Structured Maps: Modeling Explicit Semantics over a Universe of Information*

Lois M. L. Delcambre[1], David Maier[1], Radhika Reddy[1], and Lougie Anderson[2]

[1] Oregon Graduate Institute, Portland, OR 97291-1000
[2] Sequent Computer Systems, Inc., Beaverton, OR 97006

January 17, 1997

The overwhelming accessibility to data, on a global scale, does not necessarily translate to widespread utility of data. We often find that we are drowning in data, with few tools to help manage relevant data for our various activities. This paper presents Structured Maps, an additional modeling construct superimposed over available information sources, that provides structured and managed access to data. Structured Maps are based on Topic Navigation Maps, defined by the SGML community to provide multi-document indices and glossaries.

A Structured Map provides a layer of typed entities and relationships where the entities can have typed references to information elements in the Information Universe. Structured Maps can be placed over loosely structured data, e.g., document collections, with references at various levels of granularity. Structured Maps directly support new, customized, and even personalized use of the information.

In this paper, we define Structured Maps and present several examples adapted from the Sequent Corporate Electronic Library (SCEL), an intranet resource currently implemented in HTML.

1 Introduction

Imagine that you are gathering information about physicists who worked at the Institute of Advanced Studies around WW II, based on information available in your personal library, in public libraries, and at other sites. Imagine further that you are currently focused on Albert Einstein. The goal of your work is to understand how the interaction among those scientists might have influenced their individual views on atomic weapons. Your current compilation of information for Einstein, as shown in Figure 1, includes: two interviews, one from a magazine and one from a video clip in a television news archive. You have located a number of scientific contributions including five journal articles and a chapter from a physics textbook. There are photographs of Einstein from Life magazine and from a book cover. Finally, you have located biographical accounts of Einstein, including two books, an entry in Who's Who and an excerpt from an encyclopedia.

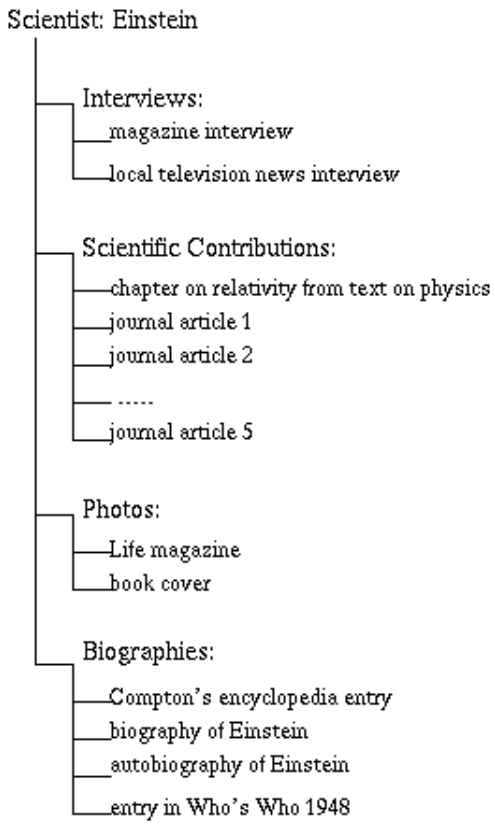

Fig. 1. Information about Einstein

You might wish to discover what sort of relationship existed among the scientists prior to the development of nuclear weapons, to distinguish those who were close

* This research was funded by a grant from the National Science Foundation, NSF Award Number: IRI-9502084. *To appear in International Journal of Digital Libraries 1997.*

friends versus acquaintances versus collaborators, etc. You might look for group photos, for example, just to see if two scientists appear together. You might look for papers co-authored by several of the scientists or references by one to another.

How can you manage your access to this diverse information? Your options include any mix of the following:

- physically assembling the material in one place,
- physically copying the material and assembling those copies in one place, and
- assembling references to the material, e.g., as bibliographic references or as location descriptions, such as folder numbers in a photo file, etc., perhaps written on 3x5 index cards.

The format and media of these various materials are quite diverse. (Will you store video clips in a file cabinet?) Each referenced item is part of some other information collection that fulfills some purpose beyond yours: the rest of the issue of Life magazine, the other entries and volumes in the encyclopedia, and so forth. Notice that when you refer to the various materials, it is important to distinguish the nature of information that is being referenced, according to your purposes: biographical items versus technical contributions versus quotations. Finally, note that it is likely you will want to organize material on other scientists along the same lines.

In organizing this information, you are developing your own framework for viewing it, which is different from the organization of any particular source you used. This framework is likely to exhibit more structure than is present simply in the pile of your collected material, and probably emphasizes similarities at a semantic level, rather than commonalities in media type or syntactic format.

1.1 Structured Maps

This paper presents Structured Maps to provide the capabilities to organize access to such diverse information for some new and unanticipated purpose *in an electronic environment*. Structured Maps are superimposed over an underlying universe of information, e.g., over the books, documents, videos, web pages, images, etc., available electronically. Structured Maps can model typed entities and relationships, e.g., Scientist and Mentor-Of, respectively. Instances of the entity types are connected to elements (often fragments of some larger information source) in the underlying universe of information. Each entity type may have one or more facet types, for example, to distinguish between biographical versus technical references to the underlying universe. Each instance of an entity, e.g., Einstein, as an instance of scientist may have any number of references for each facet type. Relationship instances will connect entity instances.

A Structured Map introduces useful information to serve as a structured guide to selected information elements in the underlying universe. Structured Maps are somewhat like structured bookmarks on the World-Wide Web.

Our research is driven by two key goals: (1) to use underlying information *in situ*, that is without disturbing the information sources nor interfering with local tools, that manage and access it and (2) to leverage database models and technology to implement Structured Maps. We are particularly interested in providing a query capability over this superimposed information.

A Structured Map is different from the result report of a text search, because the superimposed information is structured and typed (e.g., as a scientist type with additional information classified as biographical, photo, etc.). Also, Structured Maps include explicit references to information elements through facets. Structured Maps are distinct from web-based approaches to information access because Structured Maps are not limited to the structure of any particular web (e.g., WWW, or a corporate intranet). Structured Maps are distinct from models for semi-structured data because they need not translate information from the underlying Information Universe into another model. Finally, Structured Maps differ from federated databases in that they are not limited to either the structure nor the data from the participant information sources. Structured Maps can introduce new information. In the example above, the Mentor-Of relationship might not be explicitly modeled anywhere in the universe of information.

1.2 Organization

Section 2 of this paper defines Structured Maps and presents an example. The SGML/HyTime foundation for Structured Maps, the Topic Navigation Map Architecture is presented in Section 3. Section 4 describes our implementation of Structured Maps along with a discussion of issues that affect an implementation. Section 5 includes examples from a large-scale, corporate electronic library. Section 6 evaluates this work by comparing it with related topics in the database and digital library community. Section 7 concludes with a discussion of the contributions of this work and our current research plans.

2 Definition of Structured Maps

Consider the Structured Map shown in Figure 2. Three information sources are shown in the Information Universe: art.sgm, an encyclopedia containing biographical abstracts of artists; book.sgm, a biography of Gentile Bellini; and guide.sgm, a guide to Venetian art. This Information Universe is supplemented with a Structured Map definition, shown at the top of the figure and a

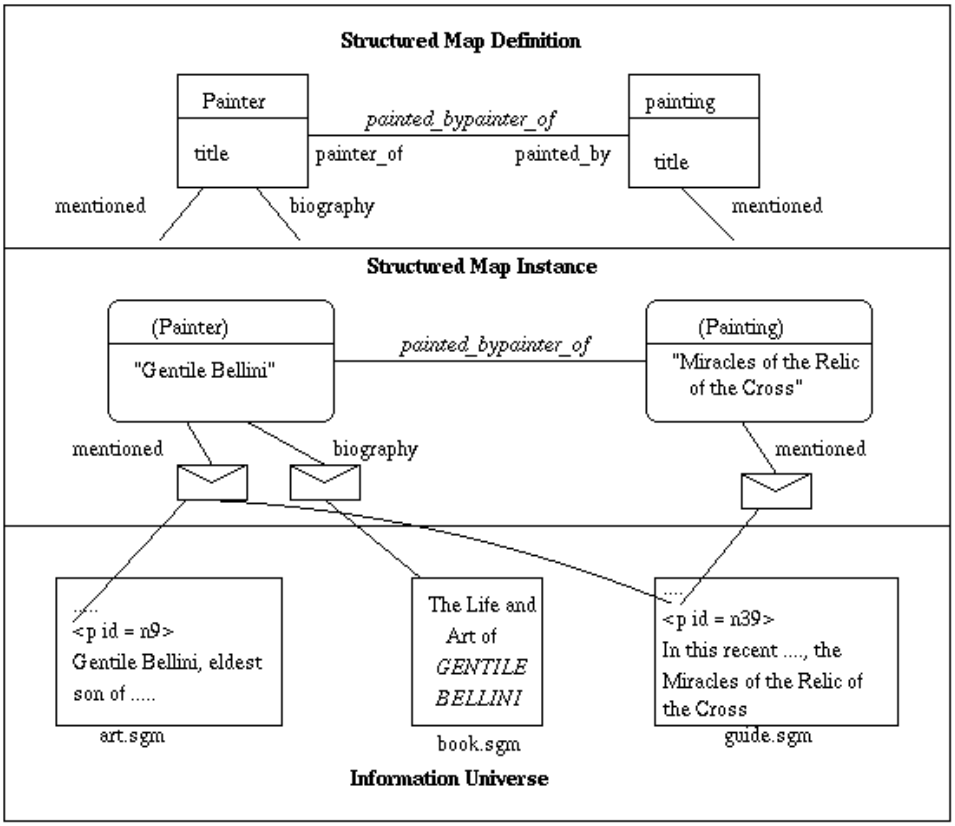

Fig. 2. Example Structured Map

Structured Map instance, shown in the middle of the figure.[1]

