09:43:21 OCA PAD AMENDMENT - PROJECT HEADER INFORMATION 10/27/93

Active

Project #: C-36-X07 Cost share #: Rev #: 3
Center # : 10/24-6-R7827-0A0 Center shr #: OCA file #:
Work type : RES
Contract#: F30602-93-C-0175 Mod #: P00001 Document : CONT
Prime #: Contract entity: GTRC

Subprojects ? : N CFDA:
Main project #: PE #: 777777

Project unit: COMPUTING Unit code: 02.010.300
Project director(s):
NAVATHE S B COMPUTING (404)894-3152

Sponsor/division names: AIR FORCE / GRIFFISS AFB, NY
Sponsor/division codes: 104 / 023

Award period: 930601 to 940228 (performance) 940330 (reports)

Sponsor amount	New this change	Total to date
Contract value	0.00	30,064.00
Funded	5,064.00	30,064.00
Cost sharing amount		0.00

Does subcontracting plan apply ?: N

Title: A METHODOLOGY FOR APPLICATION DESIGN USING ACTIVE DATABASE TECHNOLOGY

PROJECT ADMINISTRATION DATA

OCA contact: Brian J. Lindberg 894-4820

Sponsor technical contact

DR. RAY LIUZZI
(315)000-0000

RL/C3CA
BUILDING 3, 525 BROOKS AVENUE
GRIFFISS AFB, NY 13441-5700

Sponsor issuing office

LAURIE MARCHESANI
(315)330-4996

RL/PKRD
26 ELECTRONIC PARKWAY
GRIFFISS AFB, NY 13441-4514

Security class (U,C,S,TS) : U ONR resident rep. is ACO (Y/N): Y
Defense priority rating : DO-A7 N/A supplemental sheet
Equipment title vests with: Sponsor X GIT
NONE PROPOSED OR ANTICIPATED.
Administrative comments -
P00001 ADDS $5,064 IN INCREMENTAL FUNDS TO FULLY FUND CONTRACT AT $30,064.

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 07/22/94

Project No. C-36-X07_____ Center No. 10/24-6-R7827-0A0_

Project Director NAVATHE S B_____ School/Lab COMPUTING_____

Sponsor AIR FORCE/GRIFFISS AFB, NY_____

Contract/Grant No. F30602-93-C-0175_____ Contract Entity GTRC

Prime Contract No. _____

Title A METHODOLOGY FOR APPLICATION DESIGN USING ACTIVE DATABASE TECHNOLOGY____

Effective Completion Date 940228 (Performance) 940330 (Reports)

Closeout Actions Required:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	Y	_____
Final Report of Inventions and/or Subcontracts	Y	_____
Government Property Inventory & Related Certificate	Y	_____
Classified Material Certificate	N	_____
Release and Assignment	Y	_____
Other _____	N	_____

Comments_____
_____

Subproject Under Main Project No. _____

Continues Project No. _____

Distribution Required:

Project Director	Y
Administrative Network Representative	Y
GTRI Accounting/Grants and Contracts	Y
Procurement/Supply Services	Y
Research Property Managment	Y
Research Security Services	N
Reports Coordinator (OCA)	Y
GTRC	Y
Project File	Y
Other _____	N
_____	N

NOTE: Final Patent Questionnaire sent to PDPI.

A Methodology for Application Design using Active Database Technology

Status Report

June 1, 1993 - August 31, 1993

Prof. S.B. Navathe, P.I.
Ramesh Madhavan
Yee Huat Gan

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

sham@cc.gatech.edu

1 Introduction

This is a status report on the Methodology for Application Design using Active Database Technology project. This report outlines the work that has been done till now as part of the project and also the future work that is intended to be done. The acomplishments so far in this project are overall design of the active database design system, design of the intended tool for drawing the ERER (Entity Relationship diagram with Events and Rules) diagram, design of the integration of the new tool with the existing ERDRAW tool from Lawrence Berkeley Laboratory (LBL). The future work planned includes the implementation of the new tool and integrating it with ERDRAW and extending the existing Schema Translation Tool to include active behavior.

