BASIC STRUCTURAL MODELING PROJECT

Joseph J. Simpson

Mary J. Simpson

02-17-2014

CONCEPTUAL SOLUTION REPORT, FINAL

Version 1.00

Table of Contents

Note: A separate document contains the appendices: "Conceptual Solution Report Appendices"

I. Introduction

The basic structural modeling project (BSMP) Conceptual Solution Report provides example concepts, tools and procedures used to create solutions to structural modeling problems. This report examines solutions from the perspective of the selection of model types, the representation of the contextual structural relationship, and the techniques used to reflect applicable, appropriate mathematics.

The BSMP Problem Description Report discusses the general background associated with the problem space. This report provides specific features of Structural Modeling solutions. The discussion is organized around the properties associated with real-world, natural-language relationships and mathematical relations. A fundamental tool used in structural modeling is a well-formed transform between the real-world relationship and the mathematical relation that is used to model the situation of interest. Figure 1 depicts a notional representation of that transform. In Figure 1, 'Structural Integration Modeling' is the activity that supports the transformation process. Structural Integration Modeling, performed by system scientists and systems engineers, creates an abstract relation type as the primary output artifact of this process.

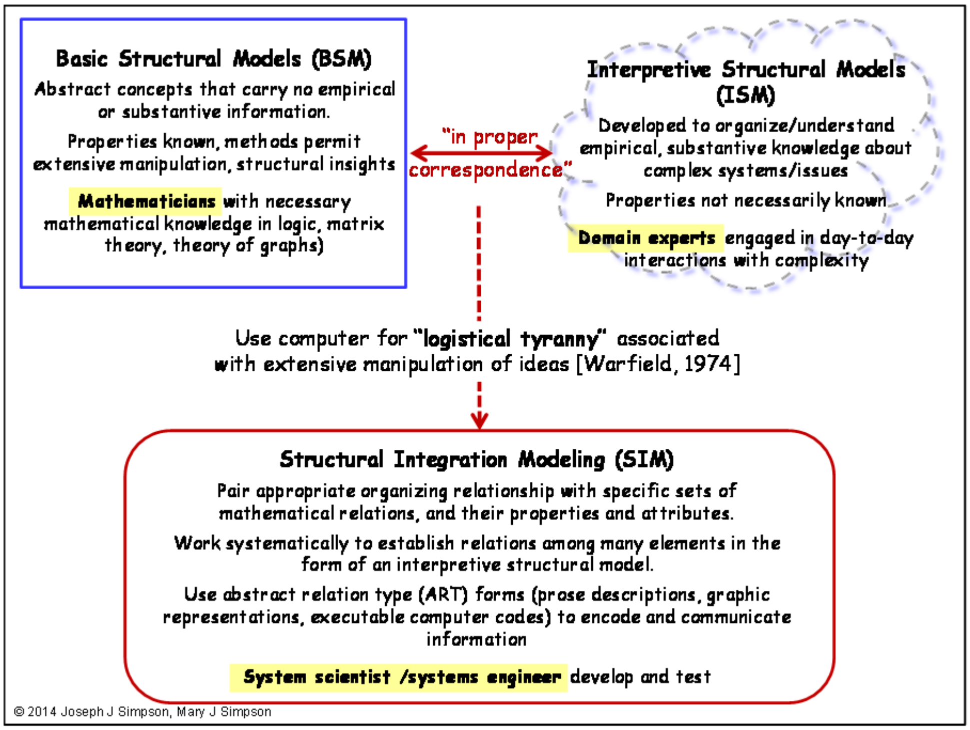

Figure 1. Structural Modeling

There are five application areas essential for structural modeling:

1. Mathematics,
2. The conversion of mathematics to computer code,
3. The generation of BSM (mathematics) computer programs,
4. Integration of BSM (mathematics) computer code into ISM (domain space) computer applications, and
5. Use of ISM techniques in Interactive Management, and other group problem solving processes.