This Structured Map definition introduces two *entity types*: Painter and Painting, with a *relationship type* to indicate the painter(s) for a painting. We use OMT notation [22] because of the strong similarity of Structured Maps with an entity-relationship-style model (ERM). We use only one attribute for each entity, called "title". The title is used during browsing as a label for each entity instance. Using a full set of attributes, as in an ordinary ERM, is also possible; see Section 7 for further discussion of this issue.

The Painter entity type in Figure 2 has two *facet types*, shown extending from the bottom of the entity symbol. They support connections from entity instances to arbitrary information elements from the universe. Facets can be of different type; two types are shown for Painter: "mentioned" to reference information elements where the painter is mentioned and "biography" to reference information elements that provide a biographical description of the painter. For Painting, we define just one facet named "mentioned".

The middle box of Figure 2 shows the current instance of the Structured Map, having one Painter instance and one Painting instance, with "Gentile Bellini" and "Miracles of the Relic of the Cross" as titles, respectively. These two instances are connected by the painted-bypainter-of relationship. Of particular interest are the references used to populate the facets. We see that Gentile Bellini is mentioned in two places: a paragraph in art.sgm and a paragraph in guide.sgm. We note that there is a reference to the entire book, book.sgm, for the biography facet. The "Miracles of the Relic of the Cross" is mentioned in the same paragraph of guide.sgm as Bellini, as indicated in the figure. Each facet instance, for an entity instance, consists of a set of zero or more addresses where each address references an information element from the universe. The facet instance is shown in the figure as an envelope icon with the facet type displayed nearby.

This Structured Map can be used as a navigational guide for the underlying Information Universe. The user can browse the Structured Map, viewing painters and paintings. Whenever relationship instances or facet instances are available, the user can follow them to the relevant entity instance or information element, respectively.

Note that a given user may have several Structured Maps, over the same or different Information Universes. The multiple Structured Maps may or may not have common structure (e.g., they may or may not have the Painter entity). Two Structured Maps could share the

[1] This example is adapted from one developed by High Text, S.A.R.L. [34].

same Structured Map definition. A user might have a Structured Map to support comparative evaluation of artists and another one to describe the accommodations, travel arrangements and tour stops on a European tour, focused on art.

A Structured Map need not be complete nor exhaustive. There is no inherent requirement that Gentile Bellini entity instance reference every information element where the string B-e-l-l-i-n-i appears. Structured Maps contain (only) the information explicitly recorded by the user (or tools the user employed). A Structured Map can represent semantic connections, given that they have been discovered or confirmed by a human user.

2.1 Structured Map Components

Here we define Structured Maps more formally.

A Structured Map has a three-level model, as shown in Figure 2: the Structured Map definition, the Structured Map instance (i.e., the populated instance of the Structured Map definition), and the underlying universe of information with various information elements highlighted by the facets of the Structured Map.

Structured Map Definition - The Structured Map definition follows the normal conventions of an ERD, except that it is limited to entity types and relationships types, e.g., there are no generalization or aggregation links. Each entity type includes one attribute definition that will hold the user-visible title or name for an entity instance. This attribute can be viewed as the label beside the entity instance, seen while browsing. Each entity type also can define one or more facet types. Each facet instance connects an entity instance to zero or more information elements in the Information Universe. Each facet type, such as Mentioned and Biography, indicates the reason or purpose for referencing information elements. Relationship types can be defined among entity types.

Structured Map Instance - The entity and relationship instances of a Structured Map are populated much like a conventional database instance that conforms to its schema.

The title attribute value is represented as a character string. References to arbitrary information elements in the Information Universe can appear in the instance for each of the facet types.

Information Universe Elements - An information element within a given information source in the Information Universe must be *delimited, addressable*, and *renderable*. By delimited, we mean that the bounds or extent of an information element must be known. As an example, SGML, HTML and Latex elements are delimited by start and end tags. By addressable, we mean that there exists one (or more) addressing schemes for the information source and that this particular information element is addressable by one of those schemes. Finally, by renderable we mean that the information element can be viewed during browsing. Each underlying information source must support atleast one addressing scheme as well as a convention for delimiting and rendering the information elements.

2.2 An Analogy

Structured Maps share certain properties with conventional road maps. A map serves as a window onto some portion of the world; the information content of the map is clearly less than the total detail of the corresponding world slice. The legend defines the types; instances of these types can appear in the map, usually with a label displayed nearby.

A road map describes a variety of potential destinations: the non-road objects, such as schools, cities, parks, etc. And the roads provide various navigational paths among the non-road objects. The objects on a map are represented iconically and displayed visually. They are usually correctly positioned in geographic space according to the stated scale of the map. The implied equivalence between an icon on a map and some real-world object is roughly analogous to the use of facets in Structured Maps. Facets are more powerful because they are not limited to equivalence nor to being single-valued. Also, an entity type can have more than one facet type.

The same collection of entity types could be used to model different slices of the world. We can have maps for each state using the same legend, for example. Conversely, the same slice of the world can be modeled with a different collection of entity types, for another purpose. For instance, we might want a map that indicates car rental locations, filling stations, and auto repair shops, or power generation facilities, substations, and transformers. Note that a map need not include every possible entity instance. A car rental company might produce maps showing only their own car rental locations. A state map generally doesn't include every single road.

Structured Maps can be viewed as an analogue of a conventional map, where the physical world (where we use maps) corresponds to the space of online information, where entities are analogous to non-road objects such as schools, landmarks, airports, cities and so forth and relationships are analogous to roads. The legend sets forth the icons for road and non-road types analogous to the Structured Map definition; the graphical part of the map includes instances of the various icons, analogous to the Structured Map instance. The graphical part of the road map corresponds to the instances of entity and relationship types, where the entity instances and relationship instances have labels. As an example, each airport icon that appears on a road map is generally labeled with the name of the airport and many roads are labeled with their street name or highway number. The connection between the map and the real world (anal-

..... (standard SGML declaration) .....

```
<!ELEMENT document - O (painter|painting|painted_bypainter_of)*>
<!ATTLIST document
               id         ID        #IMPLIED
               HyTime     NAME      HyDoc
               boslevel   NUMBER    #IMPLIED >
<!ELEMENT  painter - - (TITLE)* >
<!ATTLIST  painter
               %topic;
               anchrole  CDATA  #FIXED "painter mentioned #AGG biography #AGG">
<!ELEMENT  painting - - (TITLE)* >
<!ATTLIST  painting
               %topic;
               anchrole  CDATA  #FIXED "painting mentioned #AGG">
<!ELEMENT  painted_bypainter_of - O EMPTY >
<!ATTLIST  painted_bypainter_of
               %topicrelation;
               anchrole  CDATA  #FIXED "painted_by painter_of">
]>
<document>
<painter id=painter-BartolomeoVeneto mnemonic=BartolomeoVeneto universe=all
linkends="painter-BartolomeoVeneto adr-BartolomeoVeneto-mention
adr-BartolomeoVeneto-biography"
<title>Bartolomeo Veneto</title>
</painter>
<nameloc id=adr-BartolomeoVeneto-mention> <nmlist docorsub=art>n7</nameloc>
<nameloc id=adr-BartolomeoVeneto-biography> <nmlist>null</nameloc>

<painter id=painter-GentileBellini mnemonic=GentileBellini universe=all
linkends= "painter-GentileBellini adr-GentileBellini-mention
adr-GentileBellini-biography"
<title>Gentile Bellini</title>
</painter>
<nameloc id=adr-GentileBellini-mention> <nmlist docorsub=art>n9
<nmlist docorsub=guide>n39</nameloc>
<nameloc id=adr-GentileBellini-biography> <nmlist docorsub=book></nameloc>

<painting id=painting-MiraclesoftheRelicoftheCross mnemonic=MiraclesoftheRelicoftheCross
universe=all linkends= "paintingMiraclesoftheRelicoftheCross
adr-MiraclesoftheRelicoftheCross-mention"
<title>Miracles of the Relic of the Cross</title>
</painting>
<nameloc id=adr-MiraclesoftheRelicoftheCross-mention>
<nmlist docorsub=guide>n39</nameloc>

<painted_bypainter_of linkends = "painting-MiraclesoftheRelicoftheCross
painter-GentileBellini">
</document>
```

Fig. 3. Excerpt from a Topic Map SGML Document

ogous to the facet instances) is only implicit, through spatial correspondence.

