The Briefing Associate: A Role for COTS applications in the Semantic Web

Marcelo TALLIS *mtallis@teknowledge.com*
Neil M. GOLDMAN *ngoldman@teknowledge.com*
Robert M. BALZER *bbalzer@teknowledge.com*

Teknowledge Corporation, 4640 Admiralty Way, Suite 231, Marina del Rey, CA, U.S.A.

Abstract. This paper identifies a set of semantic markup capabilities designed to benefit the author rather than the consumer of manually composed documents. By doing so it addresses one of the major challenges facing the semantic web vision – the generation of ontologically encoded descriptions of the content of manually produced documents. It also presents a novel approach to eliminating the currently large and tedious overhead required to produce such markup by augmenting the COTS tools that users already use to produce these documents so that the semantic markup is derived as a byproduct of composing the document. These ideas are currently being implemented in a tool called the Briefing Associate that augments Microsoft's PowerPoint to support the authoring of semantically grounded briefings.

1. Introduction

The semantic web promises to expand the services of the existing web by enabling software agents to automate procedures currently performed manually and by introducing new applications that are infeasible today. The enabling factor to materialize this vision is the availability of web documents containing ontologically encoded information that software agents and tools can accurately and reliably interpret. A major challenge facing the semantic web vision is the generation of this encoding, especially for encoding or summarizing the content of documents composed by people. The mark-up of such documents is currently a tedious and sometimes complex activity. Because the benefits of these markups accrue most immediately to the agent-assisted *consumers* of the web content, content *producers* are not highly motivated to undertake this extra effort.

Although there has been considerable technical progress in supporting other portions of the semantic web lifecycle, there has been little progress in the markup of manually composed documents. The prevalent approach is to create specialized tools that specifically support the association of semantic markups with the content of existing documents [1],[2]. These tools provide a GUI that permits an author to browse ontologies, find appropriate terms, generate syntactically correct markups,

and associate them with (portions of) the document's content. This activity remains an "extra" effort that does not directly reward to the person performing it.

We are experimenting with a different approach. Rather than add ontological encoding to completed documents, we propose to *augment* the COTS tools that users *already* use to produce these documents to produce the ontological encoding as a *byproduct* of document composition. The intent of such augmentation is to (nearly) eliminate any cost of producing ontologically encoded documents beyond the costs inherent in producing the equivalent semantics-free version. We are also exploring the incorporation of analysis and synthesis tools that utilize these semantic markups during document composition to improve the resulting document's accuracy, quality, and/or speed of production. Authors themselves will thus reap a direct benefit from creating documents with associated ontological encoding. Integrating this functionality into the COTS tools that authors already use, without restricting their use of the tools' existing functionality, obviously simplifies the transition to this paradigm.

This paper describes the Briefing Associate, an application of this approach tailored to the creation of briefings using Microsoft PowerPoint. The Briefing Associate augments PowerPoint's native GUI with graphics that represent concepts and relations imported from an ontology. The concepts and relations from the ontology also define a set of attributes authors can fill in through popup dialog boxes. The author builds a briefing in PowerPoint using a combination of these ontology-related shapes and connectors and native PowerPoint elements. As a byproduct of building the briefing the author is also describing the relationships among concept instances. Each ontology-related graphic represents an instance of a concept. The ontology-related connectors between graphics stand for relations between the associated instances. The ontological encoding created as a result of using the augmented GUI is stored persistently within the PowerPoint document.

As a briefing is being composed, this evolving semantic model of instances, relations, and attribute values is shared (through Microsoft COM interfaces) with external tools called *analyzers.* These tools can process the semantic model to determine whether it is consistent and complete, perform an analysis to determine derived properties, or augment it with additional information. The Briefing Associate provides a mechanism for these external *analyzers* to add visual annotations (e.g., highlights) to the graphics that stand for model elements to provide feedback to the author. It also provides a synthesis mechanism to these tools for augmenting the author's semantic markups.

These analyzers can be general or domain specific and can be individually activated or deactivated by the author through the Briefing Associate's enhanced GUI. One particular generic analyzer is a *publisher* that exports the semantic content (i.e. the markup) of the briefing in the DAML+OIL language [3]. Using a generic *briefing* ontology, it also exports the briefing meta-information (author, date, size, etc) and all *titles* and *text* appearing in the briefing.

We are also implementing a generic analyzer that imports the ontological encoding resulting from a semantic web query into a briefing and renders it graphically. It will persistently associate the query with the resulting model elements, allowing the query to be reused in the future to keep the briefing up to date.

