The use of conceptual structures for handling the inference problem

Bhavani Thuraisingham

The MITRE Corporation, Burlington Road, Bedford, MA 01730

Abstract

The inference problem compromises database systems that are usually assumed to be secure. Here, users pose sets of queries and infer unauthorized information from the responses they obtain. The insidious nature of the inference problem is especially threatening to the Armed Forces. In this paper we describe the use of conceptual structures in handling the inference problem. Conceptual structures have many applications in artificial intelligence systems and natural-language processing. They have been used extensively for representing and reasoning about real-world knowledge. We develop multilevel conceptual structures and show how they could be used to represent multilevel applications. We also show how the reasoning strategies for multilevel conceptual structures could be used to detect security violations via inference during database design.

1. INTRODUCTION

The word inference is commonly used to mean "forming a conclusion from premises," where the conclusion is usually formed without expressed or prior approval. That is, without the knowledge or consent of anyone or any organization that controls or processes the premises or information from which the conclusion is formed. The resulting information that is formed can be innocuously or legitimately used, or it can be used for clandestine purposes with sinister overtones, threatening the security of the system. The term "information" is broadly defined to include raw data as well as data and collections of data which are transformed into knowledge.

It is possible for users of any database management system to draw inferences from the information they obtain from the databases. The inferred knowledge could depend on the data obtained from the database system only, or it could depend on some prior knowledge possessed by the user in addition to the data obtained from the database system. The inference process can be harmful if the inferred knowledge is something the user is not authorized to acquire. This has come to be known as the inference problem in database security [DENN86].

We are particularly interested in the inference problem which occurs in a multilevel operating environment. In such an environment, the users are cleared at different security levels and they access a multilevel database where the data is classified at different sensitivity levels. A multilevel secure database management system (MLS/DBMS) manages a multilevel database where its users cannot access data to which they are not authorized. However, providing a solution to the inference problem, where users issue multiple requests and consequently infer unauthorized knowledge, is beyond the capability of currently available MLS/DBMSs.
Recently some approaches have been proposed to handle the inference problem. Most of these approaches fall into two major groups. They are: (i) approaches that handle inferences during database design and (ii) approaches that handle inferences during query processing. The work reported in [MORG87, HINK88, SMIT90] focuses on handling inferences during database design where suggestions for database design tools are given. They expect that security constraints during database design are handled in such a way that security violations via inference cannot occur. The thesis for handling inferences during database design is also supported by others (see for example [LUNT89]).

In contrast, the work reported in [KEEF89, THUR87, THUR90] focuses on handling inferences during query processing. Their approach is to augment the query processor with a logic-based inference engine. The inference engine will attempt to prevent users from inferring unauthorized information. They argue that inferences can be most effectively handled and thus prevented during query processing because most users build their reservoir of knowledge from responses they receive by querying the database. It is from this reservoir of knowledge that they infer unauthorized information. It is their contention that no matter how securely the database has been designed, users could eventually violate security by inference because they are continuously updating their reservoir of knowledge as the world evolves. It is not feasible to have to redesign the database simultaneously.

While most approaches focus on handling the inference problem either during query processing or during database design, in [BUCZ89] an expert system tool is proposed which could be used by the system security officer off-line to detect and correct logical inferences.

Due to the complexity of the problem, we believe that a three-way approach to research is needed to combat the inference problem; one is to build inference controllers which act during transaction processing, the second is to build inference controllers for database design, and the third is to build inference controllers to act as advisors to the systems security officer (SSO). While our previous work has described how the problem may be handled during transaction processing, this paper describes some of our research on handling inferences during database design. In particular, the use of conceptual structures to handle the inference problem is described. Conceptual structures have many applications in artificial intelligence systems and natural language processing. They have been used extensively for representing real-world knowledge. Several reasoning tools for conceptual structures have also been developed. Some types of conceptual structures, such as Sowa's conceptual graphs [SOWA84], have been shown to be as powerful as first-order logic. It has also been argued that conceptual graphs can be naturally extended to cope with modality and time. The primary motivation for using conceptual structures is as follows:

(i) The use of conceptual structures for representing applications is consistent with the way humans view the world, whereas humans rarely view the world as a set of formulas of first-order logic.
(ii) It is more convenient to analyze the application manually when it is represented as a conceptual structure.
(iii) A representation of the application using conceptual structures can be used as a front-end subsystem to a logic programming system.
(iv) Reasoning strategies for representations based on conceptual structures are well developed.

1 A transaction is a program unit that is executed in its entirety or not executed at all. It can be regarded as a series of query and update requests. Inferences should be handled both during query and update processing.
To our knowledge, the use of conceptual structures to handle the inference problem was first proposed by Hinke [HINK88]. Hinke's work was on the use of graphs for representing the application. He showed how inferences may be detected by traversing alternate paths between two nodes in the graph. While this technique enables simple inferences to be detected via the transitivity property, it does not enable the detection of more complex inferences. Further work on the use of conceptual structures for inference handling was proposed by Smith [SMIT90]. Smith suggests extensions to the semantic data model discussed in [URBA89] to represent multilevel applications. He has developed a fairly substantial model to represent some complex situations. However, reasoning techniques have not yet been developed for this model. Smith states that eventually the representation should be translated into statements of a logic programming system. The techniques developed for such logic programming systems could then be used to detect inferences.

