NetworkedPlanet

White Paper: Topic Maps In Web-site Architecture

An overview of approaches to apply topic maps to improve site cohesion, navigation and search

White Paper : Topic Maps In Web-site Architecture

Abstract

This paper outlines the role that topic maps can play in the Information Architecture and Systems Architecture of web-sites. The challenges in creating and maintaining content-rich web-sites are outlined from the perspective of the Information Architect, the Systems Architect and the Content Creator.

The main bulk of this paper focuses on the strategies for managing site navigation using topic maps and for integrating content from multiple sources on a topic maps-driven web-site. Our goal is to encourage further discussion about these strategies and the implementation patterns that support them.

Introduction

Three Challenges

For The Information Architect

One of the major challenges for Information Architects is the search for the "perfect" site architecture, however there is never any one right solution and the goal of the IA has to be to consider the needs of both the information publisher and the information consumer in developing an overall information architecture that is manageable from a creation and maintenance standpoint but also provides consumers with findable and accessible content.

For The System Architect

One of the principal challenges facing the System Architects responsible for the implementation of a web-site is in bringing together disparate sources of content, including written content; content from databases, partner feeds and other external systems; and public content from the Internet. While topic maps are not a data integration solution, they do provide a platform on which data integration can be easily built and which support a variety of strategies. We present 5 data integration strategies in this paper and discuss their relative merits and the situations under which each is most useful.

The other major challenge is change. As a web site matures new requirements come to light resulting in the need to make changes, not only in the content but also in the content organization and site navigation. By tackling the initial design with sensitivity to this issue and with a flexible information organization tool such as topic maps available, it is possible to create sites that can support ongoing IA requirements with minimal engineering impact.

For The Content Creator

In many cases, content is created on a "publish-and-forget" basis. The author of today's market report is not too worried about last weeks report. However, from the consumer's point of view, that "old" content is still accessible – from an archive of articles; from a search on the site; or from a search through an Internet Search Engine such as Google. For the content of an article itself, good writing style and clear document labeling can provide a stand-alone document that has a date context.

But, is stand-alone content enough ? A challenge facing writers for the web now is that an increasingly more sophisticated readership is looking for those links that the content creators can provide and is always interested not only in the document as a source of information, but also in the document as a launch-pad to related content. This is a problem for documents that are created in a write-and-forget mode. While the links added by an author might be the most relevant links for that week; there is no guarantee that they are still relevant a year later, or even a month later. Revisiting every document to keep links up to date is really not a solution either, so instead sites are turning to the use of content meta-data to generate relevant links.

Topic Map Solutions

A solution based on the ISO standard Topic Maps can help address each of the challenges. While not a complete solution in itself (as the Topic Maps standard is far more general and wide-reaching than a solution for web-site architecture!), the Topic Maps standard can provide a solid platform for the development of these solutions.

For The Information Architect

A Topic Map can be used to record the core ontology for a site – the things that the site is about and the relationships between them. Topic Maps do not force a particular model on the ontology, but many of the traditional IA organizational aids such as thesauri, code lists, hierarchies and faceted classification can all be modeled using a topic map. The decision about whether to surface some or all of the organizational structure of the site to the site user can be made independently of the creation of the ontology – in some cases the ontology is used primarily to drive content classification, rather than to inform site organization.

It is important to note that the Topic Maps standard is not based in any formal Information Architecture methodology and is general enough to support the chosen methodology of the practitioner, through the development of **Topic Map Patterns** that provide the structure for modeling a particular methodology in topic maps. NetworkedPlanet has already developed simple patterns for hierarchical classification schemes and extended these to support faceted classification. It is our belief that by developing more of these patterns and tools to support the patterns we can extend the palette of options for an IA using topic maps.

For The System Architect

As we will show in this paper, a topic map can serve as the data hub of a loose-coupled information architecture, allowing new data sources to be added and merged with the content and other data sources that drive the site. This sort of information integration is not new and has been the foundation of a number of different approaches to information portals over the years. Where we believe the Topic Maps standard does have some significant advantages are:

1. In flexibility – the Topic Maps model is more than capable of modeling entity-relational data and more complex associative data models. When there is a need for the central data hub to get down to this level of complexity in representing information from an external data source, topic maps are not lacking.