On a road map, we can travel from a particular city to a particular park by selecting from among the available roads that interconnect them. In an analogous manner, using a Structured Map we can navigate from a city to its artists (if the Structured Map included a City entity type and the Born-In relationship type) and then from the artist to his or her works of art (e.g., through a "created-by" relationship). Using a road map, we can travel, physically, to the cities or airports or schools shown on the map. Using a Structured Map, we can "visit" each entity by traversing the facet instances to the relevant information elements.

3 The Basis: Topic Navigation Maps

The modeling capability of a Structured Map is fairly elementary compared to most ERD models. But we are currently guided by the definition of the Topic Navigation Map, defined as part of the working group on the Conventions for the Application of HyTime (CApH). A Topic Navigation Map is represented as an SGML document [4]. A Topic Navigation Map uses the terms topic, topic relation, topic title, and anchor role as the analogous terms for entity, relationship, title, and facet in Structured Maps. As an example, the Topic Navigation Map SGML document that corresponds to the Topic Navigation Map of Figure 2 is shown in Figure 3. Note, the explanation of Figure 3 assumes some familiar-

ity with SGML. The Document Type Definition (DTD) for the Topic Navigation Map declares the desired topic and topic relation types (i.e., the entity and relationship types, in Structured Map terminology) for the application. The content model for the document instance of the Topic Navigation Map consists of a disjunction of all topic and topic relation types. This content model allows any number of topic and topic relation instances to appear in any order in the document instance.

Topic Navigation Maps depend on the use of several HyTime constructs, particularly those for linking and addressing. HyTime (ISO 10744:1992) [38] extends the semantics of SGML but is expressed in ordinary SGML syntax.

In the Topic Navigation Map of Figure 3, each topic type is declared through the %topic parameter entity; each topic instance has a topic title (shown in the content model for each topic declaration). Each topic type declares the names of its anchor roles through the value of the `anchrole` (SGML) attribute. The name "anchor role" is intended to indicate the role played by the referenced information elements, addressed as anchors in the information source comprising the universe. By convention, the first entry in the `anchrole` attribute value is the topic type followed by the names of the anchor roles. The `#AGG` following an anchor role name means that the anchor role is multi-valued. As an example, the Painter topic type in Figure 3 has an anchor role for "Biography" and an anchor role for "Mentioned". Each topic instance has a corresponding set of zero or more addresses for each anchor role. The Painter instance for Gentile Bellini (with SGML ID `painter-GentileBellini`) provides one SGML ID for each anchor role through the values of the linkends (SGML) attribute. The SGML element with this ID consists of the list of addresses for the anchor role. Thus, `adr-GentileBellini-mention` is the SGML ID for the element that contains the list of addresses, each of which references an information item in the Information Universe.

In HyTime, the document identifier is given in the `docorsub` SGML attribute. The referenced information element, in this case, is the information element where Bellini is mentioned with the stated ID in the referenced SGML document in the underlying universe of information. As an example, the Painter instance for Gentile Bellini has the address for the entire "book" SGML document on the linkend for the "biography" anchor role and the ID `n9` in the "art" SGML document as well as the ID `n39` in the "guide" SGML document on the linkend for the "mentioned" anchor role. The way in which information references can appear is defined as part of the various HyTime addressing modes, including address by name, address by location, etc. In this example, all of the underlying information sources are SGML documents. Note that information elements are addressable only if they include the SGML ID attribute as declared in the DTD for the document.

The HyTime standard is currently being revised and the Topic Navigation Map is being proposed as an independent standard. Both of these may change the way in which Topic Navigation Maps are represented in the standard. The SGML/HyTime representation of a Topic Navigation Map shown here uses the earlier, published version of HyTime.

As declared in the Committee Draft submitted to ISO, a Topic Navigation Map provides a mechanism to define tables of contents, subject indexes, glossaries, and multi-lingual thesauri for a single document or for a set of documents [35]. The purpose of a Topic Navigation Map is to highlight individual topics that appear in a set of documents and also to establish relationships among topics. As suggested by the name, a Topic Navigation Map is intended to support navigation, like a general index or glossary in a conventional book, but extended to multiple documents.

Our motivations for following the definition of Topic Navigation Maps are:

- Topic Navigation Maps are currently being proposed as an ISO standard to provide multi-document indices, glossaries and table of contents [35]. The standard could provide Structured Maps with an ISO standard interchange format.
- Topic Navigation Maps, although developed outside of the database community, use a basic Entity-Relationship model at the core. The Entity-Relationship model has proven to be widely understandable and useful for information representation and access, using database technology.
- Topic Navigation Maps are defined using SGML [37] and HyTime [38]. Together they provide a model to describe the structure and semantics of complex documents, a foundation currently lacking in conventional database models.
- Topic Navigation Maps use a DTD to describe the structure of the document instance, analogous to the database schema and instance.

Key features of Topic Navigation Maps that are preserved in Structured Maps are:

- the information represented in a Structured Map is expressible using SGML with HyTime and CApH [4] constructs,
- a Structured Map declares entity and relationship types (in the Structured Map definition),
- an instance of a Structured Map establishes the (typed and titled) entities and (typed) relationship instances as a database extension,
- the entity instances include the facet instances and their references, and
- the addressing mechanism and the interpretation of addresses are both expressed separately from the definition of a Structured Map; the Structured Map simply holds addresses.

Structured Maps differ from Topic Navigation Maps in that Structured Maps limit the participants in a relationship to the declared entity types. Topic Navigation Maps, on the other hand, are unconstrained regarding the participants in a topic relation. We also find it useful to assume that the title attribute is a key for the entity type. In Topic Navigation Map the title is optional and multi-valued. Finally, since Structured Maps are implemented using relational database technology, we use sets rather than lists, for example, for the references that appear on a facet instance.

One of the major advantages of Structured Maps, based on the database semantics, is the query capability over the database extension. The challenges of Structured Maps, from a database point of view, are to define the semantics of the structural model including the query language for the three-level system and to implement the referenced information elements in a general-purpose manner, managed by the local information source.

4 Implementation of Structured Maps

We have implemented two prototype browsers for Structured Maps. Our first prototype was designed to show the feasibility of representing the Topic Navigation Map definition and instance in a relational database. Figure 4 gives an overview of the first prototype. We have transferred information from an SGML Topic Navigation Map document to a database and, in the reverse direction, transferred Structured Map instance information back into a SGML document that represents a Topic Navigation Map.

4.1 Initial Prototype

This first prototype used a commercial product, called EnLIGHTeN[2] [34], as the baseline for the exercise. EnLIGHTeN supports an integrated browser for Topic Navigation Maps over SGML, HTML, and text documents. With EnLIGHTeN, the Topic Navigation Map SGML document plus all of the documents in the underlying Information Universe are parsed by an SGML parser and then stored. The EnLIGHTeN engine supports bidirectional navigation across facet and relationship instances. EnLIGHTeN resolves all references that appear on the facet instances.

In our first prototype, we parsed a Topic Navigation Map SGML document from EnLIGHTeN and populated an Informix[3] [36] relational database with both the Structured Map definition and instance information. We defined a generic schema for this purpose, i.e., with relations for entity types and entity instances. The referenced (SGML) documents were not placed in the Informix database, although the SGML addresses on the facet instances were included in the database.

We issued SQL queries against the Informix database and were able to freely join across relationships. This facility is useful for query answers that consist only of entity titles. But when a query answer included addresses (from the references on facets) in the form of HyTime addresses referencing the SGML documents from the underlying Information Universe, there was no direct support in our prototype for interpreting, dereferencing, or displaying the contents of these addresses. We could do matching for equality on addresses but we were not able to navigate to or otherwise interpret the referenced information elements in the underlying SGML documents. This limitation is in direct contrast to EnLIGHTeN, which provides a fully integrated Topic Navigation Map browser with a built-in SGML browser for the underlying documents.

4.2 The CARTE System

We built our second prototype to further explore the use of conventional database technology, particularly for query processing and schema management. Our second prototype, called CARTE 1.0, also uses an Informix database as the repository for Structured Map definition and instance information. Like its predecessor, CARTE does not store the documents from the underlying Information Universe. In CARTE, we used HTML pages as the underlying information sources in the Information Universe rather than SGML documents. We used URLs and the `NAME` HTML attribute to formulate addresses for the references on the facets in the Structured Map and we used Netscape Navigator 2.0[4] to render the information elements.