To successfully handle inferences, it is not only important to be able to represent the application semantics, but it is also essential that appropriate reasoning strategies be applied. Therefore we have investigated conceptual structures which are not only powerful representation schemes, but a complete set of reasoning strategies have also been developed for them. Our main focus is on the use of semantic nets. This is because semantic nets have been used extensively for a variety of data modelling, artificial intelligence, and natural language processing applications. However, since we are dealing with a multilevel environment, we have developed the concept of multilevel semantic nets and show how multilevel information can be captured by such a representation. We then make use of the complete set of reasoning strategies that have been developed for semantic nets to detect security violations via inference. The SSO could use the technique we have developed to analyze the application manually. On the other hand, automated tools based on our approach can also be developed which could be used by the SSO.

While semantic nets are powerful for representing and reasoning about a variety of applications, it has been shown that they do not have the capabilities of first-order logic. As a result, several extensions to semantic nets have been proposed. One particular extension which is theoretically complete is the conceptual graph of Sowa [SOWA84]. We have also briefly investigated the use of conceptual graphs in handling inferences. In particular, we have investigated issues on developing multilevel conceptual graphs and shown how the reasoning techniques such as restriction, joining, and simplifying could be applied to handle our problem.

The organization of this paper is as follows: In section 2, we discuss the use of semantic nets in detail. Our investigation on the use of conceptual graphs is described in section 3. The paper is concluded in section 4. We assume that the reader is familiar with concepts in multilevel database management systems. A good discussion of this topic is given in [AFSB83]. We assume that the reader is also familiar with concepts in conceptual structures, in general, and semantic nets, in particular. An excellent introduction to these topics is given in [RICH89, RING88].

---

2 Hinke [HINKE89] also suggests the possible use of semantic nets. However, neither multilevel semantic nets nor reasoning strategies for handling the inference problem are addressed.

4 The term "conceptual graph" has been used often in the literature to mean different things. We assume the definition of conceptual graph given by Sowa [SOWA84].
2. SEMANTIC NETS AND THE INFERENCE PROBLEM

2.1 OVERVIEW

Quillian was the first to use semantic networks in problem solving [QUIL66]. In his work, Quillian's aim was to find a representation format for storing words so that a human-like interpretation of the words is possible. Basically, a semantic net can be regarded as a collection of nodes connected via links. The nodes represent entities, events, and concepts and the links represent relationships between the nodes. Since Quillian's work on semantic networks, numerous variations of the semantic networks have been proposed. These include conceptual dependencies of Schank [SCHA72], Brachman's KL-ONE [BRAC85], conceptual graphs of Sowa [SOWA84], and Wood's work on the foundations [WOOD75]. Such representational schemes are called conceptual structures. What is useful about these conceptual structures is that they can represent and reason about the real-world like humans. Therefore, they are increasingly used for a variety of artificial intelligence and natural language processing applications.

We are interested in conceptual structures because they can be used to represent a multilevel application. Such a representation can be used by the SSO to manually analyze the application to ensure that users cannot draw unauthorized inferences. On the other hand, we can build a system which processes the knowledge using strategies that have been developed for conceptual structures and performs automatic security analysis of the applications. Our initial focus is on semantic nets, due to their simplicity and human-like reasoning power.

Since the application under consideration is multilevel, standard semantic nets cannot represent it. In other words, extensions to these semantic nets are needed to support multilevel security. In section 2.2, we introduce the notion of multilevel semantic nets. Reasoning in a multilevel semantic net is the subject of section 2.3. Enforcing security constraints, which are rules which assign security levels to the various concepts and relationships, is discussed in section 2.4. Universal and existential conditionals are treated in section 2.5. Interpretations of multilevel semantic nets are discussed in section 2.6. A refutation procedure for multilevel semantic nets is given in section 2.7.

We consider a semantic net to be a collection of nodes connected via links. The nodes represent concepts, entities, etc. and the links represent relationships between them. Our treatment of semantic nets is influenced by the work reported in [RICH89].

2.2 MULTILEVEL SEMANTIC NETS

A multilevel semantic net (MSN) is a semantic net with nodes and links classified at different security levels. Figure 1 shows some simple multilevel semantic nets. We assume that there are only two security levels, Unclassified and Secret. Note that the darkened shapes and lines are assumed to be Secret. Our discussion can easily be extended to support multiple security levels.

Consider figure 1a. It states that all ships carry some weapons. In figure 1a, both the node and link are Unclassified. That is, the fact that ships carry weapons can be seen by all. In figure 1b, the concepts SHIPS and WEAPONS can be seen by all. However, the fact that ships carry weapons cannot be seen by Unclassified users as the relationship is classified Secret. In figure 1c, the Unclassified users know that something carries weapons. But they do not know that ships carry weapons. In figure 1d, the Unclassified users know that ships

---

4 In section 2.5, we discuss universal and existential conditionals in more detail.
carry something. They do not know what is carried by the ships. In figure 1e, the Unclassified users know that something is carried by something else. They do not know anything about ships and weapons. In figure 1f, Unclassified users know only about ships. In figure 1g, Unclassified users know only about weapons. In figure 1h, nothing is visible to the Unclassified users.

Figure 1. Multilevel Semantic Nets
It needs to be determined whether all the links described in figure 1 should be permitted. For example, it may make sense to classify a link at a level which dominates the levels of the nodes associated with the link. That is, the level of the CARRY relationship must dominate the levels of SHIPS and WEAPONS.

