19

Inference Analysis During Multilevel Database Design*

R. K. Burns
AGCS Inc., ESC/ENS, Hanscom AFB, MA 01731
(617) 377-9501 burns@stars1.hanscom.af.mil

Abstract

Automated analysis tools offer potential solutions to many of the difficult inference problems found in multilevel database applications. The work described here brings together two interactive tools to assist the designer in creating a multilevel database application that is free of known inference problems. One tool, the SRI inference tool (Stickel, 1994b), is a research prototype developed under contract to Air Force Rome Laboratory. The other tool is a commercial product, Software Through Pictures (StP), developed and marketed by Interactive Development Environments, Inc. The integration of these two tools provides a robust and modern development environment that is tailored to address multilevel database security.

Keywords

Multilevel security, database design, computer-aided software engineering.

1 INTRODUCTION

The design of database applications is a complex process that benefits significantly from the use of automated diagramming and analysis tools. This paper describes two tools that are being integrated to aid in the design of Multilevel Secure (MLS) database applications. The Database Inference System Security Tool (DISSECT), developed by SRI International, is a prototype inference analysis tool that identifies a class of potential design problems in a multilevel database schema. The information modeling (IM) component of Interactive Development Environment's Software through Pictures (StP) provides Entity-Relationship (E-R) diagramming tools and a variety of analysis tools to create a normalized relational database schema from an E-R

* This work was sponsored by Air Force Rome Laboratory under contract number F19628-92-C-0006.

D. L. Spooner et al. (eds.), *Database Security IX*

conceptual schema. The integration of the DISSECT tool with this commercial Computer Aided Software Engineering (CASE) product combines the strengths of both tools, guiding a database designer towards a multilevel relational schema that is free of inference channels.

This paper first briefly describes the two tools and then identifies the work that was done to integrate the tools into a coherent multilevel database design environment. The remainder of the paper describes an example database schema and how the combined tools are used during the database design process. A final section of the paper summarizes the work to date and describes possible future work.

2 THE DISSECT INFERENCE TOOL

The DISSECT tool was developed by SRI International as a research prototype under contract to Air Force Rome Laboratory (Stickel (1994a, 1994b), Garvey (1992), Qian (1993)). The tool analyzes a multilevel relational database schema and identifies potential inference channels that result from foreign key references and joins on common datatypes. The tool is written in Common Lisp and makes use of SRI's GRASPER-CL tool for graph manipulations and analysis. The relational schema is transformed into a graph, where the nodes represent groups of attributes and the edges represent relationships between the groups. Foreign key relationships are represented by a directed arc from the primary key attribute group of one relation to the foreign key attribute group of the other relation. Attribute dependencies are represented by directed arcs from the primary key attribute group to the other attribute groups within a single relation. Datatype dependencies are similarly represented. The attribute groups are annotated with a sensitivity label range assigned to the group by the database designer. The arcs are annotated with the least upper bound of the labels of the connected groups. DISSECT also recognizes *near keys* (Smith 1990), which are non-key attributes that may provide uniqueness. (For a detailed discussion of the mapping of a multilevel relational schema to the directed graph, refer to the SRI Final Report (Stickel, 1994b).) Using the graph model, DISSECT is able to identify compositional channels, where multiple paths exist between two nodes (groups of attributes) but have different classifications.

Based on the SeaView multilevel relational model (Lunt, 1990) and the compositional channels identified in the directed graph, DISSECT defines seven classes of constraints it attempts to satisfy during its analysis of the schema.

1. **Uniform Primary Key:** All attributes in the primary key have the same classification.
2. **Uniform Foreign Key:** All attributes in a foreign key have the same classification.
3. **Uniform Group:** All attributes in a group have the same classification range.
4. **Non-key Dominates Primary Key:** The classifications of all attribute groups in a relation dominate the classification of the primary key group.
5. **Foreign Key Dominates Referent:** The classifications of all foreign key attribute groups dominate the classification of the primary key to which they refer.
6. **Paths of Foreign Keys:** The classifications of all foreign key paths between the same two attribute groups are equal.

7. **Paths of Type Overlap:** The classifications of all type overlap paths between the same two attribute groups are equal.

Any constraint that is not satisfied represents a potential inference channel in the database design.

To analyze the schema and eliminate inference channels, DISSECT applies the Davis-Putnam theorem proving procedure (Davis, 1962) to a set of formulas representing the classification of the attributes. (See the SRI Final Report (Stickel, 1994b) for the details of this procedure.) DISSECT identifies attribute group upgrades that would cause the complete set of constraints to be satisfied. In order to optimize the solution, costs are associated with upgrading attribute groups, so that the database designer can selectively encourage/discourage specific upgrades. DISSECT searches for the minimal cost solution and provides a set of solutions that are within a range of minimal. The designer can then analyze the solutions and determine which upgrades are appropriate for the specific application. Alternatively, the designer can specify that specific instances of constraints need not be satisfied (e.g., that a particular foreign key need not dominate its referent). In this case, DISSECT would not include that specific constraint in its generation of formulas for the Davis-Putnam procedure.