2 Overall design of the Active Database design system

The proposed architecture for the various tools is given in Figure 1 . The ERDRAW tool can be used to create an Extended Entity Relationship diagram (EER diagram) graphically. The EER diagram created can be stored in a .erd file. The .erd file contains all the required information about the EER diagram. This diagram can be displayed back and modified using ERDRAW.

The ERDRAW tool doesn't support generalization. There is no concept of an aggregate entity type in ERDRAW. However, relationship can be regarded as an abstraction of aggregation. Classification is supported in terms of an entity type being a class for which multiple entities (instances) are members. Association in the sense of defining sets of entities is not supported in ERDRAW.

The new tool will be able to read in a .erd file created using ERDRAW and display the diagram. It also allows the user to add active objects (rules, events and triggers) to the diagram graphically. These two tools together will allow the user to create a (ER)2 diagram as described in [1]. Any modifications that has to be done on the basic EER diagram will have to be done using ERDRAW.

The new tool will append the descriptions about the active objects on to the same .erd file. This .erd file thus will have two portions, one relating to the EER objects (like entities, relations, attributes and arcs) and another part relating to the active objects (like events, rules and triggers). Both ERDRAW and the new tool will be able to read-in the respective parts of this file and work on it.

In addition to the .erd file, ERDRAW produces a .sdt file which is used by the Schema Translation Tool (SDT) from LBL. The new tool will also produce a .er2 file which will contain the behavior of the active objects specified in the Behavior Specification Language [1]. The SDT will convert the .sdt file into a meta description of the database.

Both the meta description of the database and the .er2 file is given as the input to the Active Behavior Translation Tool. This tool will create a description of the active database in the SNOOP event specification language which is used in the SENTINEL database.

3 Design of the New Tool

The proposed new tool for adding active objects will work as a extension tool for ERDRAW. It reads-in the output of the ERDRAW (.erd file) and will display the diagram. The user can add and delete **active objects** from the diagram graphically. The same conventions as that used in ERDRAW for adding and deleting objects will be followed in this tool for consistency.

In addition to adding and deleting active objects, the tool also allows to interactively query details about the diagram created. This facility allows the user to see details. For example, an easy to use selection mechanism will allow the user to see all the rules that are fired by a particular event or all the entities that are affected by a rule etc. This will help the user in making design decisions about the database.

4 Integration of the new tool with ERDRAW

The new tool will add the description of the active object in the same format as used by ERDRAW to store description about its object. The format used by ERDRAW is given in Figure 2, Figure 3, Figure 4 and Figure 5.

The description of the active objects will be appended to the .erd file which has already been created. This will make sure that whatever that has been added by the new tool will not interfere with the way ERDRAW will be interpreting the file. So a user will have to first create an EER diagram using ERDRAW and save it in a .erd file. The same file will have to be displayed using the new tool to attach active objects. Once the user has done the attaching of the active objects to the diagram, the diagram can be saved in the same file. Subsequent modification to the diagram

should be done using ERDRAW (in case the modification is to EER objects) or the new tool (in case the modification is to the active objects).

The new tool will also produce .er2 file which will contain a description of the active object description in the Behavior Specification Language. This will be one of the inputs to the Active Behavior Translation Tool.

5 Present State of the Project

As mentioned earlier, the overall design for the tool architecture is complete as well as the design for the new tool. The new tool can be integrated with the ERDRAW which is already existing. We are trying to co-ordinate with Lawrence Berkeley Lab in the development of the new tool so that both of them would be compatible. The new tool will be implemented on motif (as the ERDRAW) and will be very much consistent with ERDRAW.

The translation of the output of the two tools into a SNOOP specification needs to be done. This requires some more input about the latest status of the SENTINEL work from Dr. Sharma Chakraborthy.

References

[1] Asterio Kiyoshi Tanaka. On Conceptual Design of Active Databases. *Ph. D Thesis, Georgia Institute of Technology*, Novemeber 1992.

