

## A Novel Expert System for Reasoning Based on Semantic Cancer

Taghi Karimi

Department of Mathematics, Payame Noor University, Tehran, Iran

### Abstract

This paper describes a programmatic framework for representing, manipulating and reasoning with geographic semantics. The framework enables visualizing knowledge discovery, automating tool selection for user defined geographic problem solving, and evaluating semantic change in knowledge discovery environments. Methods, data, and human experts (our resources) are described using ontologies. An entity's ontology describes, where applicable: uses, inputs, outputs, and semantic changes. These ontological descriptions are manipulated by an expert system to select methods, data and human experts to solve a specific user-defined problem; that is, a semantic description of the problem is compared to the services that each entity can provide to construct a graph of potential solutions. A minimal spanning tree representing the optimal (least cost) solution is extracted from this graph, and displayed in real-time. The semantic change(s) that result from the interaction of data, methods and people contained within the resulting tree are determined via expressions of transformation semantics represented within the JESS expert system shell. The resulting description represents the formation history of each new information product (such as a map or overlay) and can be stored, indexed and searched as required. Examples are presented to show (1) the construction and visualization of information products, (2) the reasoning capabilities of the system to find alternative ways to produce information products from a set of data methods and expertise, given certain constraints and (3) the representation of the ensuing semantic changes by which an information product is synthesized.

### I. Introduction

The importance of semantics in geographic information is well documented [1-4]. Semantics are a key component of interoperability between GIS; there are now robust technical solutions to interoperate geographic information in a syntactic and schematic sense (e.g. OGC, NSDI) but these fail to take account of any sense of meaning associated with the information. describe how exchanging data between systems often fails due to confusion in the meaning of concepts. Such confusion, or semantic heterogeneity, significantly hinders collaboration if groups cannot agree on a common lexicon for core concepts. Semantic heterogeneity is also blamed for the inefficient exchange of geographic concepts and

information between groups of people with differing ontologies [5-9].

Semantic issues pervade the creation, use and re-purposing of geographic information. In an information economy we can identify the roles of information producer and information consumer, and in some cases, in national mapping agencies for example, datasets are often constructed incrementally by different groups of people with an implicit (but not necessarily recorded) goal. The overall meaning of the resulting information products are not always obvious to those outside that group, existing for the most part in the creators' mental model. When solving a problem, a user may gather geospatial information from a variety of sources without ever encountering an explicit statement about what the data mean, or what they are (and are not) useful for. Without capturing the semantics of the data throughout the process of creation, the data may be misunderstood, be used inappropriately, or not used at all when they could be. The consideration of geospatial semantics needs to explicitly cater for the particular way in which geospatial tasks are undertaken [10]. As a result, the underlying assumptions about methods used with data, and the roles played by human expertise need to be represented in some fashion so that a meaningful association can be made between appropriate methods, people and data to solve a problem. It is not the role of this paper to present a definitive taxonomy of geographic operations or their semantics. To do so would trivialize the difficulties of defining geographic semantics

### II. Background and Aims

This paper presents a programmatic framework for representing, manipulating and reasoning with geographic semantics. In general semantics refers to the study of the relations between symbols and what they represent [11]. In the framework outlined in this paper, semantics have two valuable and specific roles. Firstly, to determine the most appropriate resources (method, data or human expert) to use in concert to solve a geographic problem, and secondly to act as a measure of change in meaning when data are operated on by methods and human experts. Both of these roles are discussed in detail in Section 3. The framework draws on a number of different research fields, specifically: geographical semantics [11-13] ontologies computational semantics, constraint-based reasoning and expert systems and visualization to represent

aspects of these resources. The framework sets out to solve a multi-layered problem of visualizing knowledge discovery, automating tool selection for user defined geographic problem solving and evaluating semantic change in knowledge discovery environments. The end goal of the framework is to associate with geospatial information products the details of their formation history and tools by which to browse, query and ultimately understand this formation history, thereby building a better understanding of meaning and appropriate use of the information.

The problem of semantic heterogeneity arises due to the varying interpretations given to the terms used to describe facts and concepts. Semantic heterogeneity exists in two forms, cognitive and naming. Cognitive semantic heterogeneity results from no common base of definitions between two (or more) groups. As an example, think of these as groups of scientists attempting to collaborate. If the two groups cannot agree on definitions for their core concepts then collaboration between them will be problematic. Defining such points of agreement amounts to constructing a shared ontology, or at the very least, points of overlap [12-17].

Naming semantic heterogeneity occurs when the same name is used for different concepts or different names are used for the same concept. It is not possible to undertake any semantic analysis until problems of semantic heterogeneity are resolved. Ontologies, described below, are widely recommended as a means of rectifying semantic heterogeneity. The framework presented in this paper utilizes that work and other ontological research to solve the problem of semantic heterogeneity.

The use of an expert system for automated reasoning fits well with the logical semantics utilized within the framework. The Java Expert System Shell (JESS) is used to express diverse semantic aspects about methods, data, and human experts. JESS performs string comparisons of resource attributes (parsed from ontologies) using backward chaining to determine interconnections between resources. Backward chaining is a goal driven problem solving methodology, starting from the set of possible solutions and attempting to derive the problem. If the conditions for a rule to be satisfied are not found within that rule, the engine searches for other rules that have the unsatisfied rule as their conclusion, establishing dependencies between rules. JESS functions as a mediator system, with a foundation layer where the methods, data, and human experts are described (the domain ontology), a mediation layer with a view of the system (the task ontology) and a user interface layer (for receiving queries and displaying results).