Figure 2. Complex Multilevel Semantic Net

Figure 2 shows a more elaborate multilevel semantic net. The Unclassified interpretation of this figure is as follows: CHAMPION carries passengers. Its captain is Smith, who has 20 years' experience. The ship is located in the Mediterranean Sea on 16 June 1990. Its destination is Greece. The Secret interpretation is as follows: CHAMPION carries SPARK which is an explosive. Its captain is Smith, who has battle management experience. The ship is located in the Mediterranean Sea on 16 June 1990. Its destination is Libya.

We can see that certain information is polyinstantiated. Note that polyinstantiation occurs when users at different security levels have different views of the same "thing" in the real world. By "thing" we mean concept, entity, event, or any relationship. Cover stories result in polyinstantiation. Figure 2 illustrates how cover stories may be represented.

---
5 Throughout this paper, we assume that CHAMPION, OHIO, and FLORIDA are ships and SPARK and STAR are weapons.
6 In this example, we assume that Libya, SPARK, Explosive, and Battle Management are also Secret. We assumed this only to show that concepts as well as links could be classified. In reality, some of this information could be public knowledge. What is important is to classify the non-public knowledge appropriately.
The links we have defined in the semantic nets, considered so far, illustrate special relationships. In addition to such links, a semantic net also has two standard links. They are: ISA links and AKO links. An ISA link is used to specify that a particular individual belongs to a particular group. Figure 3a shows an ISA link where CHAMPION is defined to be a particular ship. An AKO link defines a subset of a collection of individuals. Figure 3b defines the collection of ships to be a subset of a collection of water vehicles. Note that the AKO relationship is assigned to be Secret.

It does not make sense to classify CHAMPION at the Secret level and SHIP at the Unclassified level. This is because CHAMPION is an instantiation of SHIP. By classifying SHIP at the Secret level we are implicitly assuming that any ship must be classified at least at the Secret level. It should also be noted that it does not make sense to classify SHIP at the Unclassified level and WATER VEHICLE at the Secret level. This is because classifying SHIP at the Unclassified level implicitly assumes that a ship should be classified at least at the Unclassified level. Classifying WATER VEHICLE at the Secret level implicitly assumes that any water vehicle (including a ship) should be classified at least at the Secret level. This results in a conflict. Therefore, we enforce the following rules for consistency:

Rule A1: If X ISA Y, then Level(X) ≥ Level(Y).

Rule A2: If X AKO Y, then Level(X) ≥ Level(Y).

2.3 REASONING WITH MULTILEVEL SEMANTIC NETS

In order to reason with multilevel semantics nets, we first need rules which can be used to reason. In this section, we describe some rules that can be used for reasoning purposes.

2.3.1 Implicit Information

Most real-world applications deal with very large quantities of information. Therefore, capturing all of the information in a semantic net would make the net extremely complex. What we need is a minimal semantic net with a powerful set of reasoning strategies so that other information can be deduced. The information that is deduced is implicit information. For a multilevel application it should be ensured that the level of the implicit information that can be deduced by a user at level L should be dominated by L.
Some rules for deducing implicit information are the following:

Rule A3: If $X$ AKO $Y$ and $Y$ AKO $Z$ then $X$ AKO $Z$. The level of the AKO link from $X$ to $Z$ is the least upper bound of the levels of AKO links from $X$ to $Y$ and $Y$ to $Z$. 

Figure 4. Sample Rules
Figure 4a illustrates an example. The semantic net has SHIP AKO WATER-VEHICLE and WATER-VEHICLE AKO VEHICLE, and the AKO link from SHIP to WATER-VEHICLE is Secret. Then, at the Secret level, one can conclude that SHIP AKO VEHICLE.

Rule A4: If X AKO Y and Y has relationship R with Z, then X has relationship R with Z. The level of the relationship R that X has with Z is the least upper bound of the levels of the AKO link from X to Y and the relationship R from Y to Z.

Figure 4b illustrates an example. The semantic net has SHIP AKO WATER-VEHICLE and WATER-VEHICLE has captain PERSON. Then SHIP has captain PERSON.

Rule A5: If X ISA Y and Y AKO Z, then X ISA Z. The level of the ISA link from X to Z is the least upper bound of the levels of the AKO link from Y to Z and the ISA link from X to Y.

Figure 4c illustrates an example. CHAMPION ISA SHIP. This link is Secret. SHIP AKO WATER-VEHICLE. Therefore, there is a Secret ISA link from CHAMPION to WATER-VEHICLE.

Rule A6: If X ISA Y and Y has relationship R with Z, then X has relationship R with Z. The level of the relationship R that X has with Z is the least upper bound of the levels of the AKO link from X to Y and the relationship R from Y to Z.

Figure 4d illustrates an example. The semantic net has CHAMPION ISA SHIP. SHIP has captain PERSON. Therefore, CHAMPION has captain PERSON.

Rule A7: If X ISA Y and Z has relationship R with X, then Z has relationship R with Y. The level of the relationship R that Z has with Y is the least upper bound of the levels of the ISA link from X to Y and the relationship R from Z to X.

Figure 4e illustrates an example. The semantic net has Libya ISA COUNTRY. The ship CHAMPION's destination is Libya. Therefore, the destination of CHAMPION is a country.

