

Using Topic Maps

for the representation, management & discovery of knowledge

Eric Freese

ISOGEN International, Dallas, USA

eric@isogen.com

<http://www.isogen.com>

Abstract:

In the AI (artificial intelligence) arena, there is a knowledge representation technique called a semantic network. A semantic network is created using a structure consisting of nodes and links. The nodes represent objects, concepts, or situations within a specific domain. The links represent and define relationships between the nodes. Semantic networks are often used to represent the knowledge of human experts in AI applications called inference engines or expert systems.

In 1999 an international standard was developed to describe a mechanism for representing information about the structure of information and organizing it into "topics". These topics have occurrences and associations that represent and define relationships between the topics. Information about the topics can be inferred by examining the associations and occurrences linked to the topic. A collection of these topics and associations is called a topic map.

Even at a high level there is an apparent similarity in the structure of these concepts. This similarity led the author to explore some interesting possibilities:

Is it possible/reasonable to build a semantic network from a topic map?

Is it possible/reasonable to store semantic network information in a topic map?

Would it be possible to design a computer program that identifies the knowledge contained within chunks of text?

If such a program could be built, would a computer be able to identify and interpret the knowledge found within a collection of documents?

In such a system, a user might be able to query the database for specific information. This system could be used to interpret the knowledge contained within the nodes. The user could begin a browsing session based on a piece of knowledge desired. The user could also request that the system interpret the knowledge in the database without manually browsing through the nodes.

This paper will discuss topic maps and semantic networks and how the two concepts may interrelate. Issues with the topic map standard that make knowledge representation more difficult will be discussed. Also a semantic network system built on topic maps will be presented.

Example application - the family tree

For illustration throughout this paper, a genealogical chart (i.e. family tree) will be used to explain topic maps and semantic networks and how they could be used to model a knowledge base. Family trees are used to express relationships between people, where topic maps and semantic networks are used to describe relationships between data items. Examining and compiling the relationships between the nodes of any of these networks can make certain inferences. For example, in the diagram below, Eric, Becky and Dawn are siblings because they share the same parents. Keri and Olivia are cousins because their parents are siblings. Cara is Carmen's grandparent because Carmen's parent is Cara's child.

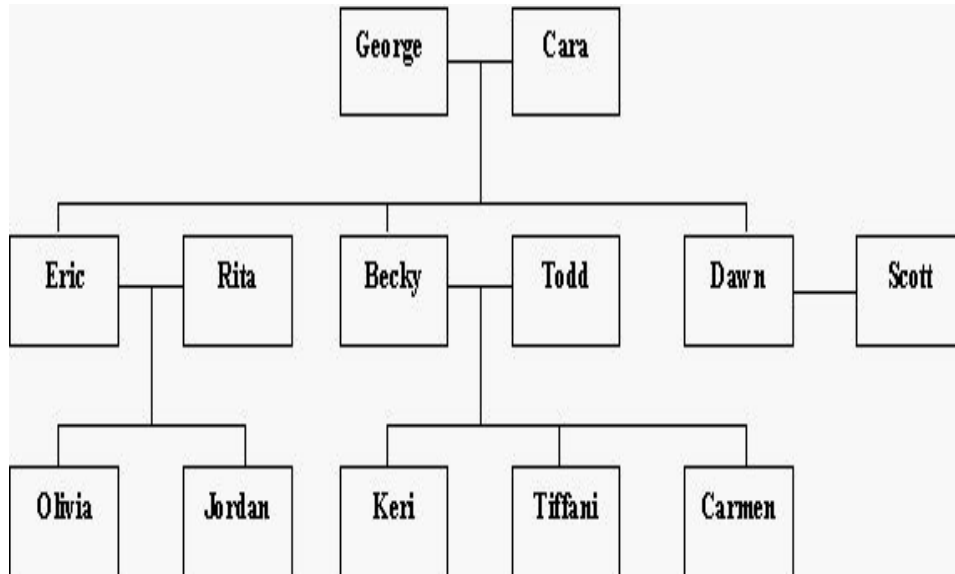


Figure 1. Genealogical chart

Semantic networks - an introduction

The semantic network is a representation formalism used in AI research. Semantic networks consist of nodes and links. Nodes usually represent objects, concepts, or situations within a specific domain. Links represent semantic relations between the nodes. Both the nodes and the links can have labels. Using the genealogical chart (Figure 1), it is possible to represent a simple fact such as "father is a parent" in a semantic network. This is done by creating two nodes to designate "father" and "parent". A link is created specifying an "is-a" relationship between the nodes (Figure 2).