### 1.1 Ontology

In philosophy, Ontology is the “study of the kinds of things that exist” In the artificial intelligence community, ontology has one of two meanings, as a *representation vocabulary*, typically specialized to some domain or subject matter and as a *body of knowledge* describing some domain using such a representation vocabulary [18]. The goal of sharing knowledge can be accomplished by encoding domain knowledge using a standard vocabulary based on an ontology. The framework described here utilizes both definitions of ontology.

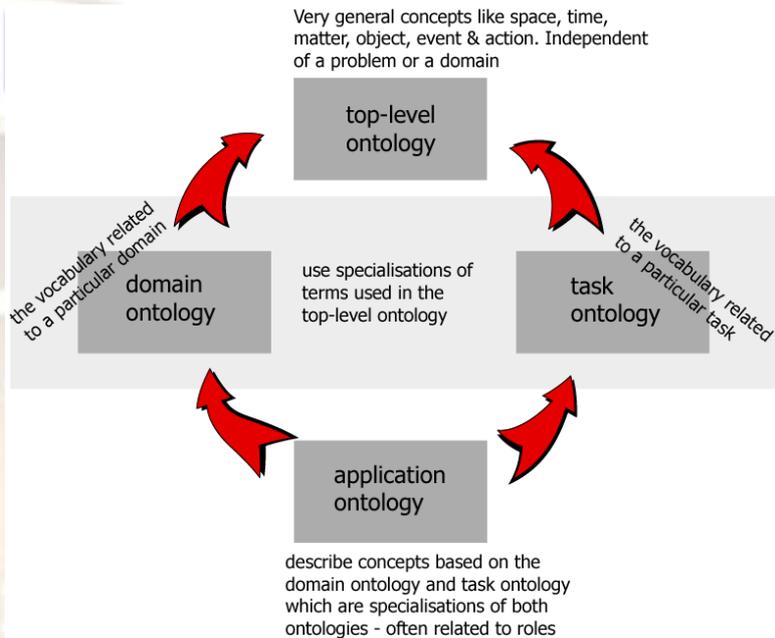


Figure 1 – Interrelationships between different types of ontology.

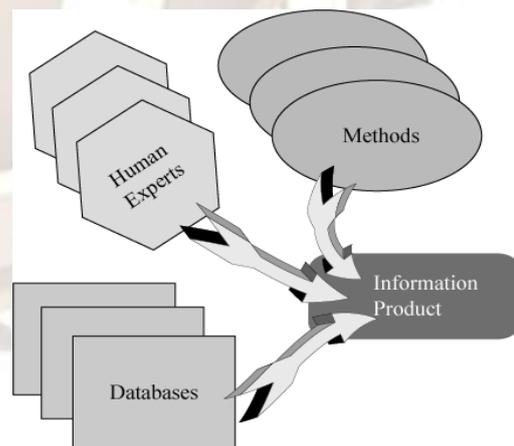


Figure 2 – Information products are derived from the interaction of entities.

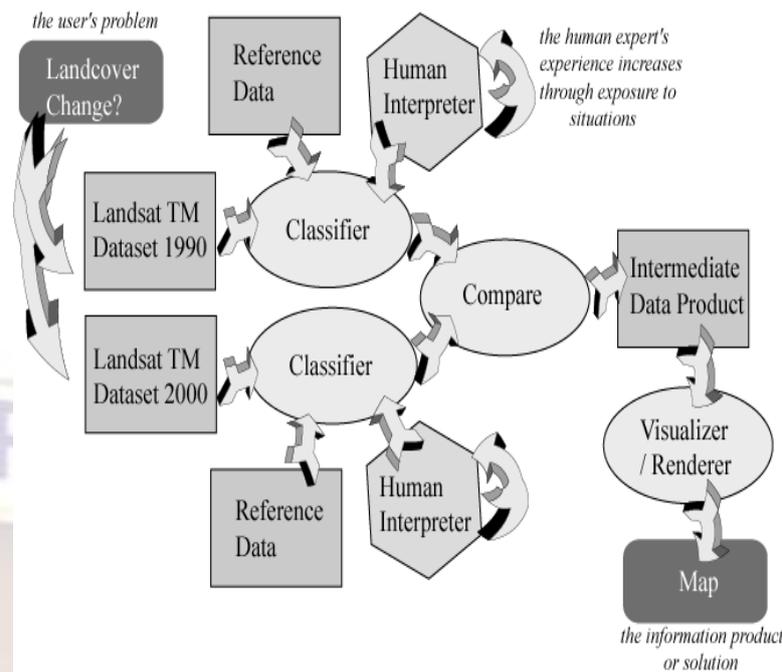
The representation vocabulary embodies the conceptualizations that the terms in the vocabulary are intended to capture. Relationships described between conceptual elements in this ontology allow