3 STP/IM TOOL

Software through Pictures (StP) is a set of CASE tools developed and marketed by Interactive Development Environments (IDE). We are using the Information Modeling component (StP/IM) (IDE, 1994a-c). All of the StP tools are based on a common set of functions and rules, part of the StP Core system (IDE, 1994d-f). Each individual tool manages a set of objects that are unique to the functionality provided by the tool. For example, the Chen E-R diagramming tool supports the entity and relationship notations defined by Chen (Chen, 1976), but stores the information about the entities and relationships as StP objects within the StP repository. Other information modeling tools retrieve the information from the repository and perform specific analysis and/or transformations on the objects. For example, the Structured Query Language (SQL) generation tools read the specifications of the entities and relationships from the repository and generate DBMS-specific SQL to create the necessary tables.

StP can be customized for different development environments by adding user-defined rules to each StP tool and by developing programs in the IDE Query and Reporting Language (QRL). The StP system architecture facilitates such customization by supporting user-defined search paths for the rule and QRL files and user-defined extensions to the object types supported by the Core system. The integration of DISSECT with the StP/IM tools was heavily dependent upon this customization support provided by the StP Core system.

4 TOOL INTEGRATION

This section describes the specific customization and enhancements that were made to both StP/IM and DISSECT. The following section explains how these customizations are used during the development of a multilevel database.

To integrate DISSECT into the StP environment, we performed the following customizations:

1. **The addition of a Label Group specification for each attribute:** We defined a Label Group annotation for the StP/IM attribute object type. Using the StP Annotation Editor, the designer assigns a specific Label Group to each attribute of an entity, as illustrated in Figure 1. All of the attributes of an entity could belong to the same Label Group, or there could be any number of Label Groups used by a single entity. These Label Groups provide the grouping of attributes for the DISSECT tool.

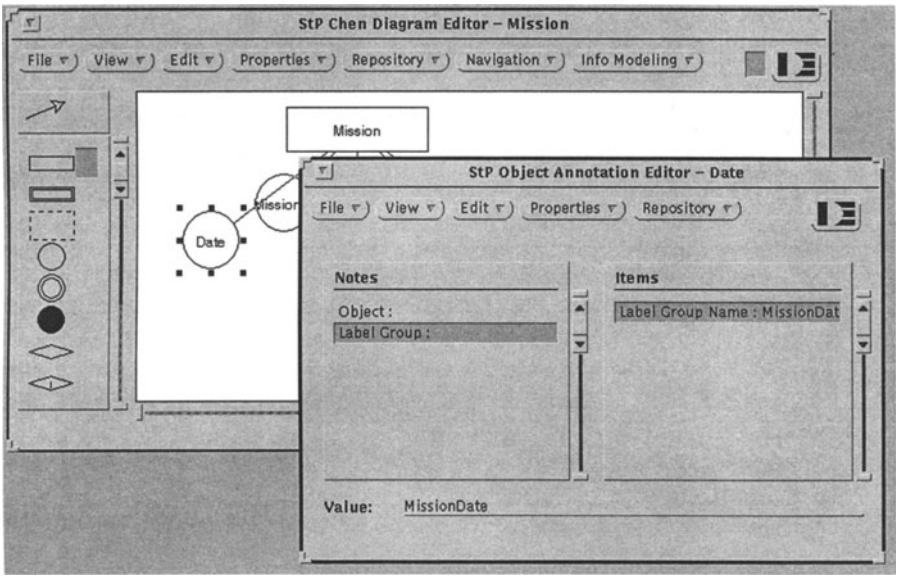

Figure 1 Entering a Label Group Annotation

2. **The definition of the sensitivity labels associated with a Label Group:** We made use of an existing StP/IM editor (the Domain Editor) to define a set of sensitivity labels associated with each Label Group. To distinguish the Label Groups from other domains that might be defined for a database, we appended an "_LBL" suffix to the names of all Label Groups. Ideally, we would have liked to create a new type of editor (e.g., a Label Group Editor), but StP does not directly support that type of customization. Instead, we are using the existing Domain Editor in a way that does not conflict with its original intended use. With the Domain Editor, the database designer specifies the set of sensitivity labels that are valid for each Label Group defined for the schema, as illustrated in Figure 2.

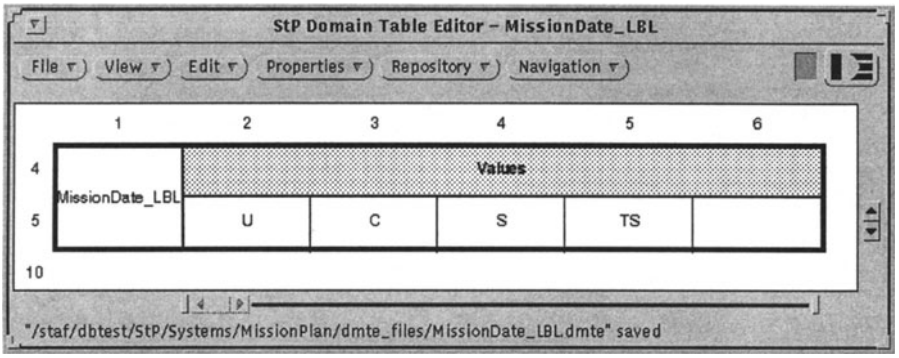

Figure 2 Defining Sensitivity Labels

3. The generation of DISSECT input: Based on the QRL code provided with StP/IM for SQL generation, we developed QRL code that generates the schema definition format needed by DISSECT. Instead of generating standard SQL CREATE TABLE statements, our code uses the information stored in the StP repository, including the Label Group annotations, to generate the Lisp-formatted schema input required by DISSECT. The StP/IM menu items available to the designer were extended to include a "Generate DISSECT" option, as well as the standard "Generate SQL" options, as illustrated in Figure 3. The output of the DISSECT generation is automatically copied to a file within the DISSECT directory structure for convenient access by DISSECT.