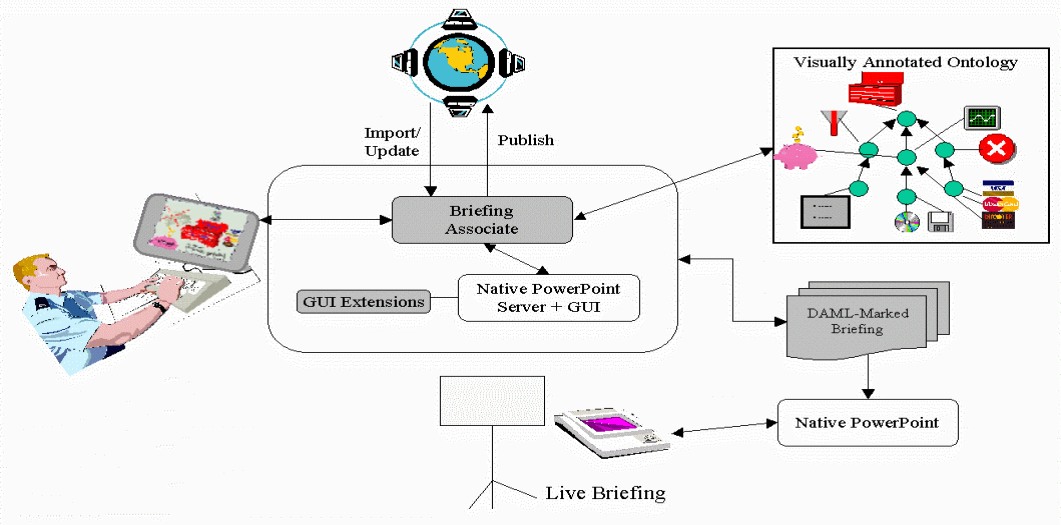

Figure 1 Briefing Associate Software Architecture

In the following sections we describe the Briefing Associate and how the ontologically encoded descriptions are introduced into a briefing, how the Briefing Associate interacts with external analyzers, and the current implementation of these ideas.

2. Briefing Associate

The *Briefing Associate* (BA) facilitates the composition and publication of semantically grounded briefings. The briefings contain markups that describe the domain-specific content matter of the briefing and are linked at a fine granularity to units of visual content in the briefing. A briefing may contain both *original* and *imported* semantic content. The BA generates DAML descriptions of a briefing's original content as a *byproduct* of creating that content's visual depiction. The creation of DAML markup for original content is mediated by *visually annotated* DAML ontologies (VAOs) from which authors select ontologically defined objects as predefined graphic shapes or icons to include in their briefing. These visually annotated ontologies are demand-loaded into the BA to specialize it to a particular subject-matter domain. They also permit the BA to generate graphical depictions of imported semantic content. Content imported from agents will be marked with the source agent and query used to obtain the content, permitting the BA to obtain, on request, an updated version of that content from the same agent.

The BA is implemented as an extension of Microsoft PowerPoint. Briefing authors familiar with that product can continue to rely on the native user interface tools, menus, and direct-manipulation actions to edit visual content. Extended interpretation of these tools and actions, and additional tools created from the ontology annotations, simplify the creation of new content, while simultaneously creating DAML markup. Figure 1 depicts the BA's architecture and major information flows.

The Briefing Associate augments PowerPoint with graphics-bearing ontological categories. These graphics represent instances of domain concepts, attributes

