# Semantic Web Data Visualization with Graph Style Sheets

Emmanuel Pietriga

## ▶ To cite this version:

to read, understand and modify them. Yet, these diagrams can be enhanced significantly by making a better use of available visual variables (layout, color, font, stroke, icons and shapes, etc.) which are greatly underutilized in default node-link representations.

## 2 Graph Style Sheets

Most tools representing RDF graphs as node-link diagrams either render them in a predefined way or offer very limited customization capabilities. The Graph Style Sheet language (GSS) is a style sheet language that makes it possible to transform the default node-link diagrammatic representation through the declarative specification of visibility, layout and styling rules applied to its nodes and arcs. Graph Style Sheets can be considered as graph transformations of a specific kind, although much simpler than most of these systems: Graph Style Sheets are not general graph rewriting systems in the sense that GSS rules do not replace matched subgraphs with other arbitrary subgraphs; the GSS transformation process is about taking the "default" representation of an RDF graph displayed as a node-link diagram, as defined by [W3C 2004], and modifying the visual appearance of its elements. It is thus conceptually much closer to the process of applying the now ubiquitous CSS style sheets [W3C 2006] to the default rendering of HTML pages.

In the "default" node-link diagram representation of RDF graphs, resources are depicted as ovals, literals as rectangles, and statements as arcs. The path of arcs, as well as the layout of elements, are highly dependent on the underlying graph layout algorithm. Topological relationships do not bear any semantics (except for the connection of nodes by arcs). Neither do several visual variables such as color, font, shape border thickness and stroke pattern (solid, dotted, dashed, etc.), thus giving significant freedom to customize the visual rendering of RDF graphs. A GSS rule is composed of a *selector* on the left-hand side, associated with a *styling instruction set* on the right-hand side. GSS's execution model is similar to CSS's, though applied to a directed labeled graph instead of a tree: given the set of rules defined in a stylesheet, the GSS processor walks the entire graph (i.e., all nodes and arcs), and evaluates relevant rules on them. If the selector of a rule matches the current node (or arc) in the graph, the corresponding set of styling, visibility and layout instructions is applied to that node (or arc).

Stylesheet designers are typically expected to create one stylesheet per RDF vocabulary, though this is not a requirement. As RDF models often make use of several vocabularies to describe resources, GSS features a cascading mechanism similar to that of CSS that makes it possible to apply several style sheets to the same model. These style sheets will often act on different elements of the graph, but conflicts between rules may arise and are handled through the computation of a specificity metrics for each rule.

Significant changes have been made to the GSS language over the last three years, aimed at enhancing the language, based on user feedback and on our own experience developing style sheets. The original selector language used in the first version of GSS [Pietriga 2003] has been replaced by a more expressive and more programmer-friendly language [Pietriga 2005]. Additional styling capabilities have been introduced, including the customization of
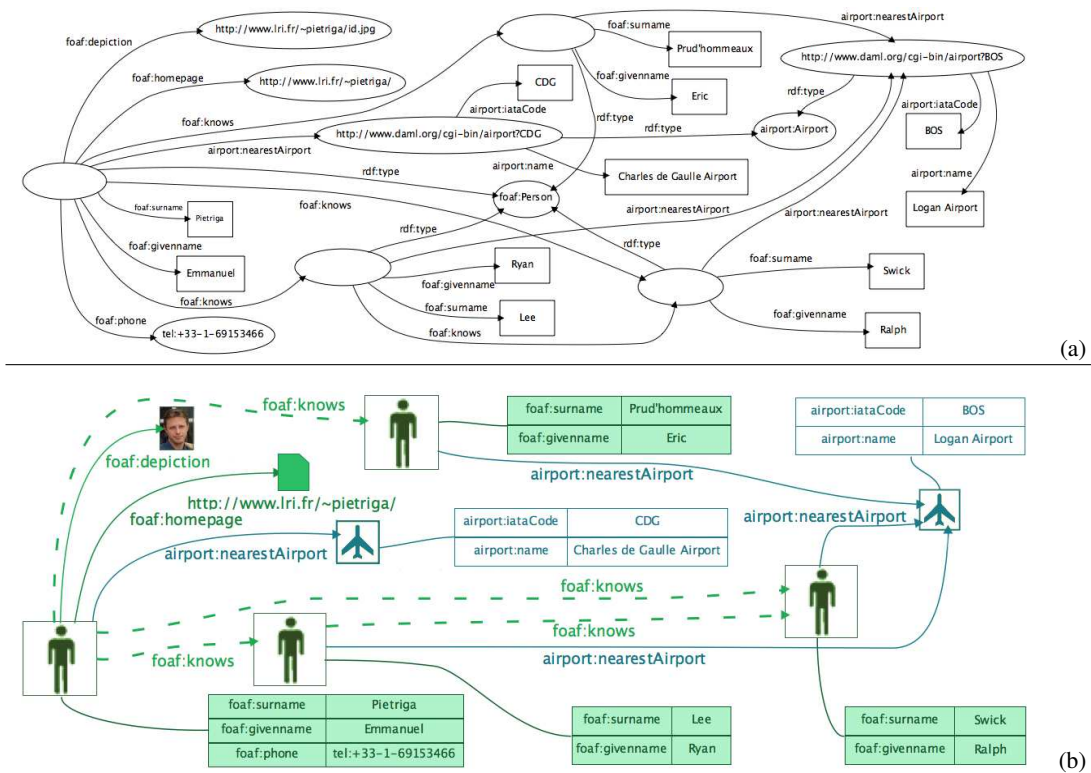
Figure 1: Default (a) and GSS (b) rendering of a Friend-of-a-Friend model with some DAML Airport data