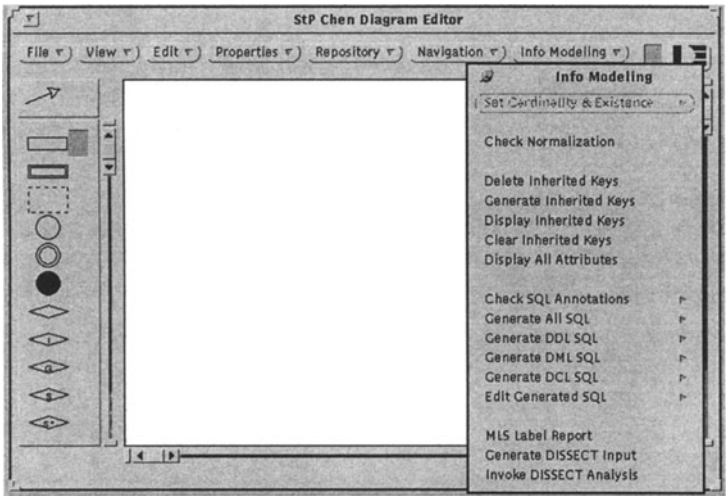

Figure 3. StP Menu Customization

4. The invocation of the DISSECT tool: The final customization involved both StP/IM customization and a minor addition to DISSECT. Another StP/IM menu item (see Figure 3) was added to allow the designer to invoke DISSECT directly from within StP, and the invocation of DISSECT was streamlined to not require any direct user input. Once the DISSECT window is created, the designer is interacting with the SRI tool, independent of StP. While DISSECT was designed to load the schema definition from a filename specified by the user; we added an option to load the schema from the standard filename used for the DISSECT generation. All of the original DISSECT functionality remains available to the designer.

These four customizations were reasonably straight-forward to implement, once a general familiarity with StP file structures and rule syntax was acquired. Section 6 identifies additional customizations that would extend the usefulness of the integrated toolset.

5 EXAMPLE MULTILEVEL DATABASE SCHEMA.

We used a small Mission Planning database schema (Burns, 1992) as the test vehicle for the integration effort. This E-R conceptual schema contains three entities (Missions, Aircraft, and Locations) and two relationships (a Destination relationship between Missions and Locations and a Posted relationship between Aircraft and Locations). Figure 4 is an E-R diagram of this schema as generated by the StP/IM E-R diagramming tool.

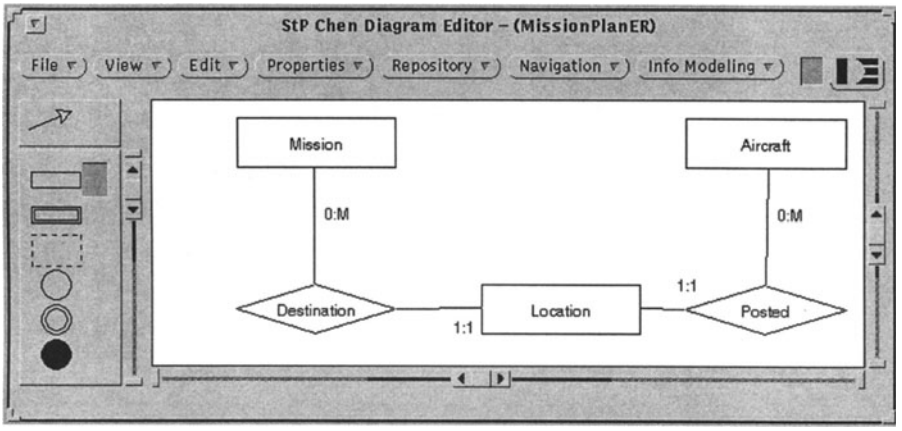

Figure 4 E-R Diagram of Mission Planning Schema

The numbers on the lines connecting the entities to the relationships indicate the existence requirements (the first 0 or 1) and the cardinality (the second 1 or M). In this schema, there is a One-to-Many relationship between Location and Mission and a One-to-Many relationship between Location and Aircraft (e.g., a single Location may be the Destination of many Missions and/or the Posted Location of many Aircraft). For existence, every Mission must have a Destination and every Aircraft must be Posted to a Location.

To represent relationships in a relational schema, StP/IM uses a concept of *inherited keys* for One-to-One and One-to-Many relationships. (For Many-to-Many relationships, StP/IM creates

an additional schema table to represent the relationship.) Inherited keys form the basis for foreign key references in the relational schemas generated by StP. Figure 5 is the StP representation of the Location entity; it has a primary key of LID. This primary key becomes an inherited key for the Mission and Aircraft entities, as illustrated in Figure 6 and Figure 7 with dashed lines.