The first two areas will use the computer-based SAGE math program from the University of Washington. Evaluated during the problem definition phase, SAGE was selected for use in this area. The third and fourth applications areas can be supported with a number of different types of computer-based programs. Python-based systems, as well as HTML5 and JavaScript-enabled web applications, will be used in these areas. Web2py has been chosen as the web framework to support work in the fifth area.

The main techniques used in this conceptual solution report are based on Warfield's basic structural modeling methods, and have been combined with other methods and processes to achieve an overall solution approach. The main operations taken from Warfield are:

- A binary matrix that contains Boolean operators;
- A zero (0) that represents either false, or unknown, in the binary matrix ; and
- Matrix multiplication that creates a reachability matrix.

The main techniques combined from other areas, or created by the authors, are:

- System and matrix reconfiguration (from automated N-Squared charts, Hitchins);
- System configuration values (from Hitchins);
- The two- or three-step process to place a matrix in 'lower triangular configuration' (Simpson and Simpson); and
- The process of effectively integrating the methods and techniques listed above (Simpson and Simpson).

Examples of each type of modeling activity will be presented for the 19-city problem initially discussed in the Problem Definition Report. Examples of each type of conceptual solution step will be demonstrated and discussed. Only those mathematical operations directly connected to the 19-city problem will be addressed.

A. Model Type Selection

Warfield, Sage, Steward and Hitchins developed distinct ISM techniques. Each of these techniques has a significantly different approach to structural modeling and basic structural modeling. Sage makes no distinction between the basic structural modeling approaches, and the ISM applications. Steward creates an ISM approach based on a natural language relationship that transitions into a mathematical relation that has a problematical 'real-world' interpretation. Hitchins uses the Automated N-Squared Chart approach as the foundation for his ISM approach. Each of these techniques provides interesting insights into the use of prose, graphics and mathematics for large-scale problem solving. One example: the system structural relationship and the 'matched' mathematical relation must have the same properties and attributes to correctly apply the selected technique. The focus for the conceptual solution report is on natural language relationships that have an asymmetric set of properties. For clarity, the "Directed Graph Model Exchange Isomorphisms" graphic from the BSMP Problem Definition Report is shown below, as Figure 2.

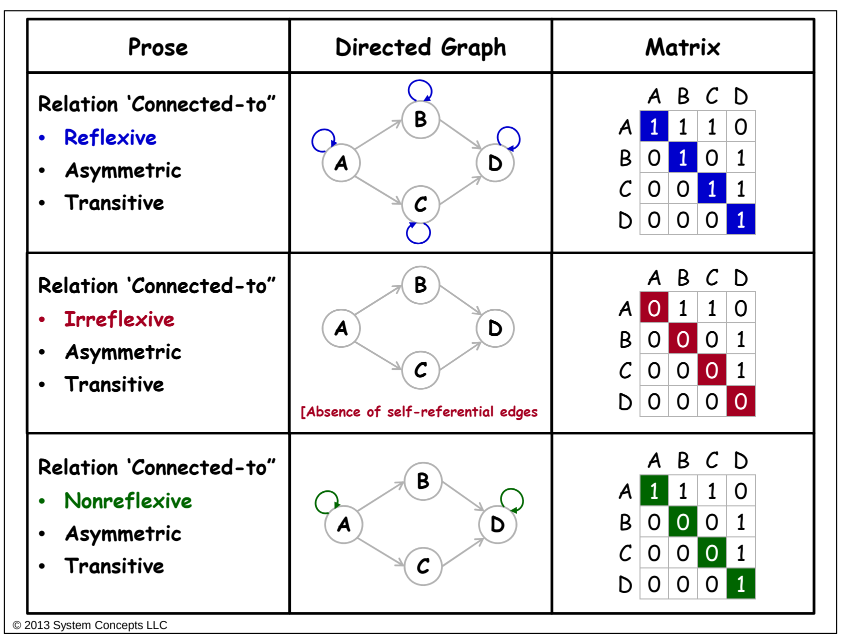

Figure 2. Directed Graph Model Exchange Isomorphisms

This limits the primary discussion to those ISM techniques developed by Warfield and Sage. The ISM types developed by Steward and Hitchins will be addressed only in terms of modifying these approaches to more closely align with the type of ISM methods and techniques developed by Warfield.