Figure 5 presents a screen image of the CARTE system. We use the multiple frame capability of Netscape 2.0 to present three different, synchronized frames. Within CARTE, the upper-left frame shows the Structured Map Definition (labeled as schema in the figure) in OMT notation. The right frame in CARTE displays the instance information from the Structured Map. In Figure 5, the instance frame is showing all of the available entity types, in list form. Figure 5 shows the initial instance screen contents, when the user begins viewing a Structured Map and has not yet selected an entity type or navigated to a particular entity instance. When the user clicks on one of the entity types, the instance screen shown on the right side of Figure 6 is presented. The entity type is shown followed by the titles of all entity instances of that type. The user can click in this frame to navigate to an individual entity instance. Such a selection results in the instance screen shown in Figure 7, with one entity title listed with all its available relationships

[2] EnLIGHTeN is a trademark of High Text S.A.R.L.
[3] Informix is a trademark of Informix Software Inc.
[4] Netscape Navigator is a trademark of Netscape Communications Corporation.

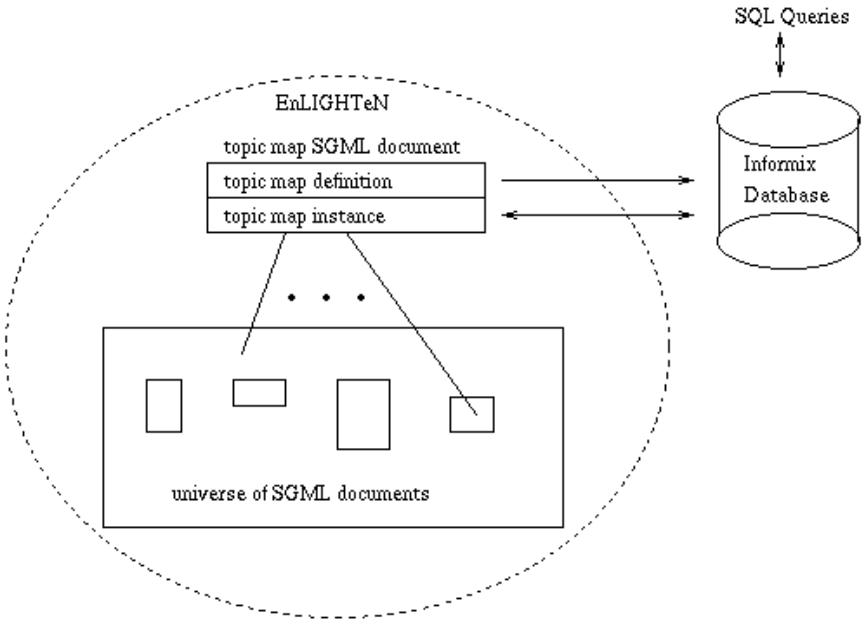

Fig. 4. Overview of our First Prototype

(to navigate from one entity to another) and facets (to navigate to underlying HTML pages). Note that facets for an instance are labeled Reference-Valued Attributes in Figure 7.

When a user navigates across a relationship, the user sees the entity instance screen for the target instance. When a user navigates to any of the references appearing on any of the facet instances, another Netscape process is invoked to view the underlying, referenced HTML page, as shown in Figure 8. At present, we do not support inverse links for the references. That is, it is not possible to navigate from the underlying HTML page "upwards" into the Structured Map from an information element whose address appears on one or more facets. But any time, while the second Netscape browser is operational, the user can return to the CARTE interface and proceed with other navigational steps. In this implementation of CARTE, the underlying HTML pages have no knowledge of the Structured Map.

The lower-left frame in the main CARTE screen is intended to give the user a sense of "you are here" during navigation by describing the current context. CARTE appends an informational message to a scrolling list in the context frame each time the user takes a navigational step. Thus, the context frame shows the progression, for example, from an entity instance, across a relationship instance, to another entity instance, down to an underlying HTML page, and so forth.

CARTE provides a navigational browser for Structured Maps. The Informix database for CARTE contains the Structured Map definition and instance as well as the references appearing on the facets. Each user action taken in the CARTE interface results in an SQL query being issued to the Informix database, followed by the appropriate presentation of information in the CARTE frames. The three CARTE frames are always synchronized with the latest user action.

4.3 Discussion

Table 1 summarizes a number of aspects of the Topic Navigation Map specification, the EnLIGHTeN product, and our two prototypes. Each aspect is listed in the left column of Table 1 and discussed below.

Any implementation of Structured Maps must deal with the addresses of referenced information elements for two different purposes. First, when populating a Structured Map, it is necessary to select information elements of interest and place their addresses on the selected facets for the appropriate entity instance. Second, for browsing a Structured Map, any traversal of a facet instance must present the address for interpretation and rendering of the information element. Tools for authoring and browsing may take various approaches for establishing and interpreting addresses.

To populate a facet (during Map creation or modification), one possibility is to generate the references using an automatic indexing technique (that might be adjusted by a knowledgeable human user). Another possibility is to mark information elements in the underlying universe with the entity type, entity instance title, and name of the facet type. This approach would allow the establishment of an address on the proper facet for the proper instance; it has been implemented this way in EnLIGHTeN [34]. Yet another possibility would be to support visual display of both the Structured Map and the

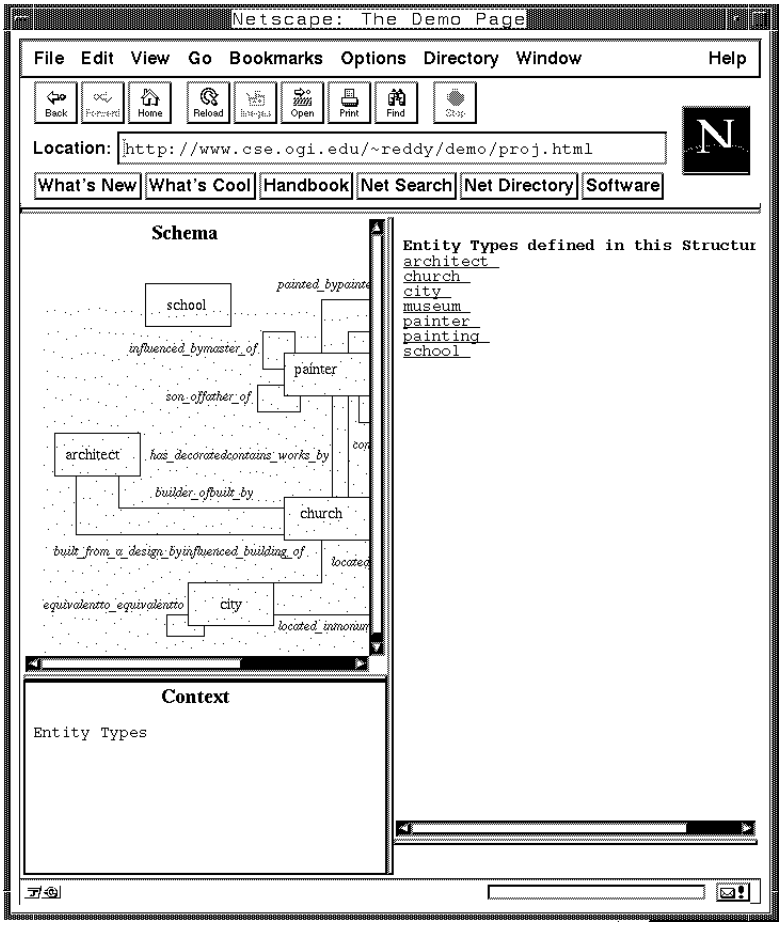

Fig. 5. CARTE Screen

information sources with an easy point and click identification of referenced information elements. Finally, it is always possible to create Structured Maps by hand, including the placement of addresses on facets, if the addresses of information elements are known to the user. The support for map creation of the various systems is shown on line 1 of Table 1.

A closely related issue, relevant to any browsing or navigational capability is: How will the information be viewed? It is not immediately obvious how various users or application domains might want to see Structured Map definitions and instances. For CARTE, we present the schema (i.e., Structured Map definition) explicitly in the user interface. Perhaps because of our history with working with databases, it seems quite natural for us to show the navigation paths through the definition as well as the instance of the Structured Map. Note that EnLIGHTeN currently does not show the Topic Navigation Map definition to the user. There is also the issue of how to present instance information for the Structured Map instance. In CARTE, we generate lists of possible traversals that are appropriate at each navigational step. For an entity instance, we list all available relationship traversals, by type, and all references that appear on a facet instance, labeled by the facet name. In general, we can imagine a set of tools that allow an interface designer to freely configure the way in which Structured Map instance information is displayed. The second line of Table 1 summarizes the current display choices of the various systems.