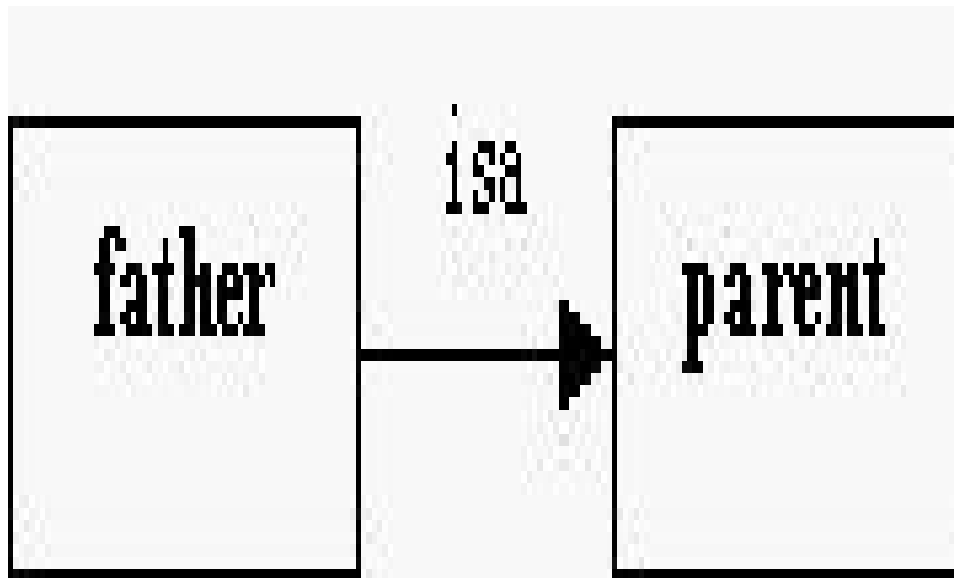


Figure 2. Simple fact

If George were a particular individual who we wished to assert is a father, we could add a node for George to the network as shown in Figure 3:

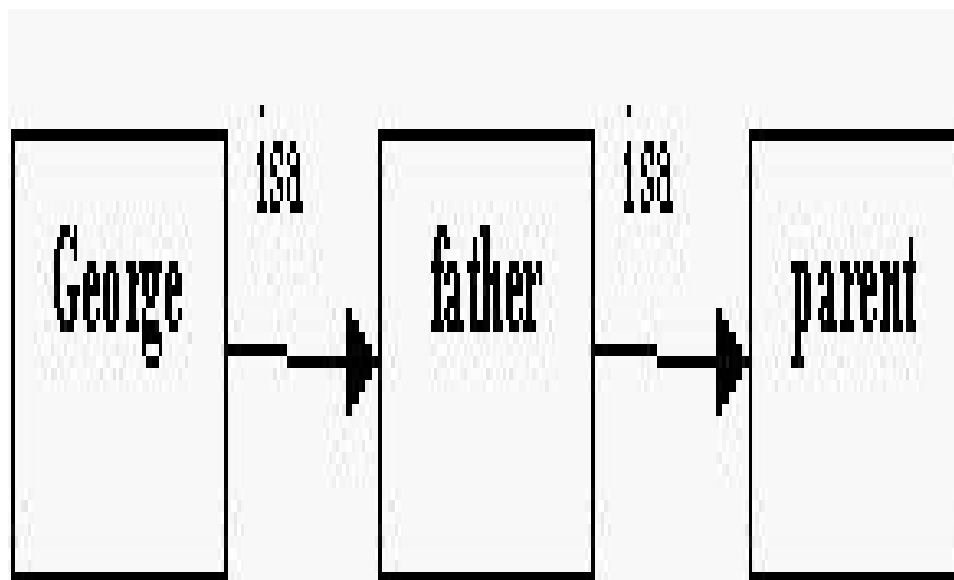


Figure 3. Inherited (transitive) fact

Notice that in Figure 3 we have not only represented the two initial facts ("father is a parent" and "George is a father"), but also deduced a third fact that "George is a parent" by simply following the links. The ability to deduce new facts based on semantic relationships is called "transitivity". Transitive relationships allow new relationships to be derived by simply creating new links. However, transitive relationships usually go in one direction. So we can say that "George is a father", but we would be incorrect to imply that "father is a George". It is possible, though, to create links that do flow in the opposite direction. Based on this, a new link could be established which says that "parent might be a father" or "father might be George".

The best way to test whether a transitive relationship exists is to test whether the following statement can be truthfully stated: "All instances of topic A have a specific relationship with Topic B." The statement: "Father is a parent" is true since "all fathers are parents." However the statement "Parent is a father" is not true since not all instances of parents are fathers.

Reflexive relationships occur when the link can be applied in all directions within set of nodes being related. Within the genealogy chart a statement such as "person is related to person" can be considered reflexive. This can be illustrated since both of the following statements are true: "Eric is related to Cara" and "Cara is related to Eric."

Symmetric relationships occur when the positioning of the nodes within the relationship does not affect the truthfulness of the resulting statement. For example, the following symmetrical statement can be made: "Parent has a child" and "Child has a parent". The "is related to" relationship mentioned above is also symmetrical.

Semantic networks make it easy to model inheritance hierarchies. By tracing through the hierarchy, facts asserted in higher nodes can be asserted about the lower ones without having to represent these assertions explicitly.

Computer languages such as Prolog have been designed which are able to model the logic contained within a semantic network. They allow the programmer to define the semantics of links programmatically so that a computer can understand and process the links and make inferences about the nodes based on the link semantics.

Semantic networks are frequently used to model the knowledge stored within expert systems. Expert systems use facts and rules to analyze complex set of data and make inferences based on the data and other input stimuli. The bits of knowledge that are stored within the semantic network are combined in such a way that a computer program can infer information about a node by following the links within the network.

Topic Maps - an introduction

Topic maps, as defined in ISO/IEC 13250, are used to organize information in a way that can be optimized for navigation. Topic maps were designed to solve the problem of large quantities of unorganized information. Information is not useful if it cannot be found or linked. In the paper publishing world, there are several mechanisms to organize and index the information contained within a book or document. Indexes allow readers to go directly to the portion of the document that is relevant to their information need. Topic maps can be thought of as the online equivalent of printed indexes. Topic maps are also a powerful way to manage link information, much as glossaries, cross-references, thesauri, and catalogs do in the paper world.

