Data Reverse Engineering: A Historical Survey

Kathi Hogshead Davis
Department of Computer Science
Northern Illinois University
DeKalb, IL 60115
kdavis@cs.niu.edu

Peter H. Aiken
Institute for Data Research
Department of Information Systems
Virginia Commonwealth University
Richmond, Virginia 23284
paiken@acm.org

Abstract

Data reverse engineering is a rapidly growing field which is sometimes misunderstood. In our effort to promote the realization that data reverse engineering is a valuable and essential part of reverse engineering and is now finding its place in software engineering, we present a summary of the history of data reverse engineering. In this paper, a definition of data reverse engineering, a summary of the history of what has been done and published, a statement about the state-of-the-art today, and a comment on the future of DRE is presented.

1. Introduction

The term "data reverse engineering" evolved from the more generic term "reverse engineering." Data reverse engineering techniques consist of a more restrictive subset than those used in reverse engineering. As Elliot Chikofsky stated in his preface to the book *Data Reverse Engineering: Slaying the Legacy Dragon* [CHIK96] "Reverse engineering is a process to achieve understanding of the structure and interrelationships of a subject system. It is the goal of reverse engineering to create representations that document the subject and facilitate our understanding it – what it is, how it works, and how it does not work. As a process, reverse engineering can be applied to each of the three principal aspects of a system: data, process, and control. Data reverse engineering concentrates on the data aspect of the system that is the organization. It is a collection of methods and tools to help an organization determine the structure, function, and meaning of its data." It is the restriction of data reverse engineering to the data portion of a software system that makes it a complex and most interesting activity.

Data reverse engineering (DRE) grew up through two different (and for the most part exclusive) communities: 1) the database community and 2) the software engineering community. The way DRE was performed and which aspects of DRE were emphasized differed as much as the focus of the two communities differed. To illustrate, we can use the technique of "figure/ground" shown in Figure 1. What do you see: Vase or faces--Data or software?

Figure 1: Data or Software?

Over the years, the research and publications in DRE by both communities has been mainly in three areas as illustrated in Figure 2.

- DRE translation and methodologies algorithms
- DRE tools
- The DRE of specific applications and experiences in DRE

Proceedings of the Seventh Working Conference on Reverse Engineering (WCRE'00)
1095-1350/00 $10.00 © 2000 IEEE

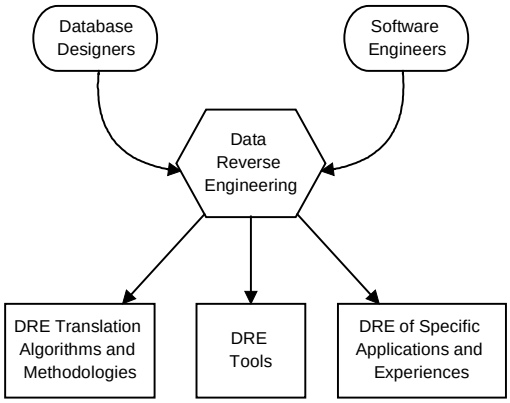

Figure 2: Data Reverse Engineering Publications come from Database Designers and Software Engineers

We need to make a disclaimer before proceeding with this paper. The papers and authors listed are not meant to be a complete list—just a representative example of each topic discussed.

2. Definition of Data Reverse Engineering

Jean-Luc Hainaut and his fellow authors in "Contribution to a Theory of Database Reverse Engineering" [HAIN93] define *database reverse engineering* as "identifying the possible specification of a specific database implementation and possibly also identifying how the implementation got to be what it is". However, this definition is too restrictive for *data* reverse engineering since many legacy systems do not utilize a database management system. Previously, *data reverse engineering* has been defined as

> "the use of structured techniques to reconstitute the data assets of an existing system" [AIKE96].

To further understand this definition, we need to look at the terms structured techniques, data assets, and reconstitute. *Structured techniques* is defined by Pressman as "model building techniques involving model construction and analysis of the existing situations and proposed solutions prior to the actual system development" [PRES93]. Structured techniques are extremely important in DRE because it is the discipline of these techniques that makes the process of DRE economically viable.

