科技资料

Proceedings of the 10th International Conference on Information Systems

PROCEEDINGS

of the Tenth

International Conference

on

Information Systems

December 4-6, 1989 Boston, Massachusetts

Editors

Janice I. DeGross
University of Minnesota

John C. Henderson
Massachusetts Institute of Technology

Benn R. Konsynski
Harvard University

Sponsored by
The Society for Information Management

and
College on Information Systems, The Institute of Management Sciences

in cooperation with
Special Interest Group on Business Data Processing, Association for Computing Machinery

and
Technical Committee 8: Information Processing, International Federation for Information Processing

On behalf of the Tenth International Conference on Information Systems, the Conference Committee would like to acknowledge and thank the following organizations whose support and financial contributions helped make this Conference possible:

CIO Magazine

Computer Services Corporation/Consulting Group

Comshare, Inc.

Digital Equipment Corporation

Ernst & Young

IBM Corporation

The MAC Group

McKinsey & Company

Mitchell/McGraw-Hill

NYNEX

Price Waterhouse

Texas Instruments, Inc.

Xerox Corporation

FOREWORD

The Tenth Annual International Conference on Information Systems represents an opportunity to reflect on a decade of growth and progress in the field of information systems. This volume of Proceedings provides evidence that our field is vibrant, diverse, rigorous, and relevant, informing both theory and practice. Emerging themes reflect the changing nature of theory and practice in areas such as international issues, social issues, ethics, advanced technologies, and management practice. These Proceedings provide evidence of the growth of the field -- growing numbers of involved individuals, diversity of acceptable research issues and perspectives, and increasing importance to effective management of organizations and industries. The progress in each of the past ten years has been the result of contributions by many; the program for this tenth ICIS is no exception.

Direction for the cumulative tradition of the Conference continues to come from the ICIS Executive Committee which provides the leadership and ongoing management of ICIS. The membership of this committee rotates on an overlayed three-year cycle through the three principal chairpersons for each conference, representatives for the sponsoring societies, plus a committee secretary and a finance chairperson. The members of the Executive Committee over the two year planning horizon for this conference were:

- Maryam Alavi (Program, 1990)
- Niels Bjørn-Andersen (Conference, 1990)
- Christine V. Bullen (The Institute of Management Sciences)
- Gordon B. Davis (Conference, 1988; Executive Committee Chair, 1988-89)
- Jane Fedorowicz (The Institute of Management Sciences)
- Blake Ives (Doctoral Consortium, 1990)
- William R. King (Conference, 1987; Executive Committee Chair, 1987-88)
- Benn R. Konsynski (Program, 1989)
- Charles H. Kriebel (Program, 1987)
- Lewis E. Leeburg (Secretary, Finance)
- Henry C. Lucas (Doctoral Consortium, 1987)
- James L. McKenney (Doctoral Consortium, 1988; Conference, 1989)
- Barbara C. McNurlin (Association for Computing Machinery)
- Margrethe H. Olson (Program, 1888)
- Hans Oppelland (Program, 1990)
- John F. Rockart (Conference, 1989)
- James H. Scott (Society for Information Management)
- E. Burton Swanson (Doctoral Consortium, 1989)

Ultimately, the quality of the conference is dependent on the quality of the papers and panel presentations and the processes that govern their review and selection. In the Fall of 1988, a Call for Papers was mailed to the broadly defined "MIS community" and printed in key journals. The deadline for submission of original research manuscripts was March 31, 1989. As established in previous years, the deadline was strictly adhered to and papers received after that date were not accepted for processing. A total of 189 papers and 36 panel proposals were received for review.