for the production of rules governing how these elements can be “connected” or “wired together” to solve a geographic problem. In our case these elements are methods, data and human experts, each with their own ontology. In the case of datasets, a *domain ontology* describes salient properties such as location, scale, date, format, etc., as currently captured in meta-data descriptions (see Figure 1). In the case of methods, a *domain ontology* describes the services a method provides in terms of a transformation from one semantic state to another. In the case of human experts the simplest representation is again a *domain ontology* that shows the contribution that a human can provide in terms of steering or configuring methods and data. However, it should also be possible to represent a two-way flow of knowledge as the human learns from situations and thereby expands the number of services they can provide (we leave this issue for future work). The synthesis of a specific information product is specified via a *task ontology* that must fuse together elements of the domain and application ontologies to attain its goal. An innovation of this framework is the dynamic construction of the solution network, analogous to the *application ontology*. In order for resources to be useful in solving a problem, their ontologies must also overlap. Ontology is a useful metaphor for describing the genesis of the information product. A body of knowledge described using the domain ontology is utilized in the initial phase of setting up the expert system. A task ontology is created at the conclusion of the automated process specifically defining the concepts that are available. An information product is derived from the use of data extracted from databases and knowledge from human experts in methods, as shown in figure 2.

By forming a higher level ontology which describes the relationships between each of these resources it is possible to describe appropriate interactions. As a simple example (figure 3), assume a user wishes to evaluate landuse/landcover change over a period of time. Classifying two LandsatTM images from 1990 and 2000 using reference data and expert knowledge, the user can compare the two resulting images to produce a map of the area(s) which have changed during that time. One interesting feature demonstrated in this example is the ability of human experts to gain experience through repeated exposure to similar situations. Even in this simple example a basic semantic structure is being constructed and a lineage of the data can be determined. Arguably an additional intermediate data product exists between the classifiers and the comparison; it has been removed for clarity.

### 1.2 Semantics

While the construction of the information product is important, a semantic layer sits above the operations and information (figure 4). The geospatial knowledge



**Figure 3 – A concrete example of the interaction of methods, data and human experts to produce an information product.**

obtained during the creation of the product is captured within this layer. The capture of this semantic information describes the transformations that the geospatial information undergoes, facilitating better understanding and providing a measure of repeatability of analysis, and improving communication in the hope of promoting best practice in bringing geospatial information to bear.

Egenhofer (2002) notes that the challenge remains of how best to make these semantics available to the user via a search interface. Pundt and Bishr, (2002) outline a process in which a user searches for data to solve a problem. This search methodology is also applicable for the methods and human experts to be used with the data. This solution fails when multiple sources are available and nothing is known of their content, structure and semantics. The use of pre-defined ontologies aids users by reducing the available search space. Ontological concepts relevant to a problem domain are supplied to the user allowing them to focus their query. A more advanced interface would take the user’s query in their own terms and map that to an underlying domain ontology (Bishr, 1998).

As previously noted, the meaning of geospatial information is constructed, shaped and changed by the interaction of people and systems. Subsequently the interaction of human experts, methods and data needs to be carefully planned. A product created as a result of these interactions is dependent on the ontology of the data and methods and the

epistemologies and ontologies of the human experts. In light of this, the knowledge framework outlined below focuses on each of the resources involved (data, methods and human experts) and the roles they play in the evolution of a new information product. In addition, the user's goal that produced the product, and any constraints placed on the process are recorded to capture aspects of intention and situation that also have an impact on meaning. This process and the impact of constraint based searches are discussed in more detail in the following section.

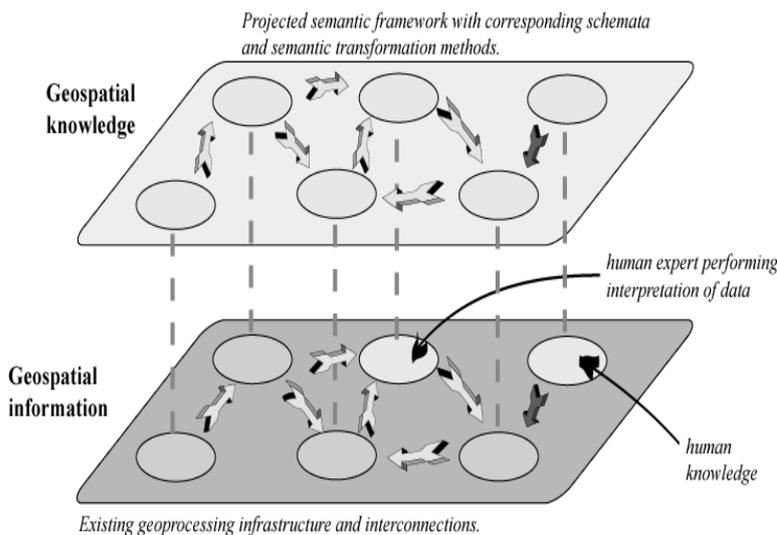


Figure 4 – Interaction of the semantic layer and

### III. Knowledge framework

The problem described in the introduction has been implemented as three components. The first, and the simplest, is the task of visualizing the network of interactions by which new information products are synthesized. The second, automating the construction of such a network for a user-defined task, is interdependent with the third, evaluating semantic change in a knowledge discovery environment, and both utilize functionality of the first. An examination of the abstract properties of data, methods and experts is followed by an explanation of these components and their inter-relationships.