The problem with data being termed an asset is that not all organizations view data as such (although this seems to be changing). As defined by Aiken in [AIKE96], *data assets* are useful or valuable data. Data assets might be student information, patient history, billing, or simply a mailing list of potential customers.

The term *reconstitute* adds the implication to the definition of DRE that the process incorporates something of value to the original data assets to make them more useful. "Thus, data reverse engineering can be regarded as adding value to the existing data assets, making it easier for organizations to use and more effective as a tool" [AIKE96].

3. Why DRE?

At the Data Reverse Engineering Workshop in Zurich in March 2000, Jean-Luc Hainaut presented the following list of reasons why we practice DRE [HAIN00]:

- Knowledge acquisition in system development
- System maintenance
- System reengineering
- System extension
- System migration
- System integration
- Quality assessment
- Data extraction / conversion
- Data Administration
- Component reuse

The need for knowledge acquisition in the system development process is prevalent during the reverse engineering of legacy systems. As we all know, legacy systems are usually poorly documented jumbles of unintelligible code that really must be understood in order to proceed correctly with new systems development. As a part of the reverse engineering process the data, be it flat files or a database, must be mastered. DRE can greatly assist in this process no matter the reason for the knowledge acquisition-whether it be system maintenance, reengineering, extension, migration, or integration.

Another nice thing that DRE assists us in performing is quality assessment. We can use the techniques involved to determine the quality of vendor databases [BLAH98]. DRE can also assist us in converting from any current data environment (flat files, databases, etc.) to new database technology. This

is desperately needed with the rapid changes in database technology.

The last two feature of the list of reasons to perform DRE, allow companies to correctly and efficiently manage their data. As we use the data more and more as information, we must be able to perform data administration and component reuse easily and pragmatically.

4. A Historical Survey of DRE

In this section, we present a synopsis of the history of DRE. We have included what we consider significant topics that were presented in the literature over that last twenty-five years. We hope that the reader gets a sense of the transformation that DRE has undergone with the purpose being to understand DRE as it is today.

4.1 In the beginning

DRE actually started manually long before anything was published. We know of one such project that began one of the authors' journeys into DRE. A university was attempting to consolidate their student information system files (some 36) into a database. The only "tool" that was available when the project started in 1975 was a text editor called "WYLBUR". The team assigned the task of getting a handle on the student data created a WYLBUR dataset in a table format. The columns of the table were used to represent the thirty-six distinct files at the university that contained student information. In the first column, the name of every field in the thirty-six files was placed. In cell where the field intersected the file, a number was placed that represented the physical length of the field as it resided in the corresponding file. (Social security number appeared in three different lengths: 1) nine digits, 2) eleven characters (dashes included), and 3) five PACKED decimal digits.

After putting all the fields from all the files into the WYLBUR dataset, over 1500 fields were listed. At the end of four agonizing years of discussing each and every field looking for commonality, the final number of fields was 634. It was at this point that the author decided that there MUST be an easier way to perform this task—and if not would make one.

In searching for an easier technique in the late 1970s and early 1980s, we discovered that, after the creation of the relational data model by E. F. Codd [CODD70], there was a flourish of DRE related research performed and published. At first the DRE publications focused working with relational database concepts. For example, Silva and Melkanoff wrote "Methodology for Helping Discover Dependencies in Relations" [SILV81]. Casanova, Fagin, and Papadimitriou were working on "Inclusion Dependencies and their Interaction with Functional Dependencies" [CASA82].

4.2 Appearance of translation algorithms

The next phase in the history of DRE, still in the early to mid 1980s, was the introduction of translation algorithms. Melkanoff and Zaniolo worked on obtaining entity-relationship diagrams from relations in "Decomposition of Relations and Synthesis of Entity-Relationship Diagrams" [MELK80]. Entity-relationship diagrams were (and still are) a very popular result of DRE translation algorithms. Another example was published by Dumpala and Arora in "Schema Translation using the Entity-Relationship Approach" [DUMP83].