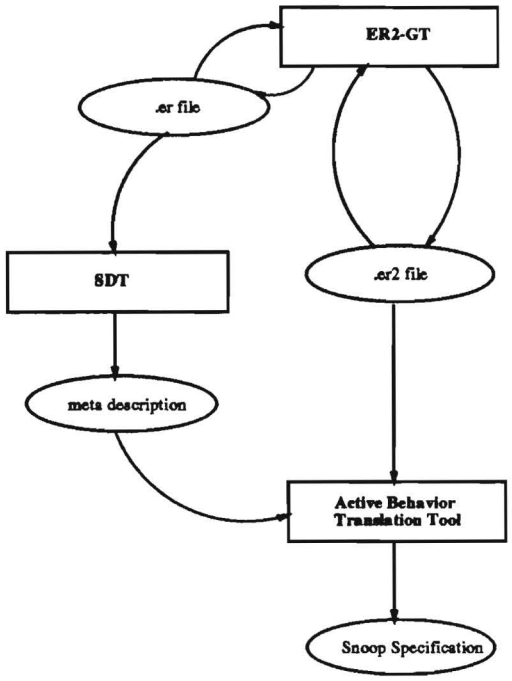

Figure 1: Proposed Tool Architecture

maxERIndex	Indicates the number of ER objects (Entities and Relationships) in the diagram. The index starts from 0.
maxArcIndex	Indicates the number of Arcs in the diagram. The index starts from 0
maxAttrIndex	Indicates the number of Attributes in the diagram. The index starts from 0

Figure 2: General Information format for ERDRAW

erIndex The number of the ER object starting from 0	**erTable.deleted** * indicates that this object has been deleted. Else it is blank
erTable.type Type - E for Entity and R for Relation	**erTable.name** Name of the object as displayed
erTable.descr Description of the object as entered	**erTable.x** The x co-ordinate of the point at which the object is displayed
erTable.y The y co-ordinate of the point at which the object is displayed	**erTable.pageNo** The page number in which the object is displayed

Figure 3: Format for ER objects in ERDRAW

attrIndex Number of the attribute starting from 0	**attrTable.erIdx** Index of the ER object to which the attribute is connected
attrTable.attrSeq Sequence number of the attribute in the given ER object	**attrTable.name** Name of the attribute as displayed
attrTable.descr Description of the attribute as displayed	**attrTable.type** Data type of the attribute
attrTable.length The length of the data type for the attribute	**attrTable.valSet** The value set of Sybase DB if used. It will override type and length if used
attrTable.null Indicates whether the attribute can be null or not	**attrTable.keyType** Indicate whether the attribute is a primary or alternate key etc

Figure 4: Format for Attribute objects in ERDRAW

arcIndex Number of the arc starting from 0	**arcTable.deleted** * indicates the this arc has been deleted. Else it will be a blank
arcTable.type Indicates the arc is one or may relation.	**arcTable.mand** Indicates whether the arc is mandatory
arcTable.updRule Indicates the type of update applicable for the relation	**arcTable.rolename** Name of the arc as displayed
arcTable.x1 The x co-ordinate fo the point at which the arc begins	**arcTable.y1** The y co-ordinate of the point at which the arc begins
arcTable.mx The x co-ordinate of the mid-point of the arc where the rectangle is displayed	**arcTable.my** The y co-ordinate of the midpoint of the arc where the rectangle is displayed
arcTable.x2 The x co-ordinate of the point at which the arc ends	**arcTable.y2** The y co-ordinate of the point at which the arc ends
arcTable.pageNo The page number in which the arc is displayed	**arcTable.fromObj** The index of the object to which the source end of the arc is connected
arcTable.toObj The index of the object to which the destination end of arc is connected	**arcTable.conn1** The exact connection point of the arc at the source end
arcTable.conn2 Te exact connection point of the arc at the destination end	

Figure 5: Format for Arc Objects in ERDRAW

A Methodology for Application Design using Active Database Technology

Status Report

September 1, 1993 - October 31, 1993

Prof. S.B. Navathe, P.I.
Ramesh Madhavan
Yee Huat Gan

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

sham@cc.gatech.edu
ramesh@cc.gatech.edu
gan@cc.gatech.edu

1 Introduction

This is a status report on the Methodology for Application Design using Active Database Technology project. This report outlines the approach we have taken and the work that has been done till now. Future work that is intended is also discussed. The accomplishments so far in this project are the complete implementation of the ER-GT tool and the preliminary design for the representation for Snoop [2]. The future work planned includes the prototype implementation of the enhanced tool and the complete design for the active behavior representation. Iteration of the design will be done based on the prototype with user feedback.