2.3.2 Conditional Statements and Auxiliary Nets

Conditional statements are of the form: A if B1 and B2 and B3 and ... Bn where B1, B2, ... Bn are the antecedents and A is the consequent.

Note that conditional statements are clauses of a logic program [KOWA79]. Such a conditional statement can be represented by auxiliary semantic sets. We illustrate the essential points with an example.

Consider the following conditional statement:

CHAMPION's destination is Libya, if it is located in the Mediterranean, and it carries SPARK, which is an explosive.

The conditional statement is represented by the auxiliary net shown in figure 5a. That is, the conditions are represented by dotted lines, and the conclusion is represented by solid lines. The transfer rule is applied in order to process conditional statements. The following is the transfer rule:

7 The dotted lines are darkened if they represent Secret relationships.
Rule A8 (Transfer Rule): If all the dotted lines in the auxiliary net are shown as solid lines in a main multilevel semantic net, and the level of each solid line of the main net dominates the level of the corresponding dotted line in the auxiliary net, then the solid line in the auxiliary net is drawn as a solid line in the main net. The security level of the line drawn is the least upper bound of the levels of all the lines in the auxiliary net and the levels of all the corresponding solid lines already in the main net.

Figure 5b shows that the dotted lines in the auxiliary net occur as solid lines in the multilevel semantic net. Figure 5c shows that the solid line in the auxiliary net is added to the multilevel semantic net at the appropriate security level.
2.4 ENFORCING SECURITY CONSTRAINTS

Security constraints are rules which assign security levels to the data. We represent security constraints by what we call "constraint nets." A constraint net is a semantic net or an auxiliary semantic net which specifies only constraints. However, while semantic nets are used in general to represent application information, constraint semantic nets are used to represent the security constraints so that any security violation in the application can be detected. Similarly, while auxiliary semantic nets are used to derive implicit information, security constraints which are represented as auxiliary semantic nets are used to detect security violations. Therefore, we differentiate between ordinary auxiliary nets and constraint auxiliary nets. 8 Figure 6a classifies the fact that CHAMPION carries anything at the Secret level. Figure 6b shows a constraint which classifies the destination country of CHAMPION at the Secret level, if CHAMPION is located in the Mediterranean and it carries SPARK.

Security violations occur (either directly or indirectly) if the constraint net contradicts the multilevel semantic net which represents the application (either directly or indirectly). The semantic net of figure 7a violates both constraints of figure 6 directly. That is, in figure 7a, the fact that CHAMPION carries something is not Secret. This directly violates the constraint of figure 6a. Also, in figure 7a, CHAMPION is located in the Mediterranean and it carries SPARK. However, its destination country is Unclassified. This directly violates the constraint of figure 6b.

---

8 One can think of ordinary auxiliary nets as integrity constraints which are treated as derivation rules. That is, they are used to deduce new information from existing information. Constraint nets are used neither as derivation rules nor as integrity rules. Instead, they are used to determine whether the semantic net, which includes the explicit as well as the derived information, has any security flaws. For a discussion on integrity and derivation rules, we refer to [GALL78].
Constraints can also be violated indirectly. This occurs when the implicit information that can be inferred contradicts the security constraints. Figure 7b shows how the security constraint of figure 6b is violated indirectly. Here, CHAMPION carries SPARK and it is located in the Mediterranean. Its destination country Libya is Unclassified. Since Libya is a country, by rule A7, the destination Libya of CHAMPION must be Secret. Therefore, the constraint of figure 6b is violated indirectly.

\[ (6a) \]

```
CHAMPION \rightarrow \text{CARRIES} \rightarrow \text{SPARK}
```

\[ (6b) \]

```
\begin{array}{c}
\text{COUNTRY} \\
\downarrow \\
\text{DESTINATION} \\
\downarrow \\
\text{LOCATION} \\
\downarrow \\
\text{CARRIES} \\
\downarrow \\
\text{SPARK}
\end{array}
```

Figure 6. Representing Security Constraints

Another example of indirect constraint violation is shown in figure 8. Consider the constraint of figure 8a. This constraint classifies at the Secret level the fact that CHAMPION carries explosive weapons. Consider the semantic net of figure 8b. In this net, SHIP carries explosive weapons. This means that all ships carry explosive weapons. It is also specified that CHAMPION ISA SHIP. Then, applying rule A4, one can infer that CHAMPION carries explosive weapons. But the "CARRIES" relationship between CHAMPION and EXPLOSIVE WEAPONS is Unclassified. This violates the constraint of figure 8a. Note that, in this example, implicit information that can be inferred from the information in the multilevel semantic net violates the security constraints.

2.5 UNIVERSAL AND EXISTENTIAL CONDITIONALS

Consider the security constraint illustrated in figure 6b. This constraint specifies that whenever CHAMPION is in the Mediterranean and it carries SPARK, its destination country is Secret. Suppose we want the same constraint for any ship. This means that for every ship in the list of U.S. Navy ships, we must have an explicit constraint. It is not practical to have a constraint for each ship. What we need is a way to specify that for every ship $X$, if $X$ is in the Mediterranean and it carries SPARK, then its destination is Secret. One way to specify this is to use the type of a ship, which is SHIP, instead of the name of a particular ship. Suppose,
instead, we want to classify the fact that whenever a ship carries one or more weapons, then its destination must be kept Secret. In this case, if we use the type of weapons, which is WEAPONS, then there is ambiguity as to whether we mean all weapons or at least one weapon. In order to unambiguously represent information such as "all ships" and "some weapons," we need to introduce the notion of universal and existential conditionals.