After the translation algorithms using the relational data model as a basis, came the translation algorithms from flat file (or called conventional systems). Casanova and de Sa wrote "Designing Entity-Relationship Schemas for Conventional Information Systems" [CASA83]. Klug tackled enterprise schemas in "Entity-Relationship views over Uninterpreted Enterprise Schemas" [KLUG80]. Nilsson and also Davis in two separate papers took on COBOL in "The Translation of a COBOL Data Structure to an Entity-Relationship type Conceptual Schema" [NILS85] and "A Methodology for Translating a Conventional File System into an Entity-Relationship Model" [DAVI85].

4.3 Methodologies appearing in late 1980s

DRE methodologies differed from translation algorithms in that they tended to be applied to more than one beginning data model whereas a DRE translation algorithm was based upon only one type of data model as input. An example of this type of research was published in "Abstracting Relational and Hierarchical Data with a Semantic Data Model" by Navathe and Awong [NAVA87]. In this paper an algorithm is presented that shows a simple transformation of the input data and does not reveal the actual intent of the data itself.

There is always an exception, in "A Method for Translating Relational Schemas into Conceptual Schemas" Johannesson and Katalin [JOHA89] presented a methodology using just the relational data model as input.

When the methodologies appeared in the late 1980s, the ER model was well established as a conceptual modeling tool. Therefore, many of the methodologies used the ER model as the result of the DRE process. For example, Davis and Arora wrote

"Converting a Relational Database Model into an Entity-Relationship Model" [DAVI87]. This work was just a subset of their methodology to data reverse engineer heterogeneous legacy systems.

4.4 Gathering information with DRE

Also in the late 1980s we saw of the concept of gathering information from data. In "An Approach to Analyzing the Information Content of Existing Databases" by Boulanger and March [BOUL89], the authors presented a means of DRE that accessed existing databases to allow use the data's information content effectively.

4.5 Comparison and critiques of DRE translations and methodologies

In the late 1980s, Markowitz and Shoshani wrote "On the Correctness of Representing Extended Entity-Relationship Structures in the Relational Model" [MARK89]. This paper presented a thorough and theoretically sound process; however, they did not address the poor relational database designs that exist in reality.

Early in the 1990s, in "An approach to Eliciting the Semantics of Relational Databases" Fonkam and Gray [FONK92] pointed out the strengths and weakness of three previous DRE algorithms for converting the relational data model into the entity-relationship model: [DAVI87], [NAVA87], and [JOHA89]. The authors then continued to present a new algorithm from pieces of the predecessors. Their algorithm included possible views and a prompted interactive user input process to convert a relational data model into an entity-relationship model.

4.6 Enter the software engineers and DRE terminology is born

Finally in the early 1990s the software engineers appeared on the publishing scene in the area of DRE. Chikofsky gave support to DRE in "Untying the Spaghetti: An Expert Picks the Tools" [CHIK92]. And with Cross, Chikofsky established a taxonomy for DRE as well as reverse engineering in general in "Reverse Engineering and Design Recovery: A Taxonomy" [CHIK90].

With the taxonomy published, more papers came in the early 1990s with definitions of terms and explanation of algorithms. Besides discussing terminology, Hainaut in "Database Reverse Engineering: Models, Techniques, and Strategies" [HAIN91] presented the essential need for user interaction in DRE.

4.7 More on DRE of relational databases

In the early 1990s, there was a plethora of papers describing DRE around the relational data model. For example, "Reverse Engineering of Relational Databases: Extraction of an EER Model from a Relational Database" was written by Chaing, Barron and Storey [CHAI94]. Markowitz and Makowsky discussed "Identifying Extended Entity-Relationship Object Structure in Relational Schemas" [MARK90]. Also Shoval and Shreiber were working on DRE research as described in "Database Reverse Engineering: From the Relational to the Binary Relationship Model" [SHOV95]. Continuing their work, Fonkam and Gray wrote "An Approach to Eliciting the Semantics of Relational Database" [FONK92].

4.8 DRE of databases other than relational

Also in the early 1990s, publications included other database management systems input to DRE translations. In "Software Reverse Engineering from a Currently Existing IMS Database into an Entity-Relationship Model", Winans and Davis [WINA90] discuss the algorithm they developed to use IMS as input.

