

Heft 84

A. Hars, R. Heib, Chr. Kruse, J. Michely, A.-W. Scheer

Concepts of Current Data Modelling Methodologies

- A Survey -

1991

Table of Contents:	Page:
1. Introduction	1
2. Review of Current Data Modelling Methodologies	1
2.1 ARIS	4
2.2 Chen ERM	8
2.3 CIM OSA	12
2.4 ECER	16
2.5 ECRM	19
2.6 EXPRESS	22
2.7 IDEF1	26
2.8 ISOTEC	29
2.9 James Martin	33
2.10 Merise	36
2.11 NIAM	39
2.12 The Relational Model	43
2.13 RM/T	45
2.14 SAM*	48
2.15 SERM	51
3. Conclusion	54

1. INTRODUCTION

This paper originates from the work of the ESPRIT project CODE (Computer Supported Enterprise-Wide Data Engineering). The main focus of the CODE project is the development of a tool-based methodology to support reference models for the creation of enterprise-specific data models.

This publication is based on a paper analysing the theoretical concepts and terminology used in data modelling. [see: Concepts of Current Data Modelling Methodologies - Theoretical Foundations. In: Scheer, A.W. (Hrsg.): Veröffentlichungen des Institutes für Wirtschaftsinformatik. Saarbrücken 1991, Heft 83]

A survey of different data modelling methodologies is presented. The purpose of this survey is not to compare or rank the methodologies but to derive the core components of the methodologies and the essential principles lying behind them.

2. SURVEY OF CURRENT DATA MODELLING METHODOLOGIES

This chapter presents 15 data modelling methodologies. Figure 1 contains the key characteristics of each of the methodologies.

Subsequently each of the methodologies is briefly monitored. A uniform example is modelled with every methodology to better show the differences between the different methodologies.

Imagine the following situation:

The X Company specializes in planning large construction undertakings for any country of the world. They discern different types of construction undertakings, which internally they term "product". Examples for these products are: Power plant (coal), power plant (nuclear), pipeline (oil), airport, highway. X has a set of large international construction companies as clients for which X carries out the planning. A product for a specific company in a specific country is planned by one project group consisting of 3 to 5 employees. The employees are either engineers, technicians, financial analysts or cost accounting people. The X company cooperates with the marketing companies which sometimes carry out costly public relations campaigns for a project. As the marketing companies have all their specialities, sometimes several marketing companies campaign for the same project. Independently of the projects, there is one employee for each customer and one employee for each marketing company which is responsible for all transactions with the company. Some employees are subordinate to others. It is possible that one employee has several

superiors. Each employee has his salary account - identified by an account number and further characterized by the salary and tax of the employee. Each person has a name and a sex. Several decisions within the company depend upon the birthdate of the persons. A person might be reached under several telephone numbers, but each person has exactly one address which consists of ZIP, city and street. Every employee receives a certain salary and has a date of employment. Each project has a certain budget and a certain end date. A marketing campaign also has a fixed budget and is carried out on a certain type of media, e.g. television, radio, newspapers etc.

This example does not completely specify the company X universe of discours. It varies in detail as properties of some objects are given, while those of others are not mentioned. The purpose of this example is to show how different facts are represented in the different models, not to completely describe a certain universe of discourse.

Methodology	ARIS-ERM	CHEN-ERM	CIM-OSA	ECER	ECRM	EXPRESS	IDEF1	ISOTEC	James Martin	Merise	NIAM	Relational Model	RM/T	SAM*	SERM
Criterion															
Knowledge Represent															
- Semantic Network	X	X	X	X	X		X	X	X	X	X	X	X	X	X
- Formal-Logic												X			
- Procedural						X									
Constructs															
Entity type	X	X	X	X	X	Various concepts of programming languages	Indep. entity	Entity type	Entity type	Entity type Relationships	Non-lexical object type	Relation	E-relation	Concepts Associations	Entity type
Relationship	X	X	X	X	X		Depend. entity				Lexical object type		P-relation		Relationship
- Degree > 2	X	X	X	X	X						Role		Catalogue		Entity-relationship type
- Aggregated relationship	X													X	X
- Aggregation	X		X	X	X					X	X			X	X
- Grouping														X	X
- Generalization/Specialization	X		X	X	X		X	X	X		X			X	X
Complexity															
- Chen-notation		X			X									X	
- Min/max-notation	X		X	X	X				X	X	(X)	None	None		(X)
- Other notation							X				X				
Dependencies															
- Existence dependency	X	X	X				X						X		implicitly
- Identification dependency			X				X						X		implicitly
Attributes															
- Single-valued	X		X	X	X		X	X	X	X	X	X	X	X	
- Multi-valued			X		X									X	
- Complex			X		X									X	
Methodology															
- Procedural specification	X		(X)					X	X	X	X		X		
- Integration with other views	X		X					X	X	X					
- Calculus definition			(X)	(X)	(X)							X	X		
- Free-form integrity specification					X		X	X			X	X	X		
- Availability of tool support		X		(X)			X	X	X	(X)	X	X	X	(X)	(X)

Fig. 1: Characteristics of data modelling methodologies

2.1 ARIS

Scheer has developed an Architecture for Integrated Information Systems.¹ This approach is shown in figure 2. The model distinguishes three separate layers:

- The conceptual layer captures issues which are relevant for an enterprise without consideration of implementation requirements. The structure of the model, however, is sufficiently formalized to allow the subsequent transformation with computer support.
- The technical layer represents the business models which have been defined on the conceptual layer as structures which comply with the data processing technologies available for the realization. The technical layer takes requirements which originate from the interfaces of the implementation tools (programming languages or database systems) into account.
- On the implementation layer the concrete physical implementation of the requirements into physical data structures and program code is described. The structure upon this layer depends heavily upon the currently available data processing technologies.

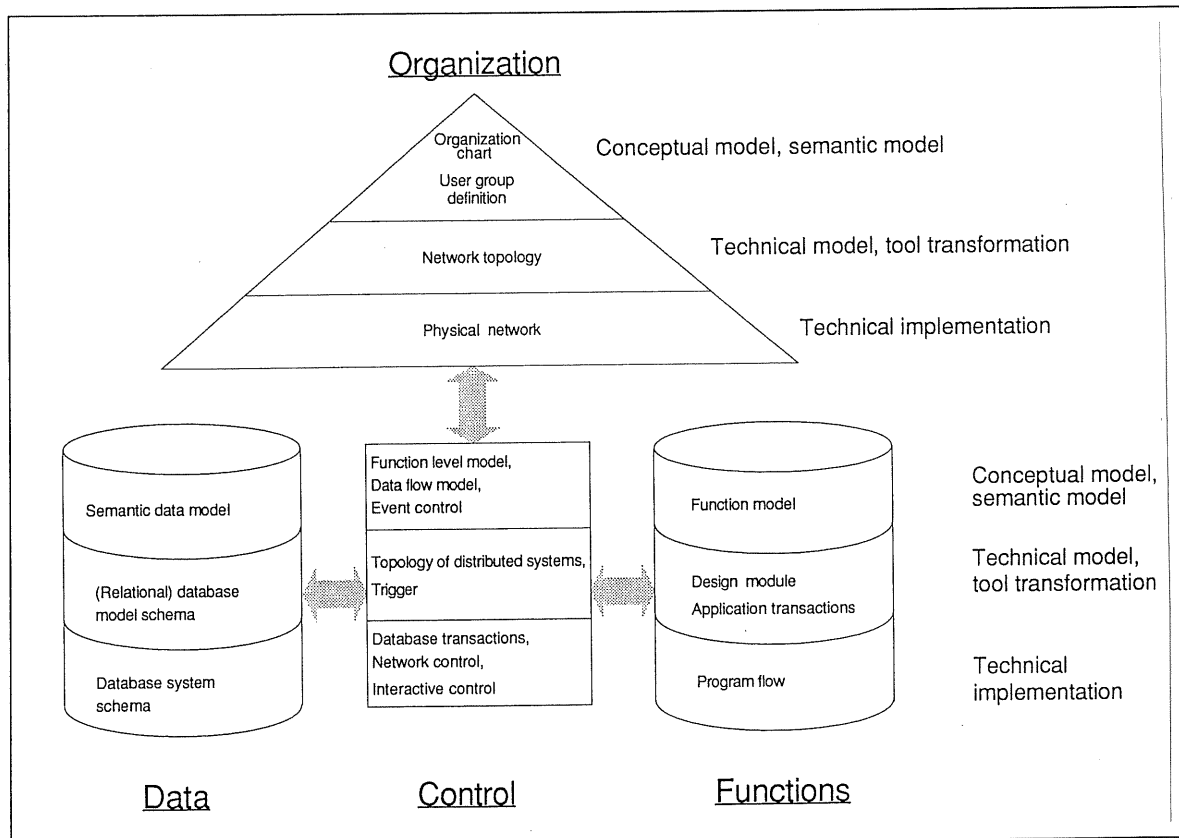


Fig. 2: Architecture of an information system²

The model of an information system can additionally be structured into the following three views and one control block:

- ❑ The data view contains the information objects and their relationships which are relevant for the business. Information objects can be externally or internally created or utilized.
- ❑ The function view contains the procedures which transform information within an information system. On the conceptual layer the activity chains which are relevant for fulfilling the enterprise objectives are described. Know-how about decision models, planning methodologies or organizational procedures is used for creating the model.
- ❑ The organization view focuses on the user and the organizational structure of an enterprise. Models for the design of user interfaces, organigrams and job descriptions are relevant under this view.
- ❑ The control block is the link between the three views. It describes the relationships between function, data and organization views.

This architecture describes the framework for the design of information systems. Each component of an information system can be located within the framework. Each cell of the framework has to be filled with the corresponding model. Within each view, the conceptual model is the initial model.

ARIS provides a meta structure definition of every layer. The data model utilized is an extension of the Chen entity relationship model. Its constructs and relationships are shown in figure 3. It incorporates the aggregated relationship type and restricts to single-valued attributes only. This model has been utilized to build an enterprise-wide data model and has been applied to several industrial enterprises.

The modelling example of figure 4 shows that the project can be defined as an aggregated relationship which shows the existence and identification dependencies from PRODUCT, CUSTOMER, and COUNTRY. There is no separate construct for a grouping relationship, instead this is represented by creating a relationship GROUP ASSIGNMENT and an entity PROJECT GROUP. A construct exists to represent the dependency between ACCOUNT and EMPLOYEE.

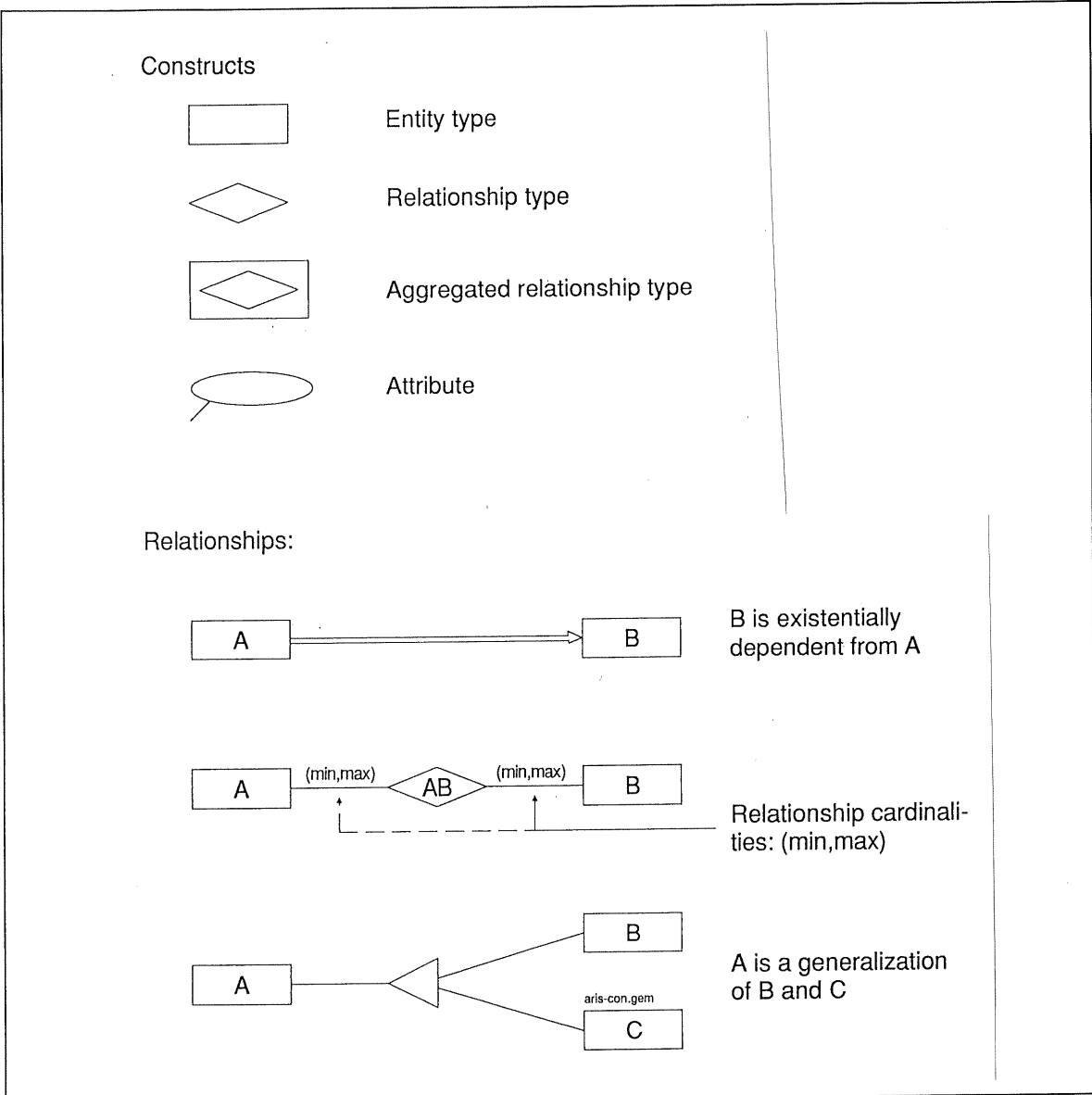


Fig. 3: ARIS constructs and relationships

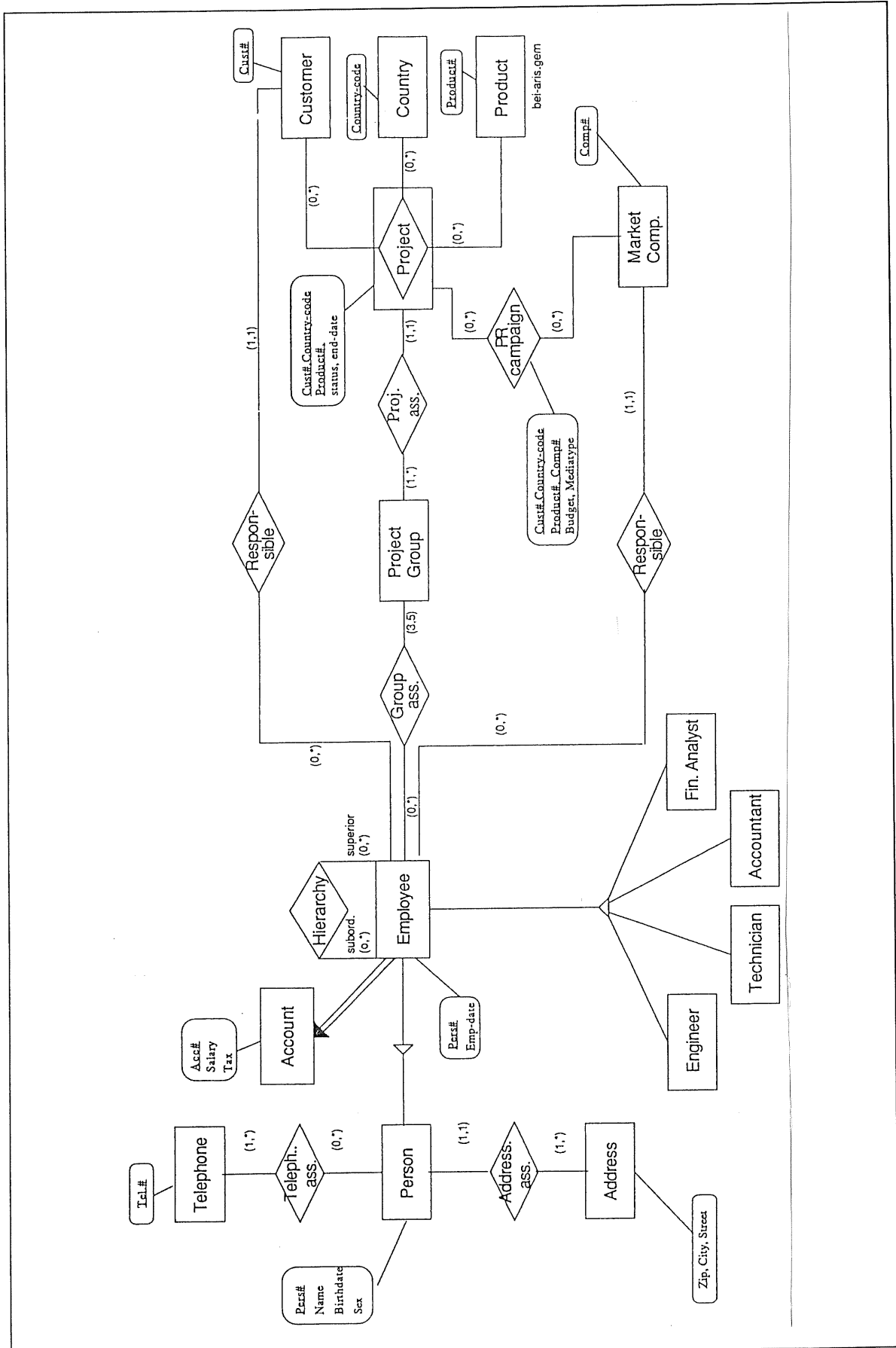


Fig. 4: The modelling example in ARIS

2.2 Chen ERM

The original Entity Relationship approach was first presented by Chen in 1976.³ Since this time the ER approach has been expanded in many ways. The main objective of Chen's ER model was to introduce a framework from which three existing data models (relational, network and entity set) can be derived (see figure 5). Additionally, Chen introduced a special diagrammatic technique, the ER diagram, to support the database design. Especially this diagrammatic technique gave rise to the success of the Entity Relationship Model.