![Diagram](image)

Figure 7. Security Constraint Violation

A conditional is an auxiliary semantic net. A universal conditional is a conditional with only universally quantified variables. An existential conditional is one with only existentially quantified variables. A mixed conditional is one with both types of variables. Universal and existential conditionals are powerful, as they can handle vast quantities of information. In order to treat universal and existential conditionals, we need to handle variables. The constraint semantic net for specifying a constraint using universal and existential conditionals is shown in figure 9a. By "SHIP: ALL X" and "COUNTRY: SOME Y" we mean "all ships X" and "some country Y" respectively. This constraint states that for all X, if X is in the Mediterranean and it carries SPARK, then its destination country is Secret.
Figure 8. Security Constraint Violation

Figure 9. Universal and Existential Conditionals
Conditionals are useful for not only specifying the constraints, but also for stating information in any auxiliary net. For example, suppose any ship which carries an explosive weapon located in the Mediterranean is destined for Libya. In addition, we assume that the destination is Secret. The auxiliary net for specifying the universal conditional is shown in figure 9b. Note that this net uses universal as well as existential conditionals. This net states that if for all ships X, if X is in the Mediterranean and it carries some weapon which is an explosive, then its destination is Libya. Furthermore, the destination is Secret.

The proof theory for variables is pattern matching and binding. We formally define pattern matching and binding as follows:

**Rule A9 (Pattern Matching Rule):**

Let A and B be subnets of two multilevel semantic nets. Note that a subnet is any subset (nodes and links) of a semantic net. Subnets are also called vectors. A matches B if the following are satisfied:

(i) The links are labelled the same and have the same security levels.

(ii) If a node in A is labelled with a constant, then the corresponding node in B is labelled with the same term,\(^9\) and the two nodes have the same security levels.

(iii) If two nodes in A are labelled with the same variable, then the corresponding two nodes in B are labelled with the same constant. Further, the security levels of the corresponding nodes in the two nets are the same.

Figure 10 shows two vectors, A and B, where A matches B.

**Rule A10 (Binding Rule):** If a vector A matches with vector B, then for any variable x in A, the corresponding node in B which matches with x is called the binding of x.

In figure 10, the binding for the variable x in A is the constant CHAMPION in B.

We now illustrate the pattern matching and binding process with an example. Consider the auxiliary net of figure 11a. Also, consider the semantic net shown in figure 11b. We can bind CHAMPION and OHIO with the variable X and SPARK, and STAR with the variable Y. The constant Explosive and the links LOCATION, CARRIES, and TYPE are matched in both nets. Therefore, applying the transfer rule, we can add Secret links from CHAMPION to Libya and OHIO to Libya. This is shown in figure 11c.

---

\(^9\) A term is a variable or a constant.
2.6 SEMANTICS

In the previous sections, we discussed representational issues as well as proof theoretic issues for multilevel semantic nets. In this section, we discuss the semantics of multilevel semantic nets. We first define the notion of a multilevel world and then discuss interpretations. Truth and satisfaction in interpretations are also discussed.

2.6.1 Multilevel Worlds

A multilevel world is the portion of the real world under consideration. It is multilevel because we assume that not all entities, concepts, events, relationships, etc. are assigned the same security level. The following is a description of an example multilevel world:

There is a ship called CHAMPION. Its captain is Smith. It is located in the Mediterranean Sea on 16 June 1990. At the Secret level, it carries SPARK which is an explosive, its destination is Libya, and its captain has battle management experience. At the Unclassified level, it carries passengers, its destination is Greece, and its captain has 20 years' experience.

Note that a multilevel world can be decomposed into single-level worlds. That is, there is a world corresponding to each security level. Information in the Unclassified world is accessible to the Secret world. But the converse is not true. Also, it is possible for conflicting information to be present in different worlds. For example, in the Secret world the destination of CHAMPION is Libya while in the Unclassified world it is Greece. Therefore, in the Secret world there is conflicting information about the destination. Whenever there is conflicting information present about some event or concept at security level \( L \), we take the information associated with that event or concept at level \( L^* \), where \( L^* \) is the highest security level dominated by \( L \) and no information about that event or concept is specified at any security level \( L^+ \) where \( L^* < L^+ \leq L \).
The set of objects associated with a world is a domain of discourse of that world. If level L+ dominates the level L*, we assume that D(L*) is a subset of D(L+) where for each L, we assume that D(L) is the domain of discourse of the world associated with L.

2.6.2 Interpretations

We first associate (or relate) multilevel semantic nets with multilevel worlds. The rules for relating the world and the net are as follows:

(i) For each constant at level L in the net, there is an object associated with the domain of discourse of the world at level L. The object is called the assignment of the constant.

(ii) For each predicate at level L in the net, there is a relationship in the world at level L. The relationship is called the assignment of the predicate.

The multilevel world, together with its denotations, is called the interpretation of the corresponding semantic net. This view of interpretation can be regarded as the proof-theoretic view. That is, the world is an interpretation of a theory which is the logic of semantic nets. Note that the multilevel semantic net of figure 2 relates to the multilevel world described in the example of subsection 2.6.1. That is, the multilevel world described in subsection 2.6.1 is an interpretation of the multilevel semantic net of figure 2.