#### 1.3 Formal representation of components and changes

This section explains how the abstract properties of data, methods and experts are represented, and then employed to track semantic changes as information products are produced utilizing tools described above. From the description in Section 2 it should be evident that such changes are a consequence of the arrangement of data, computational methods and expert interaction applied to data. At an abstract level above that of the data and methods used, we wish to represent some characteristics of these three sets of components in a

formal sense, so that we can describe the effects deriving from their interaction. One strong caveat here is that our semantic description (described below) does not claim to capture all senses of meaning attached to data, methods or people, and in fact as a community of researchers we are still learning about which facets of semantics are important and how they might be described. It is not currently possible to represent all aspects of meaning and knowledge within a computer, so we aim instead to provide descriptions that are rich enough to allow users to infer aspects of meaning that are important for specific tasks from the visualizations or reports that we can synthesize. In this sense our own descriptions of semantics play the role of a signifier—the focus is on conveying meaning to the reader rather than explicitly carrying intrinsic meaning *per-se*.

The formalization of semantics based on ontologies and operated on using a language capable of representing relations provides for powerful semantic modelling (Kuhn, 2002). The framework, rules, and facts used in the Solution Synthesis Engine (see below) function in this way. Relationships are established between each of the entities, by calculating their membership within a set of objects capable of synthesizing a solution. We extend the approach of Kuhn by allowing the user to narrow a search for a solution based on the specific semantic attributes of entities. Using the minimal spanning tree produced from the solution synthesis it is possible to retrace the steps of the process to calculate semantic change. As each fact is asserted it contains information about the rule that created it (the method) and the data and human experts that were identified as resources required. If we are able to describe the change to the data (in terms of abstract semantic properties) imbued by each of the processes through which it passes, then it is possible to represent the change between the start state and the finish state by differencing the two.

Although the focus of our description is on semantics, there are good reasons for including syntactic and schematic information about data and methods also, since methods generally are designed to work in limited circumstances, using and producing very specific data types (*pre-conditions* and *post-conditions*). Hence from a practical perspective it makes sense to represent and reason with these aspects in addition to semantics, since they will limit which methods can be connected together and dictate where additional conversion methods are required. Additional potentially useful properties arise when the computational and human infrastructure is distributed e.g. around a network. By encoding such properties we can extend our reasoning capabilities to address problems that arise

when resources must be moved from one node to another to solve a problem.

#### IV. Description of data

As mentioned in Section 2, datasets are described in general terms using a domain ontology drawn from generic metadata descriptions. Existing metadata descriptions hold a wealth of such practical information that can be readily associated with datasets; for example the FGDC (1998) defines a mix of semantic, syntactic and schematic metadata properties. These include basic semantics (abstract and purpose), syntactic (data model information, and projection), and schematic (creator, theme, temporal and spatial extents, uncertainty, quality and lineage). We explicitly represent and reason with a subset of these properties in the work described here and could easily expand to represent them all, or any other given metadata description that can be expressed symbolically. Formally, we represent the set of  $n$  properties of a dataset  $D$  as:  $D(p_1, p_2, \dots, p_n)$  (Gahegan, 1996).

#### V. Describing Methods

While standards for metadata descriptions are already mature and suit our purposes, complementary mark-up languages for methods are still in their infancy. It is straightforward to represent the signature of a method in terms of the format of data entering and leaving the method, and knowing that a method requires data to be in a certain format will cause the system to search for and insert conversion methods automatically where they are required. So, for example, if a coverage must be converted from raster format to vector format before it can be used as input to a surface flow accumulation method, then the system can insert appropriate data conversion methods into the evolving query tree to connect to appropriate data resources that would otherwise not be compatible. Similarly, if an image classification method requires data at a nominal scale of 1:100,000 or a pixel size of 30m, any data at finer scales might be generalized to meet this requirement prior to use. Although such descriptions have great practical benefit, they say nothing about the role the method plays or the transformation it imparts to the data; in short they do not enable any kind of semantic assessment to be made.

A useful approach to representing what GIS methods do, in a conceptual sense, centers on a typology (e.g. Albrecht's 20 universal GIS operators, 1994). Here, we extend this idea to address a number of different abstract properties of a dataset, in terms of how the method invoked changes these properties (Pascoe & Penny, 1995; Gahegan, 1996). In a general sense, the transformation performed by a method ( $M$ ) can be represented by pre-conditions and post-conditions, as is common practice with interface specification and design in software engineering.

Using the notation above, our semantic description takes the form:

$$M : D(p_1, p_2, \dots, p_n) \xrightarrow{\text{Operation}} D'(p_1', p_2', \dots, p_n')$$

, where *Operation* is a generic description of the role or function the method provides, drawn from a typology.

For example, a cartographic generalization method changes the scale at which a dataset is most applicable, a supervised classifier transforms an array of numbers into a set of categorical labels, an extrapolation method might produce a map for next year, based on maps of the past. Clearly, there are any number of key dimensions over which such changes might be represented; the above examples highlight *spatial scale*, *conceptual 'level'* (which at a basic syntactic level could be viewed simply as statistical scale) and *temporal applicability*, or simply *time*. Others come to light following just a cursory exploration of GIS functionality: change in spatial extents, e.g. windowing and buffering, change in uncertainty (very difficult in practice to quantify but easy to show in an abstract sense that there has been a change).