2 Enhancements to LBL ERDRAW tool

The initial plan is to directly enhance the Lawrence Berkeley Laboratory (LBL) ERDRAW Version 5.0. A discussion was carried out with the LBL staff and we were informed that they do not release the source code of the software. However, in order to ensure that the enhancements will be smoothly integrated with the existing tool, we would need access to the source code.

The alternative option that was suggested during the discussion was for them to include the drawing facilities that are needed by the new tool. They will implement the source code to draw the required symbols. Dummy functions will be then provided and then we will add the required code to these functions. For instance, when a new symbol is selected, they will activated our functions.

However, this approach presents the following disadvantages:

- There will be an overhead of communication which will inevitably slow down the progress.
- The symbols for ER^2 diagram [1] are well-defined. However, Snoop has a much richer set of active behavior specification than ER^2. Therefore, we would have to design new symbols to represent the more complex specifications. This design will be done iteratively with user feedback.

  Therefore, it is anticipated that changes to the symbols and representation will be done frequently. In order to obtain more accurate user feedback, we would need to have a working prototype. A prototype will allow us to test the ease of use and the clarity of the representation. However, the continuous of this prototype will impose an excessive burden on the LBL staff.

3 The New Approach

The current approach is to develop our tool called ER-GT. In this case, we would have a fully developed tool for our enhancement. However, the functionality and external interface to ER-

GT will be very similar to the LBL ERDRAW. This tool will be fully compatible with the LBL tool in that data files for these two tools can be freely interchanged.

Having our own ER-GT will not only allow enhancements to be made to the new active behavior functionality but to the underlying ER features as well. However, care must be taken when such enhancements so that compatibility with the LBL tool is still maintained.

An advantage of maintaining compatibility with the LBL ERDRAW tool is that whatever enhancements made to ER-GT can later be used to enhance their tool as well. By then, the design would have been complete and it would be easy for the LBL staff to perform the integration.

When the ER-GT tool is complete, we can easily develop working prototypes for the enhanced tool with active behavior specifications.

4 Design of New Representation

As mentioned, Snoop has a much more richer active behavior specification than ER^2. Therefore, we would have to design new symbols to represent the more complex specifications:

In this design, the following must be kept in mind: there is a trade-off between representing as much active behavior information as possible and cluttering the diagram with too much symbols. If too much information is shown in one diagram, the resulting picture will be too confusing to be of any use.

The current design approach is to represent the information in terms of layers. The initial diagram will show the highest level of information. The user can then request more details to be shown by zooming into the area of interest. This will produce a hierarchy of diagrams with each sub-diagrams showing more detailed information.

5 Present State of the Project

The present ER-GT tool is complete. We can now use the tool to perform the functions of the LBL ERDRAW tool. It is implemented in Motif and X-Windows. We are now in the process of designing and experimenting different representations for the active behavior specification. The user interface and ease of use will be taken into consideration. Working prototypes of the enhanced tool will be built to obtain user feedback and to validate the design.

References

[1] Asterio Kiyoshi Tanaka. On Conceptual Design of Active Databases. *Ph.D. Thesis, Georgia Institute of Technology*, November 1992.

[2] Sharma Chakravarthy and Deepak Mishra. Snoop: An Expressive Event Specification Language For Active Databases, *Technical Report UF-CIS-TR-93-007, University of Florida*, March 1993.

C-36-X07
6,7,8

A Methodology for Application Design using Active Database Technology

Status Report

November 1, 1993 - December 31, 1993

Prof. S.B. Navathe, P.I.
Ramesh Madhavan
Yee Huat Gan

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

sham@cc.gatech.edu
ramesh@cc.gatech.edu
gan@cc.gatech.edu

1 Introduction

This is a status report on the Methodology for Application Design using Active Database Technology project. This report outlines the approach we have taken and the work that has been done till now. Future work that is intended is also discussed. The accomplishments so far in this project are the complete implementation of the ER-GT tool and the preliminary design for the representation for Snoop [2]. The future work planned includes the prototype implementation of the enhanced tool and the complete design for the active behavior representation. Iteration of the design will be done based on the prototype with user feedback.