Once we have defined interpretations, we need to define the notion of truth in an interpretation. We define truth of ground vectors (which are vectors with no variables), ground conditionals, universal conditionals, and existential conditionals.

Ground Vectors: Let A and B be two nodes of a net with relationship R between them. Then the ground vector which consists of the two nodes and link is true in an interpretation I with respect to security level L, if the security levels of A, B, and R are all dominated by L, and if A*, B*, and R* are the assignments of A, B, and R, respectively, then A* is in relationship R* with B* in the world at level L.

Example: Consider the ground vector of figure 1(a). Let us assume that ships in the real world is the assignment of the node SHIPS and that weapons in the real-world is the assignment of the node WEAPONS. Let us also assume that the act of carrying is the assignment of the predicate CARRY. Suppose, in the Unclassified as well as in the Secret world, it is true that each ship carries a weapon. Then the ground vector of figure 1a is true with respect to Unclassified and Secret levels.

Ground Conditionals: A ground conditional is false in interpretation I with respect to security level L, if all of its antecedents are true in I with respect to L, and its consequent is false in I with respect to L. Otherwise, the ground conditional is true in I with respect to L.

Example: Consider the ground conditional of figure 5a. Let the assignments of CHAMPION, LOCATION, Libya, CARRIES, Mediterranean Sea, Explosive, SPARK, and TYPE be a particular ship, its location, the country Libya, the act of carrying, the Mediterranean Sea which separates Europe and Africa, an explosive, a particular weapon, and belonging to a specific kind, respectively. Suppose in the Secret world it is true that the particular ship is located in the Mediterranean Sea and is carrying the particular weapon which is an explosive.

---

10 For a detailed discussion on the various views, we refer to [GALL78].

11 We discuss negative statements in section 2.7.
Then the ground conditional of figure 5a is true if in the Secret world it is also true that the particular ship is going to Libya.

Universal Conditionals: We first define satisfaction of a universal conditional. In order to do this, we need to define the notion of a valuation.

A valuation for a universal conditional is an assignment of objects for all its variables.

Let C be a universal conditional. An interpretation I satisfies C with respect to level L and valuation V if the ground conditional which results from replacing the variables in C by the objects associated with V is true in I with respect to L.

A universal conditional C is true in I with respect to L if I satisfies C with respect to L and all possible valuations.

Existential Conditionals: We first define satisfaction of an existential conditional. In order to do this, we need to define the notion of a valuation.

A valuation for an existential conditional is an assignment of objects for all its variables.

Let C be an existential conditional. An interpretation I satisfies C with respect to level L and valuation V if the ground conditional which results from replacing the variables in C by the objects associated with V is true in I with respect to L.

An existential conditional C is true in I with respect to L if I satisfies C with respect to L for at least one valuation.

Example: Consider the conditional of figure 9b. Consider the following two possible valuations for the universal variable X of this conditional: (i) CHAMPION and (ii) OHIO. Consider the following two possible valuations for the existential variable Y of this conditional: (i) SPARK and (ii) STAR. The conditional is satisfied with respect to the Secret level for the valuation CHAMPION of X if in the Secret world it is true that CHAMPION carries either SPARK or STAR, and is located in the Mediterranean, then it is also true that CHAMPION is destined for Libya. The conditional is satisfied with respect to the Secret world for the second valuation OHIO for X if in the Secret world it is true that OHIO carries either SPARK or STAR and is located in the Mediterranean Sea, then it is also true that OHIO is destined for Libya. The conditional is true with respect to the Secret world, if it is satisfied with respect to the Secret level for both valuations of X.

2.7 REFUTATIONS

The proof theory that we have described can be used by an SSO in order to analyze the application. However, if we want to automate the analysis process, then a theory based on the refutation procedure would be more efficient. Refutation procedures have been developed for semantic nets. We describe how such a procedure could be adapted for multilevel semantic nets.

In order to develop a refutation procedure for multilevel semantic nets, we need to first introduce the notion of negation. That is, there has to be a way of specifying negative information (or statements).

Negative statements can be assumed by default. This is the closed world assumption where anything that is not specified in the semantic net is assumed to be false. On the other hand, negative statements can also be explicitly specified. When proofs are automated,
negative statements have to be explicitly specified. For example, a concept, event, or relationship can have different values at different security levels. We assumed that if in the Unclassified world the ship's destination is Greece, then in the Secret world it is assumed that the ship's destination is Libya. That is, the SSO could make such inferences when given conflicting statements. However, if the proofs are automated, then the negation of the fact that the ship is going to Greece has to be specified. Figure 12 shows how such a negative statement can be specified by an auxiliary net.

We now illustrate the refutation procedure with an example. Consider the auxiliary net, main net, and the auxiliary net for the negative statement shown in figures 13a, 13b, and 13c respectively. The auxiliary net of figure 13a states that if CHAMPION is located in the Pacific Ocean, then at the Secret level one can conclude that its destination is Japan. The main net shown in figure 13b states that the destination of Ohio is Japan. It does not give any information on CHAMPION except its name. The negative statement shown in figure 13c states that, at the Secret level, the destination of CHAMPION is not Japan. Suppose at the Secret level we want to show that CHAMPION is not in the Pacific. Then, we must add its negation (which is "CHAMPION is in the Pacific") to the main net. This is shown in figure 13d. Then applying the transfer rule, we can conclude that CHAMPION is going to Japan. This is shown in figure 13e. This is a contradiction because of the negative statement of figure 13c. Therefore, the assumption that CHAMPION is in the Pacific is false. Note that we cannot arrive at this contradiction at the Unclassified level. This is because, at the Unclassified level, we cannot apply the transfer rule, nor can we assume that the negative statement is true.