2. In interchange – the interchange syntax for Topic Maps is formalized as an ISO standard, allowing robust exchange of information between systems.

3. Support for context – the scoping mechanism of topic maps allows information stored in the topic map to be given a context. Typical uses can be to distinguish between “DRAFT” and “LIVE” data or to specify user access levels to filter out content.

For The Content Creator

For the content creator a well designed topic map solution can be extremely low-impact. In fact we will show how from the content creator’s point of view a topic map solution simply adds a requirement to tag new content with appropriate meta-data.

However, there is also more potential for tapping the knowledge of the content creators. A topic map can store information about arbitrary multi-way relationships between items (e.g. the fact that Company X and Company Y are joint partners in a venture with Government Z), and the topic map is flexible enough to allow the creation and, more importantly, the presentation of these relationships with little or no impact on the other parts of the presentation stack. What this can mean in practice is the ability to share high-level domain knowledge and to enrich that domain knowledge over time simply through the addition of data and without the need to change the engineering of the solution (with the inevitable reworking and delay that this would entail).

Taken to its logical conclusion, it has also been argued that there is a role for a content provider that makes available only topic map data that provides a high-level domain-specific index of resources that may or may not be under the control of the provider. There is a precedent for this model in the scientific publishing community where aggregators provide searchable indexes of journal articles that they do not necessarily own. The aggregator acts as a gateway to the purchase of content, taking a cut of the proceeds. An aggregator that could offer a high-level domain model, with the ease of use and increased density of highly-relevant document-to-document links that could be derived that model, might have a significant advantage in this field.

About This Paper

This paper focuses primarily on the high-level systems architecture for the integration of topic maps with a Content Management System (CMS) and some optional external data sources. We present approaches to integration of topic maps-based navigation and patterns for the use of topic maps as a data hub for the integration of information from other data sources.

Navigation Architectures

Topic maps are fundamentally about linking related items together and hence they should have a pivotal role in the navigation structure of a site. The navigation architecture determines the way in which the relationships stored in the topic map are used to present links to the user and the extent to which the topic map itself determines the structure of a page.

Topic Map On Top

In this approach, the topic map provides the structure for the site navigation and for page content.

The typical way in which this approach is applied is to have a “page per topic” and to combine generic navigation elements with more type-specific page layouts. In this approach, the names and occurrences are the principal source of data for the page content (we will discuss how data from external systems such as a CMS might be integrated later). The topic’s associations to other topics in the map are the principal source of outgoing links for the page.

The page layout template used in this approach is typically driven the type of topic being displayed, so a “Company” topic might have a different template from a “Person” topic.

The keys to a successful application of this approach for navigation are two-fold:

1. A robust method for displaying “single hop” relationships (i.e. the direct associations between the current topic and other topics in the topic map)

2. Type-specific, ontology-based queries for retrieving more deeply linked material. For example, if the focus of a page is a commercial organization, a "Related Companies" box might list all companies that are active in the same market sector as this company, so that is a two-hop relationship (company-to-sector followed by sector-to-company). An application framework should allow these queries to be created in a declarative manner, to enable new queries to be added without the need to update or rebuild the code that runs the site.

The diagram below shows a schematic view of how the *Topic Map on Top* architecture can be implemented. Serving a page request starts with a request for topic data. Based on either a request parameter or on the type of topic requested, the Extraction process queries the topic map to retrieve the appropriate data and create the page template. This gives a skeleton page, with many of the navigation links now resolved to topics in the topic map. The result then goes to the Content Aggregation process which is driven by the data retrieved from the topic map and populates parts of the page with data from other systems (e.g. CMS). Finally, a Rendition process converts the content to the presentation form required (HTML, RSS, PDF etc.).