Lines 3 through 5 of Table 1 indicate the current choice of the various prototypes regarding the method used to delimit, reference, and render the information elements from the Information Universe. Perhaps the most important point is that the various HyTime addressing modes are standard, by virtue of the fact that HyTime is a standard (ISO 10744:1992). This standardization enables the delegation of the delimiting and rendering of information elements via shared addresses. Note also that SGML implicitly provides a way to delimit information elements, through markup, and a way to address them by name, through the SGML ID attribute. Note that such a feature is currently lacking in HTML.

The final two lines of Table 1 deal with the heterogeneity of underlying information and the type of integration between the Structured Map and the underlying information sources. The heterogeneity of information is unlimited, at the conceptual level. Any type of informa-

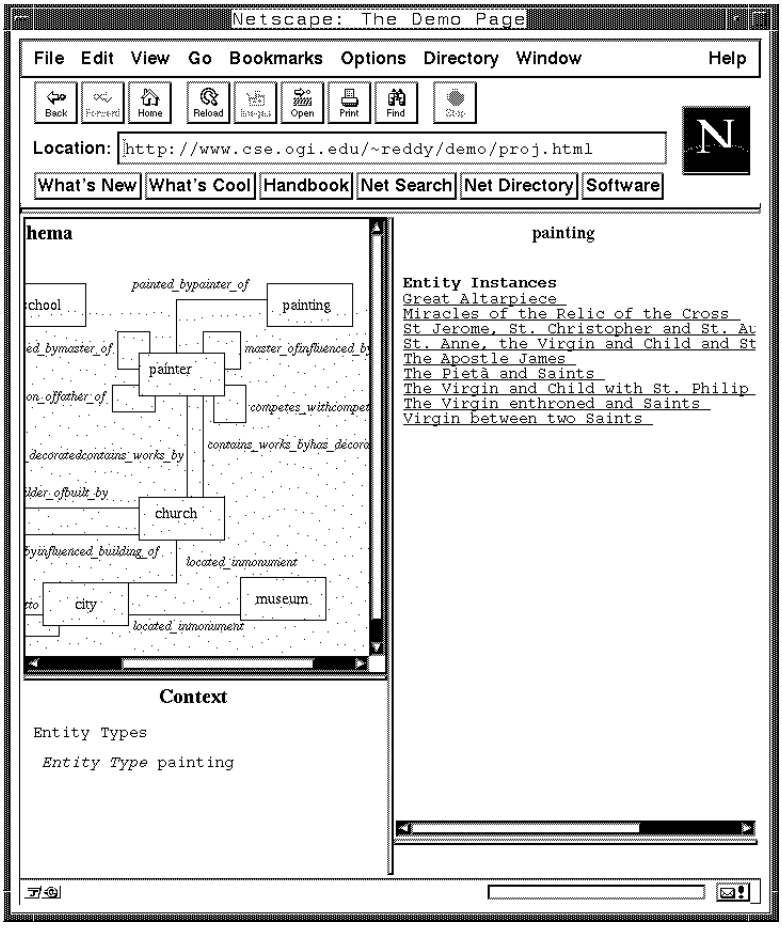

Fig. 6. CARTE Screen

tion that can be meaningfully delimited, addressed and rendered can participate in a Structured Map. The challenge of heterogeneity has to do with the availability of an addressing mechanism and the connection, at runtime, with the independent technology responsible for address interpretation and rendering. This second challenge might be addressed through various interoperation models such as CORBA [19] and COM/OLE [20].

Finally, the issue of loose or tight integration of the Information Universe with the Structured Map presents all of the classic tradeoffs between a tight or loose federation. A tightly-integrated, centrally-managed repository can offer the advantages of conventional database technology such as concurrency control during update, consistency, optimization for access, and query optimization. Also, bi-directional links can be more easily maintained, with their integrity ensured. But the scaleability of such systems may be limited. A loose integration offers the advantage of autonomy for the underlying information sources. Such a choice is particularly appropriate for an environment where we have access to information that we do not own, e.g., on the WWW.

5 Structured Maps in Digital Libraries

In this section, we highlight portions of a particular digital library, the Sequent Corporate Electronic Library (SCEL), that resemble a Structured Map. These portions of SCEL are currently represented on manually created Web pages. One of our research goals is to provide a more automated and flexible means to construct this portion of a corporate digital library.

SCEL is an intranet-based system that provides access to a rich and varied set of corporate information resources to over 2,500 Sequent employees. The system has been operational for about 24 months and the scope and utility of the resource has grown steadily. SCEL is also used to manage routine requests for services through the forms interface of HTML. SCEL is entirely implemented in HTML, relying on Web browsers to provide a uniform, easy-to-use, easy-to-learn interface. There are a number of striking analogs of Structured Maps present in SCEL.

One of the top-level pages in SCEL is shown in Figure 9. This screen presents a number of navigational choices; all of the labels shown here are clickable to proceed to more information. SCEL users can view this screen as a visually-displayed table of contents, with en-

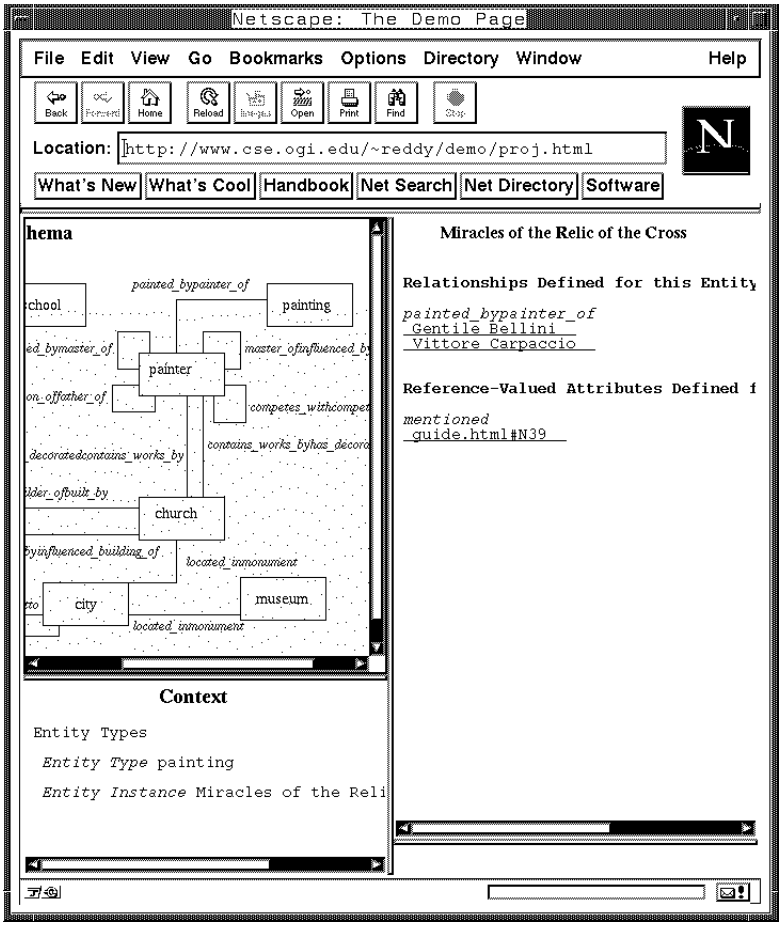

Fig. 7. CARTE Screen

tries for Sequent, Suppliers, Offerings, Partners, Channels, and Market. There are also subentries for Sequent, consisting of History and Values, Internal Processes, Organization Charts, Library Employee Services, and Education. The Market entry is further subdivided into Customers and Competitors.

If we click on Internal Processes we see the page shown in Figure 10. It shows a classical nested table of contents structure, with levels 3 and 4 shown. Levels 1 and 2 of the table of contents are shown in Figure 9. The listed entries in Figure 10 include "Admin Handbook On-Line" and "Admin Services". Each has a number of subentries such as "Business Cards" and "Name Tags" for "Admin Services". Each of these entries leads to either more detailed levels of the table of contents or directly to information sources of various kinds.

We see this nested table of contents metaphor as a powerful, recurring structure intended to organize access to information. One possible Structured Map definition capturing the same information is shown in Figure 11. This structure has been implemented in CARTE. The Structured Map Definition in Figure 11 is simple but it represents the generic nature of a nested hierarchy used as a table-of-contents. The entity title is used to contain the table-of-contents entry. The `has-subentries` relationship captures the hierarchical structure of table-of-contents entries, and the facets lead to the referenced document elements. Although this example uses just one facet, it is possible to have more than one, for different purposes. Such a Structured Map could be visualized in multiple ways. Sophisticated displays such as those used by SCEL would require additional styling and layout tools.