2 Overall design of the Active Database design system

The proposed architecture for the various tools is given in Figure 1 . The ERDRAW tool can be used to create an Extended Entity Relationship diagram (EER diagram) graphically. The EER diagram created can be stored in a .erd file. The .erd file contains all the required information about the EER diagram. This diagram can be displayed back and modified using ERDRAW.

The ERDRAW tool doesn't support generalization. There is no concept of an aggregate entity type in ERDRAW. However, relationship can be regarded as an abstraction of aggregation. Classification is supported in terms of an entity type being a class for which multiple entities (instances) are members. Association in the sense of defining sets of entities is not supported in ERDRAW.

The new tool will be able to read in a .erd file created using ERDRAW and display the diagram. It also allows the user to add active objects (rules, events and triggers) to the diagram graphically. These two tools together will allow the user to create a ER^2 diagram as described in [1]. Any modifications that has to be done on the basic EER diagram will have to be done using ERDRAW.

The new tool will append the descriptions about the active objects on to the same .erd file. This .erd file thus will have two portions, one relating to the EER objects (like entities, relations, attributes and arcs) and another part relating to the active objects (like events, rules and triggers). Both ERDRAW and the new tool will be able to read-in the respective parts of this file and work on it.

In addition to the .erd file, ERDRAW produces a .sdt file which is used by the Schema Translation Tool (SDT) from LBL. The new tool will also produce a .er2 file which will contain the behavior of the active objects specified in the Behavior Specification Language [1]. The SDT will convert the .sdt file into a meta description of the database.

Both the meta description of the database and the .er2 file is given as the input to the Active Behavior Translation Tool. This tool will create a description of the active database in the SNOOP event specification language which is used in the SENTINEL database.

3 Enhancements to LBL ERDRAW tool

The initial plan is to directly enhance the Lawrence Berkeley Laboratory (LBL) ERDRAW Version 5.0. A discussion was carried out with the LBL staff and we were informed that they do not release the source code of the software. However, in order to ensure that the enhancements will be smoothly integrated with the existing tool, we would need access to the source code.

The alternative option that was suggested during the discussion was for them to include the drawing facilities that are needed by the new tool. They will implement the source code to draw the required symbols. Dummy functions will be then provided and then we will add the required code to these functions. For instance, when a new symbol is selected, they will activated our functions.

However, this approach presents the following disadvantages:

- There will be an overhead of communication which will inevitably slow down the progress.

- The symbols for ER^2 diagram [1] are well-defined. However, Snoop has a much richer set of active behavior specification than ER^2. Therefore, we would have to design new symbols to represent the more complex specifications. This design will be done iteratively with user feedback.

  Therefore, it is anticipated that changes to the symbols and representation will be done frequently. In order to obtain more accurate user feedback, we would need to have a working prototype. A prototype will allow us to test the ease of use and the clarity of the representation. However, the continuous development of this prototype will impose an excessive burden on the LBL staff.

4 The New Approach

The current approach is to develop our tool called ER-GT. In this case, we would have a fully developed tool for our enhancement. However, the functionality and external interface to ER-GT will be very similar to the LBL ERDRAW. This tool will be fully compatible with the LBL tool in that data files for these two tools can be freely interchanged.

Having our own ER-GT will not only allow enhancements to be made to the new active behavior functionality but to the underlying ER features as well. However, care must be taken when such enhancements so that compatibility with the LBL tool is still maintained.

An advantage of maintaining compatibility with the LBL ERDRAW tool is that whatever enhancements made to ER-GT can later be used to enhance their tool as well. By then, the design would have been complete and it would be easy for the LBL staff to perform the integration.

When the ER-GT tool is complete, we can easily develop working prototypes for the enhanced tool with active behavior specifications.