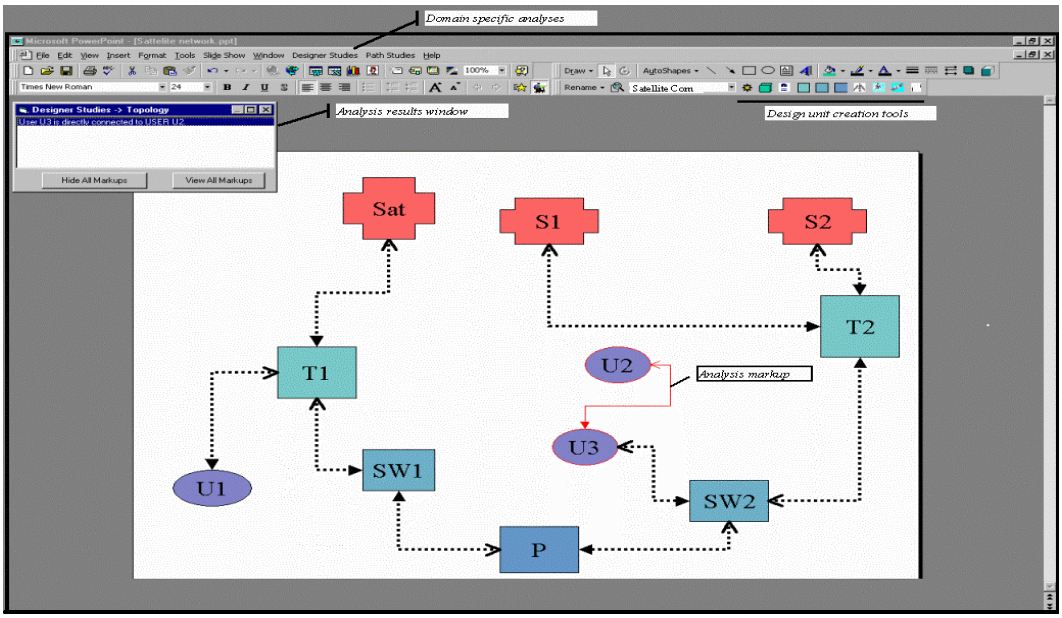

Figure 2 Ontology aware briefing editor GUI – satellite communications

(primitive data typed properties), and their relationships. The author, while composing a briefing using these graphics is indirectly constructing a semantic description of the briefing content. Besides supporting the construction of semantically grounded briefings, the Briefing Associate also exposes the briefing's emerging semantic descriptions to external modules called *analyzers* that perform specialized services or *analyzes for the author*. These analyses can provide feedback to the author, can extend or modify the briefing, or can produce external documents. One particular generic analyzer is a *publisher* that generates the semantic markups that describe the briefing content. The Briefing Associate extends the PowerPoint GUI with tools, menus, and gestures for instantiating the semantically annotated graphics, assigning attribute values to the instances and relations represented by these graphics, invoking analyses, importing and updating the graphic representations of imported semantic descriptions, and annotating domain ontology concepts and relations with their visual representations. The following subsections describe the components that achieve these added services.

3. Ontology-Aware Briefing Editor

The Ontology-Aware Briefing Editor allows a briefing author to create original content and to import and edit externally produced content. Visually annotated ontologies, discussed below, provide the means to relate DAML descriptions from a given ontology to a visual model.

Ontology-aware editing takes place through a combination of standard PowerPoint interface actions, additional GUI elements added by the BA, and extended interpretation of native controls and direct-manipulation actions. The entire native PowerPoint user interface continues to be functional. User-preference tailoring of that interface is preserved.

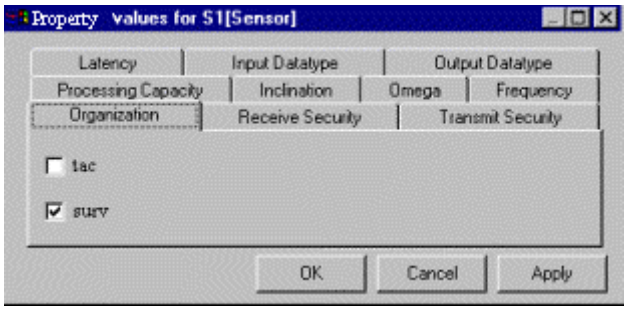

Figure 3 Property value dialog

A visually annotated ontology is the key to creating original content as well as to automatically depicting imported content. A new toolbar is added to the PowerPoint GUI for the ontology. For each concept and relation in the ontology, an *instantiation tool* is added to the toolbar. Our current implementation lays these tools out in a single list. We also plan to offer these tools in a cascading interface, mirroring the class inheritance of the ontology. Clicking on one of these tools, like PowerPoint's native autoshape tools, allows the author to insert a copy of the graphic template anywhere in his briefing. Domain relations defined in the Ontology are graphically depicted by arrows (more precisely PowerPoint connectors) whose ends are attached to the concept instances related by that relation. These instantiation tools simultaneously create the internal semantic representation for that concept or relation instance as defined by the ontology (including any default attribute values).

The Ontology-aware briefing editor also allows the author to edit these domain attribute values through a dialog box interface that is activated from the context menu of the graphic representing that instance in the briefing. A *tabbed dialog* is created for the selected instance with a tab for each attribute applicable to that instance. Each tab provides an interface, specific to the attribute type, for viewing and setting the value of that attribute.