Critical to assuring a high quality Conference Proceedings is the work of the Program Committee. This year, procedures for reviewing papers were similar to those used in the previous year. These procedures placed quality control responsibility directly in the hands of the Program Committee members. There were many individuals involved in paper and panel reviews and the development of the program. To a person, the committee was hard-working and conscientious. They deserve very special recognition:

- Maryam Alavi (University of Maryland-College Park)
- Yannis Bakos (University of California, Irvine)
- Cynthia Beath (University of Minnesota)
- Izak Benbasat (University of British Columbia)
- Robert Bostrom (University of Georgia)
- Mary Culnan (Georgetown University)
- Vasant Dhar (New York University)
- Prabuddha De (University of Dayton)
- Joyce Elam (University of Texas, Austin)

- Jane Fedorowicz (Boston University)
- P. J. Guinan (Boston University)
- Carol Hicks (Georgia State University)
- Chris Kemerer (Massachusetts Institute of Technology)
- Michael Prietula (Carnegie Mellon University)
- Sudha Ram (University of Arizona)
- John Sviokla (Harvard University)
- Joe Walls (University of Southern California)
- Richard Wang (Massachusetts Institute of Technology)

Each paper was distributed to three committee members, so that each member was expected to read as many as 25 papers. In addition, one of the three was designated the "primary reviewer" of the paper. The primary reviewer solicited one or two additional reviews. The individuals who assisted by doing these reviews are also to be commended for their professional work.

A committee member might serve as a primary reviewer for as many as ten papers. This process enables the committee to have more knowledge of the papers and leads to a more informed discussion of each paper at the program review meeting. Each paper was reviewed "blind"; the authors names and affiliations were removed prior to distribution. A subset of the Program Committee reviewed, discussed, and selected the panels.

The Program Committee met at the Harvard Business School on April 27, 28, and 29, 1989. A preliminary "sketch" of the program was developed based on the selected papers and panels. Written reviews were sent to authors of both accepted and rejected papers.

Electronic versions of all the revised, accepted papers were submitted to Jan DeGross at the University of Minnesota for final Proceedings production. Note that the copyright for each paper in the Proceedings is retained by the authors.

Thirty-two papers were accepted, providing for two parallel paper tracks. This year the panel portion of the program was expanded to include nineteen sessions. This represents an explicit strategy to increase the level of participation, as well as broaden the scope of research issues and perspectives that are addressed. Special emphasis was placed on panels that addressed emerging international issues.

Session chairs were nominated by the Program Committee at the meeting. Special consideration was given to Program Committee members. As in past years, a strict policy was enforced that no person could make a presentation in the program more than once (either a paper presenter, panel presenter, session chair, etc.). This rule has worked well in extending participation in the Conference to a broader subset of the MIS community.

The Program Committee would like to offer special thanks to the Harvard Business School for providing accommodations, meeting space, and meals for the Program Committee meetings. Jim McGee and Art Warbelow of the Harvard Business School provided extraordinary data and material management skills. The work in preparing this document from the final submitted versions of the papers was done by Steve Barstad, Jan DeGross, and Margaret Nguyen at the University of Minnesota. Particular thanks are due Jan DeGross for her role in editing these Proceedings. Finally, we would like to thank our administrative assistants, Diane Shapiro at the Harvard Business School and Barbara Finney at the Massachusetts Institute of Technology, for their considerable efforts.

Now, enjoy the Conference!

John C. Henderson and Benn R. Konsynski
Program Co-Chairs

WELCOME FROM THE CONFERENCE CHAIRS

The Boston community of colleges and universities is honored and proud to host the Tenth Annual International Conference on Information Systems.

The information systems field has expanded tremendously both in scope and depth since the first ICIS in 1980. Information technology is ever-changing. The needs of business, and thus the role of the technology, will be very different in the early 1990s than in the early 1980s. Many business schools are engaged today in re-examining their roles and the objectives of management research and curriculum development. In this changing environment, the future of our academic field, never completely certain, continues to change and evolve.

The original purpose of ICIS was to focus on research as the driving force in our ability to make an intellectual contribution, thereby increasing the recognition and respect from academic colleagues in other disciplines, as well as contributing knowledge to those who practice the management of information systems. That is still our primary objective and the conference grows more robust and productive with each passing year. Truly, ICIS is now an established tradition that has been influential in the development and maturing of a world-wide community of information systems academicians.