Page layout in the Topic Map On Top architecture depends primarily on an interaction between the Extraction and the Rendition components. The Extraction component will determine what page elements are present by the data that it extracts from the topic map. The Rendition component will then organize those page elements on the final rendered page.

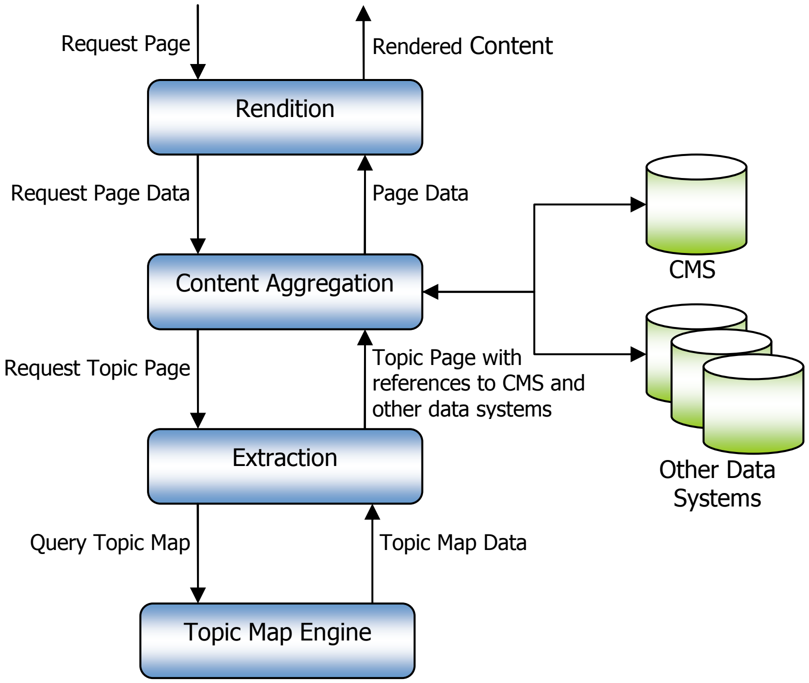

Figure 1: A sample processing pipeline for the Topic Map On Top Architecture

The diagram below shows a schematic view of a typical page with page items coloured to show where they come from.

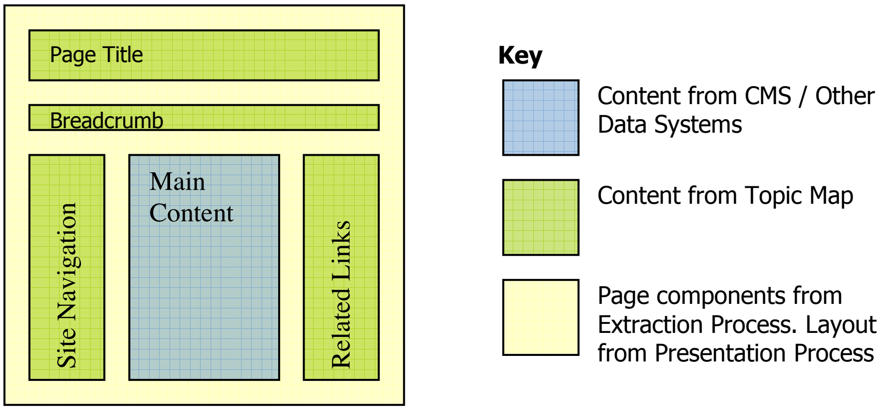
Page Title
Breadcrumb
Site Navigation
Main Content
Related Links
Key
Content from CMS / Other Data Systems
Content from Topic Map
Page components from Extraction Process. Layout from Presentation Process

Figure 2: A sample page schematic for the Topic Map On Top Architecture

Topic Map Underneath

The second approach to navigation is to use the topic map to supplement site content. In this approach, the CMS is the primary source for the page content, but the page templates maintained by the CMS include placeholders for links generated from the topic map. For this approach to be successful it is necessary for each page managed by the CMS to have meta data which can be used to select the topic (or topics) in the topic map which will contribute to page links.