We plan to augment this textual interface with a graphical one that enables some of an object's attributes to be modified by direct manipulation of the object's graphics (e.g. changing the size of an object might modify some aggregate value such as the length of a queue, and changing its color from a list of alternatives might modify some enumerated type such as its state). The correspondence of these direct manipulations to the attribute affected will be defined by additional visual annotations of the ontology.

Figure 2 is a screen shot of the ontology-aware briefing editor in a "satellite communications" domain. Everything in the figure is part of the GUI with the exception of the callouts highlighting specific elements.

In the central canvas is the depiction of a "satellite communications" configuration. The various labeled shapes represent instances of satellites, terminals, switches, processors, and users – the domain concepts defined in the ontology. They are connected by arrows representing communication links – the only (non data-typed) domain relation in this ontology.

The author created the preponderance of this briefing through the instantiation tools on the domain toolbar, on the right side of the second row of docked toolbars at the top of the figure. To the immediate left of these instantiation tools in the

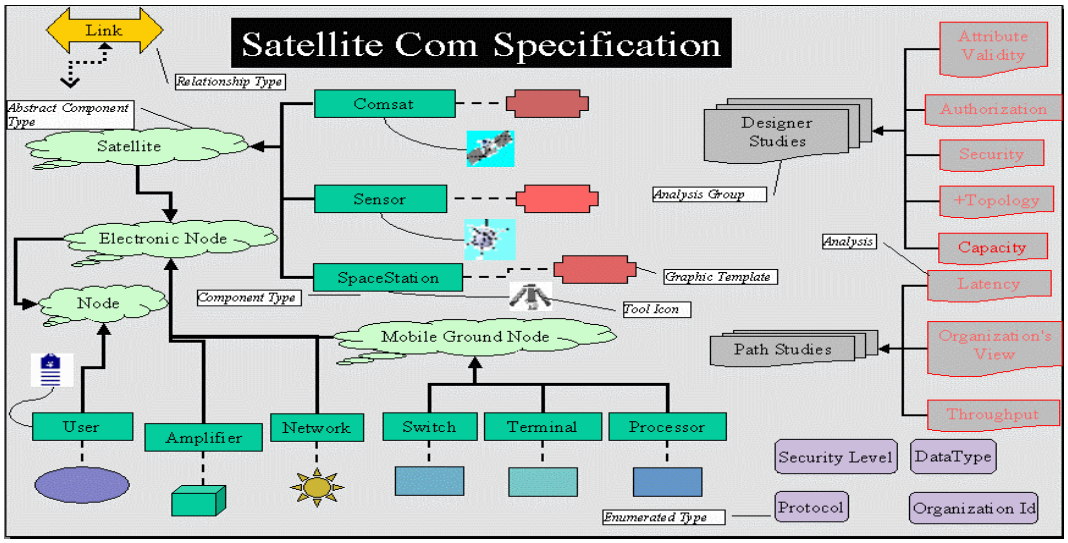

Figure 4 Visually annotated ontology – satellite communication

domain toolbar is a drop-down list box displaying the name of the current ontology ("Satellite Com"). When a briefing author starts a new briefing, this box allows him to choose an ontology. This triggers the creation and display of the appropriate domain toolbar for that ontology. Manipulation of the concept and property instances on the briefing – positioning, resizing, selecting, attaching/detaching links– is carried out through PowerPoint's native mouse gestures and/or keyboard shortcuts.

In Figure 2 the user has requested a "topology" analysis, one of the analyses in the "Designer Studies" group. The results of this analysis are displayed in a separate window, visible at the upper left of the canvas in Figure 2. The window displays a list of reports. In this example, there was just one report. Its explanation reads "User U3 is directly connected to user U2." When the user selects one of these reports, its associated *markups* are displayed as highlights. In this case, the only markup called for highlighting the communication link between U2 and U3. That is why that link has an appearance (a thin red arrow) different from the others. The effect of this highlighting is reversed when the report is deselected or the analysis window is closed.

Attribute values are viewed and assigned through dialogs, displayed on demand from the graphic instance's context menus. Figure 3 exhibits the dialog for a sensor satellite. The dialog contains a "tab" for each attribute associated with that concept in the ontology. The details of a tab depend on the value type of the attribute and on the concept specification.

Identical dialogs are used to gather the parameter values for parameterized analyses.