There is another view of Figure 9 that suggest a different metaphor. With the exception of the entries inside the box labeled Sequent, Figure 9 represents different entity types as found in an ERD. There are even some relationships suggested, such as between Offerings and Channels to indicate the distribution channels for a given product or service. Figure 9 also demonstrates several choices for visualization. Subentries inside the Sequent box, in the upper left portion of Figure 9 are placed inside the icon in list form. You could even imagine a scrolling list inside an icon. The other entity instances for a type are seen by clicking on the icon for that type. When defined as a Structured Map, the information organization and navigation can reflect the rich structure, perhaps navigating from partners who are also customers

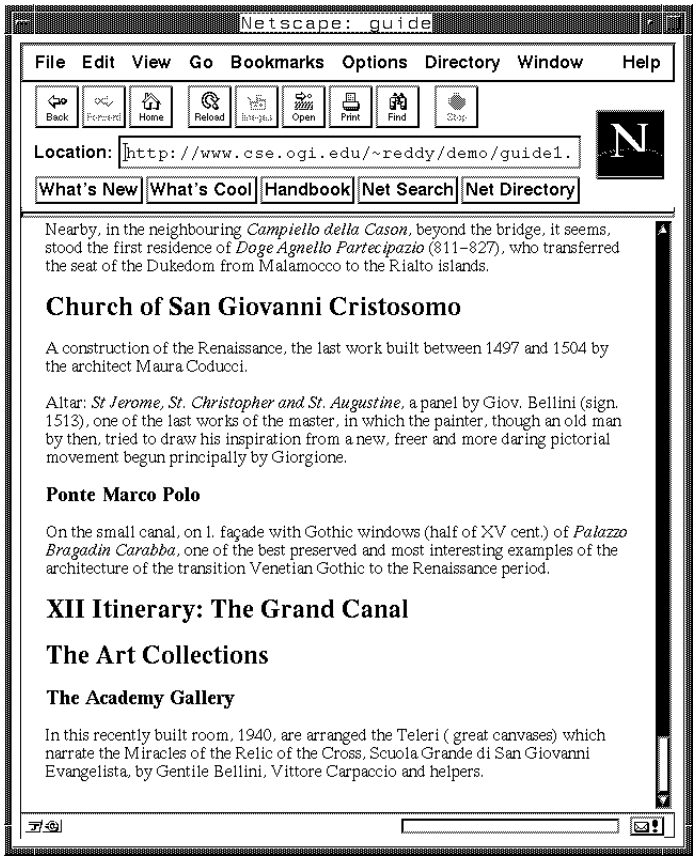

Fig. 8. CARTE Screen - Second Netscape Window (to view underlying HTML)

or partners or suppliers who also serve as channels. Figure 9 thus suggests some of the potential for introducing entity and relationship types and instances over an underlying set of information sources.

Note that Figure 9 presents a high-level picture of the value chain for Sequent. A value chain represents a view of the organization highlighting the suppliers and customers, at any level. The value chain was chosen as the highest level view of Sequent, in part, because it provides an easy way to place documents (into SCEL) and an easy way to find them. We are currently exploring the possibility of representing the value chain, in a similar form, at lower levels in the Sequent organization and reflecting it in a Structured Map.

6 Related Work and Evaluation

The most closely related work is clearly the definition of Topic Navigation Maps, the basic foundation for Structured Maps [4, 34]. As an emerging ISO standard [35], Topic Navigation Maps can provide a precise syntax and interchange format for Structured Maps. The semantics of Topic Navigation Maps derive from SGML [37] and HyTime [9, 39] in that the choice of certain SGML or HyTime elements imply certain semantics.

Within the digital library research community [32, 33], there is little focus explicitly on the conceptual model for information. Much work has focused on various aspects of searching such as the user interface and the performance. Some work has focused on spatial modeling, representation and searching, [12, 26, 27].

One approach to storing and querying documents or other loosely structured information is to define a new self-describing data model [1, 5, 6]. Such techniques require the transformation or mapping of the underlying document structure into the new model.

Other efforts store SGML documents directly in a database, capturing the DTD structure explicitly in the schema, e.g., [7, 23]. Such work is complementary to this research. We do not consider the modeling or the representation of SGML documents; we rely on the underlying information source for storage and management of the information elements. An SGML database could offer an appropriate repository for the SGML items in the Information Universe and could support a centralized approach to Structured Maps. Another research effort deals with non-SGML structured text in a digital library [6].

Table 1. Choices Concerning Implementation Issues

	SYSTEM: ASPECT:	Topic Navigation Map specification	EnLIGHTeN	First Informix Prototype	CARTE 1.0
1	support for map creation	unspecified	markup in information elements with automatic indexing	none	by hand (using SQL CREATE TABLE & Insert statements)
2	choice of display for the map	unspecified	HTML pages generated for output	none - other than relational schema	frames generated for Netscape based on Informix data
3	addresses supported	any HyTime address	SGML IDREF/ID plus docorsub reference to other documents	SGML IDREF/ID plus docorsub reference to other documents	URLs plus the HTML NAME attribute
4	identification mechanism	any identifiable element, supported by the addressing scheme	any element instance in the SGML document with an ID attribute, delimited by start and end tags	any element instance in the SGML document with an ID attribute, delimited by start and end tags	any element, in the HTML tag set, that has the NAME attribute
5	rendering mechanism	unspecified	early version: built-in, proprietary SGML browser. current version: all screens generated in HTML format; uses WWW browser.	none	WWW browser (Netscape 2.0)
6	heterogeneity of underlying information source(s) supported	any type supported by HyTime addressing	SGML, HTML, ASCII text.	SGML	HTML
7	tight/loose integration of underlying information sources	unspecified	tight – everything stored in one repository with all links maintained	loose –SGML documents outside of the Informix DB	loose – HTML pages outside of the Informix DB

Hypertext and hyperdocument models have also been considered as the basis for databases. Some of this work has focused on the file management aspects [8, 10, 17]. Such work provides data management facilities useful for query processing such as collection management and link processing, but does not focus on modeling domain specific concepts. Closely related is work that proposes a query language directly for a hyperdocument model such as the WWW [13, 16, 18]. This work exposes the hyperdocument or other aspects of the underlying model in the query language (e.g., through URLs) but does not focus on the modeling of domain-specific concepts. None of these hyperdocument model projects focus on superimposing information over the hyperdocument structure, as Structured Maps do.

More distantly related is the work on text searching and text databases [11, 14, 15, 28, 30]. Full text searching and the associated indices represent a technology that is complementary to Structured Map technology. The contrast is between automatic techniques that are complete, automatically generated and often syntactic versus Structured Maps that provide explicit connections through facet instances that may represent semantic connections, e.g., as determined by a human expert.

If we were to limit all information sources to be databases and limit the Structured Map to entities and relationships from the underlying information sources, a Structured Map can be viewed as a federated database. A federated database is where autonomous databases can be viewed conceptually as a single database with a single (integrated) schema [25, 31]. Our work focuses on three aspects of Structured Maps that are outside of the main focus of federated database research: (1) the introduction of entity and relationship types that are not necessarily present in the underlying universe of information, (2) the use of Structured Maps over loosely structured information sources such as documents, spreadsheets or video, and (3) the use of Structured Maps over in situ information where an addressing mechanism bridges the two different implementation environments.

Another area related to this work is the Intelligent Integration of Information (I3) Architecture with intelligent mediators introduced to facilitate interaction of various information sources and services [2]. The I3 program is much more ambitious in its goals and in its techniques, employing intelligent agents to analyze information sources, for example. Structured Maps adopt a simplified Entity-Relationship model, with the focus on explicit representation of information and connections. Structured Maps could conceivably be exploited in the I3 architecture because of the additional semantics they provide.

7 Conclusions and Future Work

The focus of this work is the introduction of regularly structured information to allow users to highlight information from the underlying Information Universe. The main contribution of Structured Maps is their ability to