The ERM framework consists of four different levels of data views:

1. Information concerning entities and relationships which exist in the mind of the people.
2. Information structure - organization of information in which entities and relationships are represented by data.
3. Access-path-independent data structure - the data structures which are not involved with search schemes.
4. Access-path-dependent data structure.

The network model mainly concerns the level 4, the relational model concerns level 2 and 3 and the entity set model mainly concerns level 1 which is depicted in figure 5.

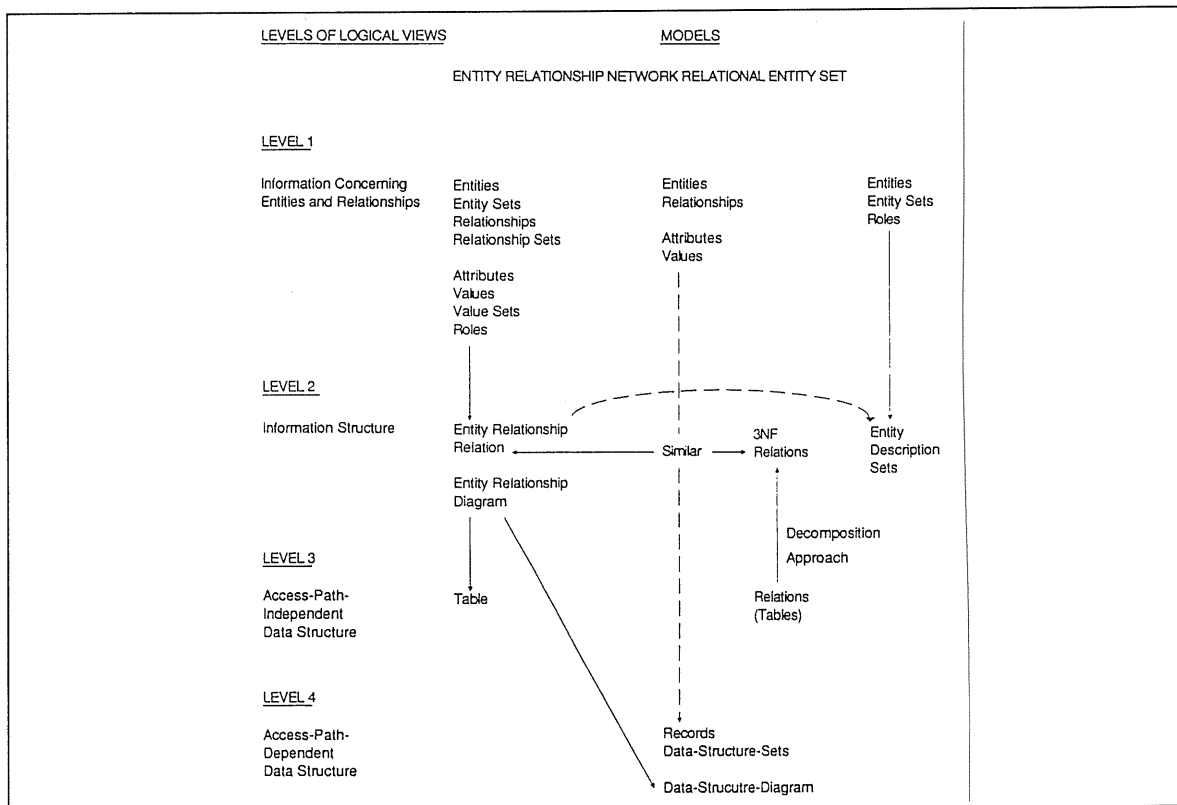


Fig. 5: Data models within the ERM framework

Chen defines entities as things which can be distinctly identified and relationships as associations among entities. Entities with common properties are classified into entity sets. A relationship set is defined as a mathematical relation among n entities each taken from an entity set. The notion "role" is used to describe the function an entity performs within a relationship. An attribute is defined as a function which maps from an entity set or a relationship set into a value set or a Cartesian product of value sets.

The ER diagram represents entity sets by rectangular boxes and relationships by a diamond-shaped box connected with the involved entity sets. Additionally, the roles of the entities in the relationship can be stated. Within the ER diagram existence dependencies between entities can be defined. Dependent entities are represented by special rectangular boxes and an arrow pointing at the dependent entity. The ER constructs and relationships are given in figure 6.

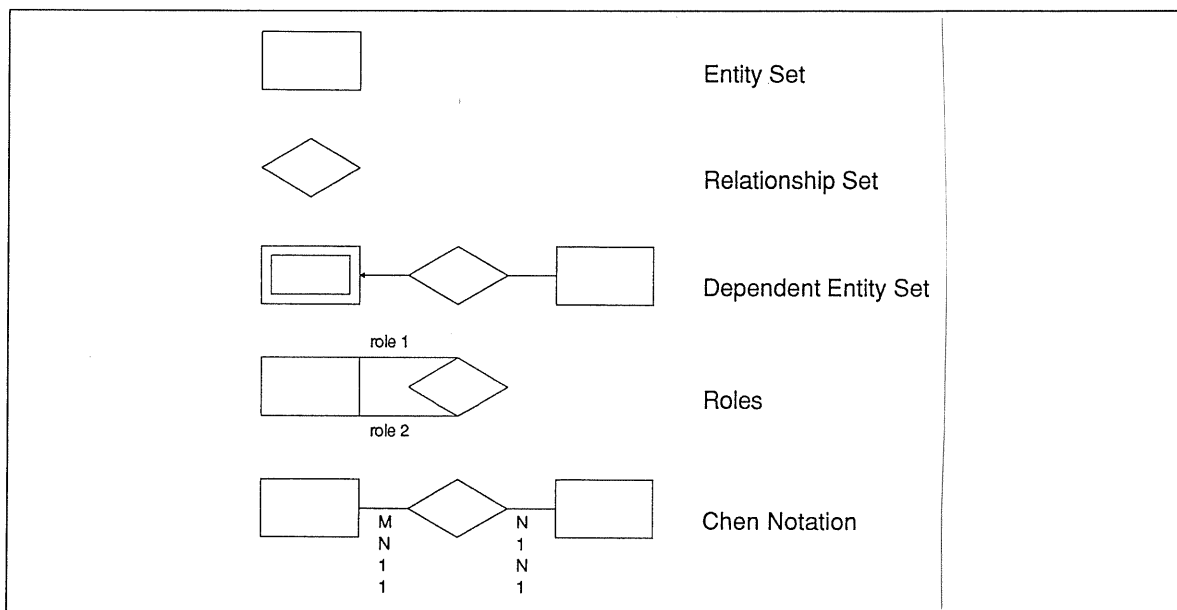


Fig. 6: Constructs and relationships of Chen's ERM

The Chen ER approach does not contain any generalization/specialization constructs. Therefore the specialization of employee into engineer, technician, accountant and financial analyst cannot be expressed in the modelling example (see figure 7). The Chen ER model does not allow aggregated relationships and attributes are not represented by the ER diagram technique. The Chen notation cannot express (min/max) cardinalities.

Besides, the Chen notation can lead to ambiguities when modelling relationships with a degree higher than two. The Entity Relationship Model by Chen has become the most common model used for conceptual data modelling because of its easy way to represent semantic networks. As a consequence, the Chen ERM was the basis for many other data modelling methods.

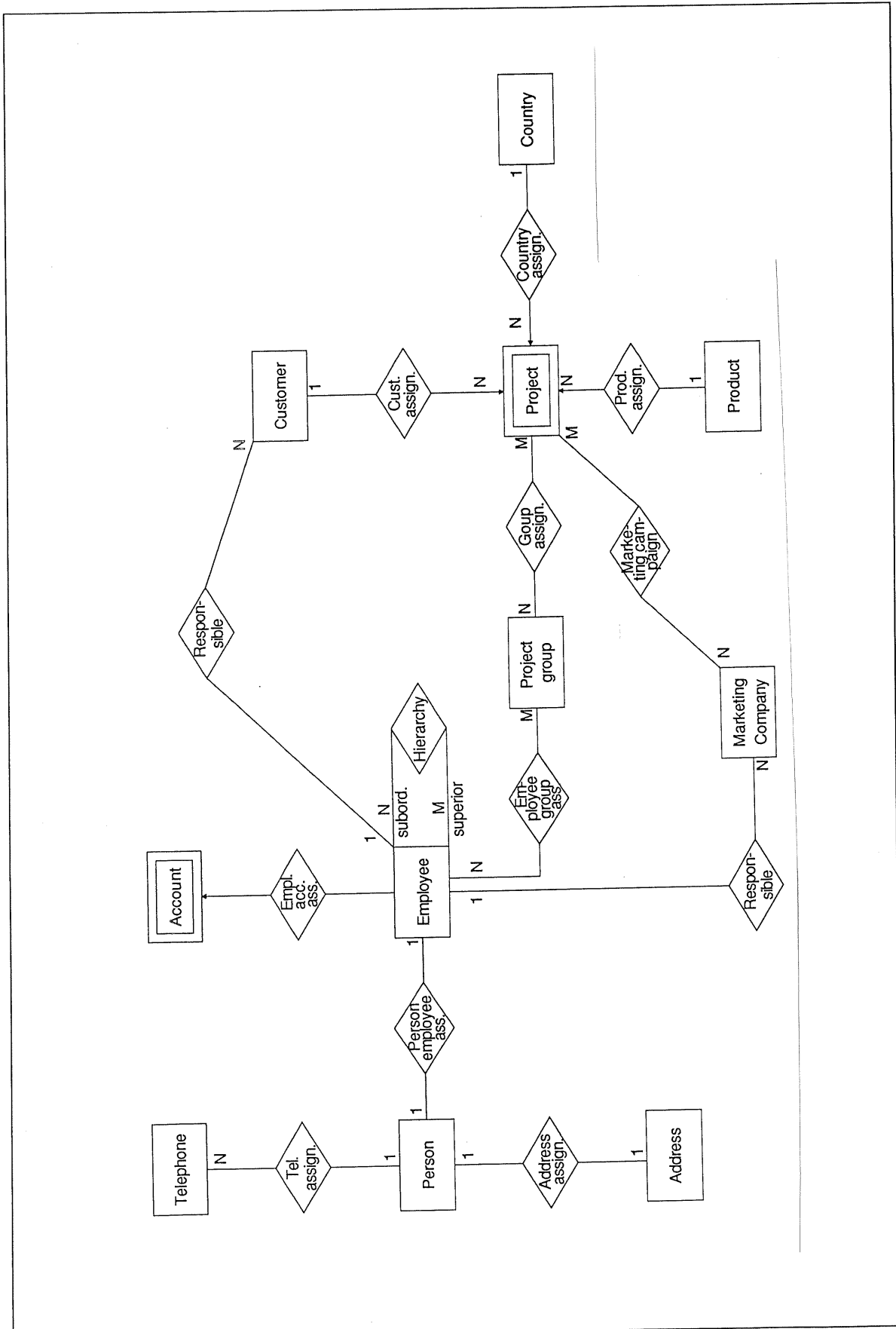


Fig.: 7 The modelling example in Chen's ERM

2.3 CIM-OSA

CIM-OSA (Computer Integrated Manufacturing - Open Systems Architecture) defines an integrated methodology to (ideally) support all phases, views and modelling activities of a CIM system life-cycle. It comprises different modelling methodologies⁴ to capture and model all the data and information in a CIM environment⁵. Due to its ambitious objective to model the highly complex and heterogeneous CIM environments, various knowledge representation schemes are utilized, for instance semantic networks (SADT-oriented, ERA) and logical representation schemes (relational model).

The basic concepts of the CIM-OSA approach are introduced by a three-dimensional framework, the so-called CIM-OSA⁶ cube, that is depicted in figure 8.

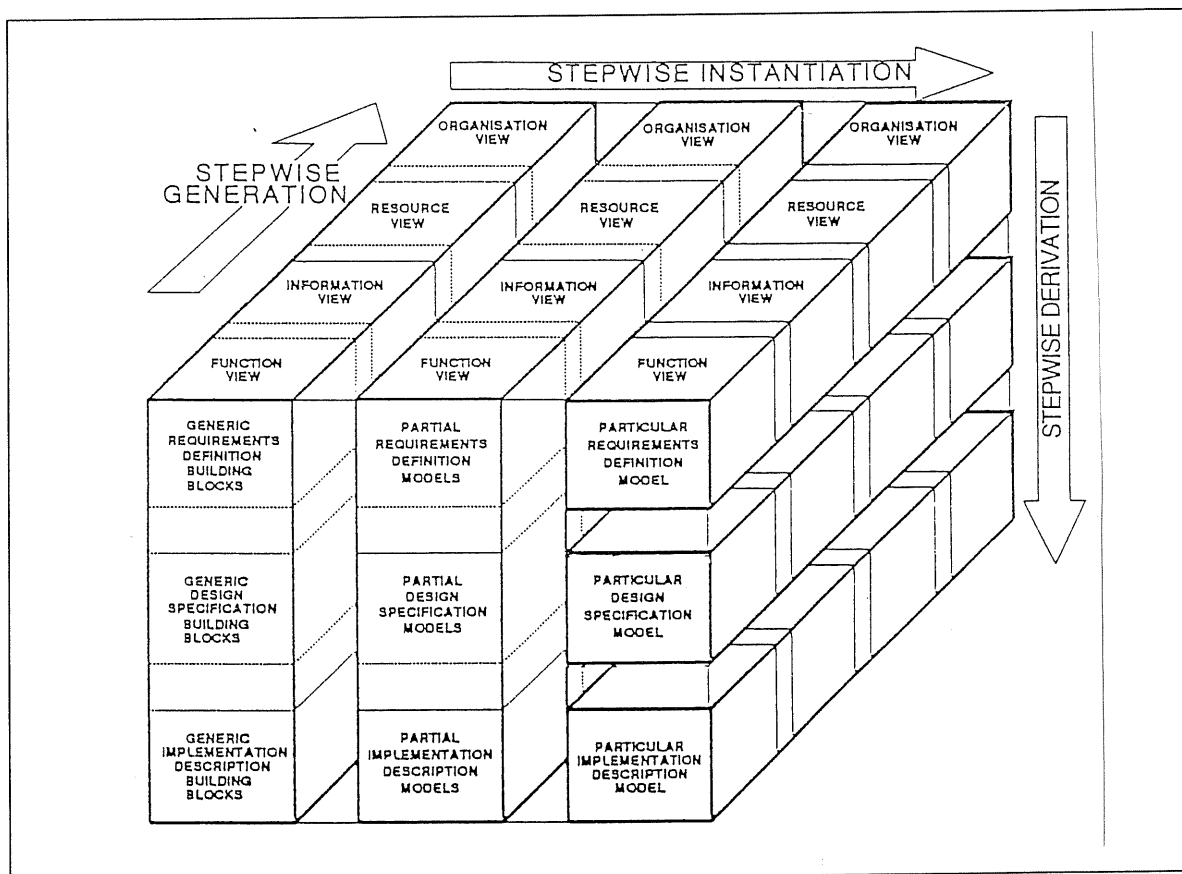


Fig. 8: The CIM-OSA cube

The CIM-OSA cube is to be interpreted according to the three axis:

- ❑ The stepwise generation consists of the function view, information view, resource view and organization view.
- ❑ The stepwise derivation comprising the requirement definition phase, the design specification and the implementation description.