The conference has grown from approximately 175 participants in Philadelphia in 1980 to over 900 in Minneapolis last year. Almost 80 papers were submitted to the second conference in 1981; 189 paper submissions were received this year. The number of corporate sponsors has increased from two the first year to thirteen for this conference. The structure of ICIS has evolved from the ad hoc group of individuals who initiated the conference to a stable yet adaptive organization. We now have a formal set of policies, an evolutionary manner of organizational change, and an expanding agenda to guide our work towards the future.

Many people have contributed to this growth in many ways -- by serving on various committees; submitting or reviewing papers; being a presenter; acting as session chair or discussant; providing funding; assisting with conference administration; or simply participating as an attendee. This tenth conference is dedicated to all of you who from 1980 to 1989 have helped to build ICIS into a successful and influential forum for information systems research.

As we mark this important tenth conference, we would like to recognize and thank four key organizations for their ongoing support:

- The Society for Information Management which helped financially support the initiation of ICIS and continues to fund the Doctoral Consortium that precedes the Conference, including the costs of conference attendance for the Consortium students.

- The Institute of Management Sciences College on Information Systems which also supported the start-up of ICIS and, over the past few years, has provided financial assistance for a number of doctoral students to attend the Conference.

- The Association for Computing Machinery Special Interest Group on Business Data Processing, which has been an "in cooperation with" sponsor since the first conference and now handles the distribution of the ICIS Proceedings to the public.

- The International Federation for Information Processing (Technical Committee 8: Information Systems), a new "in cooperation with" sponsor which will encourage the international growth of ICIS.

We would also like to gratefully recognize the corporate sponsors of ICIS '89, who are listed in the front of these Proceedings. Each firm contributed $3,000 to assist the Conference. We appreciate their financial support and share with them the recognition of the importance of linkages between information systems researchers and practitioners. A special acknowledgement goes to Digital Equipment Corporation for generously contributing equipment, software, and staff to support electronic messaging among the participants during the conference.

The members of the ICIS Conference Committee for 1989, all of whom enthusiastically took on this major effort, deserve special mention and our sincere thanks:

- John C. Henderson, Program Co-chair
- Benn R. Konsynski, Program Co-chair
- E. Burton Swanson, Doctoral Consortium Chair
- Judith A. Quillard, Planning and Arrangements Chair
- Kathleen Foley Curley, Publicity Chair
- Jane Fedorowicz, Placement Chair
- David K. Goldstein, Consortium Arrangements Chair
- Jerry Kanter, Exhibits Chair
- Janice I. DeGross, Proceedings Editor

On behalf of the entire committee, we welcome you to Boston and the tenth ICIS.

James L. McKenney and John F. Rockart
Conference Co-Chairs

TABLE OF CONTENTS

PANELS

TOWARDS MORE POWERFUL CONCEPTUAL SCHEMA LANGUAGES

Peter Creasy
Key Centre for Software Technology
Department of Computer Science
University of Queensland

ABSTRACT

One of the phases of information systems design methodologies is that of conceptual schema design. This phase involves identifying relevant objects, their properties and propositions involving these objects. These are described in a conceptual schema using a conceptual schema language.

Conceptual schema languages which permit a graphical form are becoming more widely accepted under the motivation of being able to see the data structures that exist. However the semantics are somewhat limited with, for example, few integrity constraints being represented. In this paper we show how conceptual schema languages with a graphical form can be extended to permit a wider class of semantics to be represented.

1. INTRODUCTION

The conceptual schema design phase of information systems design is one of the more important phases. A number of methodologies and languages have been proposed for this phase. In addition, languages which have a graphical form have become increasingly significant in information systems (e.g., Harel 1988). These graphical forms have enabled the user to see the data structures.

Information systems have three perspectives (Olle et al. 1988):

- the data perspective which is concerned with the data to be recorded within an information system.
- the process perspective which is concerned only with processes which operate on this data.
- the behavior (or event-oriented) perspective which is concerned with events which cause the processes to be performed.