Again, we have chosen not to restrict ourselves to a specific set of properties, but rather to remain flexible in representing those that are important to specific application areas or communities. We note that as Web Services become more established in the GIS arena, such an enhanced description of methods will be a vital component in identifying potentially useful functionality.

#### VI. Describing People

Operations may require additional configuration or expertise in order to carry out their task. People use their expertise to interact with data and methods in many ways, such as gathering, creating and interpreting data, configuring methods and interpreting results. These activities are typically structured around well-defined tasks where the desired outcome is known, although as in the case of knowledge discovery, they may sometimes be more speculative in nature. In our work we have cast the various skills that experts possess in terms of their ability to help achieve some desired goal. This, in turn, can be re-expressed as their suitability to oversee the processing of some dataset by some method, either by configuring parameters, supplying judgment or even performing the task explicitly. For example, an image interpretation method may require identification of training examples that in turn necessitate local field knowledge; such knowledge can also be specified as a context of applicability using the time, space, scale and theme parameters that are also used to describe datasets. As such, a given expert may be able to play a number of roles

that are required by the operations described above, with each role described as:

$E : \xrightarrow{\text{Operation}} (p_1, p_2, \dots, p_n)$ , meaning that expert  $E$  can provide the necessary knowledge to perform *Operation* within the context of  $p_1, \dots, p_n$ . So to continue the example of image interpretation,  $p_1, \dots, p_n$  might represent (say) floristic mapping of Western Australia, at a scale of 1:100,000 in the present day.

At the less abstract schematic level, location parameters can also be used to express the need to move people to different locations in order to conduct an analysis, or to bring data and methods distributed throughout cyberspace to the physical location of a person.

Another possibility here, that we have not yet implemented, is to acknowledge that a person's ability to perform a task can increase as a result of experience. So it should be possible for a system to keep track of how much experience an expert has accrued by working in a specific context (described as  $p_1, \dots, p_n$ ). (In this case the expert expression would also require an experience or suitability score as described for constraint management described below in section 3.3). We could then represent a feedback from the analysis exercise to the user, modifying their experience score.

#### 1.4 Visualization of knowledge discovery

The visualization of the knowledge discovery process utilizes a self organising graph package (TouchGraph) written in Java. TouchGraph enables users to interactively construct an ontology utilizing concepts (visually represented as shapes) and relationships (represented as links between shapes). Each of the concepts and relationships can have associated descriptions that give more details for each of the entity types (data, methods, and people). A sample of the visualization environment is shown below in figure 5.

Touchgraph supports serialization allowing the development of the information product to be recorded and shared among collaborators. Serialization is the process of storing and converting an object into a form that can be readily reused or transported. For example, an ontology can be serialized and transported over the Internet. At the other end, deserialization reconstructs the object from the input stream. Information products described using this tool are stored as DAML+OIL objects so that the interrelationships between concepts can be described semantically. The DAML+OIL architecture was chosen as the goal of the DARPA Agent Markup Language component (DAML) is to capture term meanings, and thereby providing a Web ontology language. The Ontology Interchange Language (OIL) contains formal semantics and efficient reasoning support, epistemological rich modeling primitives,

and a standard proposal for syntactical exchange notations (<http://www.ontoknowledge.org/oil/>).

#### 1.5 Solution Synthesis Engine

The automated tool selection process or *solution synthesis* is more complex relying on domain ontologies of the methods, data and human experts (resources) that are usable to solve a problem. The task of automated tool selection can be divided into a number of phases. First is the user's specification of the problem, either using a list of ontological keywords or in their own terms which are mapped to an underlying ontology. Second ontologies of methods, data and human experts need to be processed to determine which resources overlap with the problem ontology. Third, a description of the user's problem and any associated constraints is parsed into an expert system to define rules that describe the problem. Finally networks of resources that satisfy the rules need to be selected and displayed.

Defining a complete set of characteristic attributes for real world entities (such as data, methods and human experts) is difficult due to problems selecting attributes that accurately describe the entity. Bishr's solution of using cognitive semantics to solve this problem, by referring to entities based on their function, is implemented in this framework. Methods utilize data or are utilized by human experts and are subject to conditions regarding their use such as data format, scale or a level of human knowledge. The rules describe the requirements of the methods ('if') and the output(s) of the methods ('then'). Data and human experts, specified by facts, are arguably more passive and the rules of methods are applied to or by them respectively. A set of properties governing how rules may use them are defined for data, (e.g. format, spatial, and temporal extents) and human experts (e.g. roles and abilities) using an XML schema and parsed into facts.

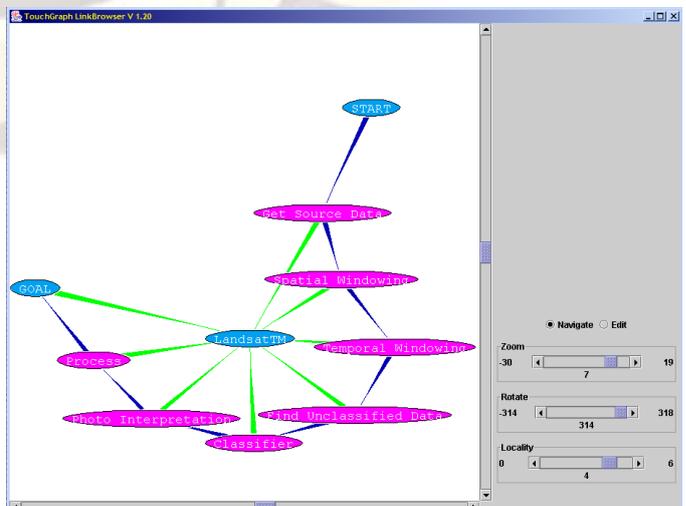


Figure 5 – A sample of the visualization environment.