Topic maps are built of units called topics. In linguistic terms, a topic can be anything that is noun. A topic can have many links that point to all its occurrences. A topic link aggregates every portion of information that is about a given subject within a given information set. Every box in the genealogical chart (Figure 1) can be considered a topic.

Topics normally have names associated with them, although not always. A simple cross-reference, such as "see page 61," is considered to be a link to a topic that has no explicit name. In a genealogical setting, there are various kinds of names, such as: given names, nicknames, maiden names, and aliases. The standard defines the following types of name: base name (required), display name (optional), and sort name (optional). The base name is a name by which a topic may be known. For example, Rita might be known as:

Rita Doe

Doe, Rita
Rita Evaline Richardson
Rita Richardson Doe
Doe, Rita R.

The display name specifies the name to be displayed by an application to a user, when the name(s) specified by the base name within should not be used for display purposes. The sort name specifies a name that is to be used to represent the topic in a sorting process that arranges a list of topics in some order, when the name specified by the base name(s) should not be used for that purpose. A limitation of the standard is that these names must be unique. In some domains this limitation might cause a problem. For example, in genealogical data, names are often reused. This limitation might also have an impact in cases where different topic maps are to be merged or where a program is attempting to locate and create topics from flowing text. Scopes, which will be discussed later, have been offered as a possible remedy to this problem. However, there are other SGML/XML capabilities that would seem to be more appropriate.

Topics can be grouped into classes called topic types. A topic type is a category to which one given topic instance belong. A topic can have one or more topic types. The topic types within a given topic map are defined by the designers of each topic map and can be treated as topics themselves. For example, when talking about a family, a gender or family role topic type can be used to group a set of topics within the topic map. In the chart above, depending on the relationship, Eric may have the type of male, father, son, or husband.

A topic can also have one or more occurrences. A topic occurrence is an occurrence (or set of occurrences) of a topic within one or more addressable information resources. In a genealogical setting, occurrences of a topic may be in various items such as birth certificates, marriage licenses, real estate titles, and published papers. Such occurrences are generally outside the topic map document itself (although some of them could be inside it), and they are pointed at using whatever mechanisms the system supports, typically HyTime addressing or XPointers. Occurrences may be of any number of different types. The standard defines the typing of an occurrence as a role. Just like topic types, occurrence roles can be treated as topics.

Topics can be related together using associations which can express a given semantic. Topic map designers can define any kind of semantics for topic associations. For example, in the genealogical data set, associations such as "Dawn is a child of Cara" or "Olivia and Jordan are siblings" can be defined. Associations are ordinary links that are constrained to only relate topics together. Because they are independent of the source documents in which topic occurrences are to be found, these associations represent a knowledge base that contains the essence of the information, actually representing the essential value of the information. An unlimited number of topics can be associated within topic associations. Within topic maps, associations are also treated and managed as topics; therefore, there may be topics for associations such as spouse, child, and parent. A possible limitation of the standard is that associations are inherently defined to be class-instance relationships. It does not define mechanisms for defining, in a standard way, other types of relationships such as superclass-subclass, but leaves them to the individual implementations.

Just as topics have type and occurrences have roles, associations between topics can be grouped according to their type. The association type for the relationships mentioned above might be:

Is child of
Is sibling of

Is married to

Is parent of

As with most other constructs in the topic map standard, association types are themselves regarded as topics.

The ability to apply types to topic associations increases the expressive power of the topic map, making it possible to group together the set of topics that have the same relationship to any given topic.

Each topic in an association has a role that states the role played by the topic in the association. In the case of the relationship "Dawn is a child of Cara," expressed by the association between Dawn and Cara, those roles might be "child" and "mother." Association roles are also treated as topics.

Topic associations are not one-way. The "is a child of" relationship between Dawn and Cara implies a "is a parent of" between Cara and Dawn. Sometimes associations are symmetrical, in the sense that the nature of the relationship is the same whichever way you look at it. For example, in the association, "Olivia and Jordan are siblings," the association "is a sibling of" would apply work in either direction between Olivia and Jordan. Sometimes the anchor roles in such symmetrical relationships are the same (i.e. "siblings"). Sometimes these anchor roles are different (as in the case of the parent and child roles). Sometimes the intended use of the information will dictate the type of anchor roles used. For example, in a marriage relationship, the association could be the same ("is the spouse of") or different ("is the husband/wife of").

Some association types can express inheritance of properties, such as those that express class/instance and part/whole relationships. For example, if we say that Rita is an instance of the mother class and that a mother is an instance of the parent class, we have implicitly said that Rita is a parent.

As may have been noticed, association roles and topic types are two different mechanisms for modeling basically the same information. It is up to the topic map designer to establish the parameters for using these different mechanisms for attaching semantic information to the topics. The manner in which the semantic information is attached to the topic or associations will have a large impact on how the processing system is able to use the topic map data. While this increases the flexibility of the standard, it could also have a negative impact on the interchangeability of topic maps.

Topics can have various characteristics assigned to them: names, occurrences, and roles. The different kinds of assertions that can be made about a topic are collectively known as topic characteristics. These characteristics are considered to be valid within certain limits. The limit of validity of such an assignment is called its scope. The concept of scope is important to avoid ambiguities between topics and their characteristics. Any assignment of a characteristic to a topic is considered to be valid within certain limits, which may or may not be specified explicitly. The limit of validity of such an assignment is called its scope. A scope is defined in terms of themes and themes are topics. For example in order to distinguish between Rome in Italy and Rome in New York, scopes of "Italy" and "New York" may be assigned. However, while scopes allow for the differentiation of topics and characteristics they still do not solve the problem cause by the required uniqueness of names.