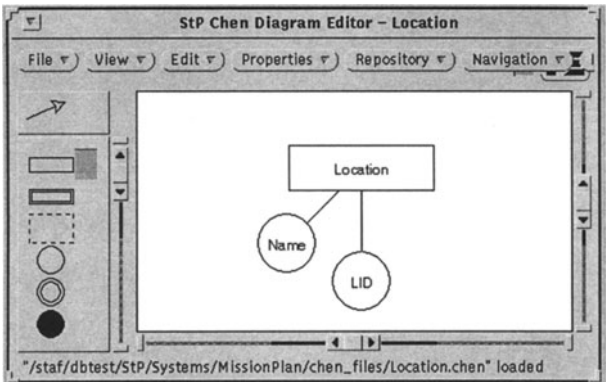

Figure 5 Location Entity and Attributes

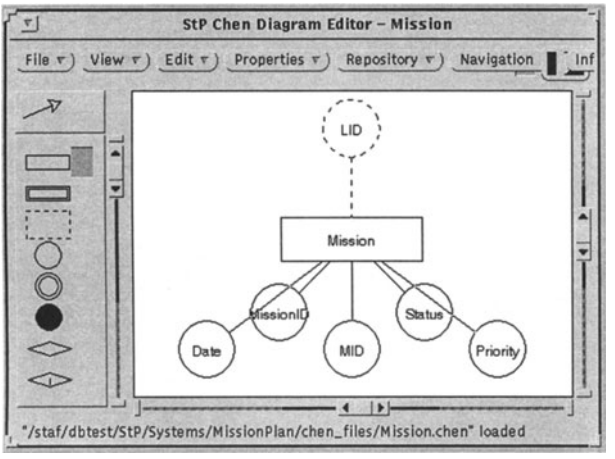

Figure 6 Mission Entity and Attributes

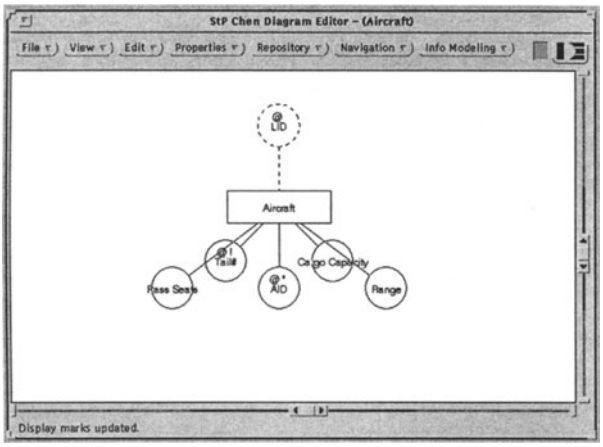

Figure 7 Aircraft Entity and Attributes

The additional attribute annotations include primary keys ("*"), not null ("@"), and unique keys ("!"). Using the StP/IM tool, these annotations and several DBMS-specific annotations (e.g., datatypes) are easily specified by the designer. For this example, we used Data Element names to map to specific ORACLE7 datatypes; each attribute was assigned to a Data Element and an ORACLE7 datatype was specified for each Data Element. Figure 8 illustrates a Data Element definition and Figure 9 shows the Data Element assignments for the Mission Entity.