The first stage of the solution synthesis is the user specification of the problem using concepts and keywords derived from a problem ontology. The problem ontology, derived from the methods, data and human expert ontologies, consist of concepts describing the intended uses of each of the resources. This limitation was introduced to ensure the framework had access to the necessary entities to solve a user's problem. A more advanced version of the problem specification is proposed which uses natural language parsing to allow the user to specify a problem. This query would then be mapped to the problem ontology allowing the user to use their own semantics instead of being governed by those of the system.

The second stage of the solution synthesis process parses the rules and facts describing relationships between data, methods, and human experts. The JESS rule, *compare* (Table 1), illustrates the interaction between the rule (or method) requirements and the facts (data and human experts). Sections of the rule not essential for illustrating its function have been removed. It is important to note that these rules do not perform the operations described rather they mimic the semantic change that would accompany such an operation. The future work section outlines the goal of running this system in tandem with a codeless programming environment to run the selected toolset automatically.

**Table 1 - JESS Sample code**

```
(1) defrule compare ;; compare two data sets
(2) (need-comparison_result $?)
(3) (datasource_a ?srcA)
(4) (datasource_b ?srcB)
(5) intersection_result <- (intersect ?srcA ?srcB)
(6) union_result <- (union ?srcA ?srcB)
(7) => ;; THEN
(8) (assert (comparison_result (inputA ?srcA)
(inputB ?srcB) (intersect ?intersection_result) (union
?union_result) ;; "perform" the operation
```

With all of the resource rules defined, the missing link is the problem to be solved using these rules. The problem ontology is parsed into JESS to create a set of facts. These facts form the "goal" rule which mirrors the user's problem specification. Each of the facts in the 'if' component of the goal rule are in the form 'need-method\_x'. The JESS engine now has the requisite components for tool selection.

Utilizing backward-chaining JESS searches for rules which satisfy the left hand side (LHS) of the rule. In the case of dependencies (rules preceded by "need-") JESS searches for rules that satisfy the "need-" request and runs them prior to running the rule generating the request. The compare rule (above) runs only when a previous rule requires a comparison\_result fact to be asserted in order for that rule to be completed.

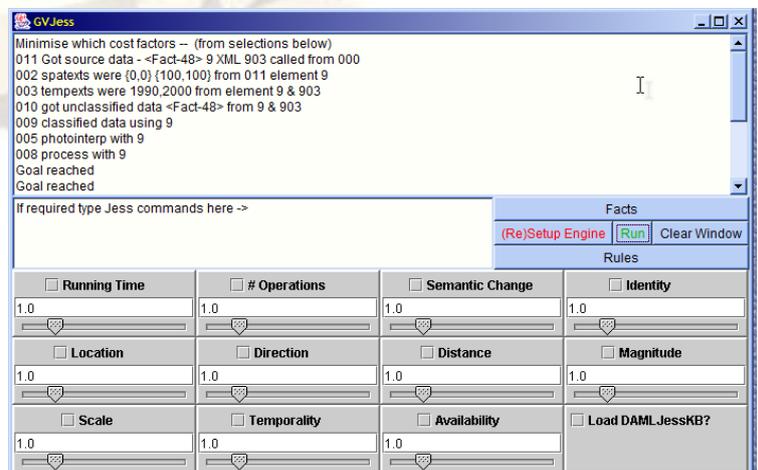
The compare rule (Table 1) has dependencies on rules that collect data sources (used for comparisons) and the rules that accomplish those comparisons (intersection and union). If each of these rules can be satisfied on the "if" side of the clause, then the results of the comparison rules are stored, together with the data sources that were used in the comparison and the products of the comparison. The results of the rule "firing" are stored in a list that will be used to form a minimal spanning tree for graphing [10-30].

As the engine runs, each of the rules "needed" are satisfied using backward chaining, the goal is fulfilled, and a network of resources is constructed. As each rule fires and populates the network a set of criteria is added to a JESS fact describing each of the user criteria that limits the network. Each of these criteria is used to create a minimal spanning tree of operations. User criteria are initially based upon the key spatial concepts of identity, location, direction, distance, magnitude, scale, time availability, operation time, and semantic change.

Users specify the initial constraints, via the user interface (figure 6) prior to the automated selection of tools. As an example, a satellite image is required for an interpretation task, but the only available data is 30 days old and data from the next orbit over the region will not be available for another 8 hours. Is it "better" to wait for that data to become available or is it more crucial to achieve a solution in a shorter time using potentially out of date data? It is possible that the user will request a set of limiting conditions that are too strict to permit a solution. In these cases all possible solutions will be displayed allowing the user to modify their constraints. The user specified constraints are used to prune the network of resources constructed (i.e. all possible solutions to the problem) to a minimal spanning tree which is the solution that satisfies all of the user's constraints.

