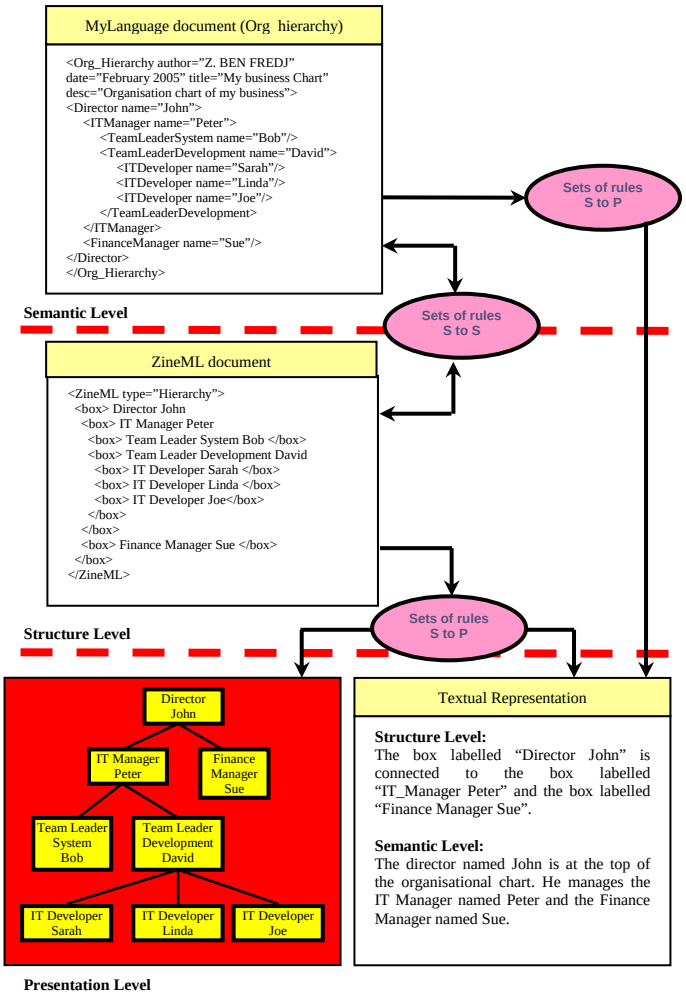

Figure 2: The Organizational Chart example

4.2 The GraSSML Family of Languages

4.2.1 Structure Level: ZineML

"ZineML" aims to be at a higher level than SVG by representing the structure of the diagram. It facilitates the creation and modification of diagrams. ZineML documents aspire to be readable by humans and to give good overviews of diagram structures. The language seeks to be rich enough to allow accessible alternatives of the structural representation of a diagram.

ZineML postulates a set of basic shapes selected to cover a wide range of possibilities for structured diagrams common in different domains (e.g. business, computing…). At this level of abstraction the language designed "ZineML" is not domain dependent, but proposes some options to express and apply rule sets when creating the diagrams. These rule sets could be used to tailor the diagram specifically to a domain.

ZineML offers the possibility to determine and to adjust positions and sizes semi-automatically, with a minimum of effort from the author. The derivation of graphical and other representations from ZineML is done in accordance with predefined rule sets (Figure 2) called "Sets of rules S to P" (Structure to Presentation).

4.2.2 Semantic Level: MyLanguage

"MyLanguage" is the XML language (e.g. XML Schema) used to capture the semantic intent behind the diagram. It does not aim to be universal at this level but domain-dependent, hence the name! It has to be applied to a specific domain where clear conventions are followed when creating diagrams (e.g. Organization charts, UML).

MyLanguage does not make any commitment to graphical presentation, but aims to capture the concepts and relationships between concepts (ontology) that are to be expressed in pictorial or other representational form. At this level, "Who knows better than the authors?" We should bear in mind that "A diagram is worth a thousand words". Indeed, the semantics of the diagram can be complex, and without the help of the author the interpretation generated may be both verbose and shallow in the sense that the author is the one who knows "exactly" what the main message behind the diagram is and therefore can give concise information.

The information concerning the different notations used, the possible relationships between them and the meaning behind each of them, is the information which is required in order to create such a language. Hence, it is necessary to first study the domain and the class of the diagrams with the aim of identifying what concepts and properties these diagrams seek to represent in the domain. As a result of this study a particular "MyLanguage" (e.g. Org_hierarchy) is created. This language aspires to be an intuitive domain specific language; it employs the notations and concepts familiar to practitioners of the domain and by doing so it makes the semantics explicit to the user. Consequently a domain expert can easily understand the information. Even if all the information needed has been gathered, an important issue remains unsolved: the semantics is only implicit for the computer. Indeed, XML covers only the syntactic level and does not provide any means of talking about the semantics of data. There is a need to make the semantics explicit to the computer.

There is an obvious link to activity in the Semantic Web area concerned with the expression of subject ontologies and relationships expressed over terms defined in ontologies (RDF / RDFS / OWL). We are currently working on creating and using ontologies to make the semantics of the diagrams explicit and to underpin "MyLanguage".

Along with the specification of a particular MyLanguage, notational conventions are created. The notational conventions govern the diagrammatic representation of the elements of MyLanguage, and are captured in sets of rules governing the generation of ZineML from MyLanguage (Figure 2: Sets of rules "S to S"). Example: Director (an element in Org_hierarchy) will be represented by a box filled in yellow and a centered label.