- The stepwise instantiation process encompasses generic building blocks, partial models and particular models.

Rather than explaining the CIM-OSA concept in its entirety, we shall concentrate at this point on the data modelling approach of CIM-OSA. The conceptual data modelling is part of the design specification phase of the information view. The information view is a structure for storing, maintaining and processing data and knowledge containing the valuable data, know-how and expertise of an enterprise⁷.

The ERA-approach (Entity-Relationship-Attribute) chosen for semantic data modelling is an extended version of the entity-relationship model. Its constructs are depicted in figure 9.

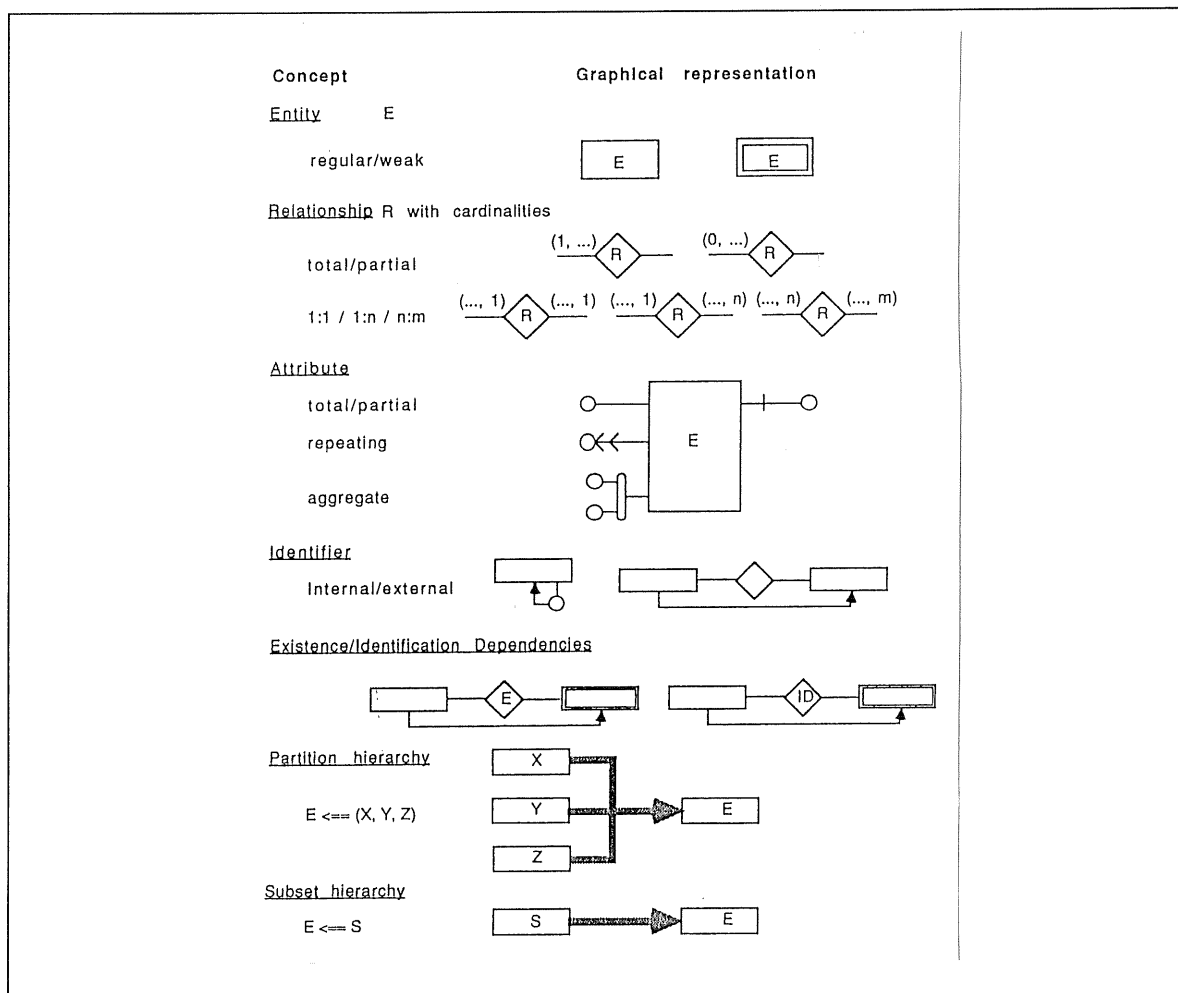


Fig. 9: CIM-OSA constructs and relationships

The major extension in comparison with the entity-relationship model is the explicit modelling of distinct attribute types. A so-called total attribute has to be assigned to an entity type, whereas a partial attribute may be assigned⁸. Furthermore, there are special

constructs for repeating and complex attributes. With respect to the entity identifiers it is distinguished between internal and external identifiers. Internal identifiers are part of the entity type they identify whereas external identifiers of weak entity types originate from the entity type the weak entity type is dependent on. In addition, existence and identification dependencies are graphically represented. Within the CIM-OSA data modelling approach it is also differentiated between a subset hierarchy and a generalization hierarchy. In the context of ERA a subset hierarchy represents the fact that an entity type S is a subset of another entity type E if every occurrence of S is also an occurrence of E. A generalization hierarchy requires the disjointness and completeness of the partitioned subtypes with respect to the supertype. In contrast to some other semantic data modelling methodologies the ERA approach of CIM-OSA does not support the concept of an aggregated relationship type (re-interpretation).

The application of the data modelling constructs of CIM-OSA to the example is shown in figure 10.

The particularity of the ERA is that it is embedded into an open framework of modelling methodologies⁹ and the CIM-OSA life cycle. Hence, it has cross-references to other concepts and methods utilized in the CIM-OSA cube. For instance, on the requirements definition modelling level for the information view an OERA-approach (object-entity relationship attribute) is applied. This approach incorporates the notion of objects and is used to capture high level semantics in the stage of requirements definition. Furthermore, some conversion rules for the transformation of the conceptual model into relations have been specified and some integrity constraints (derived from integrity rules) are described.

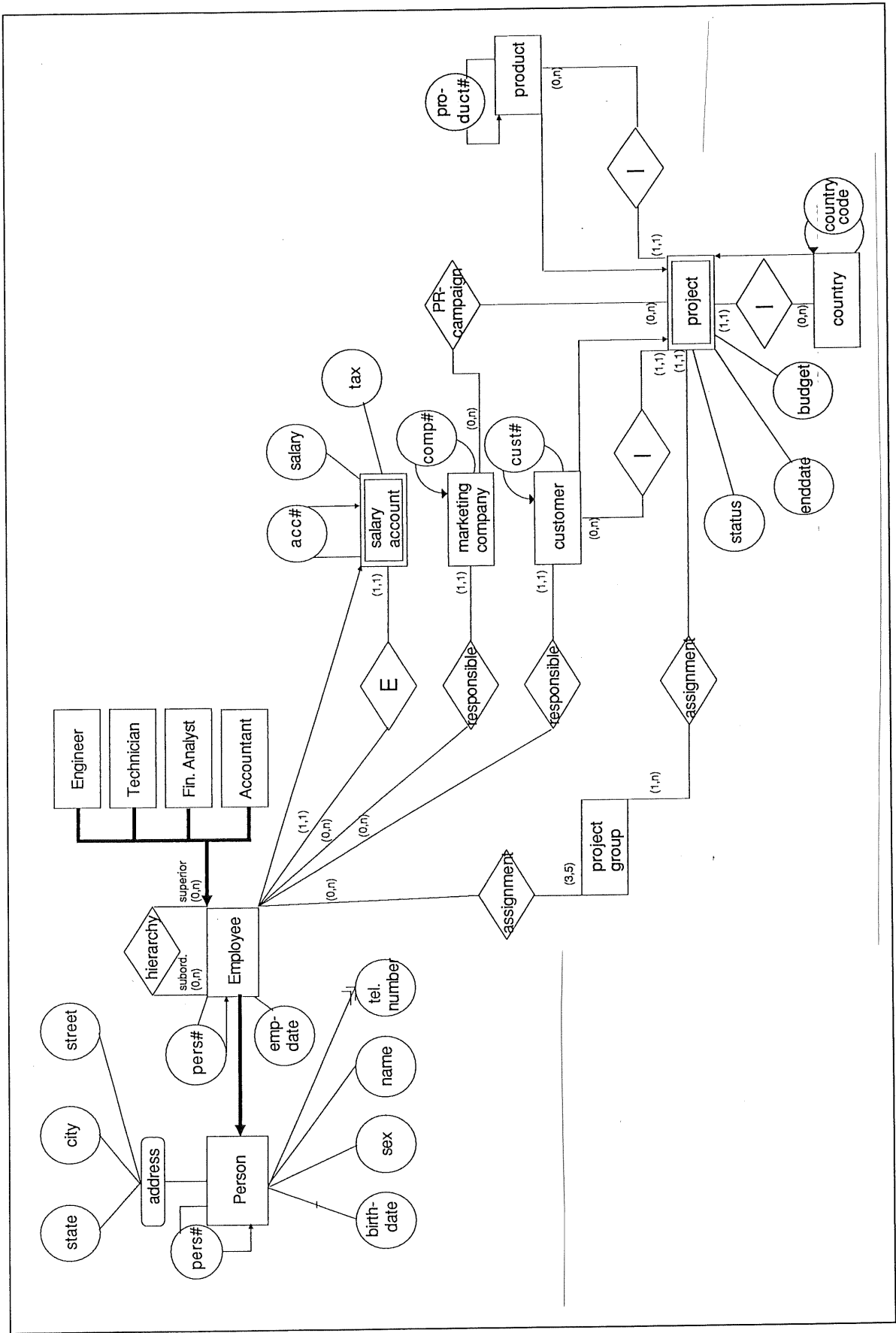


Fig. 10: The modelling example in CIM-OSA

2.4 ECER

The extended conceptual entity relationship (ECER) model¹⁰ is an extension of the entity relationship by Chen. It aims at supporting a graphical user interface to formulate queries against an underlying database. Therefore, the ECER - model is more formally defined to allow the specification of the mapping between the graphical ECER schema and the underlying database. The constructs of this approach are depicted in figure 11.

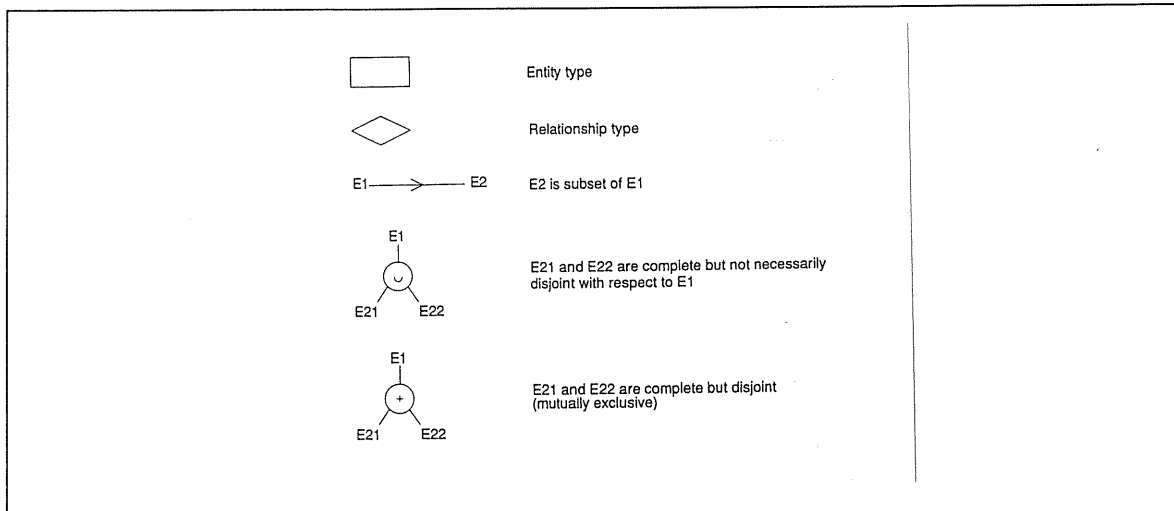


Fig. 11: ECER constructs and relationships

The so-called entity set descriptor, the relationship set descriptor and generalization/specialization descriptor are defined with the help of relational algebra expressions. With respect to the conceptual modelling the ECER model allows a more precise specification of different types of the generalization/specialization operation. Placing a subset symbol on the arc connecting two entity sets denotes a subset relation between two entity types. If the subsets are mutually exclusive and complete with respect to the superset, this is indicated by a circled plus symbol. In case the subsets are not mutually exclusive but complete, a circled union symbol is used. The assignment of keys on the conceptual level is not supported by the ECER model. It is assumed that the underlying relational model supplies surrogate keys that take care of the object identification. The surrogate key assumption resembles the RM/T approach that is more formally than the common ER-models.

Further components of the ECER model are more implementation oriented concepts such as data retrieval operators, computation operators, and update operators. These are used in connection with a relational DDL and DML¹¹. Applying the ECER-methodology to the example yields the following representation:

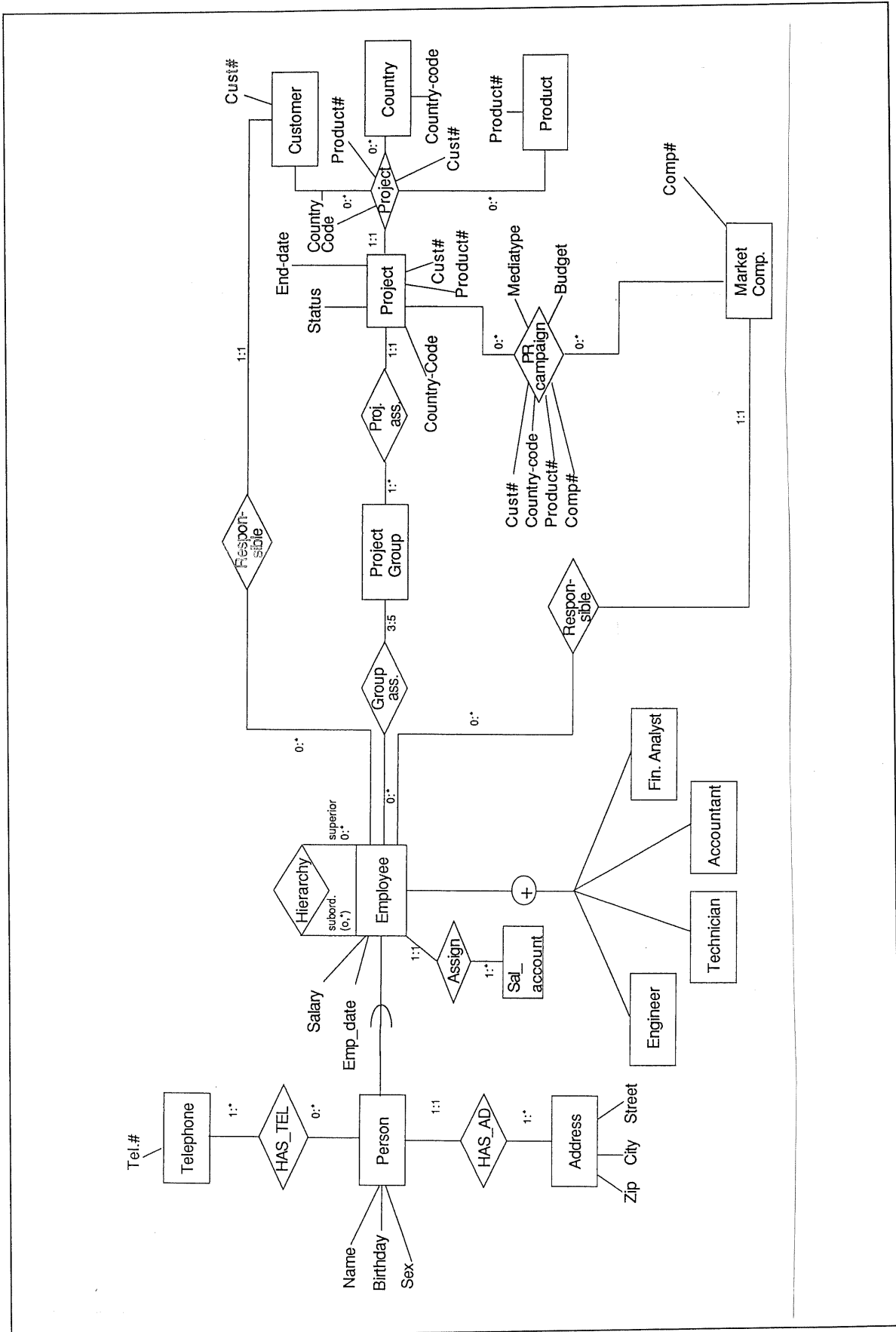


Fig. 12: The modelling example in ECER

2.5 ECRM

The Entity Category Relationship Model (ECRM)¹² is an extension to Chen's ERM. "Entities are classified into categories according to the roles they may play within relationships".¹³ Categories serve two purposes. They separate the real-world object from the roles it could play. In addition, categories can be utilized for the generalization/specialization relationship. An entity type can simultaneously be a category. Graphically, the ECRM utilizes the Chen notation to represent complexity, although its data definition language GORDAS (Graph Oriented Data Selection) allows (min,max) cardinalities. The constructs, relationships and cardinalities are shown in figure 13.