4. Visual-Annotation Ontology Editor

The Briefing Associate is not limited to any particular ontology--any DAML ontology can be annotated. Multiple annotated versions of a single ontology may be created, so that briefings can be tailored easily to different audiences with different

conventions for the visual representation of information. However, any single briefing will be based on a single visually annotated ontology.

The Visual-Annotation Ontology Editor provides an interactive means to establish a mapping between the concepts of an ontology and their visual representation. When an ontology O is imported into the VAO editor, the editor lays out O's concepts and properties depicting their hierarchical relationships (i.e., the subclass and subproperty properties). The user can assign graphic representations to these concepts and properties and assign icons to be used in the ontology tool bar used for briefings to be associated with O. The user also indicates the analyses that briefing authors will be able to invoke from the ontology-aware briefing editor through its GUI.

The VAO editor is actually a specialization of the ontology-aware briefing editor that uses the visual annotations defined for the *ontology* domain. These visual annotations allow the object and relation types in an ontology to be defined graphically. These ontology annotations are thus just briefings in this *ontology* domain and are saved as a PowerPoint presentation. They are loaded on demand by the Semantic Content Import and Update and Ontology-Aware Briefing Editor components. Importing an ontology is thus a case of content import, while adding visual annotations is a case of original content creation.

Figure 4 shows the Satellite Communications visually annotated ontology used in the example of Figure 2. The (green) rectangles labeled "Comsat", "Sensor", "User", etc. represent the leaf domain concepts. The cross shapes attached to them by dashed connections are their *graphic templates*. This determines the appearance of an instance of that concept. Any of PowerPoint's native autoshapes, formatted as desired, may be used as a graphic template. Alternatively, an image may be chosen as a graphic template.

A concept may be connected (via a curved solid connector) to an image that serves as the *tool icon* for that concept in the instantiation toolbar. Tool icons, like graphic templates, may be selected from a shape library or may be imported. If no tool icon is specified, a scaled version of the graphic template is used as the tool icon.

The (light green) clouds labeled "Satellite", etc. represent the non-leaf concepts. The (gold) arrow shape labeled "Link" defines the sole relationship type in this domain. The dashed, double-headed arrow attached to it is the graphic template for the "Link" relationship type. The user tailors the color, dashing and arrowhead styles of a relationship template in the graphic domain specification just as he tailors component type templates.

Any concept or relationship type may have initial attribute values specified through a property-editing dialog, identical to the ones used by the ontology-aware briefing editor. The default values are assigned when new instances of the type are created.

Figure 4 contains the specification of two analysis groups, "Designer Studies" and "Path Studies", and eight analyses in those analysis groups. The color and styling of the border of an analysis specify the means used to highlight components and relationships identified in *reports* in the feedback from the corresponding analyzers. For instance, the "U2-U3" connection in Figure 2 was highlighted as a thin red line because the border of the "Topology" analysis is a thin red line. Analogously, the text characteristics – font, face, size, color – of the label of an

analysis specify the textual characteristics of any markup text found in feedback from the analysis.

5. Semantic Content Import and Update

At the time this paper was written no generic semantic content import and update component suitable for any arbitrary ontology has been implemented. Instead a series of ontology-specific semantic content import and update components tailored to a particular domain have been created. The following is a description of the generic component we plan to implement.

The Semantic Content Import and Update component will allow the author of a PowerPoint presentation to contact DAML-aware agents, including search agents, and post queries to those agents. It will accept, as DAML descriptions, the results of those queries. The queries as well as the imported descriptions and meta-information will be incorporated as a persistent part of the presentation.

The component will determine how these objects should appear (i.e. be rendered) within the briefing, as specified in the respective ontology annotations to represent the imported DAML content. To do so, it will size, color, label, and place graphic renditions of these objects and interconnect them with one another. These graphic renditions will become a persistent part of the presentation, associated with the specific description units that they depict. The author will generally need to adjust the sizes and positions of these graphics to produce acceptable layouts.

The component will also provide information update capabilities, allowing information updates on demand through a menu item added to the PowerPoint user interface. Using the retained queries, the component will re-query the source agent(s) to retrieve updated content and generate an updated version. At the author's discretion, the component will visually correlate the two versions. The author may choose to incorporate the updated version as a whole, or to selectively incorporate changed information.

The author will also be able to request that any manual customization of the graphic rendition of imported content be reapplied to updates of that imported content so that it doesn't have to be reapplied manually.