Once again, having a configurable set of queries contained in the page template to generate the links from the topic map gives the flexibility to enrich cross-linking on the site without needing to rebuild the application.

The diagram below shows the pipeline for the *Topic Map Underneath* architecture. Note that while superficially similar to the *Topic Map on Top* architecture, the CMS plays a far more central role in page generation as it is the CMS template processing that drives page layout.

While the CMS could interface to the Topic Map Engine directly, the Extraction process can still play a useful role, providing a high-level interface and shielding the CMS integration from the specific implementation of the Topic Map Engine. The topic map data returned from the Extraction process to the CMS may be a transformed version of the data received from the Topic Map Engine, providing a more business oriented view of the topic map data.

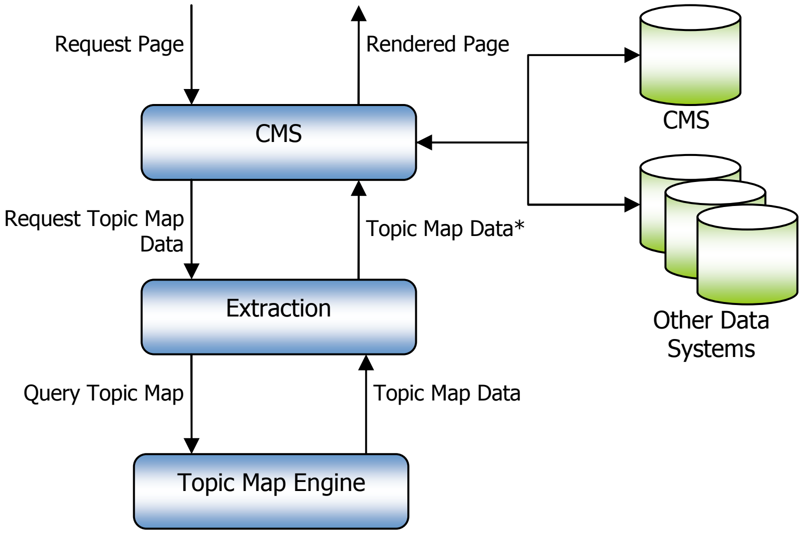

Figure 3: A sample processing pipeline for the Topic Map Underneath Architecture

The diagram below shows a schematic of a page constructed using the Topic Map Underneath architecture. Note that this is only one possible combination of sources, which minimizes the contribution from the topic map, keeping the page layout, page content and site navigation under the control of the CMS.

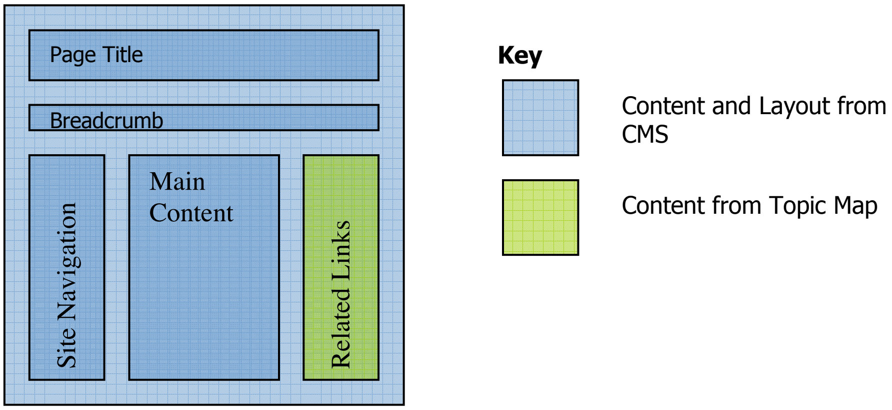

Figure 4: A sample page schematic for the Topic Map Underneath Architecture

Data Integration Architectures

Data integration is simply the task of getting the information you want to present to the user on to the appropriate page(s). We can broadly think of two types of content:

1. Authored content.
   This is content created by a writer that has one or more topics as its subject. Typically authored content can be tagged with meta-data that identifies the key subject(s) of the content.