B. Contextual Structural Relationship Representation

The selection of the structural relationship is a primary step in structural modeling. The alignment between the attributes of the natural-language relationship, and the attributes of the selected mathematical relation, are fundamental to the effective implementation of a structural modeling process. An established, well-understood set of natural-language relationship attributes provides the basis upon which a coherent structural modeling process may be developed. Relationship attributes were only briefly identified and discussed by Warfield and Steward. Abstract relation types (ART), a technique developed by the authors, places the focus of system structural modeling on the role of the natural language relationship, and the mathematical relation. The ART approach places strong emphasis on the attributes associated with both the natural language relationship, and the mathematical relation. The process for transforming a natural language relationship into a mathematical relation, is also emphasized by the ART approach.

Additional contextual information is also valuable in the structural modeling process. In the example presented in this report, the additional contextual information is related to 'how many cities can occupy the same latitude.' If only one city can occupy a specific latitude, then a strict subordination structure is developed. However, if more than one city can occupy the same latitude, then a general subordination matrix is developed. The additional contextual information impacts the conceptual solution approach. A general subordination matrix can support the encoding of a strict subordination matrix, but a strict subordination matrix cannot encode a general subordination matrix. The example presented uses a strict subordination matrix.

C. Mathematical Tools, Techniques and Solution Approaches

Properly applied mathematical tools, aligned with natural language speech and understanding, provide a mechanism to analyze problems and systems of problems. The mathematics of Boolean algebra, Boolean reasoning, binary relations, directed graphs, and matrices are the primary tools used in basic structural modeling. The selection of the structural relationship attributes has a strong impact on the specific types of mathematics selected. A relationship that has asymmetric properties is addressed in one manner, while a relationship that has symmetrical properties is addressed in a different manner. Cases where a matched set of asymmetric properties can be combined into a well-understood, natural-language, symmetrical relationship, present a possibility for the use of a combination of techniques.

II. Conceptual Solution Context

The computational solutions presented in this document were prepared using the Sage mathematics tool and the Python programming language. The graphic presentations of the solution used Open Office spreadsheet and manual techniques. These computational tools are ubiquitous, open-source tools, available on the Internet to support individual or group solutions to large-scale problems. During the problem exploration phase, the unique features of the mathematics of structure were explored. The wide range of mathematical forms integrated into Warfield's mathematics of structure process, presents some unique challenges with software-based, computational engines. The heart of the challenges revolves around the mixed use of natural numbers (non-negative integers) and Boolean constants.

While natural numbers and Boolean constants contain both zero (0) and one (1), there is a computational integration issue associated with the augmented Boolean Algebra developed by Warfield. The authors addressed this computational issue by using Sage matrix mathematics for integers, and then mapping the output from the integer solution to a Boolean solution. This type of mapping includes a number of benefits:

- Uses a standard, open-source, mathematics tool to explore the mathematics of structure;
- Reduces the amount of custom code that must be written to support the process; and
- Provides support for non-standard Boolean operators.

The transforms and mappings used to support the mathematics of structure will be discussed in the first set of conceptual solutions, and used throughout the rest of the solution process.

III. Structural Modeling

Structural modeling provides a conceptual framework that supports the evaluation and analysis of a wide range of system types. If the system of interest is unstructured during the initial phases of evaluation, then structural modeling techniques may be applied to create a more structured system for evaluation. Interpretive structural models (ISM) are used to organize, structure and support the understanding of empirical, substantive knowledge about vague, ill-defined and/or complex systems, issues and situations. Basic structural modeling techniques are the focus of this conceptual solution report. ISM examples are only used to demonstrate features of basic structural modeling.

A key issue in conceptual solution development is the ability to test any given alternative solution in a manner that indicates the proposed solution process fulfills the program objectives. Since ISM is used to structure unknown and undefined systems, testing the proposed process will require the development of a unique approach. This unique approach consists of (1) establishing a known structure to the '19-cities' problem (described in the BSMP Problem Definition Report), and (2) randomizing the data from the known structure to create an unknown city structure order, that is used as the input data to test the candidate structuring processes.