4.9 Spreading the word on DRE

The word was spreading about reverse engineering in general and DRE in specific in the early 1990s. In 1994, Diane Crawford, Executive Editor of Communications of the ACM, wrote in an issue dedicated to reverse engineering:

> "Reverse Engineering, or how your technology investment can grow, bend or respond to changing business requirements. It's not about buying new software components, it's about understanding existing components and remolding them to fit new demands"

And in the same issue, Premerlani and Blaha in "An Approach for Reverse Engineering of Relational Databases" [PREM94] presented an outline of a practical, informal, flexible, and iterative approach to extracting OMT conceptual models from existing relational databases. The algorithm worked for poorly designed databases.

The Department of Defense was also showing interest in DRE as shown by the paper "DoD Legacy Systems: Reverse Engineering Data Requirements" written by Aiken, Muntz, and Richards [AIKE94]

4.10 DRE tools, tools and more tools

In the mid 1990s, tools for DRE were highly publicized. Three such tools for translating COBOL files are described in

- "AUGUST-II: A Tool for Step-by-Step Data Model Reverse Engineering" by Davis [DAVI95]
- "DB-MAIN: A programmable CASE Tool for Database Applications Engineering" by Hainaut [HAIN95]
- "Deriving a Logical Data Model for a System Using the RECAST Method" by Edwards and Munro [EDWA95]

Two tools were also presented in this time for translating relational databases in

- "Reverse-DBMS for Windows" Chen & Associates [CHEN94]
- "A Knowledge-based System for Performing Reverse Engineering of Relational Database" by Chiang [CHAI95]

4.11 Classification and categorization of DRE tools

With the proliferation of software engineering and reverse engineering tools, as well as DRE tools, the time (mid 1990s) came to classify and categorize them. Included in "Software Engineering Tool Classification" by Sharon [SHAR93], were DRE tools. The very large report put out by the Software Technology Support Center called "Reengineering Tools Report" by Sittenauer and Olsem [SITT94] covered almost every reengineering tools known at the time including those used exclusively for DRE.

4.12 Enter object-oriented into the world of DRE

In the late 1990s with the push of object-oriented programming and design, came DRE from object-oriented systems. In "Scenarios for the Identification of Objects in Legacy Systems" by Wiggerts, Bosma, and Fielt [WIGG97], the authors researched the possibilities of discovering objects in legacy systems. Three different strategies were presented: function driven, data driven, and object driven objectification.

4.13 On to repositories

As repositories came onto the scene, naturally research was published about DRE in and around them. From the repository technology came the "mining of data". "Reverse Engineering by Mining Dynamic Repositories" by Dayani-Fard and Jurisica [DAYA98] discusses DRE as it pertains to information retrieval and knowledge-mining techniques as it pertains to reverse engineering of legacy systems.

4.14 At the end of the millenium

DRE was very prevalent during the Y2K work that was done at the end of the millenium. Currently, DRE is assisting in various areas

- analysis of legacy systems [HENS00]
- evaluation of packages [BLAH98]
- test planning [CHIK00]
- extracting of business rules [FU 00]

4.15 And the DRE of specific applications

As with story above about the experience with the DRE of the university data, and throughout all the research cited above, there are examples using specific applications in the areas of

- Government [AIKE94]
- Health Care Industry [HENS00]
- Business Applications [AKOK00]
- University Student Information [DAVI85]
- Finance [DUNC00]

5. The state-of-the-art of DRE today

As a result of the history of DRE, Chikofsky put forth a summary of DRE as "the convergence of reverse engineering, database, data design, and information engineering fields" [CHIK00].

Current research is continuing in the area of translation of legacy systems [HENS00]. An algorithm has been presented to translate network databases through DRE [AKOK00]. The main focus of the translation algorithms seem to follow the DRE process stated by Blaha in [BLAH98] of

- Implementation recovery
- Design recovery
- Analysis recover

DRE is also being used to extract constraint business rules that are unknown and hidden within legacy systems [GU 00]. This is proving to be very valuable in the area of system maintenance.