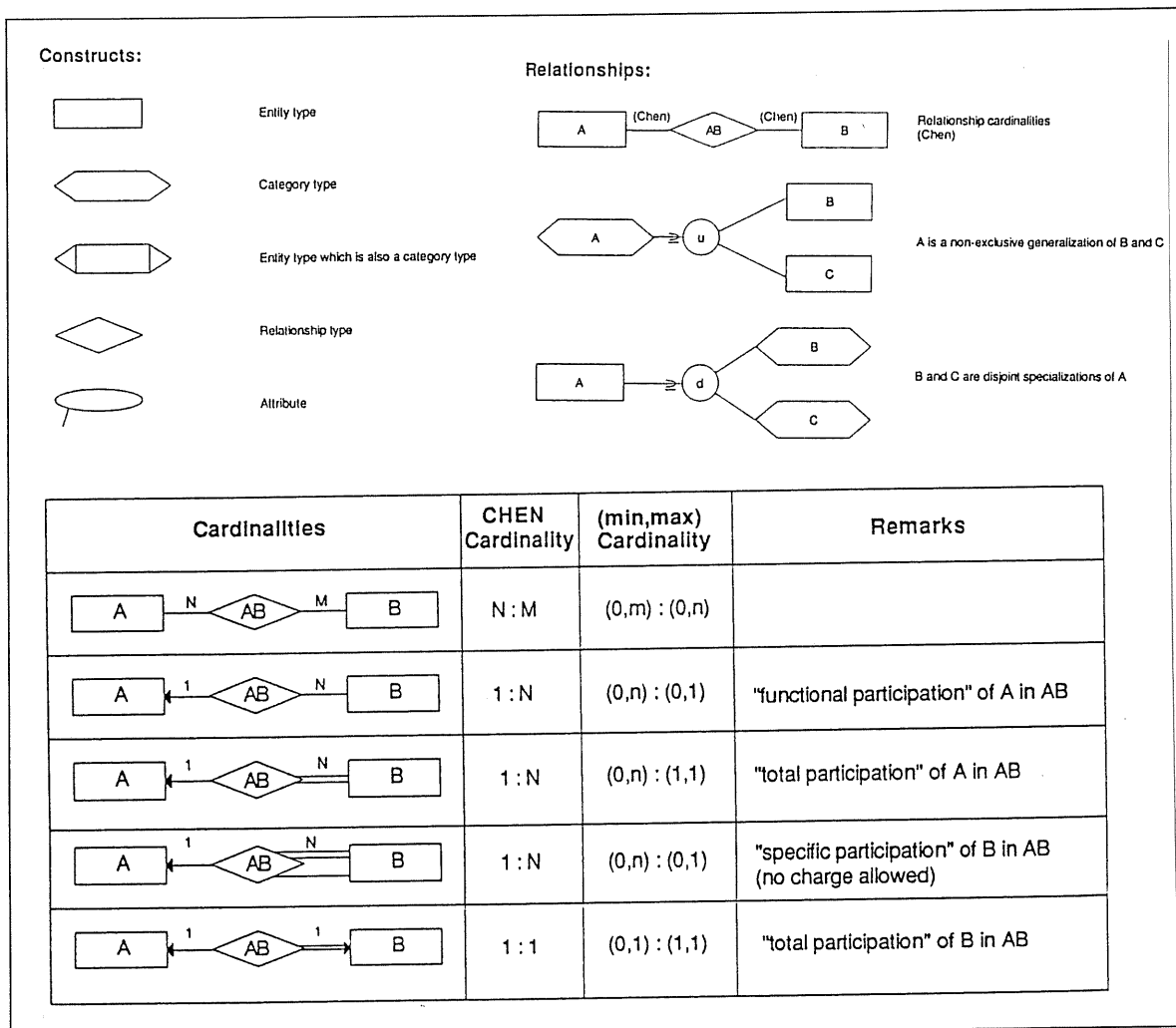


Fig. 13: ECRM constructs and relationships

There are no key considerations on the uppermost level. Transformation specifications to the relational data model exist. There is a possibility to specify free-form constraints.

The example of figure 14 shows that an aggregation of relationships is not possible. Hence, PROJECT has to be defined as an entity type. The ECRM does not model existence or identification dependencies. Hence the ACCOUNT has to be linked via a 1:1 relationship with EMPLOYEE. Multivalued and complex attributes are allowed. Both the telephone and the address can be declared as attributes.

The Entity Category Relationship Model is a mature, thoroughly specified data model. The category is a concept which improves the definition of a conceptual schema. However, the distinction between an entity type and an entity type which at the same time is a category type remains unclear.

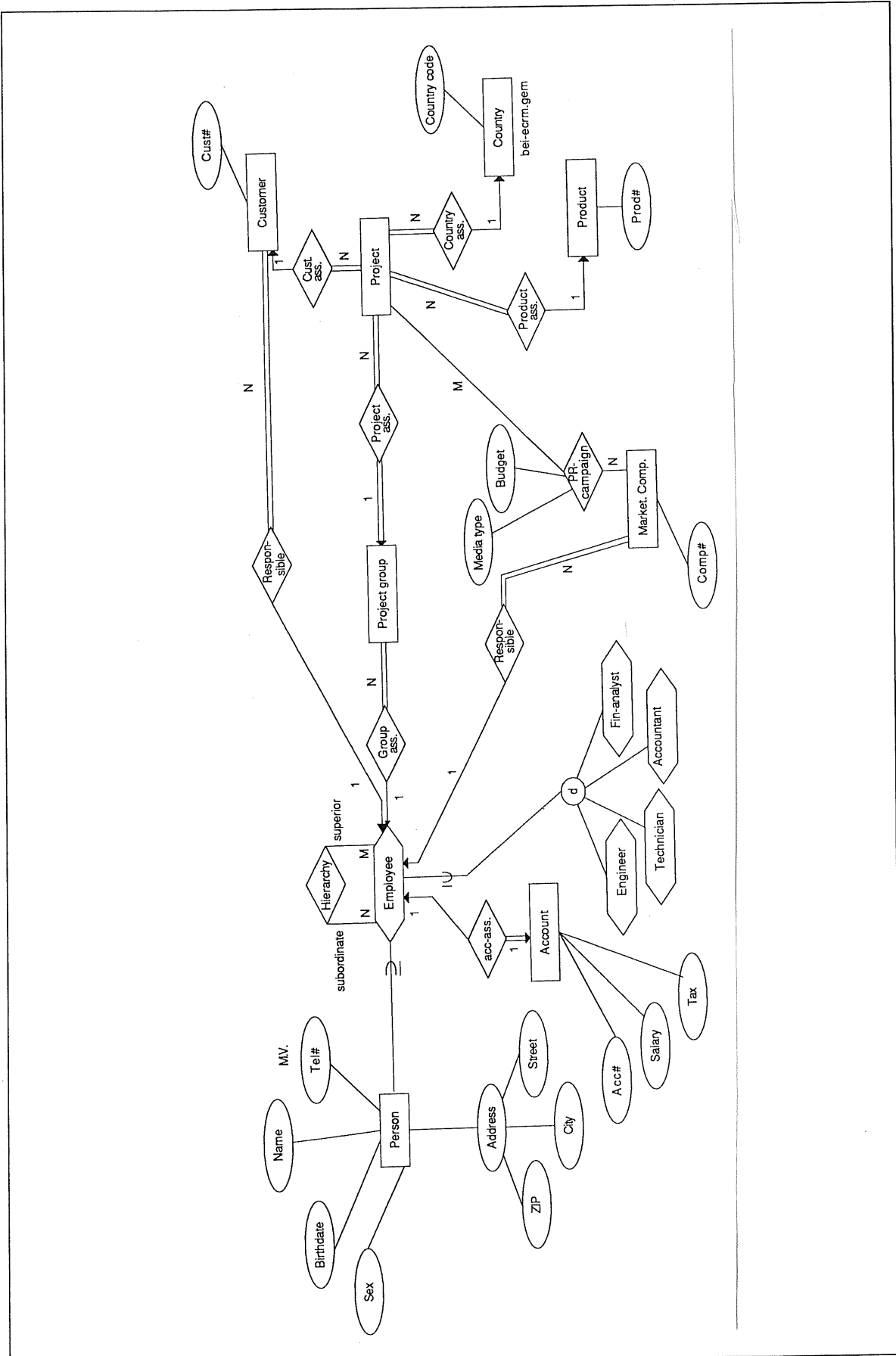


Fig. 14: The modelling example in ECRM

2.6 EXPRESS

EXPRESS is a formal information modelling language mainly used for the representation of product definition. It is currently developed by a technical committee of the international organization for standardization (ISO) in the STEP project and will soon become an ISO draft standard.

EXPRESS is parsable by computers and allows an automated checking for inconsistencies. The language focusses on the definition of entities which are specified in terms of data and behaviour. The behaviour of an entity is defined by static constraints. Constraints may be limits on the number or kind of attributes and are part of the attribute resp. entity declaration.

The language comprises a set of reserved keywords, operators, constants, standard functions and -procedures. It distinguishes between simple types, aggregation types, entity types, enumeration types, generic types and select types¹⁴. Logical and boolean data types may be defined. Aggregation data types are used to represent collections of elements of some basic types. In particular, arrays of arrays are allowed, i.e. nesting of aggregation types is allowed. The data types have to be declared, i.e. identifiers have to be assigned to them. Besides, different schema may be nested.

Entities may be structured in a hierarchical network, allowing the representation of supertypes and subtypes, i. e. the concept of specialization is supported¹⁵. This is achieved by formulating subtype clauses that allow further specification with respect to the relationships between subtypes (exclusive constraints etc.). Multiple inheritance is allowed. Attributes may either be introduced as simple data types (such as integer, real) or another entity type, i.e. an entity may have attributes that are entities. Attributes may be explicitly defined or be derived. The specification of the cardinality of relationships is similar to the (min,max)-notation. Integrity constraints may be defined with local rules that assert validity statements on the instances of attributes (uniqueness rules resp. domain rules). A EXPRESS model provides a unique identifier for each instanced item. Declared identifiers are unique and valid within the so-called scope. References within EXPRESS between declarations are established via the name of the declaration. EXPRESS allows the definition of algorithms and procedures. The concepts of functional scope, variables, and referencing mechanisms are covered in detail. Also included are logical operators such as NOT, AND, OR, string and set operations and elaborated control logic statements (e.g. if-then-else, case of, while..). EXPRESS as a formal modelling language resembles common programming languages. As such it lacks the ease-of-understanding required in the user-

oriented phase of conceptual modelling. Despite its vast amount of constructs and its expressiveness it is not an appropriate means to conduct semantic modelling unless the user is familiar with the concepts of programming languages. A major drawback of EXPRESS is that it allows only binary relationships.

The procedural system representation of the EXPRESS language can be augmented by a graphic extension, EXPRESS-G¹⁶. This graphically oriented representation scheme supports only the representation of a subset of the language constructs: EXPRESS-G provides the notions of entity, type, relationship, cardinality and schema. Constraints (rules)¹⁷, constraint mechanisms, and procedures are not supported. Entity types are depicted by different types of rectangles and relationships are represented by connecting lines, see figure 15. There are three relationship types that are displayed by different line styles. A dashed line represents an optional relationship between an entity type and one of its attributes. A generalization/specialization relationship is displayed by a thick solid line. All other relationship types are represented by solid lines. The relationships are interpreted as directed with the 'to-end' being indicated by an open circle. The composition symbol is used to mark references to either another schema or another page. Cardinalities are specified by linguistic expression on the relationship connectors. Schemas may be modelled hierarchically. EXPRESS-G also allows the design of partial models being a subset of a complete model.

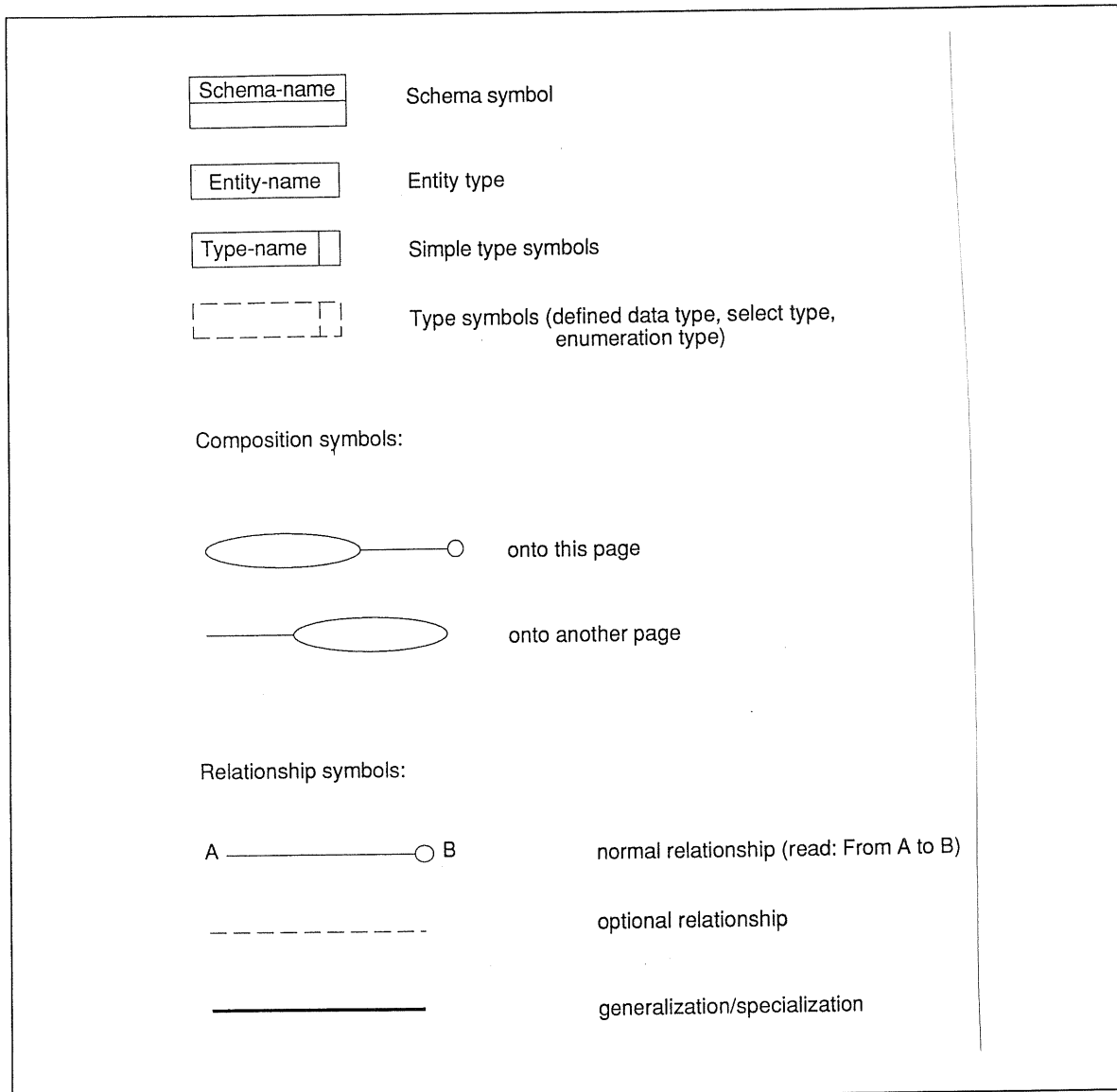


Fig. 15: EXPRESS-G constructs and relationships

Applying the EXPRESS-G constructs to the modelling example leads to the following result, see figure 16. The modelling constructs of EXPRESS-G are considerably less than the ones of the language EXPRESS. In comparison with other graphical representation schemes EXPRESS-G has only limited expressiveness.

In consequence, EXPRESS is only of limited benefit for conceptual modelling purposes in business oriented environments.