It is the first perspective which is of concern in this paper.

An essential part of any knowledge representation system is the representation of facts. As syntactic rules permit facts which are syntactically valid but semantically invalid within the universe of discourse, we additionally need constraint specifications. Correspondingly we distinguish two classes of data types -- the fact encoding construct (a data structure type which is associated with a population of instances) and validation rules. It is the latter structures which are of interest in this paper.

A number of other models/languages have been proposed, in most of which, e.g., IFO (Abiteboul and Hull 1987), the "primary focus in the development...has been on the structural component of the model" (p. 526). This paper proposes not a new language but a method of extending existing languages to include the "integrity-specification component." The method could be applied to any fact-based language. We demonstrate the method using a fact-based language which already can represent a reasonably wide class of constraints.

NIAM (Verheijn and van Bekkum 1982; Nijssen and Halpin 1989) is a methodology with an associated fact-based language with a graphical representation. It is informally described using a few examples. The methodology involves taking a significant set of sentences from the UoD and from these abstracting to obtain the fact types (the information bearing construct).

However there are a number of problems with the language. It uses only a few validation constructs (e.g., mandatory role, uniqueness). While we can represent what we regard as some of the more important constraints, there is still a large class which cannot be represented. In addition, subtype definitions and derivation rules have no formal representation.

The entity-relationship (E-R) model (Chen 1976) is a similar model, although without the methodology of NIAM. It has even fewer validation constructs than NIAM, although many proposals have been made in this area.

Most of the presentations in the integrity constraint area (e.g., Reiter 1988) use first order predicate logic (FOPL), a language unfamiliar to most analysts. Existential graphs (Peirce 1960; Roberts 1973) have been developed in a number of forms which correspond to propositional, first order and higher order logics. The graphs provide a very readable logic representation in graphical form which can be readily integrated with graphical knowledge representation languages. We discuss the first order form which is represented graphically using negation, conjunction and the existential quantifier.

We use the existential graphs to extensively extend the power of these languages to include a graphical first order logic language which permits the subtype definitions, derivation rules and constraints to be formally graphically represented. We show this for NIAM and E-R.

Finally, we discuss an extension to the NIAM methodology which guides analysts/users in the expression of constraints in the extended language.

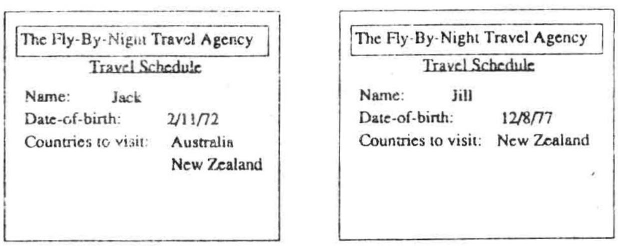
2. NIAM

We informally describe NIAM using a few examples. The methodology involves taking a significant set of sentences from the UoD and from these abstracting to obtain the fact types which, together with the database, is the fact encoding mechanism. The set of sentences should be significant in the sense that it will enable us to determine all constraints that can be represented in NIAM.

For example, let us take a travel agency business. Suppose we have forms such as in Figure 1. The NIAM methodology uses what Nijssen (1986) has called the "cookbook" approach. With this approach the users take familiar examples of facts from their application, for example, *Jack visits Australia*. We need now to take the user from the instance level to the type level. Nijssen suggests the heuristic of telephoning someone and informing him/her of the fact and "in this way the user is forced to make a visual to auditory transformation." With this verbalization process, the sentences are expressed in natural language rather than in a formal artificial language prescribed by "computing experts." There are some semi-formal rules for specifying the sentences, but this helps the users to formalize the problem in their minds.

The user may specify the statement

> Jack is going to Australia.