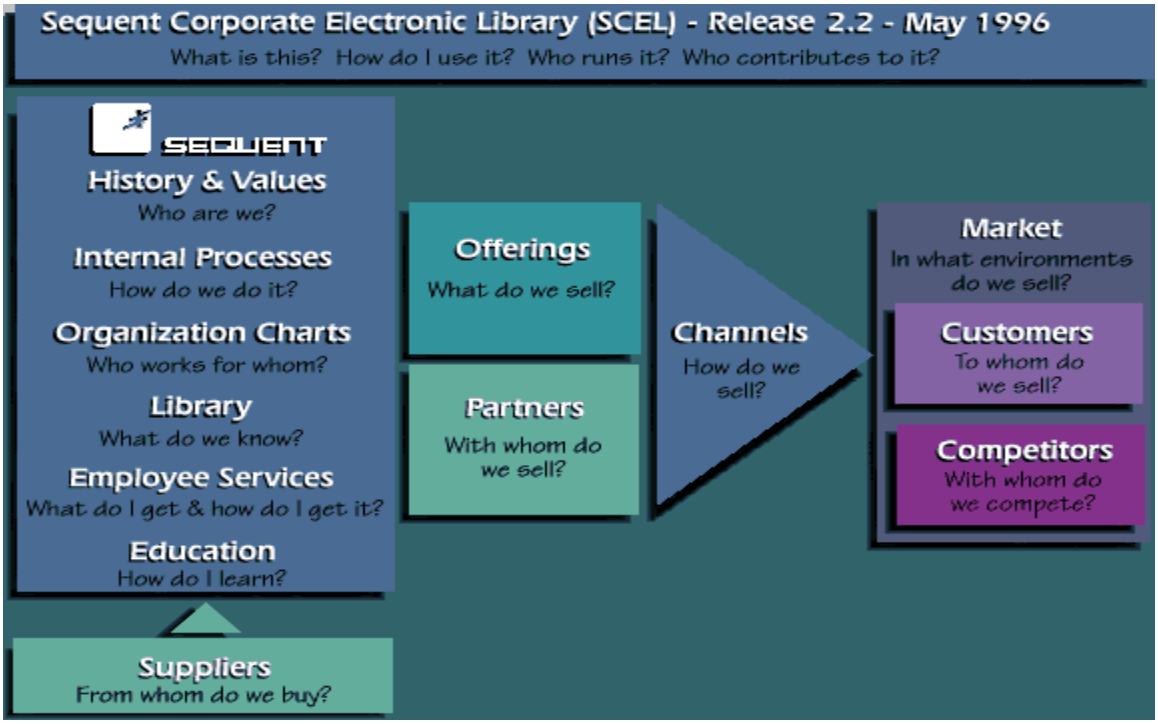

Fig. 9. Highest Table of Contents Level Screen in SCEL

provide personalized, relevant guidance (i.e. a "map") to arbitrary information. Our work is to superimpose regularly structured information over the universe so that we can leverage database concepts and technology. Note that Structured Maps differ from conventional databases through the facet types and facet instances that connect entity instances to underlying information elements. This aspect of Structured Maps requires extension of basic database modeling as well as database technology.

Topic Navigation Maps are defined as an SGML document. But there is no specification of tools nor uses for Topic Navigation Maps in the standard. However it does provide a standard data interchange format that benefits from SGML generality, such as the ability to declare notations for content as well as for the markup.

We briefly discuss several open issues with regard to Structured Maps. The first issue concerns our choice to use information sources *in situ*. We believe that this choice matches a significant portion of the information environment currently available, e.g., through the Internet or the WWW. One alternative would be a tightly integrated system where the storage and management of both the information sources as well as the Structured Map is provided by a single technology. A tightly integrated approach would likely support certain optimizations for access and query processing, such a bidirectional links for facet instances; facilitate update and concurrent access; and provide a semantically richer query language because the semantics of the underlying information sources would be known. We prefer to embrace a loosely coupled approach in order to leverage information-source-specific technology. We envision that new information source types could be easily added. Thus we see our approach as providing scaleable and generic technology.

We intend to define the capability of the underlying information sources as a set of interfaces, with varying sophistication. This set of interfaces will include the definition of the minimal capability necessary to support Structured Maps. But it will also include additional, optional interfaces that allow the Structured Map technology to support functions with better semantics or better performance. As an example, an information source might be able to compare two information source address and tell us whether they are related (e.g., equivalent, one contained inside the other). An information source might

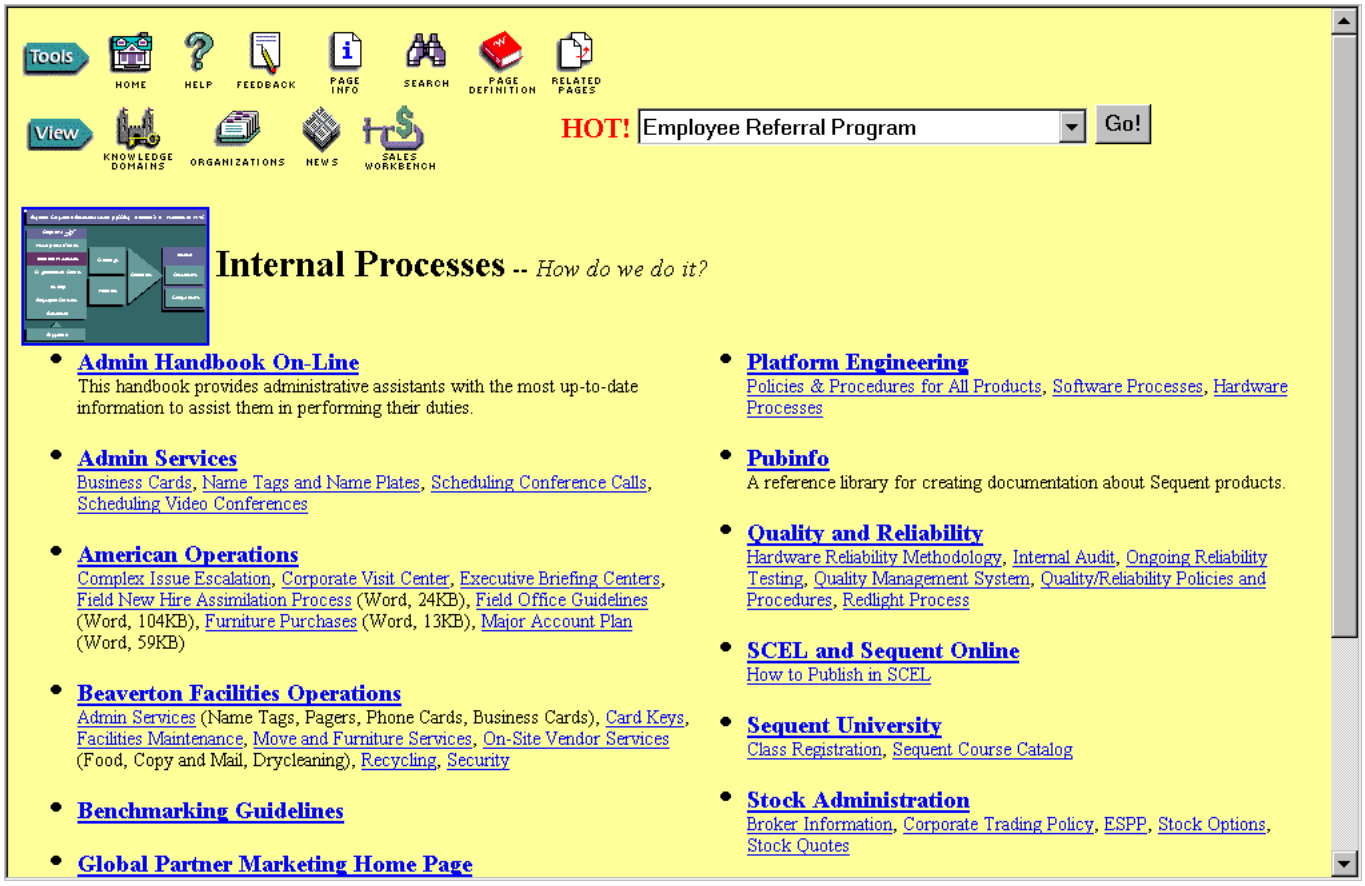

Fig. 10. Second Level Table of Contents in SCEL

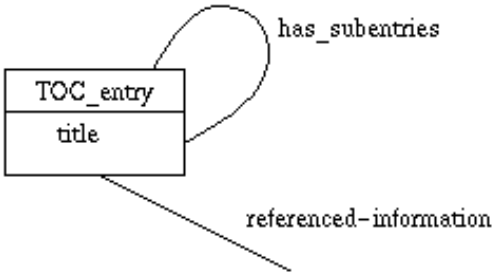

Fig. 11. Simple Structured Map definition for Table of Contents

be able to remember the downward traversals over facet instances in order to support upward traversal into the Structured Map. An information source might be able to set triggers, e.g., to inform the Structured Map technology when a given information element is touched during navigation. In general, we envision Structured Map technology where we can leverage such additional capabilities when they are available in the underlying information sources and compensate for them, in some cases, when they are not. As an example, we can build a secondary index structure to deliver all Structured Map facet instances where a given information element address appears, in order to support upward traversal of facet links.

Another issue concerns the data definition and query language for Structured Maps. For the moment we have not defined a query language; we have simply placed Structured Map definition and instance information into an Informix database. We are less inclined to define a new language and more inclined to focus on the semantics and processing of a loosely coupled environment. We will likely use a database language that support objects and navigation, e.g., ODL and OQL [3]. From a lan-

guage point of view, we envision Structured Maps using embedded languages for addressing and even for querying the Information Universe, much like SGML supports arbitrary notations and HyTime supports addressing in an external syntax. Structured Maps basically include addresses that are bound outside of the system.