Practical and useful tools within the area of DRE seem to be extremely limited. DB-MAIN is one that is continuing to be productive [HENR00].

6. The future of DRE

In the area of DRE, most problems have at least been identified. We have techniques for successfully performing DRE. However, we need more useful tools especially for large-scale projects [HAIN00].

What really remains to be done can be summed up in one word: "education". We need to show the practitioners that DRE is useful and practicable. In the keynote address at the Workshop on Data Reverse Engineering in Zurich March 2000, Hainaut included the following points in "What remains to be done"

- Training
- Developing popular and powerful CARE tools
- Making tools and methods scalable
- Refining heuristics (less noise, fewer missing constructs)
- Generalizing to the system level problem: how to reverse engineer the whole IS?
- Developing techniques for reengineering legacy systems into distributed components architectures

Towards educating the populace in the area of DRE, a DRE organizational meeting was held at STEP 99 conference in August 1999 in Pittsburgh, PA. Hosted by the IWCASE association. Also, a Workshop on Data Reverse Engineering was included in the EuroRef, Seventh Reengineering Forum (Europe) as part of the Reengineering Week 2000 festivities. This workshop was a meeting of many of the leading DRE researchers. The workshop was set up to allow for discussion into the state and direction of the growing field of DRE. Future meetings of DRE researchers are being planned –we encourage the reader to keep a lookout for announcements.

7. Conclusion

The area of DRE continues to grow rapidly. The Y2K problem assisted in the software community gaining the knowledge that the data within organizations are valuable and within legacy systems essentially not understood. If we have learned anything over the last twenty-five years, it is that data reverse engineering needs to be regarded as an essential part of software engineering. Correctly done, DRE can and does benefit organizations in the understanding of their current legacy systems, in the migration of their systems to new technology, and in the quality assessment of new software systems.

8. References and bibliography of DRE papers.

The majority of papers written on data reverse engineering have been published in either of the following two conferences: the Working Conference on Reverse Engineering or the International Entity-Relationship Conference (now called the International Conference on Conceptual Modeling).

[AIKE94] – Aiken, P., A. Muntz, and R. Richards, "DoD Legacy Systems: Reverse Engineering Data Requirements", *Communications of the ACM*, May 1994, 37(5):26-41.

[AIKE96] – Aiken, P., *Data Reverse Engineering: Slaying the Legacy Dragon*, McGraw-Hill, 1996.

[AIKE98] – Aiken, Peter H., "Reverse Engineering of Data", *IBM Systems Journal*, Volume 37(2), 1998.

[AKOK00] – Akoka, Jacky and Isabelle Comyn-Wattiau, "Migration Schema Mapping vs. Reverse Engineering of Network Databases", *Data Reverse Engineering Workshop*, EuroRef, Seventh Reengineering Forum, Reengineering Week 2000, Zurich, Switzerland, March 2000.

[ANDE94] – Andersson, Martin, "Extracting an Entity-Relationship Schema from a Relational Database Through Reverse Engineering", *Proceedings of the Thirteenth International Conference on Entity-Relationship Approach*, 1994.

[BACH89] – Bachmann, Charles W., "A Personal Chronicle: Creating Better Information Systems, with Some Guiding Principles", *IEEE Transactions on Knowledge and Data Engineering*, 1,1, March 1989, pages 17-32.

[BLAH97} – Blaha, Michael, "Dimensions of Database Reverse Engineering", *Proceedings of the Fourth Working Conference on Reverse Engineering*, October, 1997, paged 176-183.

[BLAH98] – Blaha, Michael, "On Reverse Engineering of Vendor Databases", *Proceedings of the Fifth Working Conference on Reverse Engineering*, October 1998, pages 183 -190.

[BLAH99a] – Blaha, Michael, "The Case for Reverse Engineering", *IT Pro*, March/April 1999, pages 35 – 41.

[BLAH99b] – Blaha, Michael, "How to Recognize Database Winners and Losers", *IT Pro*, May/June 1999, pages 20 – 25.

[BLAH99c] – Blaha, Michael, "Reverse Engineering for Software Quality", *Software Technology and Engineering Practice (STEP 99)*, Pittsburgh, August 1999.