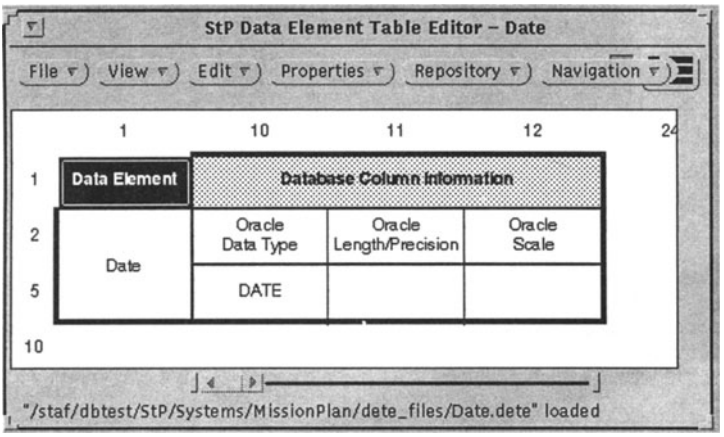

Figure 8 Data Element Definition

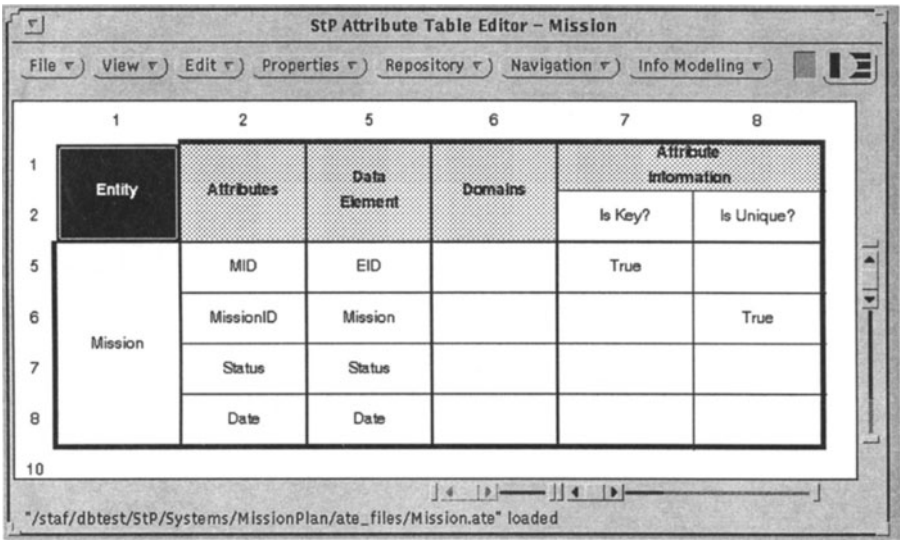

Figure 9 Data Element Assignments

Table 1 is an example of the Data Definition Language (DDL) SQL that StP/IM generates from the information entered about the Mission Planning conceptual schema. The schema generated is for ORACLE7, but several other DBMS products are supported (including Sybase and Informix).

Table 1 Generated SQL Data Definition Language

```
/*
Begin Oracle7 DDL Statements for Table: Mission
*/
DROP TABLE Mission CASCADE CONSTRAINTS
/
CREATE TABLE Mission  (MID INTEGER NOT NULL , MissionID CHARACTER (6) NOT NULL , Status CHAR (1) NOT
NULL , Date DATE NOT NULL , Priority INT NOT NULL , LID INTEGER NOT NULL )
/
ALTER TABLE Mission ADD PRIMARY KEY (MID)
/
ALTER TABLE Mission ADD  CONSTRAINT Mission_MissionID_unique UNIQUE (MissionID)
/
/*
 Begin Oracle7 DDL Statements for Table: Location
*/
DROP TABLE Location CASCADE CONSTRAINTS
/
CREATE TABLE Location  (LID INTEGER NOT NULL , Name CHARACTER (20) NOT NULL )
/
ALTER TABLE Location ADD PRIMARY KEY (LID)
/
```

To annotate the attributes of the Mission Planning entities to add the multilevel security information, the designer clicks on the target attribute (e.g., in the displayed E-R diagram) and invokes the StP Annotation Editor, as illustrated earlier in Figure 1. In addition, the designer must specify the sensitivity levels associated with each Label Group using the Domain Editor, as illustrated in Figure 2

A special Label Group, the Database group, must be specified to provide DISSECT with the valid set of labels for the database and the cost factors associated with upgrades. Figure 10 displays the Database Label group for the Mission Planning database. The order of the values in the Values section determines the dominance relationships for DISSECT. The values in the Ranges section are used as the cost factors for the DISSECT input. The top value is the cost of upgrading the minimum classification to the corresponding label; the bottom value is the cost of upgrading the maximum classification to the corresponding label.

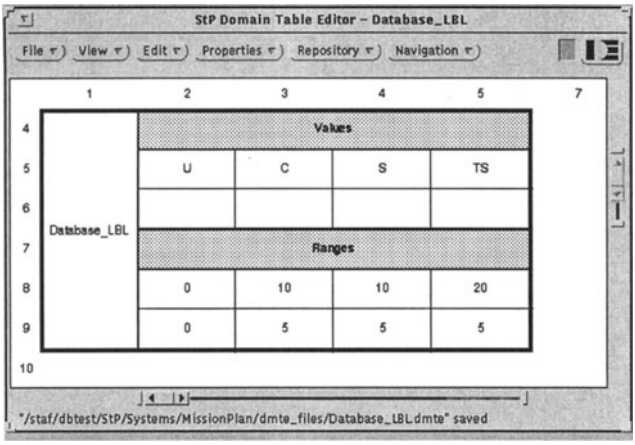

Figure 10 Database Label Group Definition

Once the attributes have been annotated and all of the Label Groups defined, the input file for DISSECT can be generated. Table 2 is the DISSECT input file that was generated for the Mission Planning E-R schema. The format is Lisp, where the ";" indicates a comment and "|" is used to delimit names. The ":SORT" field defines the (optional) datatype for the column; the values were taken from the StP/IM Domain name defined for each attribute.

After the DISSECT file has been generated, the designer invokes DISSECT from the menu and clicks on the "Load StP Schema" menu option to load the generated file into DISSECT. During the load process, DISSECT generates the set of constraints reflected in the schema. The designer may view the constraints and identify any constraints that don't need to be met for the particular application. Figure 11 illustrates the DISSECT screen after the Mission Planning schema has been loaded; Figure 12 illustrates the Foreign Key Path constraints.