For example, because of the chart above, when I refer to "Dawn," the reader knows that I am speaking of a specific person within a specific family and not the beginning of the day. The chart is itself presenting a scope. Within the topic map standard, there is a mechanism for specifying scope explicitly and handling situations in which the use of implicit scoping might otherwise lead to errors or ambiguities, such as when merging topic maps.

Figure 4 shows the genealogical chart redrawn as a topic map. Note that all the boxes from the original chart reside in the topic map chart. However, new topics have been added for the different topic types, allowing for inheritance of characteristics and inferring of information about the different topics. Now as new topics are added to the map and linked to existing topics, information can be inferred about them simply by following the links. There are also member relationships that can be modeled as associations with roles or as topics themselves. In this example, they are have been modeled as associations with roles.

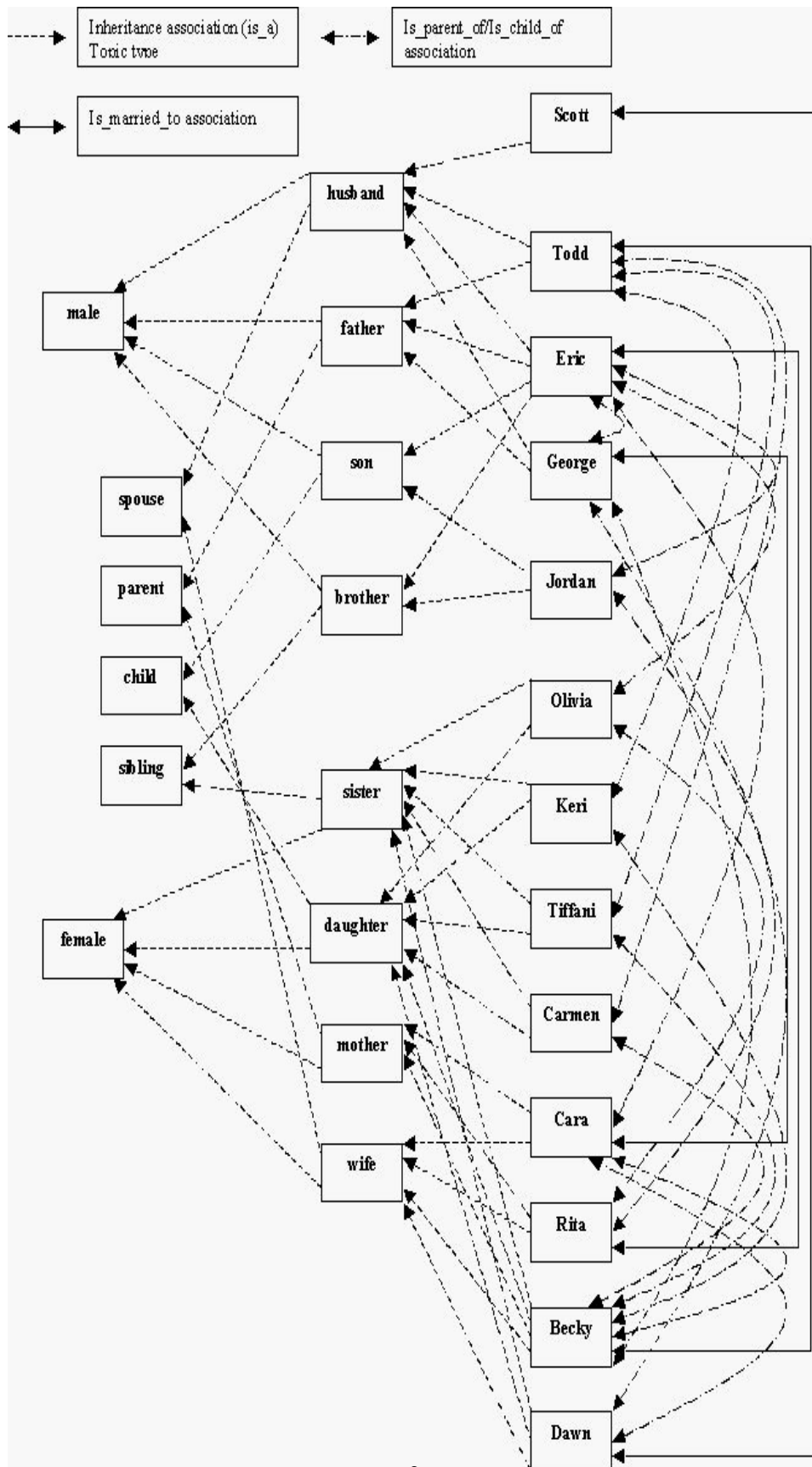


Figure 4. Genealogical Topic Map

The topic map standard uses architectures, as defined in ISO 10744, to define the topic map structures. This allows any topic map application built using these architectures to interchange data. However, any extensions an application may make to the standard may be lost in the interchange process, including any semantics assigned to associations, scopes and themes.

XML Topic Maps (XTM)

In 1999, GCA's IDEAlliance started a working group called TopicMaps.org to develop a web standard for topic maps on the web based on ISO/IEC 13250. The goal of this group to facilitate the creation and use of topic maps, focusing on but not limited to applications on the Web. The plan is to leverage the XML family of specifications as required. This group met before the conference to continue its work, which will be described during the conference.