The final issue has to do with the expressiveness of Structured Maps. In particular, are attributes allowed for entities? for relationships? The current implementation of CARTE and EnLIGHTeN provide for only a single attribute for entities, called Title. In EnLIGHTeN and in the definition of Topic Navigation Maps, the Title is optional, multi-valued, and need not be unique. The Title is only used to label the entity when the Topic Navigation Map is presented visually. In CARTE, we used the Title attribute for the same purpose but we might prefer to use the Title as a key. We are exploring a Structured Map model without any attributes (not even title) because we believe that it will simplify semantic integration of information. We propose to represent entities only by their internal identifiers and we propose to use facets as the only mechanism for elaborating and describing entities. Said another, way entities are represented in the Structured Map only by their OIDs. Semantic integration of two Structured Maps would require that identical entity instances be identified, as always [29]. But the opportunity for structural conflict is greatly reduced without attributes. In the best case, the integration of Structured Maps could reduce to the union of entity, relationship, and facet types plus the union of entity, relationship, and facet instances.

Philosophically, this type of an attribute-free model states that the essence of an entity is not in its representation. The entity instance is known only by its unique, internal identifier. Descriptive information can be found in various forms among the relevant information elements. We intend to investigate this view even when the underlying information sources are databases, with schemas, in order to support simplified semantic integration.

As a final comment, there are a number of aspects of Structured Maps that are important but outside of our current focus. These include: implementation-level connections with underlying information sources, update, and the consistency of links, etc. We do not focus on these issues, in part, because there is evidence that such work is in progress, elsewhere, e.g., through object interoperation models and various database-oriented implementations of WWW servers and other work on interoperability, e.g., [21, 24].

Acknowledgement. The authors gratefully acknowledge the technical support, through personal interaction and use of technical products, from Michel Biezunski and Catherine Hamon of High Text, S.A.R.L. and Steven Newcomb of TechnoTeacher, Inc. Both companies graciously made their products, EnLIGHTeN and HyMinder[5], available for use in this research. The authors also acknowledge the technical assistance of Roger Swanson of Sequent Computer Systems, Inc., director of the SCEL design team.

References

1. S. Abiteboul, D. Quass, J. McHugh, J. Widom and J. Wiener. The lorel query language for semistructured data. *Journal of Digital Libraries*, to appear 1997.
2. S. Adali and V. S. Subrahmanian. Amalgamating knowledge bases, II: Distributed mediators. *International Journal of Intelligent Cooperative Information Systems*, December 1994.
3. T. Atwood, et al. *The Object Database Standard:* ODMG - *93.* Morgan Kaufmann, 1994.
4. M. Biezunski. Modeling hyperdocuments using the topic map architecture. In *International HyTime Conference*, Vancouver, August 1995. available under HyTime conferences at http://www.techno.com.
5. P. Buneman, S.B. Davidson, and D. Suciu. Programming constructs for unstructured data. In *Proceedings of the Fifth International Workshop on Database Programming Languages.* Gubbio, Italy, September 1995.
6. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. In *Proceedings of the 100th IPSJ Anniversary meeting*, 1994.
7. V. Christophides, V. Abiteboul, S. Cluet, and M. Scholl. From structured documents to novel query facilities. In *Proceedings of ACM SIGMOD Conference*, Minneapolis, MN, May 1994.
8. C. Clifton, H. Garcia-Molina, and D. Bloom. Hyperfile: A data and query model for documents. *VLDB Journal.* 4(1):45-86, 1995.
9. S.J. DeRose and D.G. Durand. *Making Hypermedia Work.* Kluwer Academic Publishers, 1994.
10. U. Flohr. Hyper-G organizes the web. *Byte*, November 1995.
11. G. Gonnet and F.W. Tompa, Mind your grammar: A new approach to modeling text. In *Proceedings 13th VLDB Conference.* pages 339-346, Brighton, UK, August 1987.
12. C. Kacmar and D. Jue. The information zone system. In *Digital Library Initiative, IEEE Computer Magazine*, volume 29, pages 46–47. May 1996.
13. D. Konopnicki and O. Shmueli. W3QS: A query system for the World Wide Web. In *Proceedings of the 21st VLDB Conference.* Zurich, Switzerland, 1995.
14. W. Lee and D. Woelk. Integration of text search with ORION. *IEEE Data Engineering.* 13(1):56-62, March 1990.
15. C. Lynch and M. Stonebraker. Extended user-defined indexing with applications to textual databases. In *Proceedings of the 14th VLDB Conference.* Los Angeles, CA, August 1988.
16. S. Magavi, J. Wong, and P. Bodla. Design and implementation of heterogeneous distributed multimedia systems using mosaic GSQL. In *Software - Practice and Experience,* volume 25, pages 1223-1241, November 1995. John Wiley & Sons, Ltd.
17. H. Maurer. Hyper-G: Advancing the ideas of World-Wide-Web. available at http://www.chemie.fu-berlin.de/-outerspace/doc/hyper-g-abs.html.
18. T. Minohara, R. Watanabe, and M. Tokoro. Queries on structures in hypertext. In David B. Lomet, *Lecture Notes in Computer Science: Foundations of Data Organization and Algorithms.* 730, pages 394-411, Chicago, Illinois, October 1993. Springer-Verlag.
19. R. Ofali, D. Harkey, and J. Edwards. *The Essential Distributed Objects Survival Guide.* pages 43–216. John Wiley & Sons, 1996.

[5] HyMinder is a trademark of TechnoTeacher Inc.

20. R. Ofali, D. Harkey, and J. Edwards. *The Essential Distributed Objects Survival Guide*, pages 283–296. John Wiley & Sons, 1996.
21. A. Paepcke, S. Cousins, H. Garcia-Molina, S. Hassan, S. Ketchpel, M. Roscheisen, and T. Winograd. Using distributed objects for digital library interoperability. In *Digital Library Initiative, IEEE Computer Magazine*, volume 29, pages 61–68. May 1996.
22. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object_Oriented Modeling and Design*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1991.
23. R. Sacks-Davis, T. Arnold-Moore, and J. Zobel. Database systems for structured documents. In *Proceedings International Symposium on Advanced Database Technologies and Their Integration (ADTT'94)*, Nara, Japan, October 1994.
24. B. Schatz, W. Mischo, T. Cole, J. Hardin, A. Bishop, and H. Chen. Federating diverse collections of scientific literature, volume 29(5), *digital library initiative, IEEE computer magazine* 28-36. May 1996.
25. A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. In *ACM Computing Surveys*, volume 22, pages 183–237, Association for Computing Machinery, Inc., September 1990.
26. T. R. Smith. A digital library for geographically referenced materials. In *Digital Library Initiative, IEEE Computer Magazine*, volume 29, pages 54–60. May 1996.
27. T. R. Smith and J. Frew. Alexandria digital library. In *Digital Library Initiative, IEEE Computer Magazine*, volume 29, pages 61–62. May 1996.
28. A. Tomasic, H. Garcia-Molina, and K. Shoens. Incremental updates of inverted lists for text document retrieval. In *Proceedings of the ACM SIGMOD Conference*. Minneapolis, MN, May 1994.
29. Y.R. Wang and S.E. Madnick. The inter-database instance identification problem in integrating autonomous systems. *Proceedings of the Fifth International Conference on Data Engineering*. IEEE Computer Society Press, pages 46–55, February 1989.
30. T.W. Yan and J. Annevelink. Integrating a structured-text retrieval system with an object-oriented database system. In. *Proceedings 20th VLDB Conference*. Santiago, Chile, August 1995.
31. J. L. Zhao, A. Segev, and A. Chatterjee. A universal relation approach to federated database management. In *Proceedings Eleventh International Conference on Data Engineering*, pages 261–270, Taipei, Taiwan, March 1995. IEEE Computer Society Press.
32. Digital libraries. Communications of the ACM, 38(4), pages 22-86, (collection of papers on various aspects of digital libraries), April 1995.
33. Digital library initiative. IEEE Computer Magazine, 29(5), pages 22-76, (collection of papers on various aspects of digital libraries), May 1996.
34. High Text Web Site: http://hightext.com. *High Text Home Page*.
35. *Information Processing – SGML Applications – Topic Navigation Maps, WG8 N 1806, ISO/IEC JTC1/SC1/WG8, Document Processing and Relation Communication – Document Description and Processing Languages, (committee draft)*.
36. Informix Software Inc., http://www.informix.com. *Informix Home Page*.
37. *Modern SGML, Explicit Foundations, Create New Possibilities*. High Text, S.A.R.L., 5 rue d'Alsace, 75010 Paris France and Techno Teacher Inc., 3800 Monroe Avenue, Pittsford, New York, 14534-1330 USA.
38. *Overview of the HyTime Standard*. High Text, S.A.R.L., 5 rue d'Alsace, 75010 Paris France and Techno Teacher Inc., 3800 Monroe Avenue, Pittsford, New York, 14534-1330 USA.
39. Techno Teacher Inc., http://www.techno.com. *TechnoTeacher Home Page*.

This article was processed by the author using the LaTeX style file from Springer-Verlag.