6. Briefing Associate - analyzer interaction

Analyzers are external executable modules that process the internal semantic descriptions of the briefing content to provide an analysis, a synthesis, or some other service. An analyzer can be implemented so as to execute within the PowerPoint process, as a separate process on the same machine, or (via DCOM) on a different workstation. Analyses are associated with a particular domain and this association is indicated in the VAO and they are invoked through the BA editor menu for that domain. When an author requests an analysis, the BA creates a connection to the module implementing that analysis and passes it a reference to the briefing to be analyzed, together with any author-provided parameters for the analysis. That analyzer is subsequently expected to send the BA a set of reports

describing the analysis performed, the synthesis done, or the service rendered. The BA then presents the report(s) to the author.

For a *snapshot* analysis, the analyzer's responsibility ends with transmission of the reports detailing that analysis. An *incremental* analysis, however, is expected to send updates to its reports as the author continues to modify the briefing, until the author closes either the analysis or the briefing. To support incremental analyses, the briefing reference handed to the analyzer by the BA provides not only direct access to the *content* of the briefing, but to *events* representing changes to that content.

A *transaction* grouping is imposed on top of events. It is these transactions, not primitive events, that represent the unit of change to which an incremental analyzer commits to respond with updated analysis reports. Because the responses are permitted to be asynchronous, they are accompanied by the transaction id of the transaction that triggered them. This allows the BA to understand, and reflect in its GUI, whether a displayed set of analysis reports is *up-to-date*.

Although the briefing reference provided to an analyzer can be used to gain direct access to PowerPoint's detailed *graphic* model of a briefing, analyzers are typically interested in the *ontology-based* model information that is being automatically generated when content is imported from the semantic web or created through tools associated with the VAO. For each ontology, a COM *type library* is automatically generated. This type library reflects a straightforward mapping between classes and properties of the ontologies and the corresponding modeling concepts (classes, interfaces, and properties) of COM. Most, if not all, widely-used programming language IDEs for the Windows platform provide a declarative way to *import* such a type library, automatically building the client-side code needed to program directly in terms of the objects exposed by the library.

7. Implementation

The Briefing Associate is a descendent of the Design Editor [4], an application for producing visual domain-specific design environments. The Briefing Associate, like the Design Editor, is implemented as an extension of Microsoft PowerPoint. We regard this choice not as an implementation detail, but as central to this research. First, PowerPoint provides us as implementers with a far higher-level platform for building a briefing tool than generic middleware, such as COM/CORBA and GUI widget libraries. It provides an extensive ontology for representing the visual content of briefings, and support for making models that use that ontology persistent. Furthermore, it provides an extensive WYSIWYG user interface for viewing and editing the visual content of a briefing. This interface requires some extension, but no redesign or reimplementation, to accommodate DAML-aware briefings. Second, PowerPoint is the most widely used product for authoring briefings and hence it facilitates the adoption of the BA by briefing authors.

The BA is programmed primarily in Visual Basic. For PowerPoint 2000, this extension is a COM addin that receives "events" as the user creates, opens, closed, and modifies briefings. As a client of PowerPoint, this module can navigate through a briefing and paint analysis feedback directly onto it. For efficiency reasons, this

module runs entirely as an "in-process" component. This means it is incorporated into the PowerPoint process itself. Method calls are extremely efficient when both client and server are part of a single operating system process. Greater efficiency could be achieved by implementing the BA in C++, but the performance of the Visual Basic code has been acceptable to date.

PowerPoint's native extension mechanisms include a general, albeit low-level, ability to add arbitrary non-graphic information to a presentation and retain that information in the presentation's persistent file format. The BA implementation relies on this mechanism to retain all ontology-related information about a presentation across editing sessions – it does *not* attempt to infer ontological information on the basis of graphic attributes of existing graphic objects.

8. Rewarding the Briefing Associate Adopter

The Briefing Associate's authoring environment is an enhanced PowerPoint, the same environment most briefing authors already use. The enhancements do not *remove* capabilities, do not *necessitate* the use of new means for accomplishing old goals, do not *alter* the visual appearance of the ultimate product, and *do not* impose perceivable delays in processing speed. Thus, the Briefing Associate doesn't impose any extra impediments or costs on producing briefings with the standard tools in the standard way.

But we need to motivate the briefing author to use the Briefing Associate's markup tools. The biggest benefits of such markup will obviously accrue to the consumers of these briefings who will be able to quickly and accurately find specific content in those briefings because they have been semantically annotated. Realistically, we should never expect people to incur significant costs, whether in time, retraining, or reduced product quality, on the basis that some benefit *might* accrue in the future, especially when that potential benefit accrues to others.