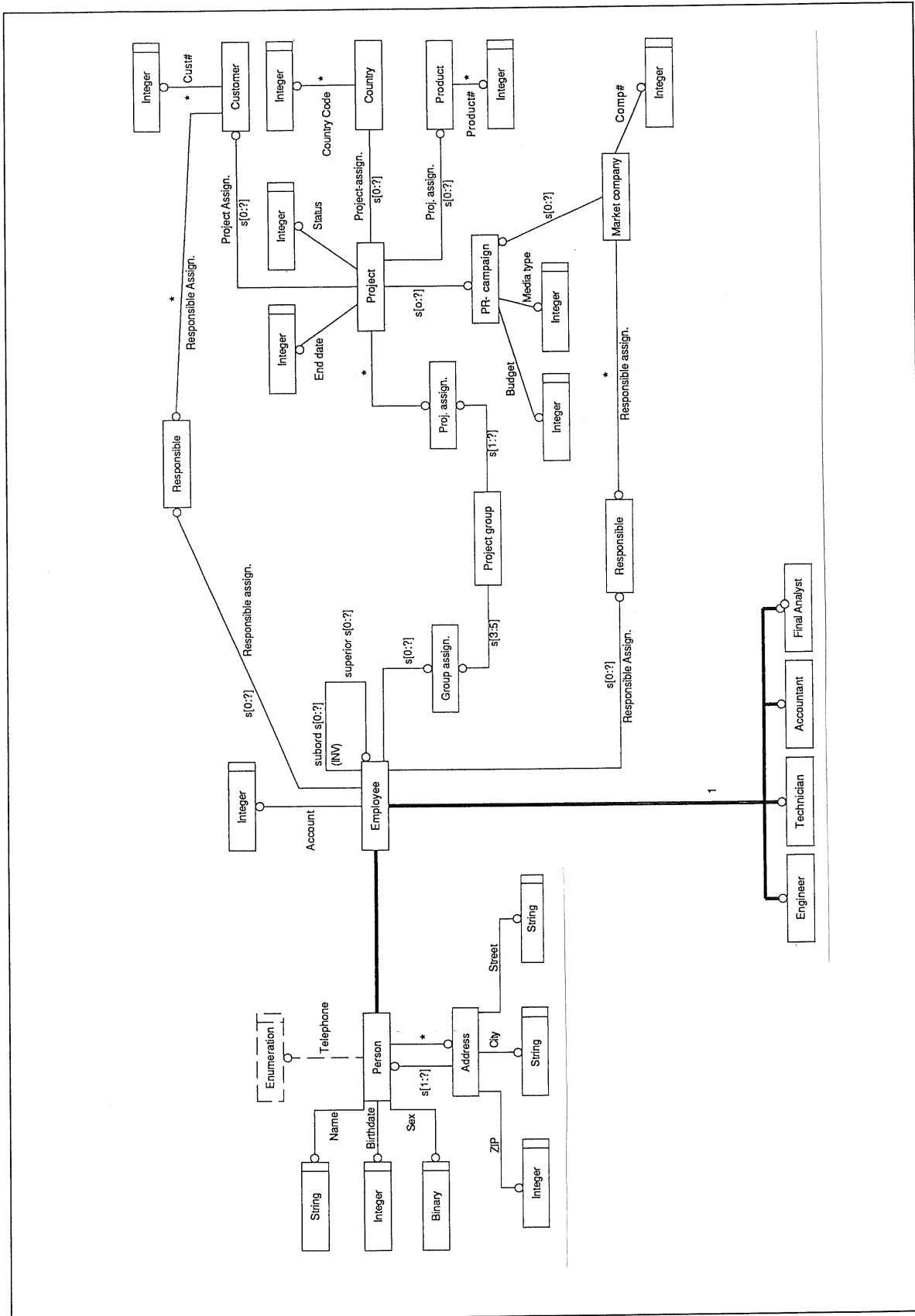


Fig. 16: The modelling example in EXPRESS-G

2.7 IDEF1

IDEF has been developed within the ICAM project (Integrated Computer Aided Manufacturing) for the United States Airforce in the late seventies. Its goal has been to improve the implementation of electronic data processing within production systems by using structured techniques. A reference model for airplane production has been designed.¹⁸ IDEF consists of three methodologies. IDEF0 bases upon SADT (Structured Analysis and Design Technique) for function modelling; IDEF1 is for data modelling, IDEF2 models processes. The linkages between the three modelling components are poor. The process description of IDEF1 is detailed. IDEF1 is close to the Entity Relationship Model. It is a binary ERM. It discerns several levels. On the entity level only entity types and relationships are described. On this level n:m relationships are allowed. On the attribute level, all n:m relationships have to be split into 1:n relationships. Key considerations and key inheritance issues are allocated on this level. The lowest level is the data manipulation language of a specific database management system.

IDEF1 gives much consideration to the identification dependency. This implies that it is conceptually close to hierarchical or network-oriented databases. It discerns independent and dependent entity types (see figure 17). M:n relationship types and information bearing relationships are modelled as dependent entity types. Weak entities are also depicted as dependent entities. IDEF1 does not use the (min,max) notation for cardinalities but a subset which is smaller (but larger than the traditional n:m notation).

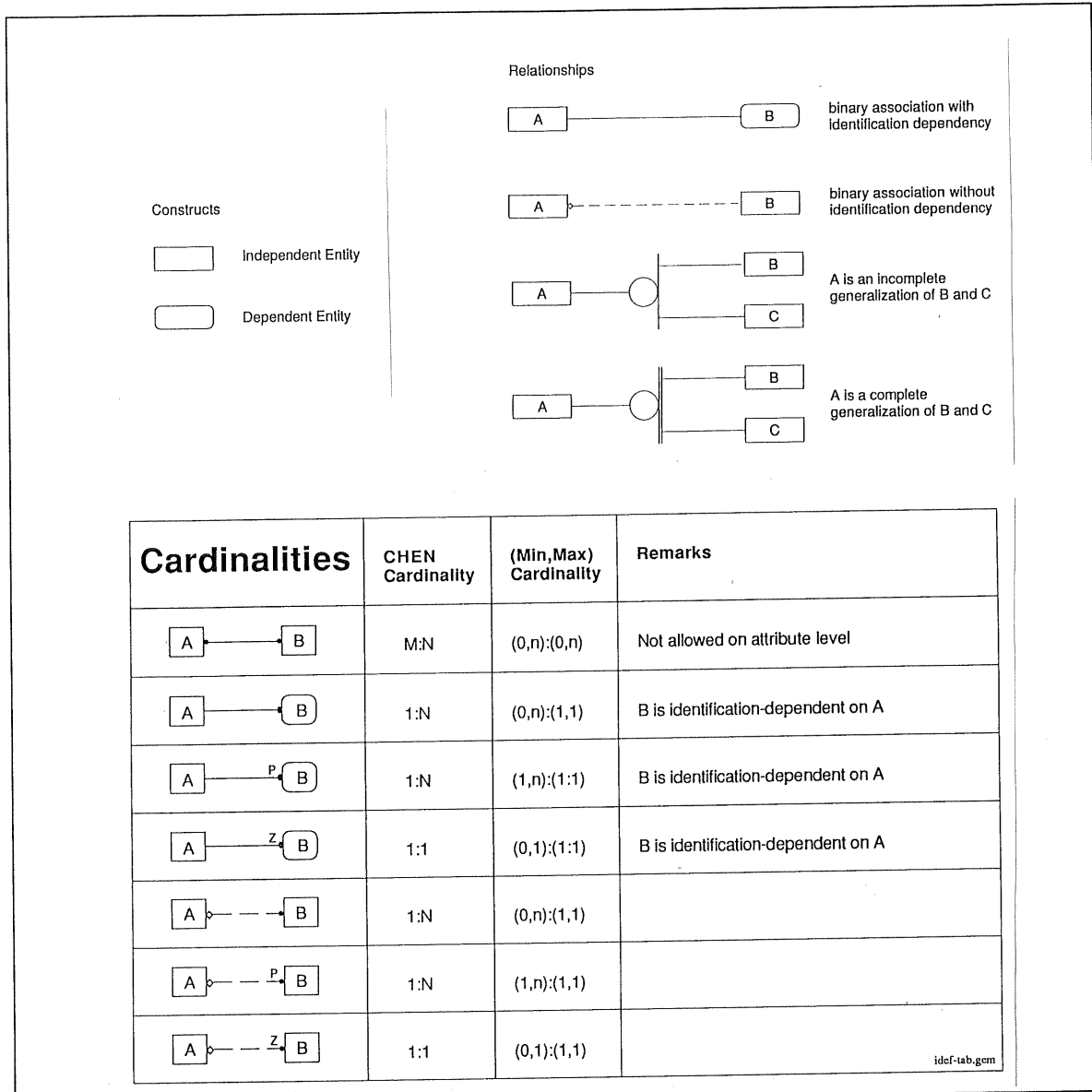


Fig. 17: IDEF constructs and relationships

The example shows that the PROJECT has to be modelled as a dependent entity type (see figure 18). Automatically the keys are inherited from CUSTOMER, COUNTRY, PRODUCT. The graphical representation is clear. On the entity-attribute level the name, the primary keys and normal attributes are clearly separated. IDEF1 discerns complete and not complete generalization/specialization relationships. We only make use of the second. In addition, only atomic attributes are allowed which implies that telephone and address have to be represented as entity types.

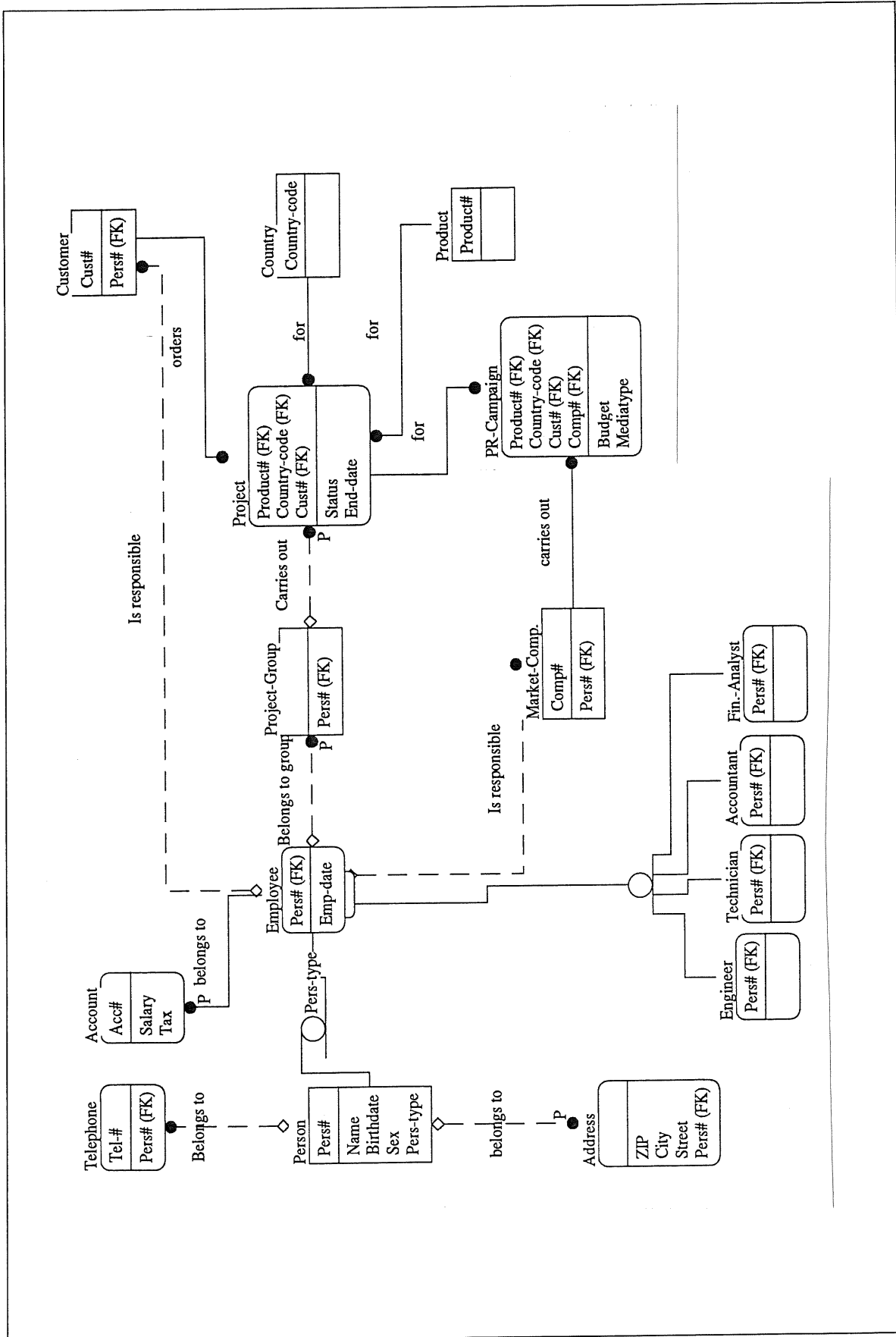


Fig. 18: The modelling example in IDEF1

2.8 ISOTEC

ISOTEC (Integrated Software TECnology) has been developed by the German software house EDV Studio Ploenzke as an integrated methodology for the development of information systems. It is implemented on the PREDICT CASE tool marketed by the Software AG. The constituent parts of ISOTEC are a procedural concept, an administration concept and a project management concept. For information modelling purposes only the procedural concept is of interest. It provides four different software development life cycle models. The activities, milestones and document types of each life cycle model are described very elaborately and in detail. Additionally, ISOTEC differentiates five methodologies. On the conceptual level there are¹⁹

- ISA (information structure analysis) for data modelling, and
- FSA (functional structure analysis) for modelling of functional aspects.

They should be carried out in parallel to ensure harmonization and validation. The remaining methodologies are more implementation-oriented:

- SSF for the specification and structuring of systems functions,²⁰
- DIA for the design of the dialogues within the scope of the user interface specification, and
- SDS for the design of a specific database.

ISA focuses on the requirements analysis and conceptual design phase. Three types of data models are developed:²¹

- The enterprise data model represents on a high level of abstraction all information objects that are important for the enterprise and - if identifiable - all their relationships. It serves as the basis for integration.
- The area data model details the object and relationship types of the enterprise model by decomposing them. The most important attributes are added to the information objects.
- That portion of the area model that is important for the application to be developed is completed in the application data model. Additional information objects and attributes are incorporated in the conceptual data model, object and relationship types are further detailed, and domains are specified.

The description language for the data structure is a binary entity relationship model. Rectangles are used as symbols for information objects. Relationship types are represented by arcs and arrows between two object types each. The arrows reflect (min; max) cardinalities. The recursive relationship type can be visualized by an arc emanating and ending in the same object type.

Furthermore, there are two possibilities to represent the generalization/specialization association: the traditional binary relationships (usually denoted "is-a") or particular graphical notations. The graphical representation is more expressive. It permits to distinguish whether the generalization/specialization association is disjoint and/or exhaustive. The attributes of the superior object type are always inherited by its subtypes. In the modelling example there are exactly four disjoint subtypes of employee: engineer, technician, accountant, and financial analyst. The completeness of the generalization/specialization is indicated by the black shaded arrow, the disjunction is shown by a box with the mark "S_x" (exclusive specialization). The specialization of person in employee is not exhaustive, this is represented by the white arrow. A box with "S_o" inside is used to depict that the subtypes need not be disjoint.

The greatest drawback of the ISA data model and the binary model in general is that ternary or more complex relationships are not allowed. A ternary or more complex relationship is transformed into an object type linked to its forming object types by binary relationships. The modelling of the ternary relationship "project" in the case study leads to the introduction of the object type "project" which inherits the primary keys of its forming entity types (customer, country, and product). The identification and existence dependency has to be expressed explicitly.

Another weakness of binary models is that attributes can only be assigned to object types. A relationship which has attributes of its own must be transformed into an object type. In the extended ER model PR campaign is a relationship between project and marketing company, identified by the key attributes of the associated entity types and described more precisely by the properties mediatype and budget. In the binary model PR campaign is objectified to permit the assignment of the attributes and referenced to the object types involved in this relationship by binary associations.

The information that "project" is originally an association is not available in the graphical notation, because binary models provide no constructs allowing to represent the aggregated relationship. To capture these information and other important semantics about the universe of discourse ISA provides the possibility to specify explicit constraints.

The data modelling constructs, integrity constraints and design rules are described very detailed in the ISA manual. The most important symbols of ISA are given in figure 19. The example is shown in figure 20.