5 Design of New Representation

As mentioned, Snoop has a much more richer active behavior specification than ER^2. Therefore, we would have to design new symbols to represent the more complex specifications:

In this design, the following must be kept in mind: there is a trade-off between representing as much active behavior information as possible and cluttering the diagram with too much symbols. If too much information is shown in one diagram, the resulting picture will be too confusing to be of any use.

The current design approach is to represent the information in terms of layers. The initial diagram will show the highest level of information. The user can then request more details to be shown by zooming into the area of interest. This will produce a hierarchy of diagrams with each sub-diagrams showing more detailed information.

6 Present State of the Project

The present ER-GT tool is complete. We can now use the tool to perform the functions of the LBL ERDRAW tool. It is implemented in Motif and X-Windows. We are now in the process of designing and experimenting different representations for the active behavior specification. The user interface and ease of use will be taken into consideration. Working prototypes of the enhanced tool will be built to obtain user feedback and to validate the design.

References

[1] Asterio Kiyoshi Tanaka. On Conceptual Design of Active Databases. *Ph.D. Thesis, Georgia Institute of Technology*, November 1992.

[2] Sharma Chakravarthy and Deepak Mishra. Snoop: An Expressive Event Specification Language For Active Databases, *Technical Report UF-CIS-TR-93-007, University of Florida*, March 1993.

C-36-X07
8,10,11

A Methodology for Application Design using Active Database Technology

Status Report

January 1, 1994 - February 8, 1994

Prof. S.B. Navathe, P.I.
Ramesh Madhavan
Yee Huat Gan

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

sham@cc.gatech.edu
ramesh@cc.gatech.edu
gan@cc.gatech.edu

1 Introduction

This is a status report on the Methodology for Application Design using Active Database Technology project. This report outlines the approach we have taken and the work that has been done till now. Ongoing work is also discussed. The accomplishments so far in this project are the complete implementation of the ER-GT tool and the preliminary design for the representation for active behavior. The implementation of the enhanced tool and the complete design for the active behavior representation is in progress. Iteration of the design needs to be done based on the prototype with user feedback.

2 Overall design of the Active Database design system

We compare our work with the tool design at Lawrence Berkeley Labs (LBL) called ERDRAW. The ERDRAW tool can be used to create an Extended Entity Relationship diagram (EER diagram) graphically. The EER diagram created can be stored in a.erd file. The .erd file contains all the required information about the EER diagram. This diagram can be displayed back and modified using ERDRAW.

In ERDRAW, relationship can be regarded as an abstraction of aggregation. Classification is supported in terms of an entity type being a class for which multiple entities (instances) are members. Association in the sense of defining sets of entities is not supported in ERDRAW.

The purpose of the new tool will be to allow the users to add active objects (rules, events and triggers) to the ER diagram graphically. This will allow the user to create a ER^2 diagram as described in [1]. The descriptions about the active objects has to be added to the basic .erd file. This .erd file thus will have two portions, one relating to the EER objects (like entities, relations, attributes and arcs) and another part relating to the active objects (like events, rules and triggers).

In addition to the .erd file, ERDRAW produces a .sdt file which is used by the Schema Translation Tool (SDT) from LBL. The new tool will have to produce a file which will contain the behavior of the active objects specified in the Behavior Specification Language [1]. The SDT will convert the .sdt file into a meta description of the database. Both the meta description of the database and the behavior description file has to be processed to create a description of the active database in the target database.

3 Enhancements to LBL ERDRAW tool

The initial architecture for the various tools is given in Figure 1. The initial plan was to directly enhance the Lawrence Berkeley Laboratory (LBL) ERDRAW Version 5.0. A discussion was carried out with the LBL staff and we were informed that they do not release the source code of the software. However, in order to ensure that the enhancements will be smoothly integrated with the existing tool, we needed access to the source code.

The alternative option that was suggested during the discussion was for them to include the

drawing facilities that are needed by the new tool. They will implement the source code to draw the required symbols. Dummy functions will be then provided and then we will add the required code to these functions. For instance, when a new symbol is selected, they will activated our functions.