To analyze the schema for inference channels, the designer chooses the "Find Solutions" menu item, and DISSECT responds with a set of upgrade solutions, as illustrated in Figure 13. The solution that DISSECT recommends is an upgrade of the foreign key LID attributes from of

Table 2 Generated DISSECT Input

```
;;; Generated Using Annotations

(SETQ LEVEL-INFO (QUOTE ((C 10 5)(S 10 5)(TS 20 5))))

(CREATE_TABLE |Mission|
 (GROUP ;;; Mission
  (COLUMN |MID| :SORT |EID| )
  (COLUMN |MissionID| :SORT |Mission| )
  (COLUMN |Status| :SORT |Status| )
  (COLUMN |Priority| :SORT |Priority| )
 (LABEL-IN U C S))
 (GROUP ;;; MissionDate
  (COLUMN |Date| :SORT |Date| )
 (LABEL-IN U C S TS))
 (GROUP ;;; MissionDestination
  (COLUMN |LID| :SORT |EID| )
 (LABEL-IN U C S))
 (PRIMARY-KEY |MID| )
 (NEAR-KEY |MissionID| )
 (FOREIGN-KEY |LID| |Location| )
)
```

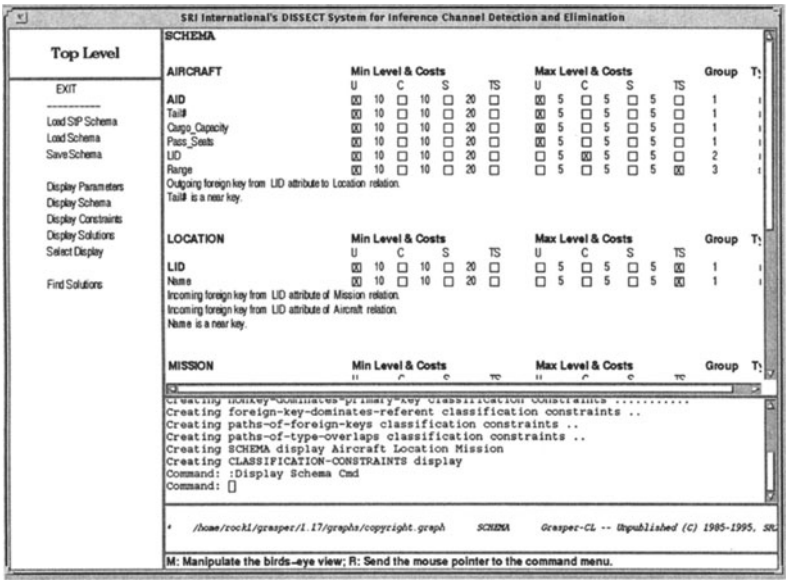

Figure 11 DISSECT Mission Planning Schema

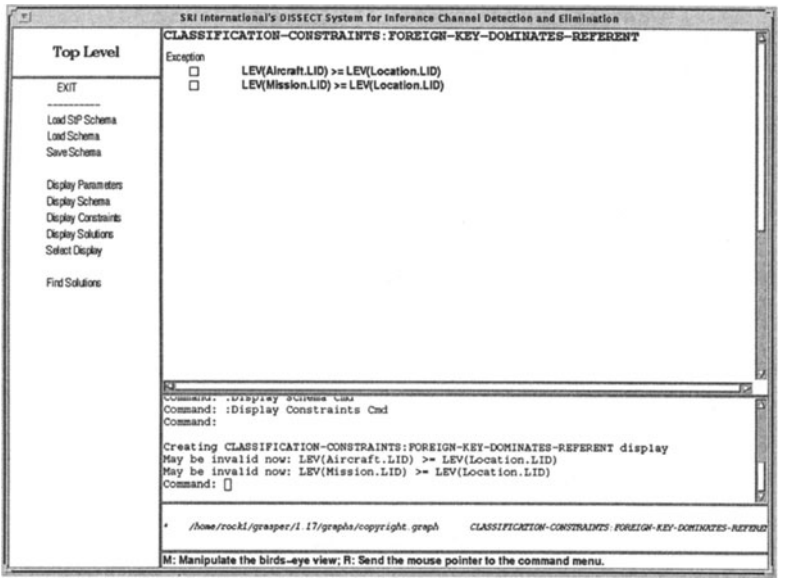

Figure 12 Mission Planning Foreign Key Path Constraints

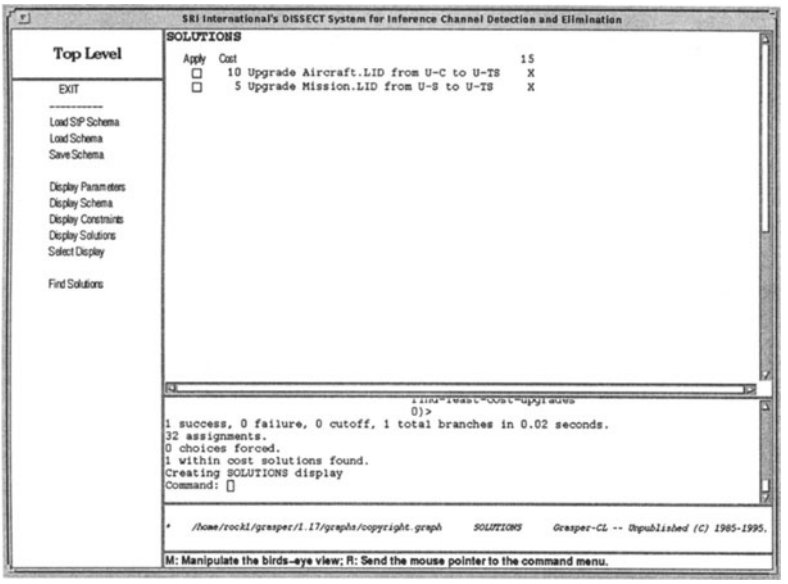

Figure 13 DISSECT Solutions for the Mission Planning Schema

both the Aircraft entity and the Mission entity. DISSECT does not directly provide the reasons it chose the specific upgrades; however, it is possible to review the constraints and the schema design to locate the cause of the problem. In Figure 12, both of the foreign key constraints were displayed in bold, indicating that they may not have been met initially by the schema design. Here the problem with the schema was actually that the label range for Location should be further constrained, not that the LID attributes need to be updated. So, instead of upgrading the AircraftPosted and MissionDestination Label Groups, the designer could choose to *downgrade* the Location Label Group to U. Either solution would be implemented by modifying the values for the relevant Label Group within StP/IM, re-generating the DISSECT input, and then re-analyzing the DISSECT schema. When the Location Label Group is set to just "U", as in Figure 14, the resulting schema "passes" the DISSECT analysis, as illustrated in Figure 15.

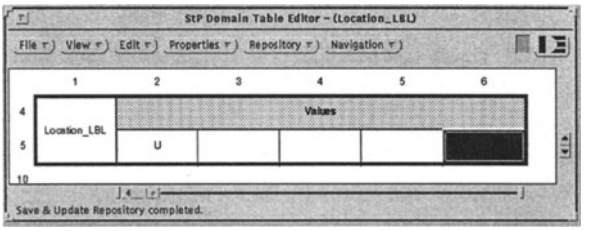

Figure 14 Revised Location Label Group

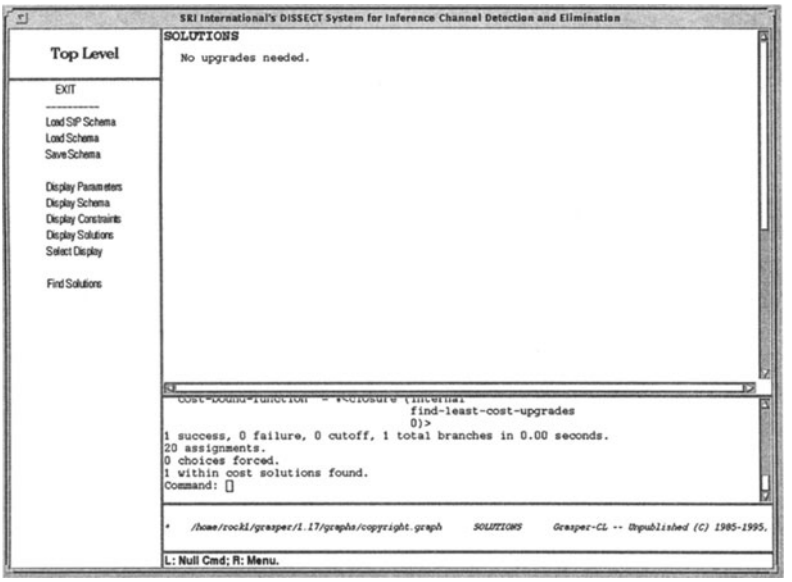

Figure 15 Final DISSECT Solution