2. Data-driven content.
   This is content from external data systems – databases, feeds, web services etc. Typically there will be at least a configuration task required to integrate such data into the site and in some cases additional development is required.

The following sections outline 5 different approaches to data integration. These approaches are not mutually exclusive – rather the system architect can choose the most appropriate integration approach for each source of data.

Content Duplication

Under this approach the content in the external system is duplicated in the topic map and is used to populate topic names and/or occurrences. The integration can be either push-based, where the data source propagates changes to the topic map, or pull-based, with the topic map engine regularly polling the data source for updates.

Most importantly, the Content Duplication architecture does not result in any new topics or associations being created in the topic map, it is purely about translating data items from the external system into the topic map.

One of the most crucial aspects to implementation of this approach is to get the direction (push vs. pull) and the frequency of updates tuned correctly. Our experience has shown that a push to a very simple web service interface is often the easiest method for handling ad-hoc updates, allowing the data source to dictate the update frequency. In more structured publishing environments however it may be desirable to perform a scheduled pull from the topic map.

The second crucial aspect to consider is where the master copy of the data will lie, and whether and to what extent changes are allowed against the duplicated data. If changes are allowed against duplicated data, it is important to also design a replication strategy. In our experience, data replication is usually best avoided. If there is a need to provide update access to the data system, this should be considered separately from the data integration architecture used for content publishing and updates should only ever be made against the master copy.

Content Reflection

The Content Reflection strategy is the next logical step on from the Content Duplication strategy. In this approach the data from the external system results in the creation or modification of topics and associations in the topic map, as well as the population of topic names and occurrences.

All of the preceding comments regarding the Content Duplication strategy apply to the Content Reflection strategy, but in addition one must consider the interaction between multiple external data sources using Content Reflection. For example, let us suppose that system A and system B both deal with an entity of type “Employee” (e.g. A is an HR system and B is a help-desk system). It is important in these cases to determine which system(s) can cause the creation of a new topic or association and the types of topic/association that can be created; and which system(s) can cause the removal of those objects.

In many cases it is possible to assign the management of these life-cycle operations for a particular class of topic to a single system, and allow other systems only to contribute to those topics. Going back to our example, we could say that the HR system is responsible for the creation/deletion of topics of type “Employee”, and the help-desk system is responsible for the creation/deletion of topics of type “Job Ticket”.

In some cases, however, things can be less clear-cut and multiple systems could potentially cause the creation/deletion of a single class of topics. There are several strategies for dealing with this issue, but these are beyond the scope of this white paper.

Content Reference

The Content Reference strategy can be used when the external systems feeding into the topic map are capable of rendering the detailed content themselves and where it is not necessary for that content to be part of the overall site navigation framework. For example links to other sites; or an online game implemented in Flash which takes over the entire browser frame, leaving no room for navigation elements.

Such content references are most easily stored as topic occurrences, where the occurrence value is the URL for retrieving the referenced content. That URL can be simply passed through to the browser and it is the browser that is responsible for retrieving the data when the user clicks on the link.

In some cases, it may also be desirable to provide some meta-data regarding the target content such as a title, the media type of the content, when the content was last updated and so on. To reflect that meta-data in the topic map it is necessary to create a topic to represent the resource itself using the Content Reflection strategy, then use the Content Duplication strategy to populate the topic with the meta-data and finally use the Content Reference strategy to provide the link to retrieve the content.

Content Aggregation

All of the strategies presented thus far rely only on the topic map system and the user's browser to provide all site content. However, in many cases the overhead of Content Duplication can be avoided by adding a content aggregation stage to the presentation pipeline. Under this model, rather than storing the content itself, the topic map stores instructions on how to retrieve the content. Those instructions are understood and processed by the content aggregation stage, which inserts the retrieved content into the stream being returned to the browser. This strategy greatly separates concerns, allowing the topic map to focus on managing the site framework and the content aggregation stage to focus on efficient access to content resources.