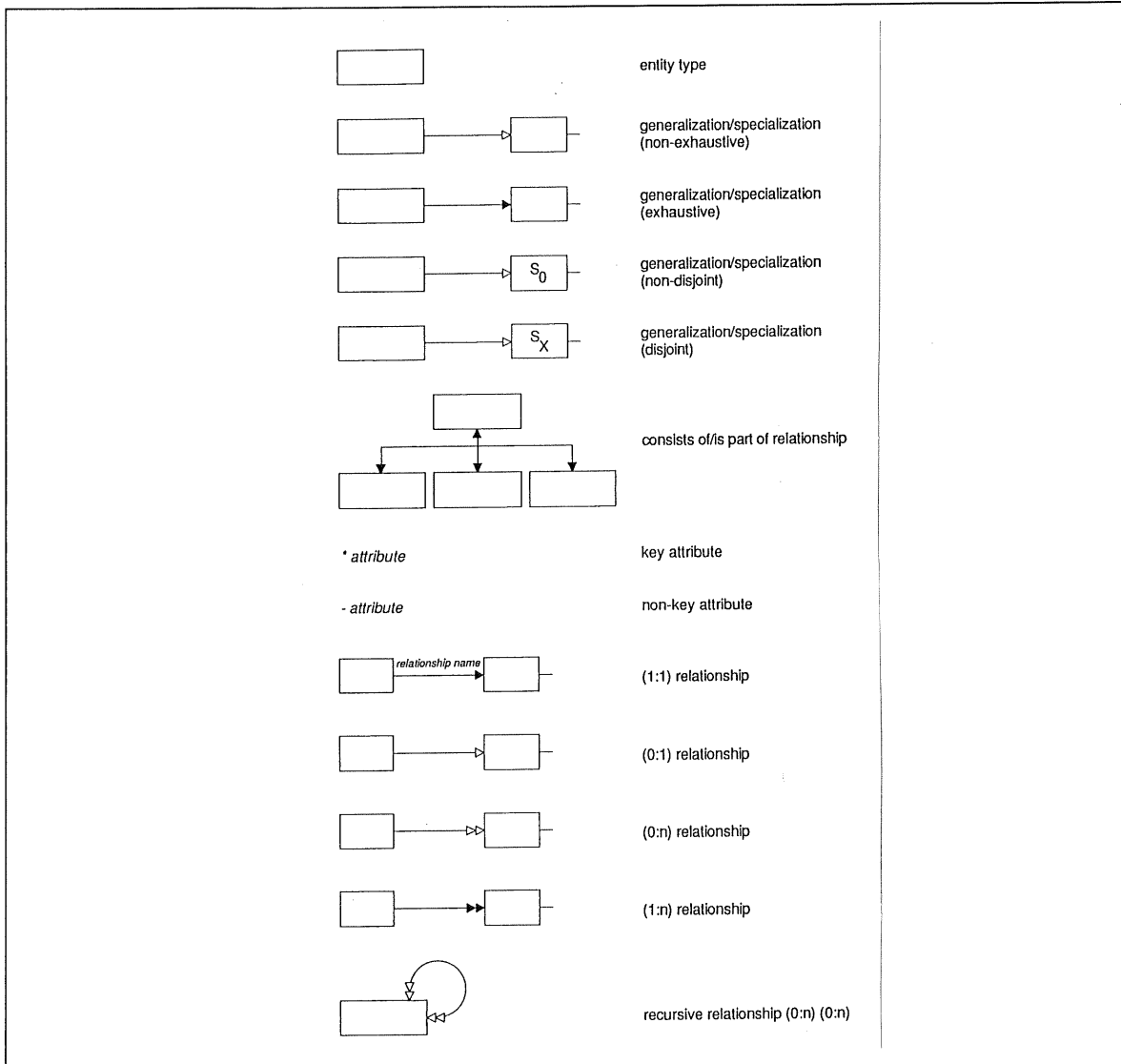


Fig. 19 ISA constructs and relationships

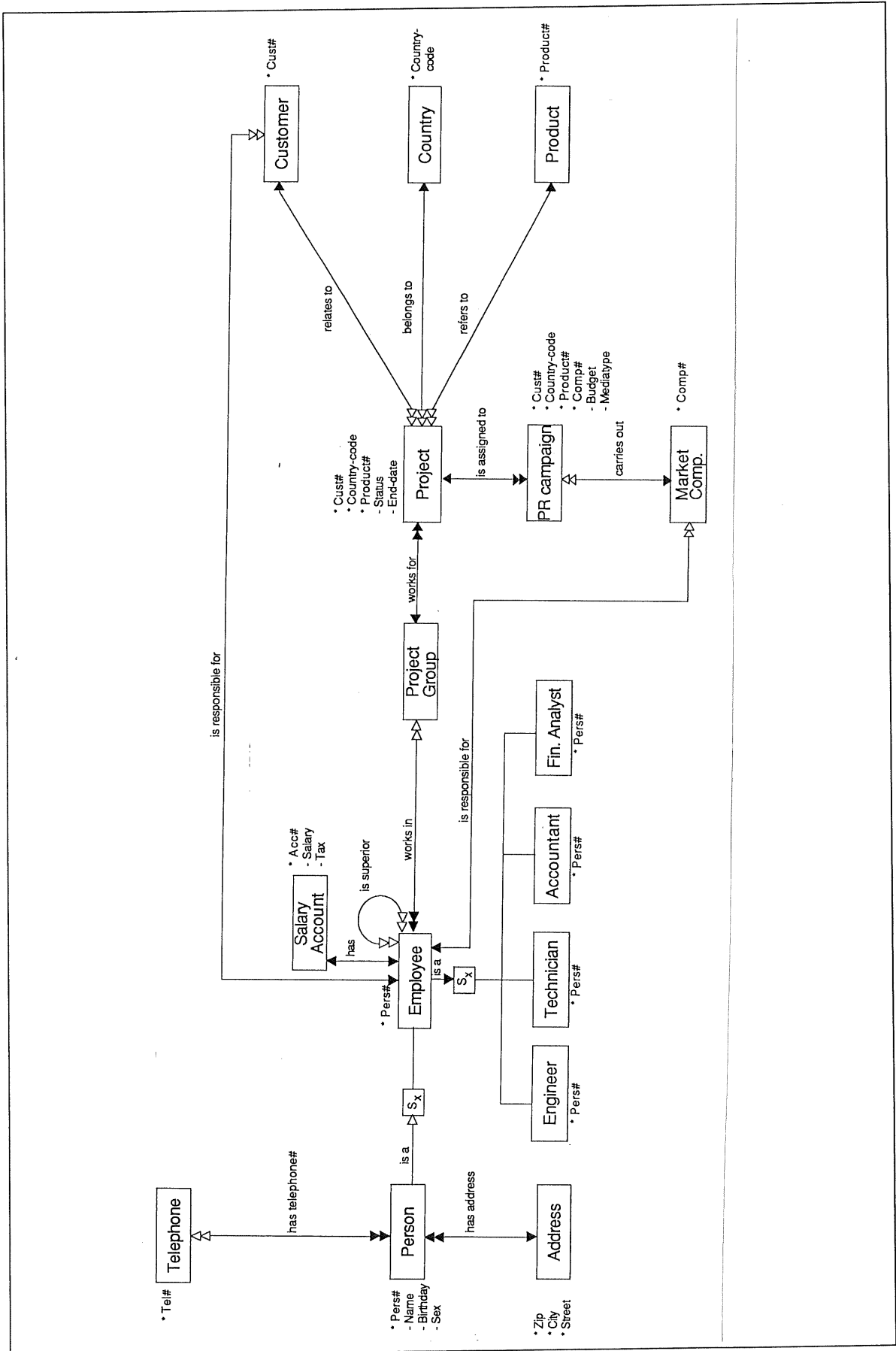


Fig. 20: The modelling example in ISA

2.9 James Martin

With the information engineering methodology (IEM) James Martin provides an architectural framework for information systems modelling. The IEM architecture divides the system development process into four stages: information strategy planning (ISP), business area analysis (BSA), system design, and construction. Each stage views data, activities, and their interaction in more detail. The emphasis is laid on the data.²²

In the ISP phase²³ the enterprise's goals, critical success factors, and information needs are identified and a high-level survey of the whole enterprise, its functions and data is created. The data model set up in this phase consists of high-level object types and their relationship types. These are further detailed during the business area analysis²⁴ for one business area at a time. Attributes are added and a fully normalized data model is built. The following phases - system design and construction - are close to implementation.

James Martin's information engineering methodology also uses a binary entity relationship model. Compared to the binary model employed by ISOTEC, the IEM data diagram is less powerful. Attributes cannot be represented graphically, they are only stored in the central encyclopedia and can be accessed, manipulated and made visible in textual form by a tool the methodology is implemented on (e.g. IEF of Texas Instruments, IEW/ADW of Ernst & Young). Furthermore, IEM distinguishes from the ISOTEC methodology in the graphical notation of the (min; max) cardinalities - the crow's foot representation is used. Explicit integrity constraints are not supported. Apart from these drawbacks the same strengths and weaknesses as described in the chapter on ISOTEC also hold for IEM.

The core component of IEM is the encyclopedia, a computerized knowledge base that not only stores development information but helps to control its accuracy and validity. It stores the content represented in diagrams, not the diagrams themselves, and enforces consistency within the representations. The encyclopedia shall be the basis for automated code generation.

In the following figure the graphical notation used in IEM is depicted. Figure 22 then shows the modelling example.

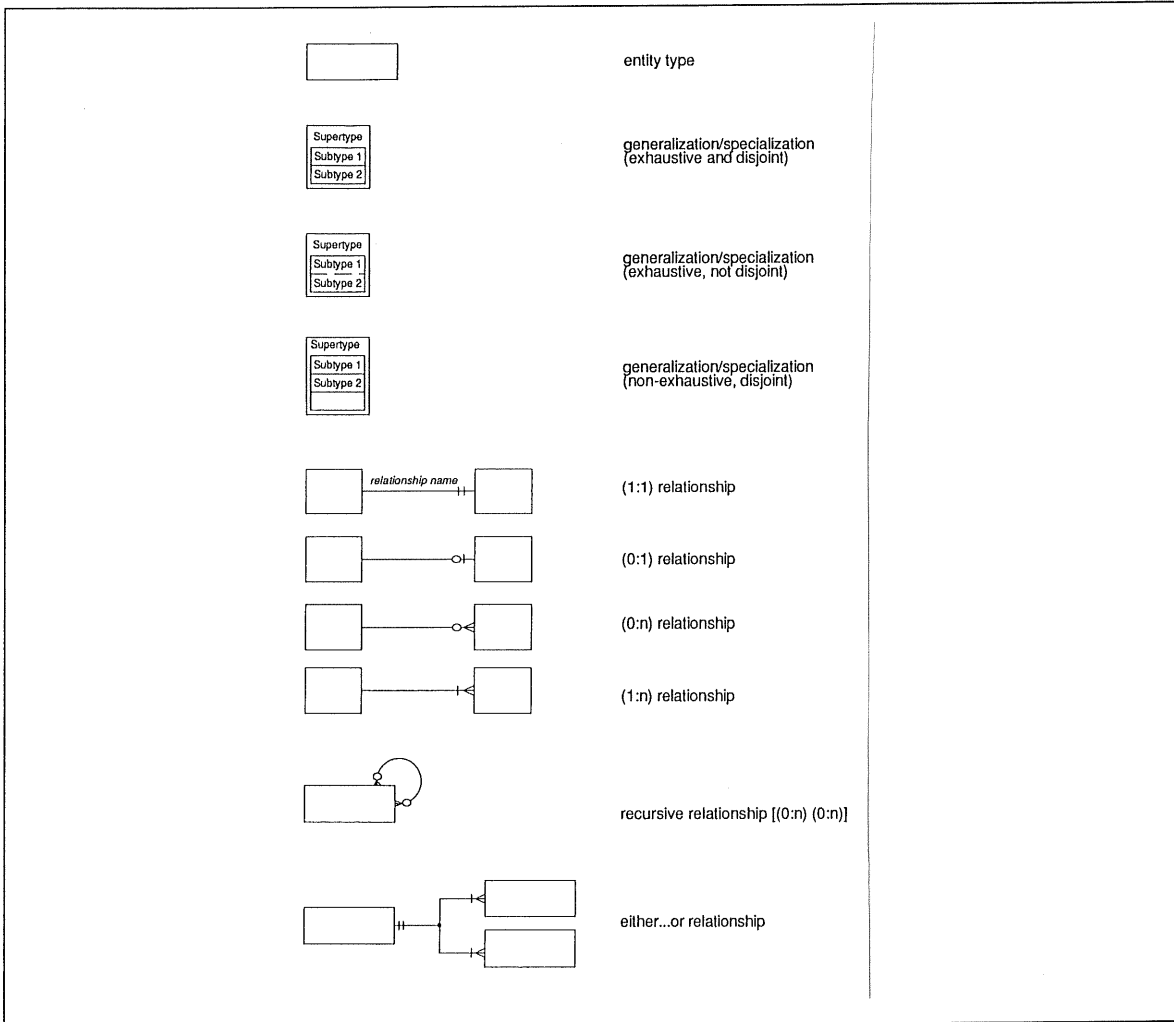


Fig.e: 21 IEM constructs and relationships

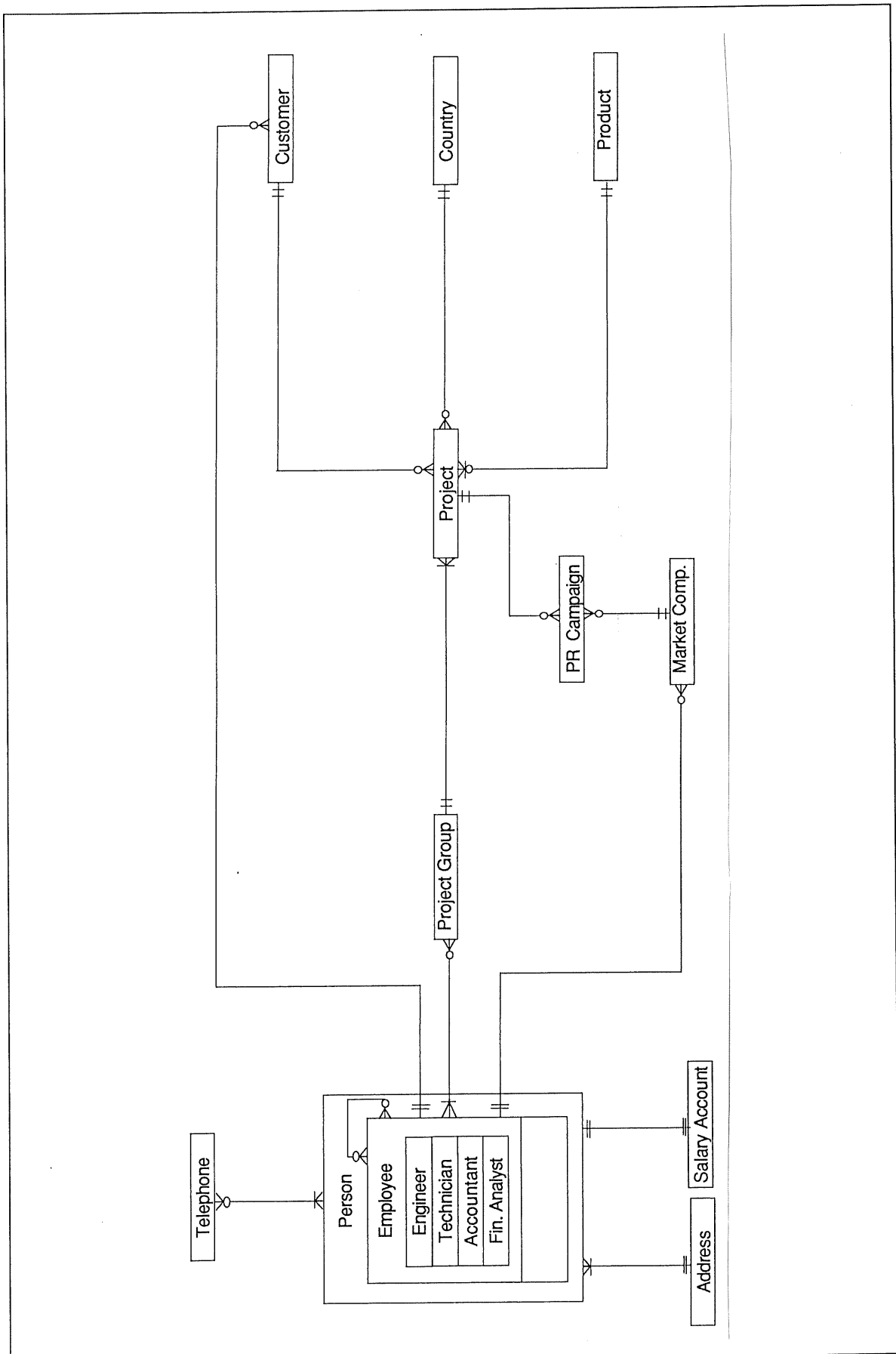


Fig. 22: The modelling example in IEM

2.10 Merise

Merise²⁵ is an information system design and development methodology. It was originally defined by several French consulting firms and university institutes and was sponsored by the French department of industry. Currently it is the far most used design and development methodology in France. Merise is mainly used to develop business oriented information systems. Merise comprises the data, function and organization view of an information system. On the data side Merise uses a kind of ERM which comes close to the original approach proposed by Chen. It uses min/max cardinalities but does not support aggregated relationships (re-interpretation) and generalization constructs although it is intended to extend the method towards this direction. The constructs of the data modelling part of Merise are depicted in figure 23.