6 SUMMARY AND FUTURE PLANS

While an increasing amount of attention is being focused on the problems of multilevel database design ((Wiseman, 1991), (Sell, 1992), (Lewis, 1993), (Marks, 1994)), few automated tools have been developed to aid the designer. Hinke (Hinke, 1992) is developing tools to identify inferences in a database schema and Pernul (Pernul, 1992) is prototyping a database design environment that includes both static and dynamic application models. The work we have undertaken represents an attempt to integrate a research prototype with a commercial database design tool in a manner that best takes advantage of the capabilities of each product. We have demonstrated that the basic integration does provide a flexible tool for multilevel database design. Now we plan to enter additional multilevel schemas to further explore the capabilities of the DISSECT tool for the elimination of inference channels in database schemas. Other StP customizations that would move the toolset towards the goal of a complete multilevel database design paradigm include the following:

1. **Graphical annotations for security**: The current StP customizations do not display the Label Groups or their classification values in E-R diagrams; the Label Group annotations must be entered as text. These annotations are relatively "hidden" from the designer, and a display within an E-R diagram would assist in communicating the overall multilevel design.

2. **Generation of SQL DDL for Trusted ORACLE7, Sybase Secure SQL Server, and Informix On-line/Secure:** Multilevel SQL generation requires that each Label Group be defined as a separate table, since the commercial products all provide sensitivity labels for rows. To improve performance, the automated SQL generation could also create clusters for the tables, so that all of the attributes for an entity could be stored in the same physical data blocks, even though they belong to different tables.

3. **Additional analysis tools:** As tools are developed to analyze multilevel database schemas for different types of problems, they could be added to the StP toolset. For example, additional classification constraints could be entered as annotations to StP, and analysis tools for specific types of constraints could be integrated into the environment.