We find that this is the best strategy for the *Topic Map on Top* integration of systems that are designed to provide content to a web server, such as a CMS. It allows the solution to make use of the performance and scalability of those systems while still keeping the topic map in control of the navigation framework for the site. In the *Topic Map Underneath* strategy, the content aggregation stage can still server a purpose in bringing together content from disparate sources, but typically this responsibility gets merged into the general responsibilities of the CMS that drives the site.

Content Query

Our final strategy for content integration is more focused on locating the relevant content. In most cases, we strongly recommend that content producers focus on the topic map as a source of managed content and managed references to content. However in some cases, this is just not achievable for some or all of the content, with legacy content being of particular concern.

However, a topic map can still provide an implicit link to content by providing the best keywords and other search criteria to locate relevant content. The task of the content creator is then to select the most suitable search criteria for each topic – including synonyms and possibly common misspellings as well as keywords for disambiguating context. This is a strategy which can also prove useful in locating relevant links from Internet search engines such as Google.

Site Search

Topic maps can benefit site search in three very specific ways: faster searching, improved search-browse cycle, and semantic enrichment of search queries.

Faster Searching

With a topic map that models the information domain and connects to related resources, the topic map can be used as the primary content index. Rather than trawl through a full-text index of the content, searches can be directed to an index of just the topic map content (names and occurrence data). The result of the search would be either a list of topics or a list of the documents that are related to the topics found by the search, depending on whether the site design calls for a topic-centric or a document-centric search results set. Indeed it is also possible to combine the two and present the user with not only the most relevant documents, but also the most relevant subject areas to which they might browse to select the documents they wish to read.

Improved Search/Browse Cycle

A search, by its very nature returns a list of links – while those links might be dressed up with relevance ranking, text snippets, user ratings and the such like, it is still at its heart just a list of links, and usually if one clicks on a link and finds that its not quite what one was looking for, the only course of action available is to search again.

With a topic map, it is possible for every piece of content found through a search to be linked not only to related content, but to related subject areas. It is rare that a search on a domain-specific site is going to be completely irrelevant to the user's query, but more common that it will be "not quite relevant". If the page that the user arrives at from the results page is richly linked to related subject areas and related documents it is far more likely that the "not quite relevant" document page might link to a more relevant document or subject area.

Semantic Searching

By "semantic searching" we mean the application of domain ontology against a query string to attempt to find more relevant results. For example a query that uses a narrow keyword (say "Georgian") could be mapped to a wider keyword (say "18th Century") or to a site-specific combination of keywords (say "(`18th Century` OR `19th Century` OR `Hanoverian`) AND `Britain`").

Such expansion might be completely behind-the-scenes or might be presented to the user as a way to communicate the vocabulary used by the site and to guide future searches.

The search vocabulary and search intentions of users can often confound domain experts, and a good site ontology will include not only the domain-specific ontology, but also mappings to the folk-ontology of the domain – the terms used by the community at large and by the general public.

Conclusion

We have presented strategies for the use of topic maps to structure the navigation, content integration and search facilities of content-rich web-sites. We believe that a topic maps-based approach to the development of web-sites brings benefits to the Information Architect, the Systems Architect and the Content Creator.

The principal benefit for the Information Architect (IA) and the Systems Architect (SA) is that topic maps provide a semantically rich basic model of topics, associations and occurrences which simplify the development of domain-specific ontologies and navigation structures while simultaneously allowing pattern-based implementations to be less tied to the specifics of the domain model. This allows the IA to evolve a site structure over time and allows the SA to keep up with the demands of the information architecture without the need for constant rebuilding of the site and with more data-driven site configuration.

The principal benefits for content creators are in content linking and content discovery. A topic map can provide rich sets of content links that are to some degree self-maintaining. There is no need to revisit old content to ensure that links remain relevant as the concerns of content creation and linking to relevant content are separated. A rich set of links between documents helps to drive users to new content, and semantically-enriched searching can assist users in locating relevant content even when they might lack the domain-specific vocabulary used to index that content.