The Fly-By-Night Travel Agency
Travel Schedule
Name: Jack
Date-of-birth: 2/11/72
Countries to visit: Australia
New Zealand

The Fly-By-Night Travel Agency
Travel Schedule
Name: Jill
Date-of-birth: 12/8/77
Countries to visit: New Zealand

Figure 1: Sample form

However, this is not structured enough. We are relying on the listener to interpret that Jack is a person and Australia is a country. As it is the user with the expert knowledge of the UoD, this domain information must come from the user. If we persuade our user to be a little more explicit, we should receive the message:

> The person with person name Jack will visit the country with country name Australia.

Sentences such as this reveal the deep structure of the sentence. We are listening for verbs, entity categories or types, label categories or types and label instances. Examining sentences in terms of these structures we obtain:

The PERSON	**entity type**
with PERSON-NAME	**label type**
Jack	**label instance**
will-visit	**verb (role)**
the COUNTRY	**entity type**
with COUNTRY-NAME	**label type**
Australia	**label instance**
and	**connective**
was-born-on	**verb (role)**
the DATE	**entity type**
with EUROPEAN-DATE-CODE	**label type**
2/11/72	**label instance**

The user (the expert of the UoD) has been able to take us from a surface structure to a deep structure. From here we proceed to step 2 which involves a quality check on the elementary facts. We ensure that the entities are fully designated and that the facts cannot be split into smaller ones without loss of information.

We carry out the abstraction process at step 3. For this we ignore the label instances and present the remaining entity types, verbs and label types in a graphical form.

From this (and the remainder of a significant sample), we can derive the schema of Figure 2, although strictly it is a subschema. We have also included part of the population of the significant sample, however this is usually only done either for explanatory purposes or to check the schema. The label types (PERSON-NAME and COUNTRY-NAME) are represented as dotted circles. The current extension of the label type PERSON-NAME is Jack and Jill. The entity types (PERSON and COUNTRY) are represented as circles. The current extension of PERSON is those people called Jack and Jill. These entities are non-lexical. We have represented them by arbitrary lexical marker. Often in the diagrams, we use the corresponding lexical representation (e.g., Jack).

The NIAM language is fact-based. It has a graphical representation which is the form generally used for conceptual schema design. The facts (fact type instances) are semantically irreducible associations between entity instances, with each entity instance playing a role in the fact (sentence). The methodology abstracts from facts to give fact types: the fact encoding construct.

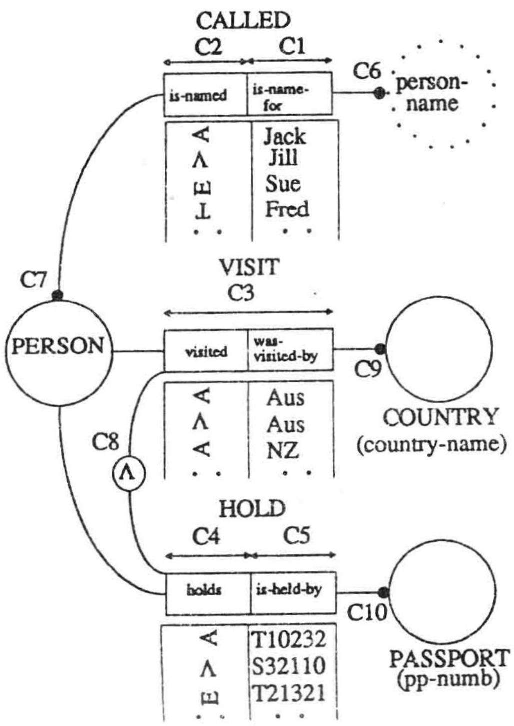


Figure 2: A NIAM Conceptual Schema

Figure 2 shows a NIAM diagram which represents people holding passports and visiting countries. The main constructs are entity types (circles), e.g., PERSON, COUNTRY; roles (rectangles), e.g., visited, was-visited-by, which correspond to verbs of a sentence; fact types (compound rectangles), e.g., VISIT, HOLD; label types (dotted circles), e.g., person-name; and constraints.

A relationship type between entity types is a fact type. Each fact or reference type consists of a number of roles. These describe the part (or role) each entity or label type (to which the role is joined) plays in the fact or reference type. For example, the role played by PERSON in CALLED is "is-named."