## VII. Results

This section presents the results of the framework's



**Figure 6 – Interface showing constraint selection.**

solution synthesis and representation of semantic change. The results of the knowledge discovery visualization are implicit in this discussion as that component is used for the display of the minimal spanning tree.

A sample problem, finding a home location with a sunset view is used to demonstrate the solution synthesis. In order to solve this problem, raster (DEM) and vector (road network) data needs to be integrated. A raster overlay, using map algebra, followed by buffer operations is required to find suitable locations, from height, slope and aspect data. The raster data of potential sites needs to be converted to a vector layer to enable a buffering operation with vector road data. Finally a viewshed analysis is performed to determine how much of the landscape is visible from candidate sites.

The problem specification was simplified by hard-coding the user requirements into a set of facts loaded from an XML file. The user's problem specification was reduced to selecting pre-defined problems from a menu.

A user constraint of scale was set to ensure that data used by the methods in the framework was at a consistent scale and appropriate data layers were selected based on their metadata and format. With the user requirements parsed into JESS and a problem selected, the solution engine selected the methods, data and human experts required to solve the problem. The solution engine constructed a set of all possible combinations and then determined the shortest path by summing the weighted constraints specified by the user. Utilizing the abstract notation from above, with methods specifying change thus:

$$M_1 : D(p_1, p_2, \dots, p_n) \xrightarrow{\text{Operation}} D'(p_1', p_2', \dots, p_n')$$

, the user weights were included and summed for all modified data sets:

$$\sum D_1'(u_1 p_1', u_2 p_2', \dots, u_n p_n') \dots D_n'(u_1 p_1', u_2 p_2', \dots, u_n p_n')$$

. As a result of this process the solution set is pruned until only the optimal solution remains (based on user constraints).

### VIII. Future Work

The ultimate goal of this project is to integrate the problem solving environment with the codeless programming environment GEOVISTA Studio currently under development at Pennsylvania State University. The possibility of supplying data to the framework and determining the types of questions which could be answered with it is also an interesting problem.

A final goal is the use of natural language parsing of the user's problem specification.

### IX. Conclusions

This paper outlined a framework for representing, manipulating and reasoning with geographic semantics. The framework enables visualizing knowledge discovery, automating tool selection for user defined geographic problem

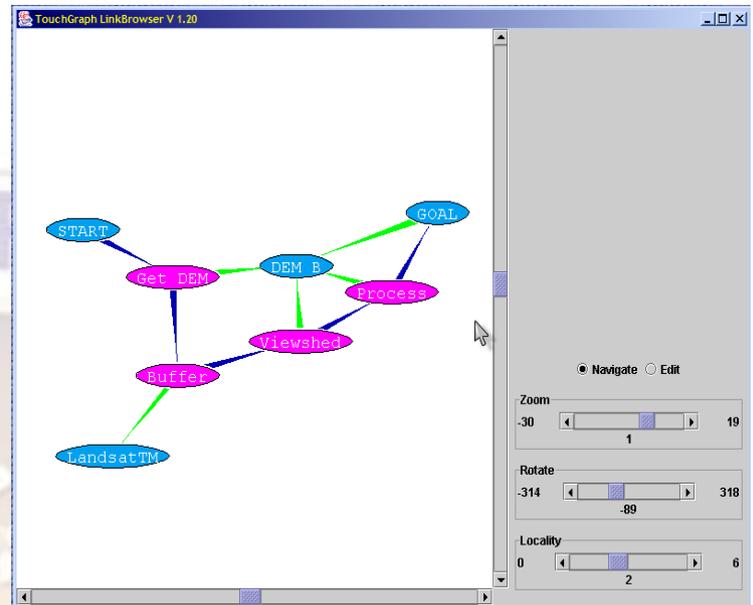


Figure 7 – Diagram showing optimal path derived from thinned network

solving, and evaluating semantic change in knowledge discovery environments. A minimal spanning tree representing the optimal (least cost) solution was extracted from this graph, and can be displayed in real-time. The semantic change(s) that result from the interaction of data, methods and people contained within the resulting tree represents the formation history of each new information product (such as a map or overlay) and can be stored, indexed and searched as required.

### X. Acknowledgements

Our thanks go to Sachin Oswal, who helped with the customization of the TouchGraph concept visualization tool used here. This work is partly funded by NSF grants: ITR (BCS)-0219025 and ITR Geosciences Network (GEON).

### References

- [1] Abel, D.J., Taylor, K., Ackland, R., and Hungerford, S. 1998, An Exploration of GIS Architectures for Internet Environments. *Computers, Environment and Urban Systems*. 22(1) pp 7 -23.
- [2] Albrecht, J., 1994. Universal elementary GIS tasks- beyond low-level commands. In Waugh T C and Healey R G (eds) *Sixth International Symposium on Spatial Data Handling* : 209-22.