A comparison - Topic Maps versus semantic networks

Although the descriptions have been brief, structural commonalities exist between topic maps and semantic networks:

Both topic maps and semantic networks are organized into a network of information nodes or modules.

Both topic maps and semantic networks allow the user to model links between the nodes.

Both topic maps and semantic networks allow the user to attach semantic information to the nodes and the links.

There is also a basic difference:

Topic maps seem to focus more on the navigation between topics than on the associations. Semantic networks focus on the links between the nodes and the knowledge that is represented by the linked nodes.

These similarities raise some interesting questions:

Is it reasonable to build a semantic network from a topic map?

Is it reasonable to store semantic network information in a topic map?

Would it be possible to design a computer program that identifies the knowledge contained within chunks of text?

If such a system could be built, would a computer be able to identify and interpret the knowledge found within a collection of documents using semantic networks and topic maps?

Before answering these questions, we first must raise some issues that may affect the ability to use topic maps to model semantic networks.

Issues in the Topic Map standard affecting the ability to model semantic networks

Limited association types

As discussed earlier, the topic map standard only defines the class-instance relationship between topics by using the types attribute. It does not define mechanisms for defining, in a standard way, other types of relationships such as superclass-subclass. Definition of these other relationships is left to the individual implementations. While this may have simplified the development of the standard, it has created a potentially huge problem for topic map application interoperability.

At XML '99, Steve Pepper and Hans Holger Rath presented a paper that detailed a set of association types that express the basic concepts of knowledge representation. These relationships include:

- component-object
- member-collection
- portion-mass
- stuff-object
- feature-activity
- place-area
- phase-process

These relationships would seem, at least a good starting point, if not likely candidates for standardization in order to maximize the interoperability between topic map applications that attempt to apply semantics to the associations.

If individual implementations are left to define common relationships and semantics, then the probability of successful interchange of more than the most rudimentary topic map is not very high, especially in cases where a great deal of machine processing is dependent on the semantic within the relationships, such as within a semantic network.

Association occurrences

As stated above, topic maps seem to concentrate more on topics, leaving associations more or less as second-class citizens. One of the example uses of topic maps is for navigation within a collection of documents, based on topics. However, the standard makes no allowance for allowing associations to have occurrences. This would allow the inference of a fact to be linked to the source document from which the association was derived.

Again, it might be possible to define a topic that represents a fact, or topics connected by an association. However, this seems rather awkward and prone to error, especially in systems with the ability to build topic maps on the fly. Also, without standardization, the possibility of interoperability of such data is questionable.

Association templates

In order to more accurately define the relationship between one or more topics, it is reasonable that some sort of mechanism be developed by which a general template for an association can be defined. This would allow other associations to reference it and inherit the rules (semantics) set forth. This would also provide a

mechanism for validity checking of associations, reducing the instances of bad topic associations within the topic map.

Given following portion of the topic map defined in Appendix A:

```
<topic id="male">
  <topname><basename>Male</basename></topname>
</topic>
<topic id="female">
  <topname><basename>Female</basename></topname>
</topic>
<topic id="parent">
  <topname><basename>Parent</basename></topname>
</topic>
<topic id="spouse">
  <topname><basename>Spouse</basename></topname>
</topic>
<topic id="sibling">
  <topname><basename>Sibling</basename></topname>
</topic>
<topic id="child">
  <topname><basename>Child</basename></topname>
</topic>
<topic id="mother" types="female parent">
  <topname><basename>Mother</basename></topname>
</topic>
<topic id="father" types="male parent">
  <topname><basename>Father</basename></topname>
</topic>
<topic id="wife" types="female spouse">
  <topname><basename>Wife</basename></topname>
</topic>
<topic id="husband" types="male spouse">
  <topname><basename>Husband</basename></topname>
</topic>
<topic id="sister" types="female sibling">
  <topname><basename>Sister</basename></topname>
</topic>
<topic id="brother" types="male sibling">
  <topname><basename>Brother</basename></topname>
</topic>
<topic id="daughter" types="female child">
  <topname><basename>Daughter</basename></topname>
</topic>
<topic id="son" types="male child">
  <topname><basename>Son</basename></topname>
</topic>
<topic id="eric" types="husband father son brother">
  <topname><basename>Eric</basename></topname>
</topic>
<topic id="rita" types="wife mother">
  <topname><basename>Rita</basename></topname>
</topic>
<topic id="olivia" types="daughter sister">
  <topname><basename>Olivia</basename></topname>
</topic>
<topic id="jordan" types="son brother">
```

```

    <topname><basename>Jordan</basename></topname>
</topic>

<assoc type="is-married-to">
  <assocrl anchrole="husband">eric</assocrl>
  <assocrl anchrole="wife">rita</assocrl>
</assoc>
<assoc type="is-parent-of">
  <assocrl anchrole="father">eric</assocrl>
  <assocrl anchrole="mother">rita</assocrl>
  <assocrl anchrole="child">olivia jordan</assocrl>
</assoc>

```

There are associations here between the members of this family. A human reader can probably figure out how the relationship works. However, the standard provides no guidance or mechanism in how such relationships to be programmatically derived. It would be helpful to have a mechanism to define how n-ary relationships can be interpreted. In such a model it would be possible to define:

- The member topic types of the association
- How many of each type can occur within the association
- The associations between the different topic types, in all directions
- The properties of the associations (reflexive, transitive, symmetrical)
- The types of the associations