Entities are non-lexical and are lexically identified by reference types (e.g., CALLED). These are special fact types which associate entity types with the identifying label type. The references can be abbreviated by writing the corresponding label type in parentheses following the entity. We have shown this for the entity type COUNTRY.

An example of a VISIT fact type instance is <Jill,USA>. We can show such populations as in Figure 2, where the non-lexical entities have been represented by the corresponding lexical representation. This fact can be read as *the person identified by person-name Jill visited the country identified by country-name USA.*

Some of the constraints we can represent in NIAM are intrafact (and intrareference), interfact uniqueness, mandatory role, equality, exclusion and subset constraints. The arrows above the roles represent intrafact or intra-reference uniqueness constraints which are placed on the instances of the role or roles. They indicate that the role or roles uniquely identify a single fact instance, i.e., they specify functional dependencies. In this case constraints C4 and C5 indicate the functional dependencies person $\longleftrightarrow$ passport while C3 indicates person, country $\rightarrow \phi$. Constraints C1 and C2 indicate unique naming for the entity type PERSON.

The constraint C7 is a mandatory role constraint which indicates that if an entity (or label) takes part in any fact (or reference) instance then it must take part in an instance of the fact (or reference) type corresponding to the constraint. The subset constraint C8 indicates that anyone who visits a country must have a passport.

The fact types can be mapped to relational structures using what is essentially a synthesis algorithm. For Figure 2 the resulting relations are VISIT(<u>person , country</u>) and HOLD(<u>person</u> , <u>passport</u>).

We have not mentioned all the features of the language. One which should be mentioned is subtyping, an important part of the language. An example is shown later. It should be noted that it is not necessary to write down the populations, fact type names, role names or constraint names unless they are needed for clarification. This results in a much less cluttered diagram.

3. EXISTENTIAL GRAPHS

Existential graphs were first proposed by Peirce (1960) at the beginning of the century. As a graphical representation of logic they were intended to be easy to learn, read and write. Many other forms have also been proposed (see Gardner 1983). The graphs have few operators, are easy to translate to the Peano-Russell notation and suit the purpose we intend for them.

The graphs have a number of parts:

- Alpha part, equivalent to propositional logic.
- Beta part, equivalent to first order predicate logic.
- Gamma part, equivalent to second, higher order and model logics.

Roberts (1973) has shown completeness and consistency for existential graphs. In particular, Roberts uses Quine's (1955) ML logic system.

We shall informally describe existential graphs and relate them, where appropriate, to the more widely understood propositional and first order logic and the Peano-Russell notation. The Alpha part has only three basic symbols: the sheet of assertion, the cut and the graph. The graphs are laid out on the *sheet of assertion* (SA), an arbitrarily large area, which equates to a model of the universe of discourse. A graph instance is something which asserts a possible state in the universe, viz. proposition. The sheet of assertion is also considered to be a graph, as the blank SA expresses whatever initially holds in the UoD.

Graph instances written on the SA are asserted to be true. By a graph is meant "any sign (symbol) which expresses in a proposition some state in the universe" (Roberts 1973, p. 32). Some examples are shown in Figure 3. It should be noted that the shape of the graphs (whether lines are straight or curved or the shape of the closures) is not important. Figure 3(a) contains two graphs which together state that the propositions represented by Raining and Cold hold, i.e., the conjunction of these holds. Peirce distinguished graph and graph instance. In his terms, there is strictly only one graph, Raining, of which there may be a number of occurrences (graph instances); e.g., we may have Raining repeated a number of times such as in Figure 3(b). In the following we shall use the term graph rather than graph instance, except where we wish to make the distinction.