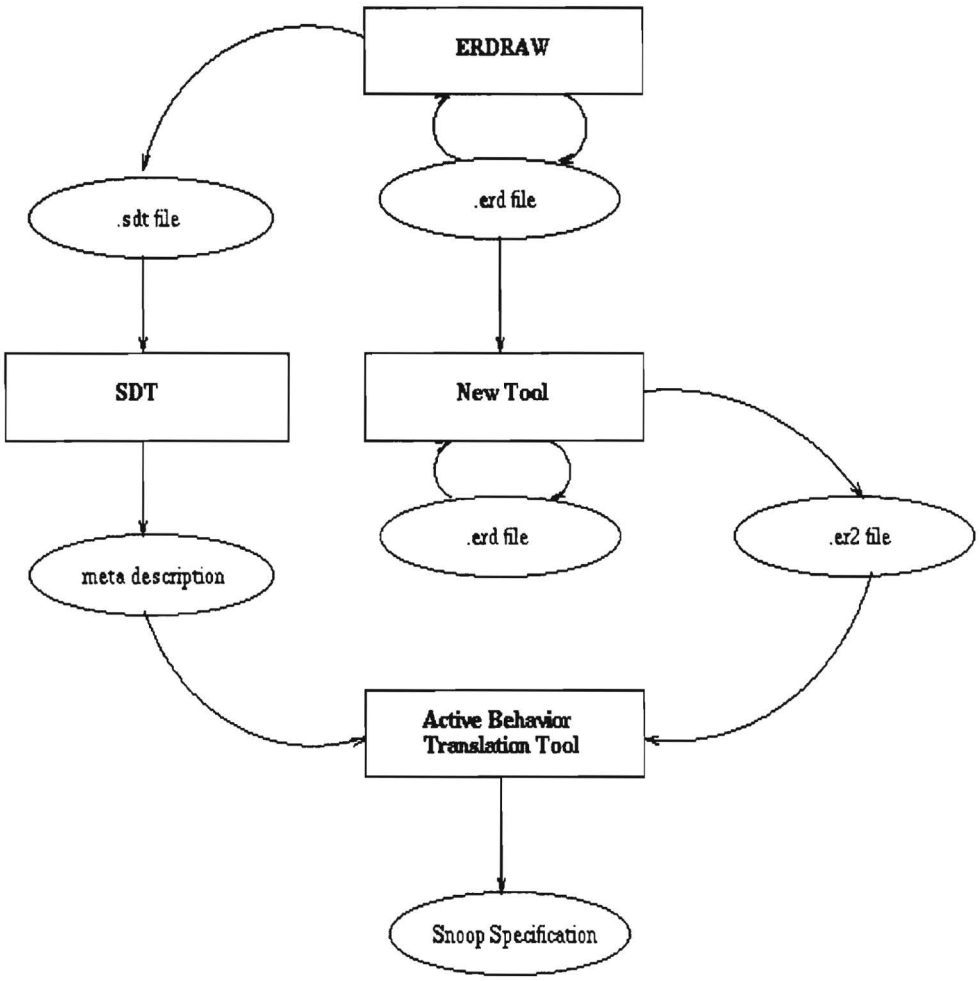

Figure 1

However, this approach presents the following disadvantages:

- There will be an overhead of communication which will inevitably slow down the progress.

- The symbols for ER^2 diagram [1] are well-defined. However, Snoop[2] has amuch richer set of active behavior specification than ER^2. Therefore, we would have to design new symbols to represent the more complex specifications once we look at accommodating complex events in the tool. This design will be done iteratively

with user feedback.

Therefore, it is anticipated that changes to the symbols and representation will be done frequently. In order to obtain more accurate user feedback, we would need to have a working prototype. A prototype will allow us to test the ease of use and the clarity of the representation. However, the continuous development of this prototype will impose an excessive burden on the LBL staff.

4 The New Approach

The architecture of the tools in new approach is shown in Figure 2. This approach is to develop our tool called ER-GT. In this case, we would have a fully developed tool for our enhancement. However, the functionality and external interface to ER-GT will be very similar to the LBL ERDRAW. This tool will be fully compatible with the LBL tool in that data files for these two tools can be freely interchanged.