We have therefore added several enhancements to the Briefing Associate that provide immediate benefit to the briefing author to motivate the use of the Briefing Associate's semantic markup capabilities:

- The Briefing Associate simplifies the construction of the briefings because authors will have readily available the graphic templates that they repeatedly use to represent objects of the domain.
- Ontology-based descriptions of a briefing's content are generated as a side-effect of briefing composition. The extent and value of such descriptions, however, depends on the extent to which the author makes use of the extensions offered by the BA.
- Generic and domain specific *analyzer* tools exploit the semantics of the briefing content to provide an analysis, synthesis, or other service for the author while the briefing is being created. Although such analyzers are not inherently tied to the semantic web, their implementations might well make use of web-based agents that *consume* the content of a briefing and provide feedback to the author.
- Ontology-based descriptions of briefing meta-data and textual content are produced at no cost and independent of the use of any extensions.

- The BA's extensions for importing and visualizing semantically marked-up content could be a significant time-saver in constructing certain classes of briefing. Since these facilities are designed to rely on queries *posted to* the semantic web, however, they server to *leverage*, rather than to *bootstrap*, the semantic web vision.
- The BA will automate the update of content that originated in the semantic web. Like import and visualization, this is a leveraging rather than a bootstrapping relationship of the BA to the semantic web.

These BA author-enhancements just embed the semantic web lifecycle into the briefing creation process so that briefing authors themselves can enjoy (some of) the benefits of semantic markup.

Ontology-based annotations will turn briefings into reusable resources. New content as well as novel aggregations of imported content will be published in a form accessible to DAML-enabled agents. Linking the graphic content to the semantic content in the published briefing will foster reuse of the visual as well as the semantic material.

The automated content update facilities of BA will transform briefings from information snapshots, whose value declines as the information in those snapshots becomes dated and obsolete, into renewable resources whose information can be automatically updated as needed.

The automatic generation of visual depictions for imported material, and ontology-specific interface editing extensions may actually *reduce* the effort needed to compose the visual content of a briefing, even though briefings will contain non-visual semantic content as well as traditional graphic content.

9. Related Work

Several initiatives aiming to establish a global semantic markup scheme for the web are currently being undertaken. The oldest and most widely adopted is the Dublin Core Metadata Initiative (DCMI) [5]. The DCMI has the goal of facilitating the discovery of electronic resources in the web. Its primary offering is the Dublin Core Metadata Element Set, a set of fifteen elements like Title, Creator, Subject, and Date that is used to describe web resources. The Dublin Core Metadata Element Set is the de facto worldwide standard for the description of information resources across disciplines and languages and has already been translated into 25 languages.

Newer undertakings like the European Community sponsored Ontobroker [6], its successor OntoWeb [7], and the DARPA sponsored DAML [8], go beyond the DCMI goals. Rather than annotating electronic resources to merely facilitate their discovery, these projects aim to describe electronic and real world entities using a machine understandable language that enables autonomous software agents to accurately understand and process their content [9]. Our BA is being developed under the DAML program.

There are two dimensions along which we can state requirements over semantic markup generator tools. The first dimension is the granularity of a description unit. This dimension ranges from coarse descriptions that relate a whole document with

a set of predefined conceptual categories to detailed descriptions of a document's content. The second dimension is the degree of regularity of the generated descriptions. This dimension ranges from highly regular data usually supported in relational databases to descriptions of highly unstructured and irregular information like the content of newspaper articles. The BA is aimed at detailed descriptions of irregular and unstructured documents.

The above dimensions are useful for comparing the BA with other tools for generating semantic markups. One kind of metadata generator tool is The Nordic DC metadata creator [10]. The Nordic DC metadata creator is a metadata editor for the Dublin Core Metadata Initiative. It consists in a Java applet that displays a form where users can fill in the values corresponding to the Dublin Core Metadata Element Set. The Nordic DC metadata creator generates a syntactically correct encoding of these values that a user can attach to the described document. This tool corresponds to the less elaborated form of semantic markups: Coarse descriptions based in a predefined set of conceptual categories. Another kind of tool is represented by Klarity [11], a metadata generator tool that supports the Dublin Core Metadata Element Set. Klarity is a tool that can automatically generate metadata for HTML pages based on the concepts found in the text. It uses statistic methods to allocate values based on the concepts it has identified from the 'seed' or exemplar documents that are significant to the concept in question. Klarity is another example of a tool that generates coarse metadata descriptions of documents.