IV. Example of Structural Modeling

Structural modeling depends on the interaction of basic structural modeling elements, the successful integration of basic structural modeling elements with computational methods, and the use of interpretive structural modeling data and decisions to create solutions to a given problem context.

A. Basic Structural Modeling Elements

The following basic structural modeling elements were addressed in the Problem Definition Report:

- Boolean algebra,
- Mathematical sets,
- Binary relations,
- Binary matrices,
- Binary matrix models,
- Directed graphs (Digraphs),
- Directed graph maps,
- Directed graph models,
- System structure,
- System complexity,
- Transitive embedding, and
- Graph cycles.

A more detailed discussion of these elements can be found in Appendix B of this report. Though the use of every element could be demonstrated using SAGE math worksheets and SAGE math notebooks, resource constraints permit a demonstration of only the most relevant subset of these elements.

The specific areas for discussion of the conceptual solution are Boolean algebra, mathematical sets, binary relations, binary matrices, and binary matrix models. Table 1, presented below, details the differences between decimal operations and the augmented Boolean operations developed by Warfield.

Table 1. Comparison of Decimal and Boolean Operations

Decimal Operations	Boolean Operations
0 + 0 = 0	0 + 0 = 0
0 + 1 = 1	0 + 1 = 1
1 + 0 = 1	1 + 0 = 1
1 + 1 = 2	1 + 1 = 1
0 x 0 = 0	0 x 0 = 0
0 x 1 = 0	0 x 1 = 0
1 x 0 = 0	1 x 0 = 0
1 x 1 = 1	1 x 1 = 1
0 < 1	0 < 1
1 > 0	1 > 0

B. Computation Approach, Sage Code

As shown in Table 1, the only difference between the decimal and augmented Boolean operations is the addition of 1 + 1. In decimal, 1 + 1 = 2, in Boolean 1 + 1 = 1. While the Sage math tool supports Boolean operations for individual variables, the support for Boolean matrix operations is missing. A decimal to Boolean mapping transform was developed for binary matrices with Boolean operators that are required by some of Warfield's structural modeling techniques and methods. This mapping transform is based on two specific observations. The first observation is the fact that both decimal and Boolean operations with zero (0) are the same. The second observation is the fact that Boolean algebra has only two constants, one (1) and zero (0). Given these observations, decimal matrix operations are used to create a decimal matrix outcome. Then, the decimal matrix is transformed by mapping all zeros (0) in the decimal matrix to zero (0) in the Boolean matrix, and mapping all numbers greater than zero (0) in the decimal matrix to one (1) in the Boolean matrix. Using this transform mapping in the Sage math tool greatly reduces the amount of effort required to create a unique separate mathematics program for Warfield's augmented Boolean algebra. The Sage code for the decimal to Boolean mapping transform is listed below.

```
def d2b(x):
if x > 0:
return 1
else:
return x
```

The function d2b (decimal to Boolean) is then mapped over each element in the decimal matrix using the apply_map(d2b) function. An example of this transform operation is presented below.

```
Matrix m = [[1, 0, 0, 0, 0, 0],
            [1, 1, 0, 0, 0, 0],
            [1, 0, 1, 0, 0, 0],
            [1, 1, 0, 1, 0, 0],
```

[1, 0, 1, 0, 1, 1],
[1, 0, 1, 0, 1,1]]

m1 = m * m

Matrix m1 = [[1, 0, 0, 0, 0, 0],
[2, 1, 0, 0, 0, 0],
[2, 0, 1, 0, 0, 0],
[3, 2, 0, 1, 0, 0],
[4, 0, 3, 0, 2, 2],
[4, 0, 3, 0, 2,2]]

m2 = m1.apply_map(d2b)

Matrix m2 = [[1, 0, 0, 0, 0, 0],
[1, 1, 0, 0, 0, 0],
[1, 0, 1, 0, 0, 0],
[1, 1, 0, 1, 0, 0],
[1, 0, 1, 0, 1, 1],
[1, 0, 1, 0, 1,1]]

As can be seen from the examples above, the d2b Sage function works well in transforming a decimal matrix into a Boolean matrix. This operation is a key operation in structural modeling, and must be supported by a verified, validated and well-tested, open-source math program like Sage math. This approach supports the encoding of natural language relationships into formal mathematical relations.