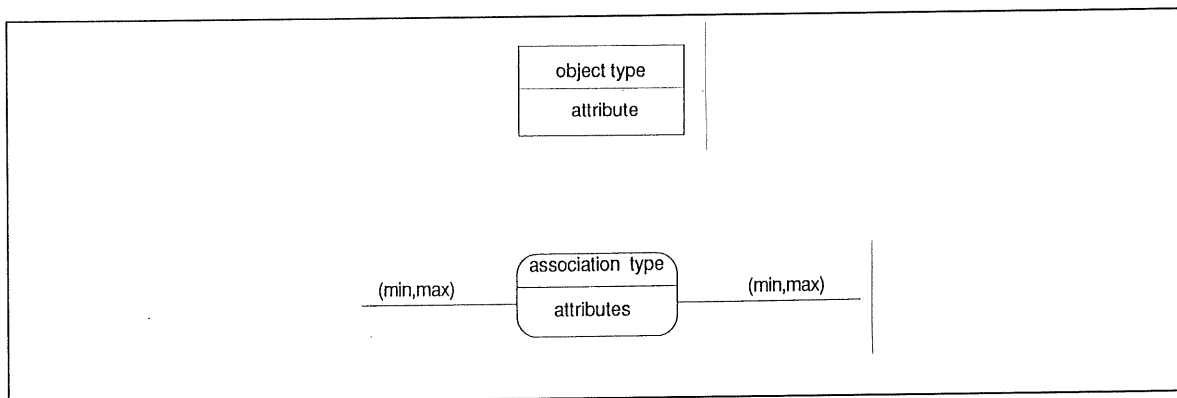


Fig. 23: Merise constructs and relationships

On the function side Merise uses a method which is influenced by Petri nets. Besides the modelling of data and functions Merise and its extensions use some other diagrammatic techniques especially to model the relationships between data, functions and also organizational units. For instance flow diagrams and flow matrices specify the information flow between the organizational units and some kind of process chain diagrams model the relationships between data, functions and organizational units. Additionally, information objects, processes, decision rules and organizational units are textually described within tables. Merise is based on a comprehensive methodological framework. This framework integrates three cycles: The abstraction cycle, the life cycle and the decision cycle. The abstraction cycle was originally influenced by the ANSI-SPARC architecture and consists of conceptual, organizational and physical level.

On each level data and function models are created. Compared to the ANSI/SPARC architecture there are large similarities. Merise differs only on the second level by explicitly considering the organizational units of an enterprise. Merise includes a very

detailed, iterative life cycle concept which does not only focus on the engineering of software but also on the design and development of a whole information system. The life cycle consists of different phases which can be used to control a concrete information modelling project. The abstraction levels are considered at each modelling step but with different emphasis. The Merise life cycle consists of the following steps: Long range planning, initial study, detailed study, technical study, programming and tests, launching and support. Quality assurance control is an activity performed in parallel with all previous steps. In addition to the definition of the life cycle phases Merise recommends a certain sequence concerning the modelling of the abstraction cycle components. For example conceptual function and data models can be generated in parallel and conceptual function model, organizational function model are used to validate the conceptual data model. The decision cycle of Merise comprises all decisions which are taken during the life cycle of the information system. The decisions can be structured hierarchically.

In the example, depicted in figure 24, one important disadvantage of Merise is the lack of a generalization/specialization construct. The possibility to express the generalization by normal relationship types is not very satisfying. Another problem of Merise is the missing of an aggregated relationship. Therefore, each relationship which should be part of further relationships has to be modelled as an entity type. This entity type has relationships with (1,1) cardinality to the entity types which participate in the former relationship.

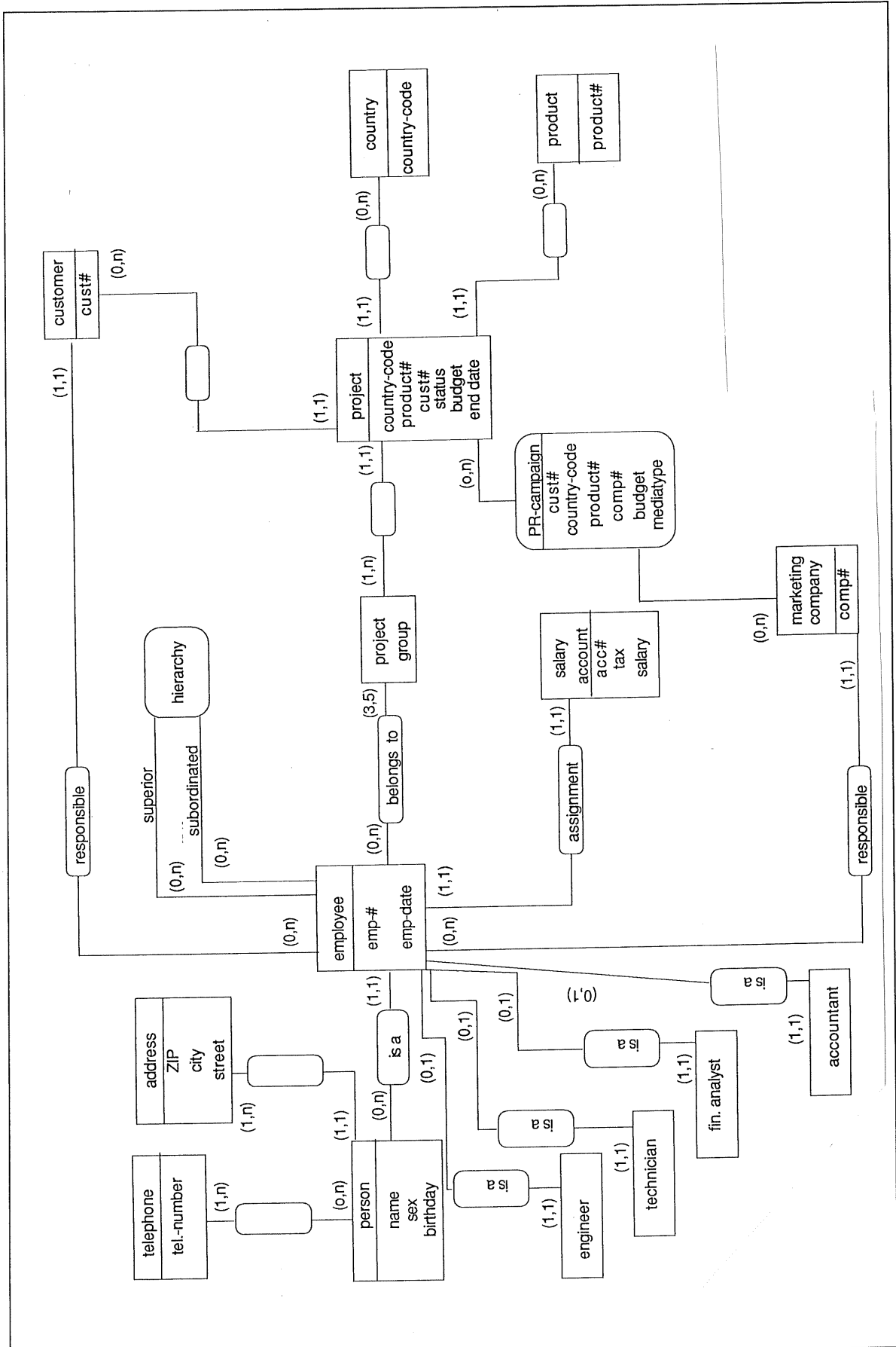


Fig. 24: The modelling example in Merise

2.11 NIAM

NIAM²⁶ - Nijssen's Information Analysis Methodology - is a fact-oriented approach for conceptual data modelling. An important part of this methodology is its design procedure which consists of nine steps:²⁷

- 1 Transform familiar information examples into elementary facts.
- 2 Draw a first draft of the conceptual schema diagram and apply a population check.
- 3 Eliminate any surplus entity types and common roles, and identify any derived fact types.
- 4 Add uniqueness constraints for each fact type.
- 5 Check that fact types are of the right arity.
- 6 Add entity type, mandatory role, subtype and occurrence frequency constraints.
- 7 Check that each entity can be identified.
- 8 Add equality, exclusion, subset and other constraints.
- 9 Check that the conceptual schema is consistent with the original examples, has no redundancy, and is complete.

The first three steps are concerned with designing

- entity types,
- relationships between entity types,
- attributes (label types),
- relationships between entity types and label types (reference types).

NIAM provides only one construct to graphically represent these elements: the fact type. It consists of one or more object types and the roles these object types play. Object types group entity and label types. The fact type linking an entity type with a label type is denoted as reference type. The fact type is a very powerful representation means, it allows to depict all constructs of the extended ER model, even the aggregated relationship type and the generalization/specialization association.

Additionally, NIAM supports the explicit specification of constraints. Certain constraint types occur so frequently and are so fundamental that they have a graphical representation as well. The most important constraint types are:

- entity type constraints
 - subtype constraints,
 - label type constraints

These restrictions concern the domain of the attributes. There is a variety of notations for label type constraints:

- braces describing the data type or listing the permissible values,
 - square brackets indicating that numeric operations are allowed,
 - angle brackets indicating that string operations are allowed.
- uniqueness constraints:
- intra-fact type constraints (refer to only one fact type)
 - inter-fact type constraints (between different fact types)
- role constraints
- mandatory/optional roles
 - occurrence frequency constraints (min; max) cardinality
 - equality constraints
 - exclusion constraints
 - subset constraints
 - reflexivity, symmetry, transitivity

Checks are performed throughout the procedure to ensure that no mistakes have been made in the modelling process.

The information given in the modelling example can be modelled by NIAM as shown in figure 26. The specialization of employee in engineer, technician, accountant, and financial analyst can be modelled in this case study in two different ways: as supertype/subtype relationship or, as represented in the diagram, as unary roles. The NIAM constructs are depicted in figure 25.

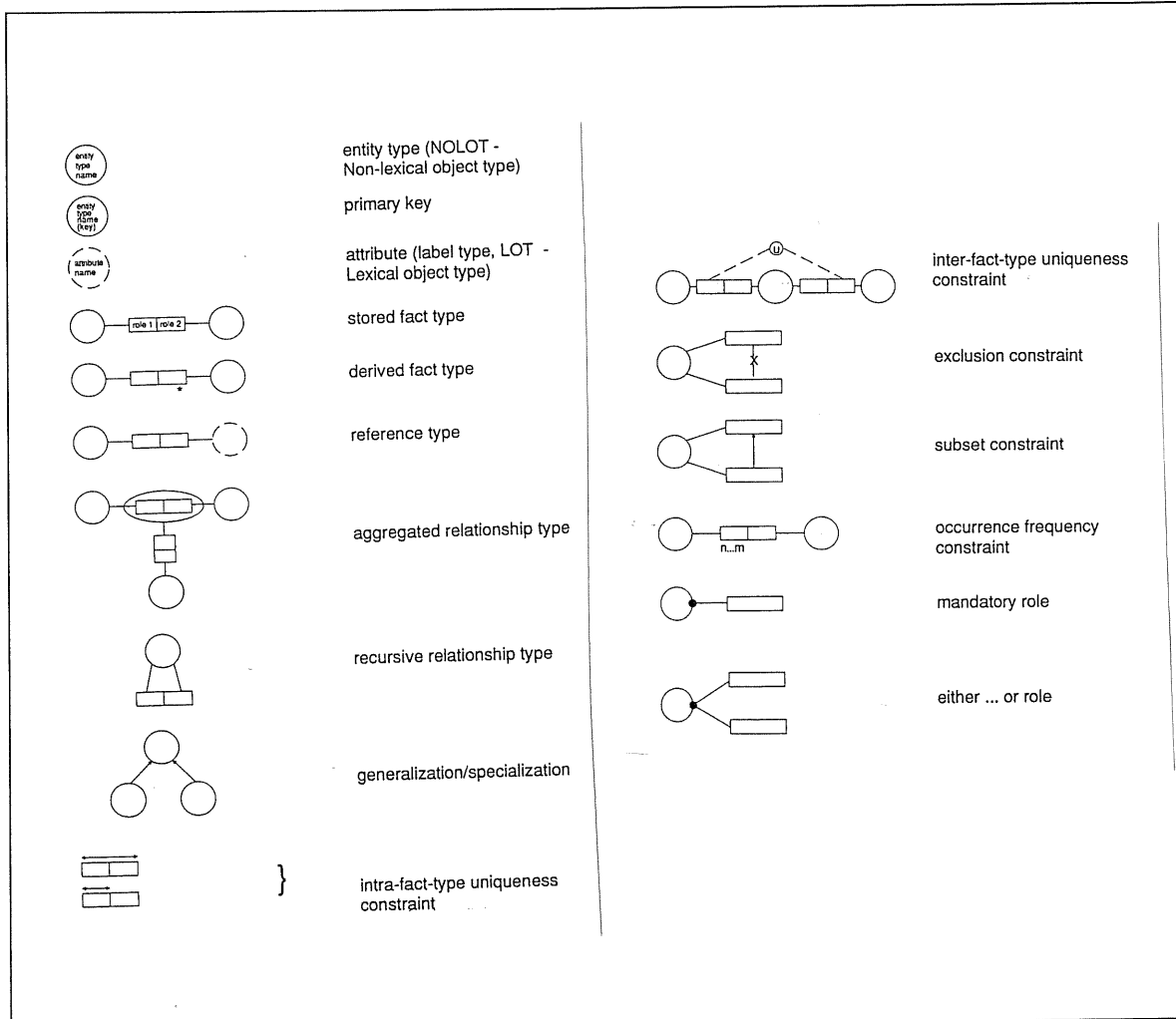


Fig.: 25: NIAM constructs and relationships

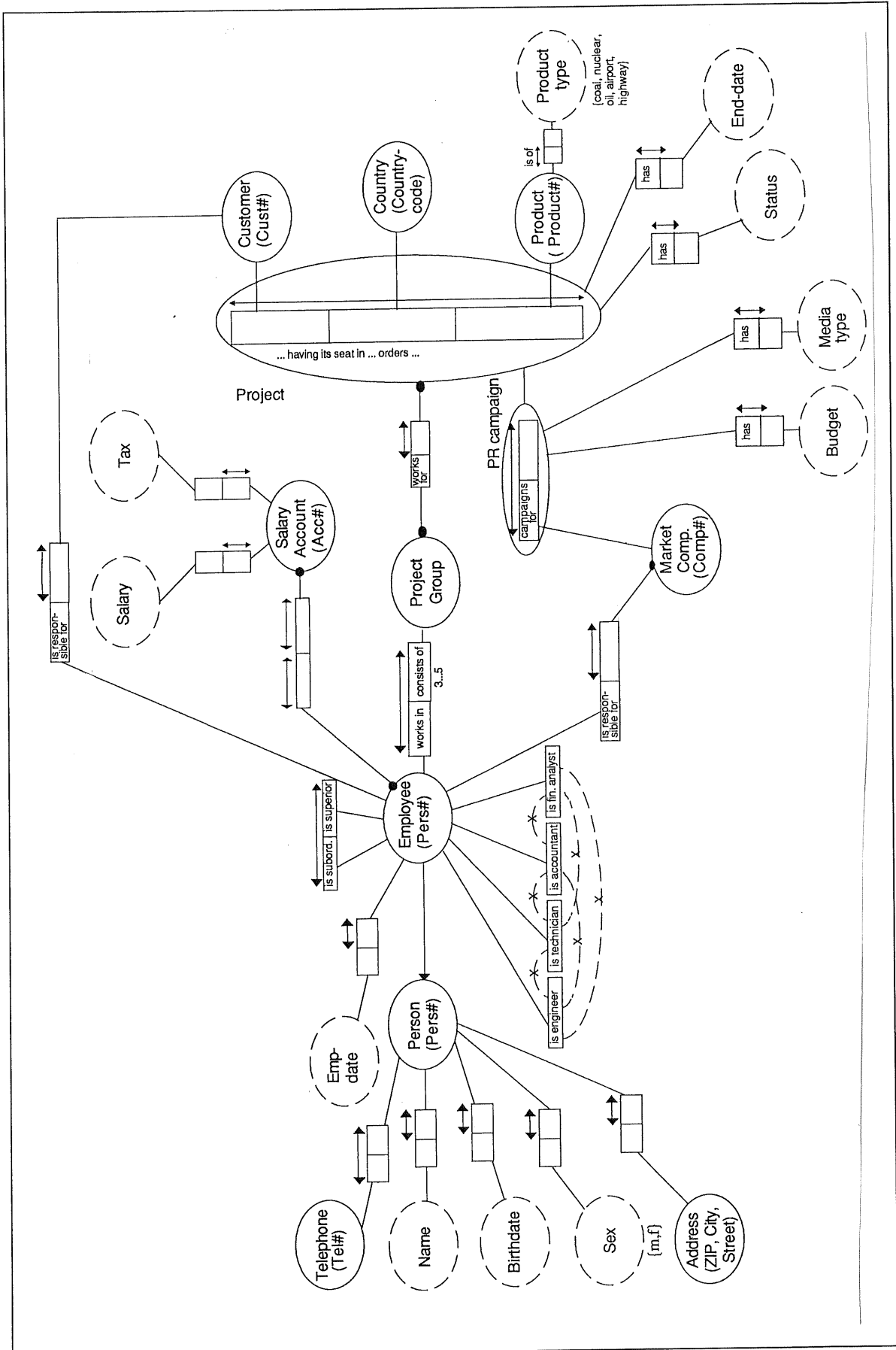


Fig. 26: The modelling example in NIAM

2.12 The Relational Model

The relational data model is a concept based upon the relational theory. The restrictive definition of mathematical relational theory is enhanced and concepts such as primary key and references between relations (referential integrity) are incorporated²⁸. However, according to the mathematical definition of a relation, no explicit references exist between relations. Hence, the referential integrity is not an original concept of the basic relational model. This is also true for the concept of primary key, since in the strict mathematical sense no element of a set has to be qualified as the identifier of a tuple to form a relation.