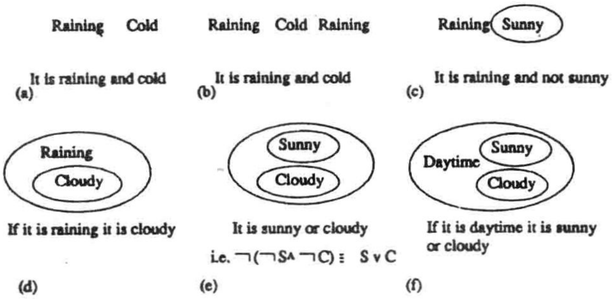


Figure 3: Existential Graphs -- Alpha Part

A single line, with no crossings, can be drawn around a graph (a cut) to indicate the negation of the graph. For example, Figure 3(c) is equivalent to Raining $\wedge$ $\neg$Sunny. For material implication (e.g., Raining $\Rightarrow$ Cloudy, or "if Raining then Cloudy"), the term "scroll" is used. This example is shown in Figure 3(d). The graphs are read from the outside inward (*endoporeutic*). In this example (Figure 3(d)), reading from the outside inward we have the antecedent ("if Raining") and then the consequent ("then Cloudy").

With only negation and conjunction we need these to represent disjunction. While this may appear awkward, we emphasize that the visual representation is important and one can quickly learn to recognize and draw this picture as a disjunction.

The Beta part additionally has a *line of identity*, used to represent an individual in the universe, and a spot which is equivalent to the FOPL predicate, e.g., Cold of Figure 4(b). In the Alpha part, the graph and the (propositional) symbol were one and the same. However, in the Beta part, a graph may consist of a number of symbols, hence the need to distinguish the symbol and the graph. The line of Figure 4(a) indicates the existence of something, e.g., $(\exists x)$.

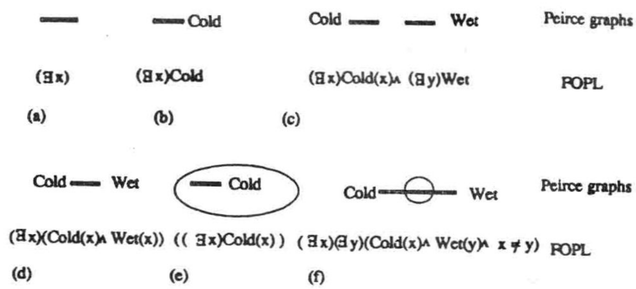


Figure 4: Existential Graphs -- Beta Part

We still have the convention that two graphs drawn on the same SA represent the conjunction of the graphs. Thus Figure 4(c) represents $(\exists x)\text{Cold}(x) \wedge (\exists y)\text{Wet}(y)$. In Figure 4(d), the line of identity between Cold and Wet indicates that the two individuals represented by the ends are identical, i.e., $(\exists x)(\exists y)(\text{Cold}(x) \wedge \text{Wet}(y) \wedge x = y))$ or $(\exists x)(\text{Cold}(x) \wedge \text{Wet}(x))$. This can be extended to n individuals.

A line of identity passing through an empty cut indicates the non-identity of the individuals at the extremities. For example, Figure 4(f) is equivalent to $(\exists x)(\exists y)(\text{Cold}(x) \wedge \text{Wet}(y)\ x \neq y)$. The negation of a ligature indicates the non-identity of the individuals denoted by the extremities of the ligature.

Just as a predicate can have more than one argument, more than one line may be attached to a predicate (symbol). The places of attachment are called *hooks*, and the meaning of the hooks must be understood just as the positions of the predicate arguments in FOPL need to be understood. Figure 5(a) represents *someone is located somewhere*. In this case we arbitrarily decided to consider the hook in front of the predicate to be the person and the other hook to be the place.

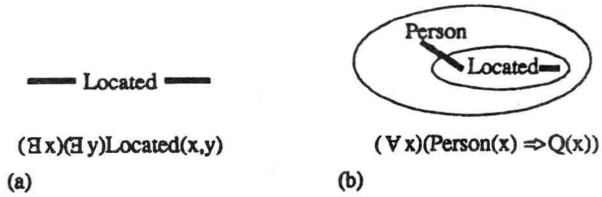


Figure 5: Binary Predicates