arc endpoints, control of node sizes, simple textual content manipulation, and the use of semantic zooming to control the quantity of information displayed for a given element depending on the screen real-estate available for its rendering. The following section briefly presents some of GSS's capabilities on a simple example.

## 3 Example of Use

The Friend-of-a-Friend (FOAF) RDF vocabulary (http://www.foaf-project.org) is used to represent information about people (contact information, current projects, workplace, etc.) and about the social connections between them. Our example RDF model makes use of both this vocabulary and the DAML Airport vocabulary to provide information about airports nearest to the people described. In the default representation of this model (Figure 1-a), all resources are depicted as ellipses, literals as rectangles and statements as arcs. Figure 1-b illustrates the GSS rendering of the same graph.

Two GSS style sheets are applied to the original model: the first one applies to FOAF elements while the second one applies to DAML Airport elements. The style sheets make it easier to get a general understanding of the graph and to extract information without having to go in the low-level details of the representation. For instance, all nodes and arcs representing class membership information have been removed, but this essential information is still conveyed, as resources are now depicted by icons or shapes representative of each class (Airport, Person, Document). Nodes representing persons have been made larger than other nodes as they represent the central elements of the network. Styling elements is however not only about making a better use of available visual variables for representing information, but also about providing visual clues to improve the legibility of representations. Other styling instructions thus include using similar color hues for the elements of a given vo-

cabulary (green for FOAF, blue for Airport), and grouping related information for a given resource in tables (e.g., contact information about a person). The code associated with these styling rules can be found in style sheets available on the Web[1][2].

## References

MC SCHRAEFEL, SMITH, D., OWENS, A., RUSSELL, A., AND HARRIS, C. 2005. The evolving mSpace platform: leveraging the Semantic Web on the Trail of the Memex. In *16th ACM Conference on Hypertext and Hypermedia*.

PIETRIGA, E. 2003. Styling RDF Graphs with GSS. *XML.com* (December). http://www.xml.com/pub/a/2003/12/03/gss.html.

PIETRIGA, E., 2005. Fresnel Selector Language for RDF, November. http://www.w3.org/2005/04/fresnel-info/fsl/.

QUAN, D., AND KARGER, D. 2004. How to Make a Semantic Web Browser. In *Proceedings of the 13th international conference on World Wide Web*, 255–265.

RUTLEDGE, L., VAN OSSENBRUGGEN, J., AND HARDMAN, L. 2005. Making RDF Presentable: Selection, Structure and Surfability for the Semantic Web. In *Proceedings of the 14th international conference on World Wide Web*.

W3C, 2004. Resource description framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/rdf-concepts/.

W3C, 2006. Cascading Style Sheets, April. http://www.w3.org/Style/CSS/.

---

[1]http://www.w3.org/2001/11/IsaViz/gss/foaf/foaf2.gss
[2]http://www.w3.org/2001/11/IsaViz/gss/airport/DAMLAirport.gss