4.2.3 Presentation Level: SVG, XHTML…

The availability of the information "behind the graphic" allows us to generate alternative representations. In order to explore this information in the best possible way and to provide representations as accessible as possible we refer to studies on what kind of information is needed and how to describe graphics textually or verbally (e.g. [13] [14]).

- Graphical representation (Figure 2): SVG is used as the graphical output renderer at this level. For complex diagrams we are aiming at allowing an interactive exploration of the diagram by using some of SVG's facilities and the available information concerning the structure and semantics of the diagram (the user can hide some details).
- Textual representation (Figure 2): a Verbalization Model has been implemented. It allows the generation of a textual representation of the structure and the semantics of the diagram.
- Query System: The structure and the semantics of the diagram being available, it becomes possible to express queries concerning specific parts of a diagram in novel ways: "smart diagrams" become a reality. At a later stage we aim at developing the query system based on the explicit structure and semantics of the diagram. Example (Figure 2): What is the total number of employees reporting to Peter? Who is reporting to Peter? Show me the levels that directly report to Peter?

4.3 Implementation

Implementation of ZineML for two classes of diagrams: process and hierarchical diagrams is in hand. The Java Programming Language has been used to implement the presentation algorithms aiming at generating the graphical representation of the diagram (SVG). The graphical representations of ZineML and MyLanguage are done by applying XSLT transformations to each document by respecting rule sets. The verbalization model allowing the generation of the structure and semantic of the diagram based on the syntax of the corresponding XML language used (ZineML or MyLanguage) has been implemented using an XSLT Transformation on ZineML and MyLanguage. The output is an XHTML document containing the textual representation of the structure and the semantic of the diagram. Other functionalities of GraSSML are at the design stage ready for implementation. Once the implementation of the first prototype is completed, experiments will be carried out in order to evaluate the usability and accessibility of the system.

5. CONCLUSION AND FUTURE WORK

Graphics on the Web have significantly improved but a number of issues still remain unsolved. This paper has outlined these issues and has presented some related work aiming at resolving them. These only address some of the problems and they emphasize the need to make the graphical information (information on the structure and the semantics of the graphics) part of the graphic.

We have looked at the problem of creation, modification, access and exploration of Web graphics from a different perspective. Using the GraSSML system, the semantic and structural information is made available at the creation stage. The availability of this information offers new possibilities. Web Graphics become "smart", they carry their knowledge with them, and intrinsically they know their structure (ZineML) and semantics (MyLanguage). Any application smart enough to access, explore and present this knowledge in the right way should be able to propose solutions for identified problems (e.g. adaptive and accessible graphics for all). This could substantially improve the accessibility of web graphics. GraSSML could be the starting point for many projects currently aiming to access, present, explore and adapt graphical information [15] [16].

6. ACKNOWLEDGMENTS

Financial support for ZBF from Oxford Brookes University is gratefully acknowledged.

7. REFERENCES

[1] Charles McCathieNevile and Marja-Riitta Koivunen, *"Accessibility Features of SVG"*, http://www.w3.org/TR/SVG-access/, 2000.

[2] Z. Mikovec, P. Slavik, *System for Picture Interpretation for Blind*.http://cs.felk.cvut.cz/~xmikovec/bis/interact99/

[3] Martin Kurze et al., *"New Approaches for Accessing Different Classes of Graphics by Blind People"*, 2nd TIDE Congress, 268-272, Paris, Amsterdam: IOS Press 1995.

[4] M. Horstmann et al., *"TEDUB: Automatic interpretation and presentation of technical diagrams for blind people"*, CVHI, 2004.

[5] Vladimir Bulatov, John A. Gardner, *"Making Graphics Accessible"*, SVG Open, 2004.

[6] John A. Gardner, Vladimir Bulatov, *"Smart Figures, SVG, and Accessibility"*, Proceeding 2001 CSUN International Conference on technology and persons with Disabilities, Los Angeles, CA, March 2001.

[7] G.J. Badros, J.J. Tirtowidjojo, K. Marriott, B. Meyer, W. Portnoy, A. Borning, *"A Constraint Extension to Scalable Vector Graphics"*, Tenth International World Wide Web Conference, WWW10, Hong Kong 2001.

[8] I. Herman and D. Dardailler. *"SVG Linearization and Accessibility"*, Computer Graphics Forum, 21(4), 2002.

[9] John A. Gardner & al., *"The Problem of Accessing Non-Textual Information On The Web"*, Proceedings of the 1997 Conference of the W3 Consortium, CA, April, 1997.

[10] J.L. Bentley, "*Little Languages*", *CACM*, **29**(8), 1986.

[11] E. Tufte, "*The Visual Display of Quantitative Information*", Graphics Press, 1983, and Edward Tufte, "*Envisioning Information*", Graphics Press, 1990.

[12] G.L. Lohse, K. Biolsi, N. Walker and H.H. Rueter, "*A Classification of visual representations*", CACM, **37**(12), 1994

[13] Web Accessibility for All, "*How to create Descriptive Text for Graphs, Charts & other Diagrams*", http://www.cew.wisc.edu/accessibility/tutorials/descriptionTutorial.htm

[14] M. Cornelis and K Krikhaar, "Guidelines for Describing Study Literature", FNB Amsterdam, 2001.

[15] K. Marriott, et al., "Towards flexible graphical communication using adaptive diagrams". Advances in Computer Science-ASIAN'04, pages 380-394., Dec. 2004.

[16] D.J. Duke, Drawing Attention to Meaning, Adaptive Displays Conference, 2004.