[BOUL89] – Boulanger, Danielle and Salvatore T. March, "An Approach to Analyzing the Information Content of Existing Databases", *Database*, 1989 pages 1-8.

[CASA83] – Casanova, M.A. and Jose Eduardo Amaral de Sa, "Designing Entity-Relationship Schemas for Conventional Information Systems", *Proceedings of the Third International Conference on Entity-Relationship Approach*, 1983.

[CHEN94] – Chen & Associates, "Reverse-DBMS (Access 2.0) for Windows Reference Manual Version 3.0", 1994.

[CHIA94] – Chiang, R.H.L, T.M. Barron, and V.C. Storey, "Reverse Engineering of Relational Databases: Extraction of an EER Model from a Relational Database", *Data & Knowledge Engineering*, 12, 1994.

[CHIA95] – Chiang, Roger H.L., "A Knowledge-Based System for Performing Reverse Engineering of Relational Database", *Decision Support Systems*, 13 (1995) pages 295-312.

[CHIK90] – Chikofsky, E. and J. Cross, "Reverse Engineering and Design Recovery: A Taxonomy", *IEEE Software*, January 1990, 7(1):13-17.

[CHIK92] – Chikofsky, Elliot, "Untying the Spaghetti: An Expert Picks the Tools", *Datamation*, April 15, 1992.

[CHIK96] – Chikofsky, Elliot, "The Necessity of Data Reverse Engineering", Preface to *Data Reverse Engineering: Slaying the Legacy Dragon*, McGraw-Hill, 1996, pages xiii – xvi.

[CHIK00] – Chikofsky, Elliot and Kathi Hogshead Davis, Preface to the proceedings of the *Workshop on Data Reverse Engineering*, Zurich, Switzerland, March 2000.

[CODD70] – Codd, E. F., "A relational model of data for large shared data banks", *Communications of the ACM*, 13(6), 1970, pages 377 – 387.

[COMY96] – Comyn-Wattiau, Isabelle and Jacky Akoka, "Reverse Engineering of Relational Database Physical Schma", *Proceedings of the International Conference on Entity-Relationship Approach*, 1996, pages 372-391.

[DAVI85] – Davis, Kathi Hogshead and Adarsh K. Arora, "A Methodology for Translating a Conventional File System into an Entity-Relationship Model", *Proceedings of the Fourth International Conference on Entity-Relationship Approach*, pages 148-159.

[DAVI87] – Davis, Kathi Hogshead and Adarsh K. Arora, "Converting a Relational Database Model into an Entity-Relationship Model", *Proceedings of the Sixth International Conference on Entity-Relationship Approach*, 1987.

[DAVI94] – Davis, Kathi Hogshead, "August-II: A Software Reverse Engineering Tool that Produces a Flexible Conceptual Data Model", *The Fourth Reengineering Forum*, Victoria, B.C., Canada, September 1994.

[DAVI95] – Davis, Kathi Hogshead, "August-II: A Tool for Step-by-Step Data Model Reverse Engineering", *Proceedings of the Second Working Conference on Reverse Engineering*, July 1995, pages 146 – 155.

[DAVI96] – Davis, Kathi Hogshead, "Combining a Flexible Data Model and Phase Schema Translation in Data Model Reverse Engineering", *The Third Working Conference on Reverse Engineering*, Monterey, CA, November 1996.

[DAVI99] – Davis, Kathi Hogshead, "Data Revere Engineering: A History – Where we've been and What we've done", *Software Technology and Engineering Practice (STEP 99)*, Pittsburgh, PA, August 1999.

[DAYA98] – Dayani-Fard, H. and I. Jurisica, " Reverse Engineering by Mining Dynamic Repositories", *Proceedings of the Fifth Working Conference on Reverse Engineering*, October 1998, Honolulu, Hawaii, pages 174-182.

[DUMP83] – Dumpala, Surya R. and Sudhir K. Arora, "Schema Translation Using the Entity-Relationship Approach", *Entity-Relationship Approach to Systems Analysis and Design*, 1983.