Figure 12. Negative Statement

Figure 13. Refutation Process
3. A NOTE ON MULTILEVEL CONCEPTUAL GRAPHS

3.1 OVERVIEW

A conceptual graph is a graph-based notation developed by Sowa [SOWA84] for representing and reasoning about knowledge. It evolved from other graph-based representation schemes, such as semantic nets. It has been found to be more flexible, more extensive, and more precisely defined than its predecessors [RING88]. It has the full representational power of first-order logic. Further, its mapping onto logic is precisely defined. It has also been argued that conceptual graphs can be naturally extended to represent modality and time. Due to all these advantages, Clancey [CLAN85] writes the following:
"Every AI and cognitive science researcher should study the conceptual graph notation and understand its foundation in logic, database, and knowledge representation research."

Although much of the development of conceptual graphs has evolved from semantic nets, their origins date back to the work of Pierce on graph structures carried out during the last century. The basic constructs of conceptual graphs are the notions of concepts and relationships between concepts. A concept could be any entity, action, or event. Concepts are related to one another by relationships. Concepts are represented as nodes and relationships as links. Each concept is an instantiation of a concept type. The concept types are ordered by the relation "<". For example, the concept types PERSON and MAN are ordered by MAN < PERSON. This means that every man is also a person. An instantiation of the type MAN is "John." The relationships are also instantiations of RELATIONSHIP TYPES. For example, if AGENT is a relationship type it can be instantiated as John is the agent of drinking. This means that John is drinking.

Our interest in conceptual graphs is to be able to represent a multilevel application using these graphs and be able to reason about the multilevel application. In section 2, we found that semantic nets were quite useful for this purpose. Furthermore, a considerable amount of representation of the application as well as reasoning can be carried out using multilevel semantic nets. However, semantic nets do not have the full power of first-order logic [RING88]. As a result, it may not be possible to be able to completely represent some applications. Therefore, if a complete analysis of any multilevel application has to be performed, a more powerful representation and corresponding reasoning strategies are needed.

Conceptual graphs, as they are defined, are not adequate to represent multilevel applications. That is, we need to develop the notion of multilevel conceptual graphs for this purpose. In section 3.1, we briefly discuss some of the essential points of multilevel conceptual graphs, and in section 3.2, we discuss some of the reasoning strategies. We assume that the reader is familiar with conceptual graphs as described in [SOWA84].

3.2 REPRESENTATIONAL ISSUES

In this section, we define the notion of multilevel conceptual graphs (MCG). A multilevel conceptual graph is a conceptual graph with security levels assigned to concept types, concepts, relationship types, and relationships. We enforce the following rules for consistency. The reasons are explained in our discussion on multilevel semantic nets.

Rule B1: If X is an instantiation of concept type C, then Level(X) ≥ Level(C).

Rule B2: If Y is an instantiation of a relationship type R, then Level(Y) ≥ Level(R).

Rule B3: If C1 and C2 are concept types and if C1 < C2, then Level(C1) ≥ Level(C2).

A basic MCG should represent assertions associated with the multilevel application being modelled. For example, consider the assertion at the Secret level that there is a battle ship named FLORIDA which is sailing to Libya. However at the Unclassified level, the assertion states that there is a passenger ship named FLORIDA which is sailing to Greece. Figure 14 illustrates this assertion. At the Secret level, the ship FLORIDA is the agent to the action of sailing. The event that results from the action is reaching Libya. An attribute of FLORIDA is battle. At the Unclassified level, the ship FLORIDA is the agent to the action of sailing. The event that results from the action is reaching Greece. The attribute of FLORIDA is passenger.

Negative statements as well as more complex assertions can be represented using the concept type PROPOSITION. The instances of PROPOSITION is any conceptual graph. It is
represented as a box. The unary relation NOT is then inserted in front of the box to imply its negation. For example, consider the MCG shown in figure 14. The information presented at the Secret level conflicts with the information presented at the Unclassified level. A human analyst at the Secret level could assume that the information at the Secret level is the valid information. However, an automated analyst needs to be told that the information represented at the Unclassified level is not valid. That is, the negative information "It is not the case that there is a passenger ship FLORIDA which is going to Greece" needs to be specified. One way of representing this negative information is shown in figure 15.

Figure 16 shows another use of the PROPOSITION type. Consider the Secret assertion "The captain knows that the enemy is going to attack on 6/16/90." The assertion "enemy is going to attack on 6/16/90" is represented as an instance of PROPOSITION type. This instance is Unclassified. The assertion that the captain knows something is also Unclassified. What the captain knows is Secret.

Universal statements can also be expressed by conceptual graphs. For example, the assertion "every ship is managed by a captain" is represented in figure 17. Representing several other types of assertions is described in [SOWA84]. It has also been shown that conceptual graphs can be naturally extended to represent beliefs, modalities, and tenses, among others.