Given this capability, it would then be possible to re-define the "is-parent-of" association above as follows:

```

<assoc-template name="parent-child">
  <topic-member topic-type="parent" occurs="+"/>
  <topic-member topic-type="child" occurs="+"/>
  <rule reflexive="0" transitive="0" symmetrical="0" type="member-collection">
    <topic-rl type="parent"/>
    <assoc-rl type="is-parent-of"/>
    <topic-rl type="child"/>
  </rule>
  <rule reflexive="0" transitive="0" symmetrical="0" type="member-collection">
    <topic-rl type="child"/>
    <assoc-rl type="is-child-of"/>
    <topic-rl type="parent"/>
  </rule>
  <rule reflexive="1" transitive="0" symmetrical="1" type="member-collection">
    <topic-rl type="child"/>
    <assoc-rl type="is-sibling-to"/>
    <topic-rl type="child"/>
  </rule>
</assoc-template>

<assoc type="parent-child">
  <assocrl anchrole="father">eric</assocrl>
  <assocrl anchrole="mother">rita</assocrl>
  <assocrl anchrole="child">olivia jordan</assocrl>
</assoc>

```

The topic type attribute within the topic-member element uses the transitivity defined within the standard to define in as general of level as possible the topics that can participate within the association. In this example

more specific instances of "parent" could be used, such as "father" or "mother." The occurs attribute specifies how many of each topic type can participate in the association, based on SGML/XML occurrence indicators, with the default value being "1." The rule element specifies the properties of the association and which type of association is being defined. Two rules are defined here to allow a reverse association to be defined for the "is-parent-of" association. A third rule allows a new association to be built without having to specifically code it in the source topic map. Some associations such as the sibling relationship can be derived from the associations within the topic map. Others, such as cousin relationships, might require specific rules to be developed and applied to the topic map information.

This template now provides all the information necessary for a system, or a human reader unfamiliar with the subject matter, to establish the relationships between the topics within the association. It also clarifies the items and the associations among them.

Building semantic networks from Topic Maps

Providing a general statement concerning the ability to create semantic networks from topic maps is difficult. While the connection can be made between the two models based on the structural similarities, the actual information stored in the topic map will largely dictate whether it can be used to build a semantic network. If the topic map is built purely for navigation, then its usefulness in building a semantic network may be limited. However, if the descriptive and associative mechanisms defined in the topic map standard are implemented, then some semantic information can be extracted from the topic map to build or add on to a semantic network.

Reconsider the family tree that shows all the topics/nodes and the relationships between them. If the topic map built from the family tree in Figure 1 only listed the topics (names) of the people, it would not be very useful as a semantic network. However, if the relationships are modeled, as in Figure 4, a great deal of semantic information could be derived from the associations and topics.

As stated previously the flexibility of the standard might also serve as a hindrance to a general method for building semantic networks from topic maps. Because of its generality, there are many ways to model the same information. This will lead to different interpretations of the standard and different implementations based on the standard. This freedom will make a generalized methodology for the creation of semantic networks from topic maps, difficult, if not impossible.

Another possible hindrance is that the associations between the topics must be human understandable, per the standard. There is no mechanism within the standard for programmatically defining the semantics of the association, making it difficult for a semantic network based system to process the topic map beyond the simplest associations. The example above illustrates this point. For example, logic statements could be developed to define a cousin in a family tree by defining the relationship based on nodes and links in the network. In a topic map, the association must be made explicitly, if it is to be made at all.

Storing semantic network information in Topic Maps

While not all topic maps can be used to build a semantic network of much value, it should be possible to model most semantic networks using the topic map paradigm. The similarities in the structures allow the mapping to be relatively straightforward. Nodes in the semantic network can be mapped to topics. Links can be mapped to associations.

It should be noted that not all the capability and flexibility of the topic map standard should be used in converting semantic networks to topic maps. Unless the network is designed to take advantage of the special features of topic maps, many things (i.e. the multiple names and ability to treat everything as a topic) will not be used. This should not be an issue unless a specific topic map system depends on the existence of specific features.

One item to be considered is whether all the semantic information can be modeled in the topic map. If special functions have been developed defining the relationships between the nodes, it may be necessary to explicitly define the relationships rather than allowing a computer to infer the relationship based on predefined functions.

Capturing the knowledge contained within text

One of the benefits of XML (Extensible Markup Language) is the ability to define a set of mark up tags that explicitly label the content of a data set rather than using formatting tags such as those in the HTML (HyperText Markup Language) . By using content tagging, programs can be developed which identify certain topics within the information to populate a topic map or semantic network. However, the associations or relationships between the topics may not be explicitly stated in the markup. Tools must be developed which allow the user to define associations and topic types so that data extracted from documents can be placed into the topic map and interpreted by the computer.

One example of this process is the tool used by Michel Biezunski to create the topic maps for GCA (Graphic Communication Association) conferences. Papers are submitted using a standard DTD (Document Type Definition) , which contains several content tags such as company, city, state, country, keyword, and acronym. Based on the specific tags, topic maps can be built based on the associations between the marked items. In general, a city occurs within a state, so topics can be defined for each city and state and associations can be built between each city/state pair.

Identifying and interpreting the knowledge found within documents

The field of knowledge management has been gathering momentum over the past year or so. The definition of knowledge management depends on the individual doing the defining. In general, it is an attempt to classify and organize information within an enterprise so that this information can be located and used. Several tools and systems have been introduced, and they claim to perform some sort of knowledge management. However these systems range from simple document management systems to advanced repositories that purport to be able to process the meaning contained within the text to classify the information.