The primary objective of encoding expert information obtained in natural language, into formal mathematical constructs, is strongly supported by the concepts of natural language relationships and their attributes. These natural language relationships are encoded into mathematical relations in a manner that supports the objective of creating a system structure model of an unknown, or poorly defined, system. As indicated, the real-world relationships used as an example in this document are spatial relationships.

C. *Application to '19-Cities' Problem*

Consider a situation in which there are 19 cities. The objective is to determine which city (and/or cities) are north of a specific city. This type of problem can be encoded into a 19 by 19 matrix that represents a binary relation 'is north of.' Using the Warfield approach of assigning the value of one (1) to true, zero (0) to false and zero (0) to unknown, a matrix containing 361 zeros would be generated. The first task is to determine what each zero (0) represents. Since there is no empirical information about the cities, no city names, no city locations – it would seem that all of the zeros would indicate an unknown state. However, this is not the case. From the structure of the matrix, each entry on the diagonal of the matrix will ask the question "is City X north of City X?" This question can be answered by reviewing the properties of the 'is north of' binary relation. From a real-world perspective, letting a city occupy space north of itself is nonsensical, and the answer should be that the city is not north of itself. From an analysis of the binary relation properties, each zero (0) on the diagonal can be correctly determined to be a false value.

The transition from real-world relationships to mathematical models presents a few conceptual hazards and traps. Some design structure matrices (DSM) contain examples of one type of conceptual hazard. Often, DSM techniques use the natural language relationship 'precedes' as the system structural relationship. However, Steward – the originator of DSM – provides an analysis of the 'precedes' natural language relationship that assigns relational properties and mathematical operations to this relation. In this case, Steward indicated: "If $x \leq x$ we say that x 'preceded' x. Note by definition that each x precedes itself." 'Real-world' events do not precede themselves, though this is allowed in mathematical relational constructs of Steward's 1981 DSM work (*Systems Analysis and Management: Structure, Strategy and Design).*

Another area of caution in the transition from natural language relationships to mathematical relations is associated with the transitive property. Most people assume that a transitive relation must be among at least three distinct, individual objects. As Warfield pointed out, the mathematical relation does not require that the objects be distinct; you could have a transitive relation on one object. In fact, that is one outcome of the manner in which Steward assigned the 'precedes' relation. Steward allows event x to precede event x which can precede event x and so on. While these arrangements will work out mathematically, these concepts are problematical when translated to real-world events and systems.

The process of identifying, defining, and communicating the structure of a system is a very demanding task in which all available types of information must be used to reduce the uncertainty associated with the system of interest. The properties of the organizing system relations are a rich source of information, that seem to be ignored by most system scientists and engineers in their evaluation of an ill-defined and/or unknown system.

For example, using one of the binary spatial relations provided in the introductory text – 'is-two-miles-from' - city A can be two miles from city B, and city B can be two miles from city C. This binary relation, 'is-two-miles-from,' was assigned a nontransitive property. If this relation is nontransitive, then the spatial configuration of the cities has a very high degree of uncertainty. However, if the relation is transitive, then the spatial configuration of the cities is precisely known to be an equilateral triangle.

The choice of the correct relational property can greatly reduce uncertainty by increasing the information content of the communication. Abstract relation types were created and developed by the authors to provide a structured means of system description that focuses on the system-defining relation, and the relation characteristics and properties.

The structural modeling approach is based on three equivalent forms of information: prose, graphics and mathematics. As the list of basic structural modeling elements indicates, there is a heavy emphasis on mathematics and graphics with the prose elements absent from the list. The natural language prose attributes must be identified, documented, communicated, and correctly translated into mathematical relations. An expanded example using the "is-north-of" natural language relationship is developed and presented next.