4. **Support for additional MLS concepts:** Concepts such as cover stories need to be implemented carefully ((Garvey, 1991), (Binns, 1992), (Burns, 1992)). Design constructs, analysis tools, and SQL generation mechanisms would assist a database designer to implement appropriate cover story mechanisms for specific application requirements.

5. **Customization of the StP Object Modeling Technique (OMT) tools:** The StP/OMT tools allow StP/IM entities to be mapped to StP/OMT object classes. By customizing the StP/OMT tools for multilevel security, it would be possible to model the dynamic aspects of a database application (e.g., events and operations), as proposed in (Sell, 1993).

StP and DISSECT provide a strong foundation on which to build sophisticated tools for designers of multilevel database applications. The work described here represents an initial step towards building a toolset that in the future may contain a number of design and analysis tools to aid in the development and deployment of multilevel database applications.

7 REFERENCES

Binns, L. (1992) Inference and Cover Stories, *Proceedings of the Sixth IFIP WG 11.3 Conference on Database Security*, North-Holland.

Burns, R.K. (1992) A Conceptual Model for Multilevel Database Design, *Proceedings of the Fifth Rome Laboratory Multilevel Database Security Workshop.*

Chen, P.P. (1976) The Entity-Relationship Model - Toward a Unified View of Data, *ACM Transactions on Database Systems.*

Davis, M. and H. Putnam, H. (1962) A Computing Procedure for Quantification Theory, *Journal of the ACM*, Vol. 5, No. 7.

Garvey, T.D. and Lunt T.F. (1991) Cover Stories for Database Security, *Proceedings of the Fifth IFIP WG 11.3 Working Conference on Database Security*, North-Holland.

Garvey, T.D., Lunt, T.F.,Qian, X. and Stickel, M.E. (1992) Toward a tool to detect and eliminate inference problems in the design of multilevel databases, *Proceedings of the Sixth IFIP WG 11.3 Working Conference on Database Security*, North-Holland.

Hinke, T. and Delugach, H. (1992) Aerie: An Inference Modeling and Detection Approach for Databases, *Proceedings of the Sixth IFIP WG 11.3 Working Conference on Database Security*, North-Holland.

Hsieh, D., Lunt, T.F. and Boucher, P.K. (1993) The SeaView Prototype Final Report, SRI International.

IDE (1993a) Creating Information Models, Interactive Development Environments, Release 1.

IDE (1993b) Getting Started with Information Modeling, Interactive Development Environments, Release 1.

IDE (1993c) Information Modeling Handbook, Interactive Development Environment.

IDE (1994d) Customizing StP, Interactive Development Environments, Release 1.

IDE (1994e) Fundamentals of StP, Interactive Development Environments, Release 1.

IDE (1994f) Query and Reporting System, Interactive Development Environments, Release 1.

Lewis, S. and Wiseman, S. (1993) Database Design and MLS DBMS: and Unhappy Alliance?, *Proceedings of the Eighth Computer Security Applications Conference*, IEEE Computer Society Press.

Lunt, T.F., Denning, D.E., Schell, R.R., Shockley, W.R. and Heckman, M. (1990) The SeaView Security Model, *IEEE Transactions on Software Engineering*, June 1990.

Marks, D.G., Binns, L.J. and Thuraisingham, B.M. (1994) Hypersemantic Data Modeling for Inference Analysis, *Proceedings of the Eighth IFIP WG 11.3 Conference on Database Security.*

Pernul, G. (1992) Security Constraint Processing during Multilevel Secure Database Design, *Proceedings of the Eighth Computer Security Applications Conference*, IEEE Computer Society Press.

Qian, X., Stickel, M. E., Karp, P.D. , Lunt, T.F. and Garvey, T.D. (1993) Detection and elimination of inference channels in multilevel relational database systems, *Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society Press.

Sell, P. (1992) The SPEAR Data Design Method, *Proceedings of the Sixth IFIP WG 11.3 Conference on Database Security*, North-Holland.

Sell, P. and Thuraisingham, B. M. (1993) Applying OMT for Designing Multilevel Database Applications, *Proceedings of the Seventh IFIP WG 11.3 Conference on Database Security*, North-Holland.

Stickel, M.E. (1994a) Elimination of Inference Channels by Optimal Upgrading, *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy,* IEEE Computer Society Press.

Stickel, M. E., Garvey, T.D., Lunt, T.F. and Qian, X. (1994b) Inference Channel Detection and Elimination in Knowledge-Based Systems, Final Report, SRI International.

Smith, G.W. (1990) Modeling Security-Relevant Data Semantics, *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society Press.

Wiseman, S. (1991) Abstract and Concrete Models for Secure Database Applications, *Proceedings of the Fifth IFIP WG 11.3 Working Conference on Database Security*, North-Holland.

8 BIOGRAPHY

Rae Burns has worked in the area of multilevel database security for over ten years, with a focus on the difficult issues inherent in the design of multilevel database applications. She received a BS degree in mathematics from Stanford University and a Masters of Software Engineering from the Wang Institute of Graduate Studies. She has consulted to DBMS vendors on implementation strategies for multilevel security and has participated in the evaluation of trusted DBMS products for the United States Department of Defense. Currently she is investigating interoperability among multilevel DBMS products and is providing guidance to Air Force programs developing multilevel database applications.