[DUNC00] – Duncan, Howard and Alan Cooke, "Some Experience in Porting a Relational Database from Concurrent OS32 to IBM DB2", *Data Reverse Engineering Workshop*, EuroRef, Seventh Reengineering Forum, Reengineering Week 2000, Zurich, Switzerland, March 2000.

[EDWA95] – Edwards, Helen M. and Malcolm Munro, "Deriving a Logical Data Model for a System Using the RECAST Method", *Proceedings of the Second Working Conference on Reverse Engineering*, July 1995, pages 126 - 135.

[FAHR95] – Fahrner, C. and G. Vossen, "Transforming Relational Database Schemas into Object-Oriented Schemas According to ODMG-93", *Proceedings of the International Conference on Deductive and Object-Oriented Databases*, Singapore, 1995.

[FONG93] – Fong, J. and M. Ho, "Knowledge-Based Approach for Abstracting Hierarchical and Network Schema Semantics", *Proceedings of the Twelve International Conference on Entity-Relationship Approach*, Dallas, 1993.

[FONK92] – Fonkam, M. and W. Gray, "An Approach to Eliciting the Semantics of Relational Databases", *Advanced Information Systems Engineering Lecture Notes in Computer Science #593*, CAiSE 92.

[FU 00] – Fu, G., X. Liu, J. Shao, S.M. Embury, and W.A. Gray, "Extracting Business Rules from Legacy Information Systems", *Data Reverse Engineering Workshop*, EuroRef, Seventh Reengineering Forum, Reengineering Week 2000, Zurich, Switzerland, March 2000.

[HAIN91] – Hainaut, Jean-Luc, "Database Reverse Engineering: Models, Techniques, and Strategies", *Proceedings of the Tenth International Conferenc eon Entity-Relationship Approach*, 1991.

[HAIN93] – Hainaut, J-L., M. Chandelon, C. Tonneau, and M. Jorris, "Contribution to a Theory of Database Reverse Engineering", *Proceedings of the Working Conference on Reverse Engineering*, May 1993, pp. 161-170.

[HAIN95] – Hainaut, J-L., "DB-MAIN: A Programmable CASE Tool for Database Applications Engineering", *Tutorial on Database Reverse Engineering*, presented at both IFORSID'95 and CAiSE95 Conferences, 1995.

[HAIN00] – Hainaut, J-L., "The Nature of Data Reverse Engineering", *Data Reverse Engineering Workshop*, EuroRef, Seventh Reengineering Forum, Reengineering Week 2000, Zurich, Switzerland, March 2000.

[HENR00] – Henrard, J., J-L. Hainaut, J-M. Hick, D. Roland and V. Englebert, "From micro-analytical Techniques to Mass Processing in Data Reverse Engineering –The Economic Challenge", *Data Reverse Engineering Workshop*, EuroRef, Seventh Reengineering Forum, Reengineering Week 2000, Zurich, Switzerland, March 2000.

[HENS00] – Hensley, Jeffrey and Kathi Hogshead Davis, "Gaining Domain Knowledge while Data Reverse Engineering: An Experience Report", *Data Reverse Engineering Workshop*, EuroRef, Seventh Reengineering Forum, Reengineering Week 2000, Zurich, Switzerland, March 2000.

[JI 92] – Ji, Wenguang, C. Robert Carlson, and David Dreyer, "An Algorithm Converting Relational Schemas to Nested Entity-Relationship Schemas", *Proceedings of the Eleventh International Conference on Entity-Relationship Approach*, 1992 pages 231-246.

[JOHA89] – Johannesson, P and Katalin Kalman, "A Method for Translating Relational Schemas into Conceptual Schemas", *Proceedings of the Eighth International Conference on Entity-Relationship Approach*, 1989.

[KALM89] – Kalman, Katalin, "Implementing and Critique of an Algorithm which Maps a Relational Database into a Conceptual Model", *1989 SYSLAB Working Paper*, 1989.

[KLUG80] – Klug, Anthony, "Entity-Relationship Views over Uninterpreted Enterprise Schemas", *Entity-Relationship Approach to Systems Analysis and Design*, 1980.