A list of 19 cities are arranged in their known spatial order, and then randomly rearranged to create a set of cities that have an unknown spatial order. Basic structural modeling techniques are then used to create a spatial ordering of the cities based on the 'is-north-of' relationship. In this example, the information about each city is limited, and full empirical knowledge of a city's position with respect to all of the other cities is not known. For each city under evaluation, empirical information is available for the relationship between that city and five (5) other cities. Techniques associated with Boolean algebra, binary relations and binary matrices are applied in this example, mapping a natural language relationship to a mathematical relation. Some concepts that need to be highlighted include the communication of binary relations using Boolean algebra and binary matrices.

The 19 cities are arranged as a Cartesian product of the vector of cities in a 19 by19 matrix. The natural language relationship 'is-north-of' is asymmetric. The inverse relationship, 'is-south-of', is also asymmetric. If a city is not north-of another city then it is south-of the other city. Empirical information associated with a city's relative location can be either, 'is-north-of', or 'is-south-of'. Five cities are selected, at random, to start the empirical information gathering process. Each selected city will be evaluated relative to five other cities. If a city is north-of another city, then a 1 is placed in the matrix cell, and the background color is changed to green to indicate empirical information. If a city is south-of another city, a 0 is placed in the matrix cell, and the background color of the cell is changed to red to indicate empirical information.

The initial city information is shown in Figure 3 and the final city structure and information is shown in Figure 4. The process steps used to structure the system are provided in 'Appendix A – Structuring Process Example.' The general process starts with a set of objects with a known structuring relation, but an unknown structure. The general steps in the structuring process are to: gather partial empirical information, structure the acquired information, and perform an inference analysis on the current system state. These three general steps are performed again and again, until the system structure is in the desired state.

In the beginning phases of the structuring process, a group of empirical data may be collected. In this example, 25 data points were selected to start the analysis. After these 25 data points were processed to structure the system, an inference analysis was performed. After the first inference analysis, 25 more data points were collected and processed to structure the system. After the second set of data points were processed, a second inference analysis was performed. At this point in the process, there is sufficient structure to specifically select the data points that will return the most information. Individual data points were selected and analyzed for a small set of empirical data points. After this process an inference analysis was performed.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
10	1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
15	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3 – Initial City Information

	17	11	13	19	14	3	7	12	2	15	1	6	10	8	5	9	18	4	16
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
15	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0
9	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0
18	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0
16	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0

Figure 4 – Final System Structure and Information

The approach used in each case is a specific design decision. The process could be applied in a manner that selected a minimum of two data points and then performed the structuring and inference analysis on the system after each new data point is selected. The software program that supports this analysis activity should support the selection of different values for these factors. The inference process is well defined by Warfield, using the binary reachability matrix. The specific structuring and ordering process is generally addressed by Warfield, but specific, detailed rule sets must be developed to analyze the system structure state as it is represented in the current matrix configuration.

The best approach will be to build software that is flexible and modular for maximum use by SE practitioners. The software system should be developed to enable the capability to "plug-in" a specific type of system structuring relationship. In the cities example, the relationship is 'is-north-of' - with asymmetric, transitive, and irreflexive relation properties. Each plug-in should be organized around a specific set of relation attributes. Abstract relation types (ART), developed by the authors, provide a standard format for documenting this type of relationship information.

V. Solution Approaches

An interesting aspect of structural modeling is the fact that the system structure is unknown at the beginning of the structuring activity. Because the structure is unknown, it is difficult to test the proposed methods to determine if an acceptable structure was produced. In the case presented here, 19 cities with known spatial arrangements were selected. These cities were then processed to produce a random spatial arrangement. The basic structural modeling techniques were applied to the random spatial arrangement in an attempt to reproduce the known spatial order. In this first test case, the proper spatial order was reproduced.

In the solution steps presented in Appendix A, a number of design decisions were made, but not discussed. These design decisions will now be addressed to document their type and nature.

The first design decision is associated with gathering the initial empirical information. In this case, data on five cities was "gathered."

- The five cities were selected at random.
- The data points associated with each city were selected at random.
- The data was placed in the matrix to create the initial structural configuration.
- Then the initial unstructured data was structured using a set of ordering rules.