There are many mechanisms that are used within these systems to classify and organize the information. Some simply match keywords and phrases; others use statistical theory to match patterns of terms and contextual relationships that represent an idea.

Whether topic maps could be used to model the knowledge managed by these systems remains to be seen. At this time, no commercially available tool or system advertises the ability to use a topic map to interchange the knowledge contained within nor do they advertise that they can export a topic map for interchange of the information.

The SemanText System

Current status

The SemanText system is a demonstration topic map based application, written in Python, which builds semantic networks from topic maps. Nodes are created from topics and topic types. Links are created from associations between the topics. Additional information can be added to allow the semantic network processor to infer additional information beyond the class-instance relationship that is defined in the standard.

The system uses a customized HTML browser interface that presents the topic map information in a manner extremely familiar and intuitive to most users. By not using a tree diagram interface, circular links do not become a confusing issue in browsing through the information. Also, by using a browser interface, occurrence links can be followed and displayed directly from the topic maps application.

The user browses the topic map by selecting a topic or topic type. All information associated with the topic, within a given scope, is displayed including any related topic and topic types, associations, and links to all occurrences.

Topic maps can be merged in two ways. A full merge combines two topic maps into one, connecting and resolving common topics with user intervention. SemanText also allows a softer merge, called a reference merge, where the topic maps remain separate, but links are made to common topics. This allows the core topic map being used to remain separate while still being able to reference one or more other topic maps.

SemanText can also be used to build topic maps. The user can build topic maps by entering the information manually using a series of dialogs. However, users can also build topic maps by parsing XML and SGML files and extracting information from them into topics and associations. This automatic method uses a tree representation of the source file where the user can specify an element and how the element and its contents should be added to the topic map.

Future plans

Previous prototypes of the SemanText system used groves to represent the structure of the information. It is planned that the grove paradigm can be included in the full system again, once the basic topic map capability has been completed. This will allow non-SGML data to be accessible to the system, both for building topic maps, and for browsing occurrences of topics.

In many semantic network applications it is possible to assign weightings to the statements modeled in the nodes and links. These weightings tell the application the certainty value of a statement: the higher the value, the more factual or certain the statement. This allows the application to build inferences that can be weighted based on the information contained within the network. In the future, SemanText will include an inference engine that will be able to take confidence weighting into consideration. In addition, the inference engine will provide a mechanism where rules can be developed which allow the semantic network to be automatically enhanced as new topics and association are added to the semantic network. Work similar to this is currently taking place in the W3C's semantic web initiative.

A great deal of research has been in the area of natural language processing. It is hoped that a natural language input interface can be implemented so that SemanText can identify new topics and associations within flowing text.

Several output formats are being explored. Included among the possibilities are Open E-book, VRML or SVG, audio input and output using Voice XML, and others. These various outputs will demonstrate new ways to access and view data.

Conclusion

This paper presents the similarities between topic maps and semantic networks. The similarities between the two concepts are explored to determine whether they are truly interchangeable. Questions raised by these similarities are addressed to demonstrate that topic maps can be used to represent the information stored within a semantic network.

However, several issues still exist:

There are several items in the topic map standard that, while adding to the standard's flexibility and power, may hinder the ability to truly interchange topic map/semantic network data. These items include topic types vs. association roles, and the ability to model almost everything in a topic map as a topic.

The required uniqueness of the different types of names across the entire topic map is problematic. Scopes have been offered up as the solution for handling topic with similar names, but the use and definition of scopes requires a great deal of forethought in the design of the topic map. In many cases, though, the developer of the topic map might not have any knowledge of the future use of the topic map and thus does not define scopes or facets for the data.

The limited standard types of associations may also hinder interchangeability of topic maps since there is no standard way to interchange the semantic processing involved in some associations.

It is unclear how well topic maps will scale in large applications. The standard is still relatively new and there are but a few implementations. Time will tell as attempts are made to merge large topic maps.

While several issues exist, as topic maps become more widely used, standardized methodologies will be developed and accepted to assure the reliable interchange of data between topic maps applications. As this happens, semantic network tools will also be able to interchange their knowledge bases.

Genealogical Topic Map

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<topicmap>

  <!-- =====
        Topic types: Relationships
  -->

  <topic id="male">
    <topname><basename>Male</basename></topname>
  </topic>
  <topic id="female">
    <topname><basename>Female</basename></topname>
  </topic>

  <topic id="parent">
    <topname><basename>Parent</basename></topname>
  </topic>
  <topic id="spouse">
    <topname><basename>Spouse</basename></topname>
```

```

</topic>
<topic id="sibling">
  <topname><basename>Sibling</basename></topname>
</topic>
<topic id="child">
  <topname><basename>Child</basename></topname>
</topic>

<topic id="mother" types="female parent">
  <topname><basename>Mother</basename></topname>
</topic>
<topic id="father" types="male parent">
  <topname><basename>Father</basename></topname>
</topic>

<topic id="wife" types="female spouse">
  <topname><basename>Wife</basename></topname>
</topic>
<topic id="husband" types="male spouse">
  <topname><basename>Husband</basename></topname>
</topic>

<topic id="sister" types="female sibling">
  <topname><basename>Sister</basename></topname>
</topic>
<topic id="brother" types="male sibling">
  <topname><basename>Brother</basename></topname>
</topic>

<topic id="daughter" types="female child">
  <topname><basename>Daughter</basename></topname>
</topic>
<topic id="son" types="male child">
  <topname><basename>Son</basename></topname>
</topic>