![Multilevel Conceptual Graph](Image)

Figure 14. Multilevel Conceptual Graph

---

12By a captain we mean "some captain," and should, therefore, be existentially quantified. Many conceptual structures implicitly assume existential quantification.
Figure 15. Negative Statement

Figure 16. Complex Assertions
3.3 REASONING STRATEGIES

Several reasoning strategies (or deduction/inference rules) have been proposed for conceptual graphs. We discuss a few of these strategies and show how inferences can be detected.

Three of the deduction rules that have been proposed are:

(i) Restriction: This rule takes a graph and replaces any of its concept nodes by a subtype concept node or by a special instance of the concept. For example, the concept type PERSON may be restricted to MAN or an instance such as John.

(ii) Joining: This rule takes two graphs with a common concept (or concept type) and joins them over this concept. Various types of joins have been proposed.

(iii) Simplifying: This rule removes duplicate relationships between two concepts (or concept types) which could possibly arise after a join operation.

We illustrate the rules with an example. Figure 18a states that the ship is sailing to a country. Figure 18b states that the battle ship is sailing fast. Figure 18c is the restriction of 18a. It states that the battle ship is sailing to a Middle East country. Figure 18d is the join of 18b and 18c. It states that the battle ship is sailing fast to a Middle East country. Figure 18e removes the redundant relationship between battleship and sailing.

The following security properties should be enforced whenever a deduction rule is applied:

Let C1 be a conceptual graph and C2 be the result of applying the restriction rule to C1. Any security level of the restricted concept (or concept type) in C2 must dominate the level of the original concept in C1.

Let C1 and C2 be conceptual graphs which are joined to form C3. Any concept, concept type, or relationship in C3 must have the security level of the corresponding concept, concept type, or relationship in C1 or C2. For example, if the concept type Middle East country in figure 18c is Secret, then this concept type in figure 18d must also be Secret.

Let C1 be a conceptual graph simplified to C2. Let R1 and R2 be identical relationships between the same two concepts in C1. The security level of this relationship in the simplified graph C2 is the greatest lower bound of the levels of R1 and R2.
Next we describe how these reasoning strategies can be used to detect inferences. First of all, we need a way to represent security constraints. We assume that constraints are also represented as conceptual graphs. Figure 19a specifies a constraint which classifies the fact that ships carry weapons at the Secret level. Consider the main conceptual graph of figure 19b. It asserts that the Ship FLORIDA carries the weapon SPARK. This graph violates the security constraint. It can be detected as follows. Apply the deduction rules to the security constraints

![Diagram](image1)

![Diagram](image2)

![Diagram](image3)

Figure 18. Applying Deduction Rules
and see whether any portion of the main conceptual graph can be derived. If so, check whether the levels of the concepts, concept types, and relationships of the derived graph dominate the levels of the corresponding constructs in the main conceptual graph. If so, there is a security violation. In this example, the constraint is restricted by substituting FLORIDA for ship and SPARK for weapon. The agent and object relationships remain Secret. The derived graph occurs in the main conceptual graph. However, the agent and object relationships in the main conceptual graph are Unclassified. This causes a security violation.
We have stated only the essential points in the representation and manipulation of the multilevel applications using conceptual graphs. More work needs to be done if useful tools for multilevel application analysis are to be developed.

![Conceptual Graphs](image)

Figure 19. Violating Security Constraints

4. CONCLUSION

Inference is the process of forming conclusions from premises. It becomes a problem if unauthorized conclusions are drawn from authorized premises. It is a complex problem because the inference process is not controlled by any individual, group, or organization. That is, it is possible for humans to make numerous inferences from the knowledge they possess.

Due to the complexity of the problem, we feel that several approaches are needed to control it. We are conducting various research and development activities to handle this problem during transaction processing as well as during database design. This paper has described some of our research in handling this problem during database design. It shows how conceptual structures could be used to handle the inference problem.

Conceptual structures have been used extensively in the past for artificial intelligence and natural language processing applications. This is because structures such as these make possible the capture of certain semantics of applications as well as languages. We have extended conceptual structures to multilevel conceptual structures in order to capture the semantics of multilevel applications. This is necessary because, in a multilevel application, not all of the concepts and relationships are assigned the same security level. We have also used various reasoning strategies in order to reason about the multilevel application represented by multilevel conceptual structures. The SSO could use the techniques that we have developed to manually analyze the multilevel application and detect possible security violations that could occur. On the other hand, a tool based on our approach could also be developed which could be used by the SSO to design the multilevel database.
In this paper, we have described a multilevel conceptual model based on semantic nets for representing and reasoning about multilevel applications. As stated earlier, we have also investigated the use of other conceptual structures such as conceptual graphs for representing and reasoning about multilevel applications. Future work should include the following:

(i) Selecting a multilevel Navy application and representing it using the model that we have described,
(ii) Developing a tool based on the model which can be used by the SSO to design the multilevel database, and
(iii) Investigating the use of other conceptual structures. This will include a more detailed investigation of structures such as multilevel conceptual graphs and other hyper-semantic data models such as the one described in [TRUE89]. Such models have the representational power of semantic data models and the reasoning power of logic.

ACKNOWLEDGEMENTS

We gratefully acknowledge the Department of the Navy (SPAWAR) for sponsoring this work under contract F19628-89-C-0001.

DISCLAIMER

All examples given in this paper are hypothetical. They do not reflect the policies of the the U.S. Government, the Department of the Navy, or of the MITRE Corporation.

REFERENCES