A different approach for metadata generation is represented by ITTALKS [12]. ITTALKS is a portal for announcements about talks, seminars and colloquia related to Information Technology that is part of the DAML program. Although not its main focus, ITTALKS is able to generate DAML descriptions from the talks contained in its database. In this sense, ITTALKS is an example of a tool that generates descriptions from highly structured data.

Closer to the scope of the BA are the Annotation Tool of the KA^2 initiative (under the Ontobroker project) [1] and the Knowledge Annotator of the Shoe project [2]. These tools offer a GUI for authoring and attaching semantic annotations to web documents. They make available context sensitive instances and ontology browsers that facilitate the authoring of semantic descriptions. A second incarnation of the KA^2 annotation tool can also generate annotations semi-automatically from lexical analysis of text plus a vast word and domain lexicon.

These approaches contrast with the Briefing Associate in that the BA generates these markups as a byproduct of constructing the document and hence do not require the users to perform any extra activity. Additionally, because the semantic annotations are embedded in the original document (instead of being inserted in a second step using a different tool) modifying the original document does not lose the existing annotations. In the other hand, the BA approach might not be adequate for marking up existing documents that do not use the BA conventions for representing ontological relationships, and for documents whose type is not supported by the underlying COTS product.

Although the semiautomatic markup generation feature of the KA^2 annotation tool simplifies the production of semantic annotations, it stills constitutes an extra activity because the users need to check and revise the generated annotations. Furthermore, this approach is limited to textual documents that contain enough information as required to infer their semantic relationships. This limitation might

certainly exclude briefing documents because they usually contain diagrams that are not self-explained within the text.

10. Conclusions

The overload incurred in annotating documents with semantic markups should be kept to a minimum. With this goal in mind we have developed a new paradigm for facilitating the generation of semantic descriptions of the document content. Our approach consists in augmenting the same COTS products that users normally use to compose these documents with natural representations of domain ontology entities. By using these representations in composing a document, the user is simultaneously generating a semantic description of its content. We also suggest that in order to motivate document author's transition to this paradigm, these added semantic descriptions should be exploited by author-oriented tools that help improve the resulting document's accuracy, quality, and/or speed of production. It should be noted that this is just embedding the semantic web lifecycle into the document creation process so that document authors can enjoy (some of) the benefits of semantic markup.

We have implemented these ideas in the Briefing Associate, an extension to the Microsoft PowerPoint that reflects an internal semantic model of a briefing from its graphic representations of domain instances and their relationships. The Briefing Associate can produce DAML descriptions of a briefing content as a byproduct of creating the graphic content of that briefing. That semantic content can be analyzed for consistency and completeness to improve the briefings accuracy, quality, and speed of production. In addition, the Briefing Associate can automatically create graphic depictions from imported DAML descriptions, and these graphic depictions can be updated on demand to reflect changes in the imported DAML content.

References

[1] Erdmann, M., Maedche, A., Scnurr, H.-P., and Staab, S. From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools. *To appear in the Linköping Electronic Articles in Computer and Information Science.* http://www.ida.liu.se/ext/epa/cis/2001/002/tcover.html. 30.December, 2001

[2] Heffin, J, Hendler, J., Luke, S. SHOE: A prototype Language fro the Semantic Web. *To appear in the Linköping Electronic Articles in Computer and Information Science.* http://www.ida.liu.se/ext/epa/cis/2001/003/tcover.html.10.March.2001

[3] DAML+OIL (March 2001). http://www.daml.org/2001/03/daml+oil.daml

[4] Goldman, N. and Balzer, R. The ISI Visual Design Editor Generator, Proceedings of the 1999 IEEE Symposium on Visual Languages, Tokyo, Japan, September 13-16, 1999, IEEE Computer Society Press, pp. 20-27.

[5] Dublin Core Metadata Initiative. http://dublincore.org

[6] OntoBroker & Ontology Related Initiatives for the Semantic Web. http://ontobroker.semanticweb.org

[7] OntoWeb. http://www.ontoweb.org

[8] The DARPA Agent Markup Language Homepage. http://www.daml.org

[9] Tim Berners-Lee, James Hendler and Ora Lassila. The Semantic Web. *Scientific American.* May 2001. http://www.sciam.com/2001/0501issue/0501berners-lee.html

[10] Dublin Core Metadata Template. http://www.lub.lu.se/cgi-bin/nmdc.pl

[11] Klarity.http://www.klarity.com.au

[12] ITTALKS. http://ittalks.org