<!-- =====
      Topic definitions: Associations
-->

<topic id="is-married-to">
  <topname><basename>is married to</basename></topname>
</topic>
<topic id="is-parent-of">
  <topname><basename>is the parent of</basename></topname>
</topic>
<topic id="is-child-of">
  <topname><basename>is the child of</basename></topname>
</topic>
<topic id="is-sibling-to">
  <topname><basename>is a sibling to</basename></topname>
</topic>

<!-- =====
      Topic definitions: People
-->

```

```

<topic id="george" types="husband father">
  <topname><basename>George</basename></topname>
</topic>
<topic id="cara" types="wife mother">
  <topname><basename>Cara</basename></topname>
</topic>
<topic id="eric" types="husband father son brother">
  <topname><basename>Eric</basename></topname>
</topic>
<topic id="becky" types="wife mother daughter sister">
  <topname><basename>Becky</basename></topname>
</topic>
<topic id="dawn" types="wife daughter sister">
  <topname><basename>Dawn</basename></topname>
</topic>
<topic id="rita" types="wife mother">
  <topname><basename>Rita</basename></topname>
</topic>
<topic id="todd" types="husband father">
  <topname><basename>Todd</basename></topname>
</topic>
<topic id="scott" types="husband">
  <topname><basename>Scott</basename></topname>
</topic>
<topic id="olivia" types="daughter sister">
  <topname><basename>Olivia</basename></topname>
</topic>
<topic id="jordan" types="son brother">
  <topname><basename>Jordan</basename></topname>
</topic>
<topic id="keri" types="daughter sister">
  <topname><basename>Keri</basename></topname>
</topic>
<topic id="tiffani" types="daughter sister">
  <topname><basename>Tiffani</basename></topname>
</topic>
<topic id="carmen" types="daughter sister">
  <topname><basename>Carmen</basename></topname>
</topic>

<!-- Associations: Married -->

<assoc type="is-married-to">
  <assocrl anchrole="husband">george</assocrl>
  <assocrl anchrole="wife">cara</assocrl>
</assoc>
<assoc type="is-married-to">
  <assocrl anchrole="husband">eric</assocrl>
  <assocrl anchrole="wife">rita</assocrl>
</assoc>
<assoc type="is-married-to">
  <assocrl anchrole="husband">todd</assocrl>
  <assocrl anchrole="wife">becky</assocrl>
</assoc>
<assoc type="is-married-to">
  <assocrl anchrole="husband">scott</assocrl>
  <assocrl anchrole="wife">dawn</assocrl>

```

```

</assoc>

<!-- Associations: Parent/Child -->

<assoc type="is-parent-of">
  <assocrl anchrole="father">george</assocrl>
  <assocrl anchrole="mother">cara</assocrl>
  <assocrl anchrole="child">eric becky dawn</assocrl>
</assoc>
<assoc type="is-parent-of">
  <assocrl anchrole="father">eric</assocrl>
  <assocrl anchrole="mother">rita</assocrl>
  <assocrl anchrole="child">olivia jordan</assocrl>
</assoc>
<assoc type="is-parent-of">
  <assocrl anchrole="father">todd</assocrl>
  <assocrl anchrole="mother">becky</assocrl>
  <assocrl anchrole="child">keri tiffani carmen</assocrl>
</assoc>

</topicmap>

```

Bibliography

- [BARR81] Barr, Avron and Feigenbaum, Edward A.: **The Handbook of Artificial Intelligence**, Reading, Massachusetts, 1981.
- [BIEZ99] Biezunski, Michel: **Topic Maps at a Glance**, Granada, Spain, 1999.
- [DEWD89] Dewdney, A. K.: **The Turing Omnibus**, Rockville, Maryland, 1989.
- [HARM85] Harmon, Paul and King, David: **Expert Systems**, New York, NY. 1985.
- [ISO13250] International Organization for Standardization: **ISO/IEC 13250:1999 Document description and processing languages - Topic Maps**, Geneva, 1999.
- [PEPP99] Pepper, Steve: Euler, **Topic Maps and Revolution**, Granada, Spain, 1999.
- [RATH99] Rath, Hans Holger and Pepper, Steve: **Topic Maps: Introduction and Allegro**, Philadelphia, PA. 1999.

Author

Eric Freese

Director of Professional Services - Midwest Region

ISOGEN International

Postal Address:

2200 N. Lamar Street #230

75202 Dallas

TX

USA

Telephone: +1 319 338 1333

Fax: +1 319 338 2207

E-mail: eric@isogen.com

Web: www.isogen.com

Eric Freese - Mr. Eric Freese is the Director of Professional Services - Midwest Region for ISOGEN International, a DataChannel company. Previously, he was the president and founder of the Electronic Data Foundry which was recently acquired by DataChannel. Mr. Freese has over 12 years of experience in the area of information and document management. His specific expertise is in the development of Standard Generalized Markup Language (SGML) products and implementation of SGML technologies including the Extensible Markup Language (XML), Document Style Semantics and Specification Language (DSSSL), Hypermedia/Time-Based Structuring Language (HyTime), Topic Maps, HyperText Markup Language (HTML) and the World Wide Web (WWW). This experience includes research, analysis, specification, design, development, testing, implementation, integration and management of database systems and microcomputer technologies in business, education and government environments. Mr. Freese has developed and implemented training programs and materials from elementary to graduate level. He also has research experience in human interface design, graphics interface development and artificial intelligence.