NOTE: *The final software must have a set of well-defined rules that are used to select and evaluate the system configuration in an automated or semi-automated fashion.*

Once the initial data was ordered, then the inference process was executed and the inferred data points were entered into the matrix. After the initial round of 25 data points were processed, another set of 25 data points were processed in the same manner.

After 50 data points had been selected at random to create a semi-populated matrix, the process was switched to *selecting the data points that would have the greatest inference potential*.

NOTE: *The rules for selecting data points with the greatest inference potential, need to be documented and discussed during the software development phase.*

Though there are other types of basic structural modeling approaches that could be used, the current approach will provide the basis for the techniques used in the software product.

This type of test-driven design is an effective approach to the design, development and deployment of software systems. The BSMP will use these types of processes whenever possible.

The coordinated application of logic and empirical data is used to great effect in these processes.

NOTE: *A key design criteria is selecting a natural language structuring relationship that is easy for people to understand and provide the necessary empirical data*.

Various combinations of data collection and logic application will be explored in future research activities.

Types of Truth

Three basic types of truth are used in the application of basic structural modeling techniques. These types are:

1. **Formal** truth (True or False), the truth of formal sentences depends only on the form of the logical connectives in the sentence.
2. **Factual** truth, the truth of factual sentences depends on the state of the real world.
3. **Value** truth, the truth of a value sentence is determined in different ways, including formal and factual methods.

In basic structural modeling techniques, formal and factual truths are combined in the same operation. This operation is associated with assigning a zero (0) for the value 'False' (*formal truth*), or assigning a zero (0) for the unknown 'state of the real world' (*factual truth*). The transition from basic structural modeling (BSM) to computer-augmented interpretive structural modeling (ISM) is not well defined, and there are a number of software programs that 'implement' ISM techniques. In the current work, abstract relation types will be used to structure and implement the code to support BSM methods and an integrated ISM approach.

Abstract Relation Types

Abstract relation types have three primary components:

1. A prose description of the ART form including formal and informal prose.
2. A graphic representation of the ART form including formal and informal graphics.
3. A mathematical representation of the ART form including equations and computer code.

A basic ART mathematical representation is composed of three computational and evaluation spaces:

1. The marking space, used to encode the system structure.

2. The value space used to encode the ART form value system.
3. The outcome space used to record the ART form outputs and outcomes.

In the current example, the marking space reflects the cities' names in the top row and first column. Then the relationship between each city pair is recorded in the cells of the marking space matrix. The empirical data and the inferred information are both entered into the marking space matrix in this case. The ART form value space and the ART form outcome space were not used in this example. Matrix transformation rules are used to provide the required matrix operations. While the value space and the outcome space could be used to support the implementation of these transformation rules, they were not used in the examples in this paper.

Structural Modeling

Structural modeling, as developed by Warfield, has two primary components: 1) basic structural modeling and 2) interpretive structural modeling. The application of structural modeling techniques by individuals over the past decades has shown an interesting variety of very creative and unique application approaches and techniques. This wide range of techniques indicates a need for a third component of structural modeling, structural integration modeling. The authors have developed structural integration modeling techniques over a period of years. These integration techniques are not the subject of the current work, but may be mentioned in the examples associated with basic structural modeling.

The classical methods of system engineering and science that are discussed here all have common graphical representations, and similar types of solution methods. The specific integration and application of these methods to solve a given type of system problem is the primary purpose of the ART form. Each ART form then becomes a pattern that may be adapted to explore the solution of certain problem types.

The modeling approach used in this project generally consists of the following steps:

1. Evaluate individual mathematical components using SAGE math,
2. Model the graphic structure in a spreadsheet first,
3. Evaluate the structural information in the spreadsheet using SAGE math,
4. Write a standalone HTML5, JavaScript web application to process a solution,
5. Write a SAGE math program to generate random systems and test solution approach,
6. Write a network enabled web application to implement a distributed solution process.

Steps one, two and three were accomplished in the problem definition report and this report. Step four was completed and used as a basis for the second video series. Step five is in work and developing. Step six has not been addressed.