- [3] Bishr, Y., 1997. *Semantic aspects of interoperable GIS*. Ph.D Dissertation Thesis, Enschede, The Netherlands, 154 pp.
- [4] Bishr, Y., 1998. Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, 12(4): 299-314.
- [5] Brodaric, B. and Gahegan, M., 2002. Distinguishing Instances and Evidence of Geographical Concepts for Geospatial Database Design. In: M.J. Egenhofer and D.M. Mark (Editors), GIScience 2002. *Lecture Notes in Computing Science 2478*. Springer-Verlag, pp. 22-37.
- [6] Chandrasekaran, B., Josephson, J.R. and Benjamins, V.R., 1997. Ontology of Tasks and Methods, AAAI Spring Symposium.
- [7] Egenhofer, M., 2002. Toward the semantic geospatial web, Tenth ACM International Symposium on Advances in Geographic Information Systems. ACM Press, New York, NY, USA, McLean, Virginia, USA, pp. 1-4.
- [8] Fabrikant, S.I. and Buttenfield, B.P., 2001. Formalizing Semantic Spaces for Information Access. *Annals of the Association of American Geographers*, 91(2): 263-280.
- [9] Federal Geographic Data Committee. FGDC-STD-001-1998. Content standard for digital geospatial metadata (revised June 1998). Federal Geographic Data Committee. Washington, D.C.
- [10] Fonseca, F.T., 2001. *Ontology-Driven Geographic Information Systems*. Doctor of Philosophy Thesis, The University of Maine, 131 pp.
- [11] Fonseca, F.T., Egenhofer, M.J., Jr., C.A.D. and Borges, K.A.V., 2000. Ontologies and knowledge sharing in urban GIS. *Computers, Environment and Urban Systems*, 24: 251-271.
- [12] Fonseca, F.T. and Egenhofer, M.J., 1999. Ontology-Driven Geographic Information Systems. In: C.B. Medeiros (Editor), 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO, pp. 7.
- [13] Gahegan, M., Takatsuka, M., Wheeler, M. and Hardisty, F., 2002. Introducing GeoVISTA Studio: an integrated suite of visualization and computational methods for exploration and knowledge construction in geography. *Computers, Environment and Urban Systems*, 26: 267-292.
- [14] Gahegan, M. N. (1999). Characterizing the semantic content of geographic data, models, and systems. In *Interoperating Geographic Information Systems* (Eds. Goodchild, M.F., Egenhofer, M. J. Fegeas, R. and Kottman, C. A.). Boston: Kluwer Academic Publishers, pp. 71-84.
- [15] Gahegan, M. N. (1996). Specifying the transformations within and between geographic data models. *Transactions in GIS*, Vol. 1, No. 2, pp. 137-152.
- [16] Guarino, N., 1997a. Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In: M.T. Paziienza (Editor), *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*. Springer Verlag, pp. 139-170.
- [17] Guarino, N., 1997b. Understanding , building and using ontologies. *International Journal of Human-Computer Studies*, 46: 293-310.
- [18] Hakimpour, F. and Timpf, S., 2002. A Step towards GeoData Integration using Formal Ontologies. In: M. Ruiz, M. Gould and J. Ramon (Editors), 5th AGILE Conference on Geographic Information Science. Universitat de les Illes Balears, Palma de Mallorca, Spain, pp. 5.
- [19] Honda, K. and Mizoguchi, F., 1995. Constraint-based approach for automatic spatial layout planning. 11<sup>th</sup> conference on Artificial Intelligence for Applications, Los Angeles, CA. p38.
- [20] Kokla, M. and Kavouras, M., 2002. Theories of Concepts in Resolving Semantic Heterogeneities, 5<sup>th</sup> AGILE Conference on Geographic Information Science, Palma, Spain, pp. 2.
- [21] Kuhn, W., 2002. Modeling the Semantics of Geographic Categories through Conceptual Integration. In: M.J. Egenhofer and D.M. Mark (Editors), GIScience 2002. *Lecture Notes in Computer Science*. Springer-Verlag.
- [22] MacEachren, A.M., in press. An evolving cognitive-semiotic approach to geographic visualization and knowledge construction. *Information Design Journal*.
- [23] Mark, D., Egenhofer, M., Hirtle, S. and Smith, B., 2002. Ontological Foundations for Geographic Information Science. UCGIS Emerging Resource Theme.
- [24] Pascoe R.T and Penny J.P. (1995) Constructing interfaces between (and within) geographical information systems. *International Journal of Geographical Information Systems*, 9:p275.
- [25] Pundt, H. and Bishr, Y., 2002. Domain ontologies for data sharing—an example from environmental monitoring using field GIS. *Computers & Geosciences*, 28: 95-102.
- [26] Smith, B. and Mark, D.M., 2001. Geographical categories: an ontological investigation. *International Journal of Geographical Information Science*, 15(7): 591-612.
- [27] Sotnykova, A., 2001. *Design and Implementation of Federation of Spatio-Temporal Databases: Methods and Tools*, Centre de Recherche Public - Henri Tudor and Laboratoire de Bases de Donnees Database Laboratory.
- [28] Sowa, J. F., 2000, *Knowledge Representation: Logical, Philosophical and Computational Foundations* (USA: Brooks/Cole).
- [29] Turner, M. and Fauconnier, G., 1998. Conceptual Integration Networks. *Cognitive Science*, 22(2): 133-187.
- [30] Visser, U., Stuckenschmidt, H., Schuster, G. and Vogege, T., 2002. Ontologies for geographic information processing. *Computers & Geosciences*, 28: 103-117.