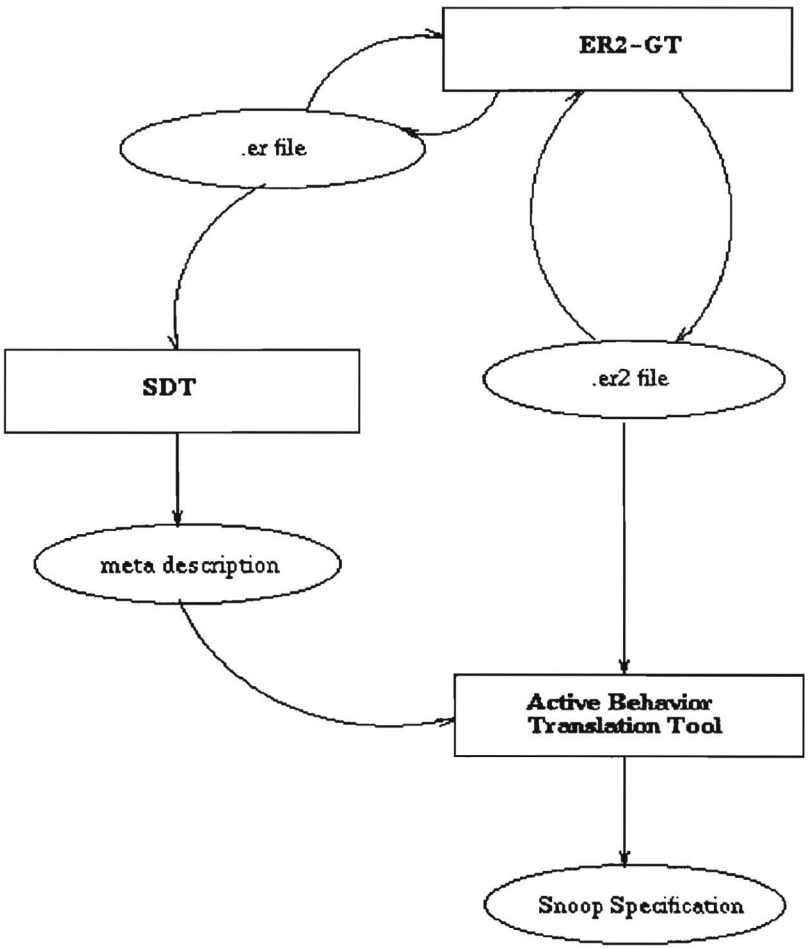

Figure 2

Having our own ER-GT will not only allow enhancements to be made to the new active behavior functionality but to the underlying ER features as well. However, care must be taken when such enhancements so that compatibility with the LBL tool is still maintained.

When the ER-GT tool is complete, we can easily develop working prototypes for the enhanced tool with active behavior specifications.

5 Design of New Representation

As mentioned, Snoop has a much more richer active behavior specification than ER^2. Therefore, we would have to design new symbols to represent the more complex specifications:

In this design, the following must be kept in mind: there is a trade-off between representing as much active behavior information as possible and cluttering the diagram with too much symbols. If too much information is shown in one diagram, the resulting picture will be too confusing to be of any use.

The current design approach is to represent the information in terms of layers. The initial diagram will show the highest level of information. The user can then request more details to be shown by zooming into the area of interest. This will produce a hierarchy of diagrams with each sub-diagrams showing more detailed information.

6 Present State of the Project

At present the ER-GT tool is complete. We can now use the tool to perform the functions of the LBL ERDRAW tool. It is implemented in Motif toolkit. We have done the design of the representation for the active behavior specification. We are in the process of completing the implementation of the active behavior representation facility on ER-GT and enhancing it into ER2-GT. User feedback has to be obtained once the implementation is complete to validate the design. Graphical representation of the Snoop event specification language has also to be considered and the ways of integrating it with the other functionalities of the tool.

References

[1] Asterio Kiyoshi Tanaka. On Conceptual Design of Active Databases. *Ph.D. Thesis, Georgia Institute of Technology*, November 1992.

[2] Sharma Chakravarthy and Deepak Mishra. Snoop: An Expressive Event Specification Language For Active Databases, *Technical Report UF-CIS-TR-93-007, University of Florida*, March 1993.