[LING88] – Ling, Tok Wang, "External Schemas of Entity-Relationship Based Data Base Management Systems", *Proceedings of the Seventh International Conference on Entity-Relationship Approach*, 1988, pages 495-508.

[MARK89] – Markowitz, Victor M. and Arie Shoshani, "On the Correctness of Representing Extended Entity-Relationship Structures", *SIGMOD 89*, 1989.

[MARK90] – Markowitz, Victor M., and Johann A. Makowsky, "Identifying Extended Entity-Relationship Object Structure in Relational Schemas", *1990 IEEE Transactions on Software Engineering*, 1990.

[MELK80] – Melkanoff, Michel A. and Carlo Zaniolo, "Decomposition of Relations and Synthesis of Entity-Relationship Diagrams" *Entity-Relationship Approach to Systems Analysis and Design*, 1980.

[NAVA87] – Navathe, S and A. Awong, "Abstracting Relational and Hierarchical Data with a Semantice Data Model", *Proceedings of the Sixth International Conference on Entity-Relationship Approach*, 1987.

[NILS85] – Nilsson, Erik G., "The Translation of COBOL Data Structure to an Entity-Relationship Type Conceptual Schema", *Proceedings of the Fourth International Conference on Entity-Relationship Approach*, 1987, pages 170-177.

[PREM94] – Premerlani, W. and M. Blaha, "An Approach to Reverse Engineering of Relational Databases", *Communications of the ACM*, May 1994, 37(5):42-49, 134.

[PRES93] – Pressman, R., *Software Engineering: A Practitioner's Approach* (3rd Edition), New York, McGraw-Hill, 1993.

[SHAR93] – Sharon, David, "Software-Engineering Tool Classification", *IEEE Software*, September 1993, pages 106-109.

[SHOV95] – Shoval, P. and N. Shreiber, "Data Reverse Engineering: From the Relational to the Binary Relationship Model", *Data & Knowledge Engineering*, 10, 1993.

[SILV81] – Silva, A.M. and A. Melkanoff, "A Method for Helping Discover the Dependencies of a Relation", *Advances in Database Theory*, 1981.

[SITT94] – Sittenauer, C., M. Olsem, and D. Murdock, "Reengineering Tools Report", *Software Technology Support Center*, Ogden ALC/TISE, Hill AFB, Utah 1994.

[SOUT96] – Soutou, Christian, "Extracting N-ary Relationships through Database Reverse Engineering", *Proceedings of the International Conference on Entity-Relationship Approach*, 1996, pages 392-405.

[SOUT98] – Soutou, Christian, "Inference of Aggregate Relationships through Database Reverse Engineering", *Proceedings of the Seventeenth International Conference on Entity-Relationship Approach*, 1998, pages 135-149.

[SPRI90] – Springsteel, F. and C. Kou, "Reverse Data Engineering of ER Designed Relational Schemas", *1990 Proceedings of the International Conference on Databases, Parallel Architectures, and their Applications*, 1990.

[TEOR86] – Teorey, T., D. Yang, and J.P. Fry, "Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model", *ACM Computing Surveys*, Vol 18(2), June 1986.

[TILL98] – Tilley, Scott R., "A Reverse-Engineering Environment Framework", *Software Engineering Institute Technical Report*, (CMU/SEI-98-TR-005), 1998

[TRYT00] – Trythall, Steve and Sian Hope, "Model Mapping: Tools and Method", *Data Reverse Engineering Workshop*, EuroRef, Seventh Reengineering Forum, Reengineering Week 2000, Zurich, Switzerland, March 2000.

[WIGG97] – Wiggerts, Theo, Hans Bosma, and Erwin Fielt, "Scenarios for the Identification of Objects in Legacy Systems", *Proceedings of the Fourth Working Conference on Reverse Engineering*, October 1997, Amsterdam, The Netherlands, pages 24 - 32.

[WINA90] – Winans, John and Kathi Hogshead Davis, "Software Reverse Engineering from a Currently Existing IMS Database to an Entity-Relationship Model", *Proceedings of the Ninth International Entity-Relationship Conference*, Lausanne, Switzerland, pages 345-360.