Major terms used in conjunction with the 'enhanced' relational model are the relation itself, tuple, attribute, domain and primary key²⁹. A relation consists of two parts, a heading and a body: The heading consists of a fixed set of attributes that correspond to exactly one of the underlying domains. The body is a time-varying set of tuples with each tuples being a set of attribute values pairs. Within a given relation there are no duplicate tuples; the tuples and attributes are unordered and attribute values may not be a repeating group³⁰. Domains can be interpreted as pools of values from which the actual values appearing in attributes are drawn. Every attribute must be defined on exactly one underlying domain.

The 'enhanced' relational data model includes two general integrity rules³¹ that address the aspect of data integrity: the entity integrity rule and the referential integrity rule. The entity integrity rule is expressed by the primary key concept. A primary key is a unique identifier for a relation and satisfies the uniqueness and minimality condition. The referential integrity rule is realized by the foreign key concept which requires that for every foreign key value there must exist a matching primary key value in a target relation. Thus, foreign key values represent entity references.

A further part of the relational model is its manipulative part that consists of the traditional set operations such as union, intersection, Cartesian product and special relational operations such as restriction, projection or join. These operations form the so-called relational algebra³². The relational language SQL (Structured query language) is used to formulate these relational operations on an implementation level. SQL comprises operations that define and manipulate data in relational form. It is both an interactive query language and a database programming language (since it can be embedded into programs) and always yields relations. The transformation of the theoretically defined relational operations into executable SQL statements is one of the major strengths of the relational model.

The relational data model as a data modelling concept is based upon a sound theoretical foundation. Nevertheless, it lacks both the expressiveness required in the phase of semantic data modelling (e.g. aggregation, hierarchical relations, generalization/specification, semantic integrity rules) and the support of application-specific, dynamic integrity constraints (e.g. automatic update of foreign key values, control of values assignment). These concepts cannot be expressed in the flat data structure of the relational model. The representation of relationships between information objects is restricted to the available construct, the relation itself. Thus, any (logical) relationship can only be expressed within a relation or by using the same values in two or more relations. Further structural information about data objects has to be represented through additional, mainly text-based constraints that are not part of the relational model.

In figure 27 the modelling example is depicted. The relations are represented by the attributes without specifying tuples.

<p>R. Person (<u>Pers#</u>, Name, Address, Birthdate, Sex, ...)</p> <p>R. Telephone (<u>Tel#</u>, ...)</p> <p>R. Telephone assignment (<u>Tel#</u>, <u>Pers#</u>, ...)</p> <p>R. Employee (<u>Pers#</u>, Emp-date, Acc#, ...)</p> <p>R. Salary Account (<u>Acc#</u>, Salary, Tax)</p> <p>R. Hierarchy (<u>Sup-Pers#</u>, <u>Sub-Pers#</u>, ...)</p> <p>R. Engineer (<u>Pers#</u>, ...)</p> <p>R. Technician (<u>Pers#</u>, ...)</p> <p>R. Accountant (<u>Pers#</u>, ...)</p> <p>R. Financial Analyst (<u>Pers#</u>, ...)</p> <p>R. Customer (<u>Cust#</u>, <u>Pers#</u>, ...)</p> <p>R. Country (<u>Country-code</u>, ...)</p> <p>R. Product (<u>Product#</u>, ...)</p> <p>R. Project (<u>Cust#</u>, <u>Country-code</u>, <u>Product#</u>, Group#, Status, End-date, ...)</p> <p>R. Project group (<u>Group#</u>, ...)</p> <p>R. Group assignment (<u>Pers#</u>, <u>Group#</u>, ...)</p> <p>R. Marketing company (<u>Comp#</u>, <u>Pers#</u>, ...)</p> <p>R. PR campaign (<u>Cust#</u>, <u>Country-code</u>, <u>Product#</u>, <u>Comp#</u>, Budget, Mediatype, ...)</p>	
---	--

Fig. 27: The modelling example in Relational expression

2.13 RM/T

The RM/T method introduced by Codd³³ is an extension of the relational model. In contrast to graphically-oriented network based representation schemes, the RM/T method is a logical representation scheme. It has no graphical constructs except the symbol for different types of relations. This is depicted in figure 28.

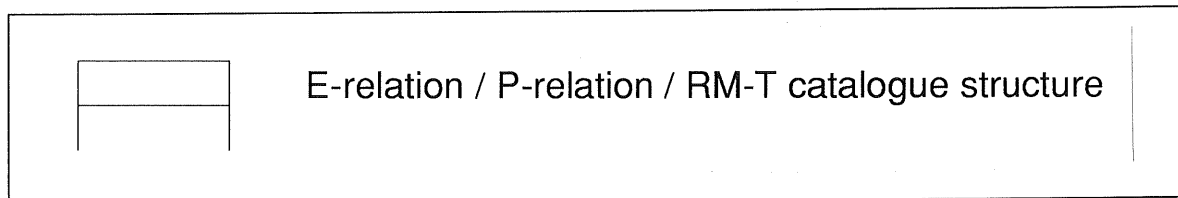


Fig. 28: Relation symbol used in RM/T

In comparison with the relational model the RM/T method encompasses a more extensive treatment of structural and integrity aspects and has its own set of special operators defined. The RM/T model's object modelling is based on an entity classification scheme to distinguish different semantic modelling constructs (e.g. kernel entities, characteristic entities and associative entities). With the help of these three entity classes, an expressiveness similar to the constructs introduced by Chen can be realized³⁴. However, the different entity types have to be further described with DDL expressions to make them distinguishable. The RM/T method has no distinction between an entity type and a relationship type. With respect to the system representation criteria, the RM/T method is rather comprehensive. It explicitly supports most of those criteria. The phenomenon of multi-valued attributes, repeating groups or existence dependency, for instance, is addressed with the notion of characteristic entity³⁵.

One specific feature of the RM/T method is its formal representation scheme. Abstract objects (i.e. entities) are represented by permanent surrogates (i.e. system controlled primary keys). Each entity type has an associated one-column E-relation that holds the surrogates of the objects currently populating that type. Besides, there are so-called P-relations that represent the properties (or attributes) of a given entity type. Beyond these two constructs, there are no additional means of describing the RM/T, i.e. there are no graphical diagramming techniques that support the RM/T approach except the representation in tables.

Concerning the degree of integrity support there is a vast set of formally defined integrity rules available in RM/T that restrict the utilization of the RM/T objects and their interactions³⁶. These integrity rules have to be specified in the appropriate schema

description language of the database model. Furthermore, a formal set of operators is defined and a so-called RM/T catalogue is introduced to store all the constructs and their relationships. This catalogue includes a set of graph-relations whose function is to represent the various connections among relations in the database - for example the connection between an E-relation and its corresponding P-relation(s). Figure 29 depicts a partial representation of the modelling example.

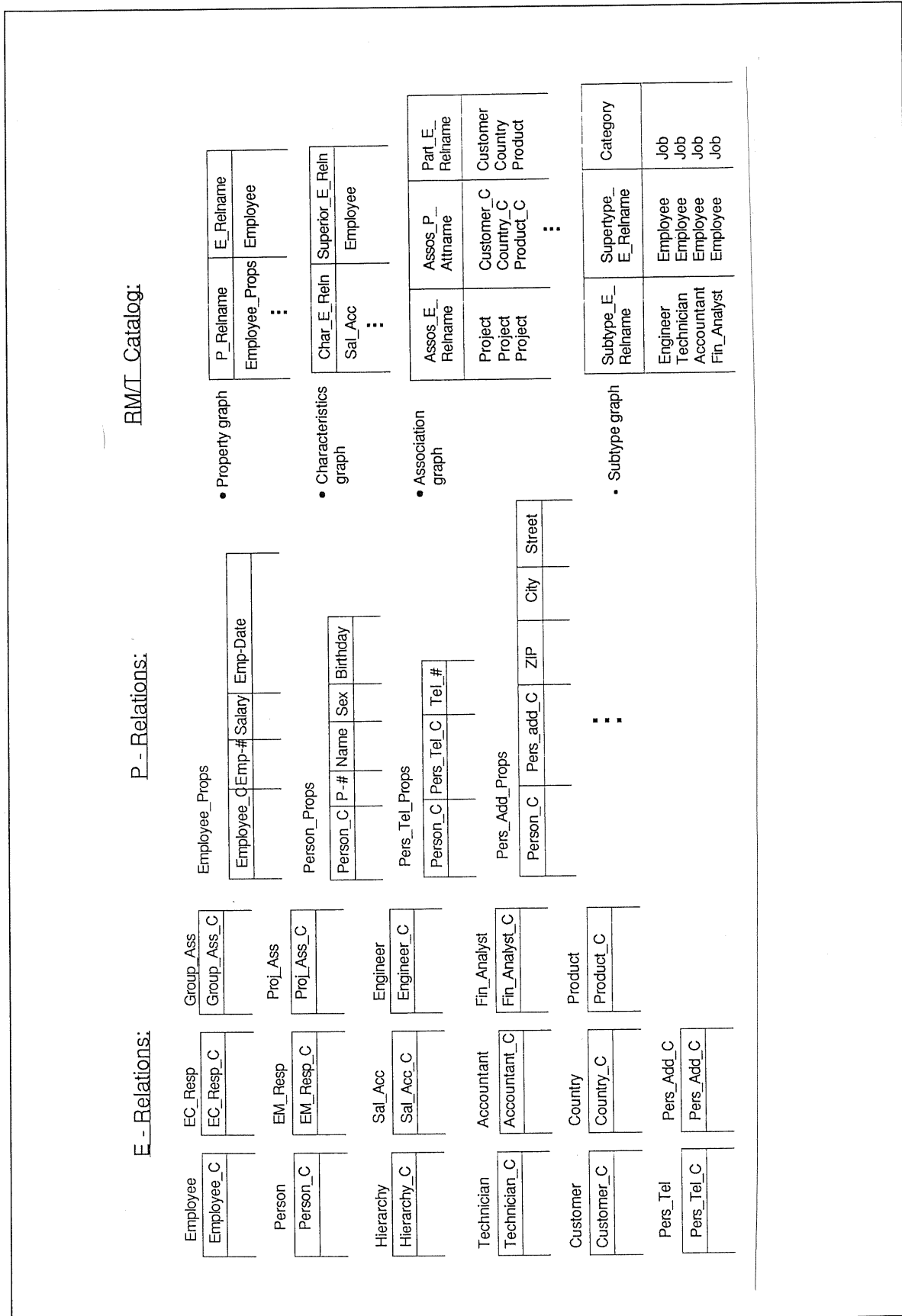


Fig. 29: The modelling example in RM/T

2.14 SAM*

The Semantic Association Model (SAM*)³⁷ is a highly structured semantic modelling methodology. It is particularly tailored to incorporate complex data types used in engineering applications. The graph-based network representation scheme is based on the notion of concept and association. Its constructs are depicted in figure 30.

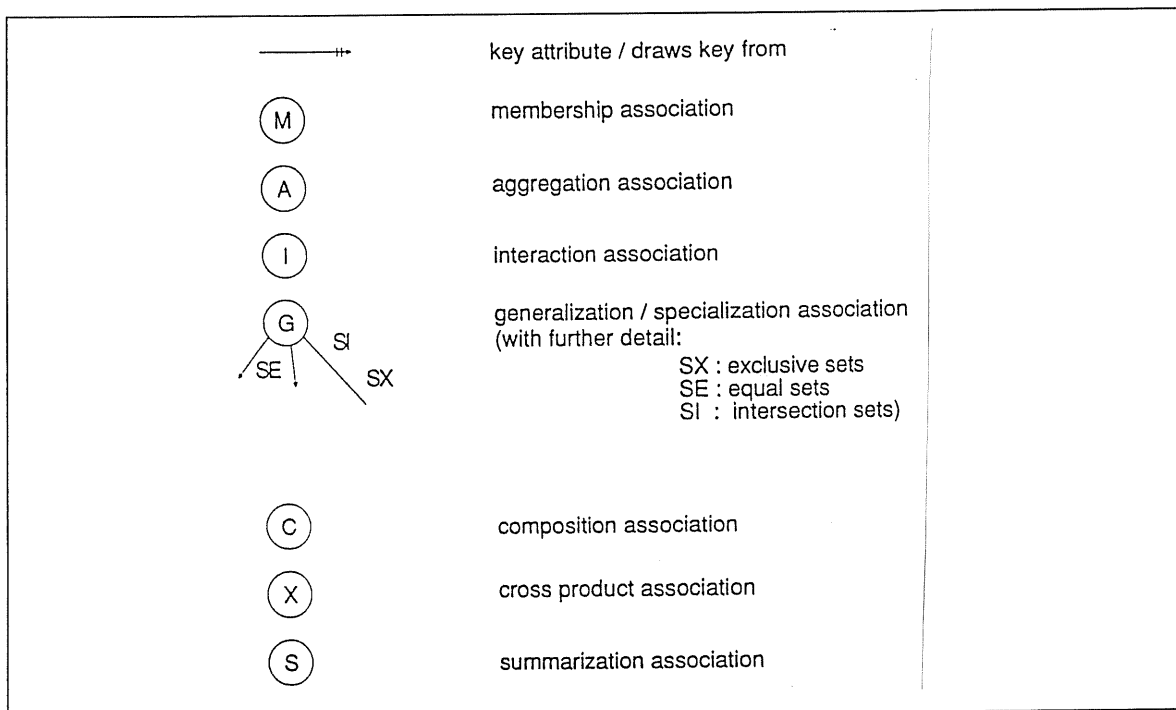


Fig. 30: SAM* constructs and relationships

Atomic and non-atomic concepts are distinguished³⁸. Atomic concepts include integer, real and character types as well as structured programming language data types such as arrays, time serious and sets³⁹. The idea of atomic concepts allows the definition of complex data types (e.g. vectors) and the provision of procedural operators and functions (e.g. three-dimensional representation and rotation) in a data language for testing and processing data of these types. This comprehensive data type definition functionality of SAM* is one of its particularities. While atomic concepts cannot be further decomposed and have no meaning independent of an application, non-atomic concepts are described in terms of atomic or other non-atomic concepts. Physical or abstract objects are represented using non-atomic concepts. The grouping of atomic or non-atomic concepts to describe another non-atomic concept is called association. There are seven different types of associations⁴⁰ defined that have specific uses and (database-) operations specially tailored to them. The membership associations at the root of the graph, indicated by M in the node, allows the specification of domain or attribute range of a type. It models the natural language expression

"is_a_member_of". In figure 31, for instance, the M-node City shows that an address draws its city-instances out of a set of different cities.

To provide a description of an entity type, the so-called aggregation association is introduced, denoted by an A in the node. The key of an aggregation association is indicated by the two dashes on the connector pointing to the M-association of the key attribute. The interaction association (denoted by an I) resembles the aggregation operation in other semantic modelling methodologies and is equivalent to the relationship type as defined by Chen. However, interaction associations can use other interaction associations as their components⁴¹. Cardinality conditions may be assigned to the interaction. Furthermore, the generalization association is used to model is-a relationships. These relationships, that not only allow the inheritance of attributes but also of knowledge and procedural rules, can be specified with respect to the disjointness of the subtypes. The composition, cross-product and summarization association complete the seven association types of SAM*. The composition association is a grouping of similar or dissimilar concept types with each occurrence of that node holding a set of sets⁴². The cross-product- and summarization association are typically used in statistical databases⁴³. The proposed association types can be nested and recursively defined to express highly complex data relationships.

Compared to other semantic modelling methodologies SAM* offers a comprehensive set of explicitly labelled constructs and operators supporting the database oriented schema definition. It addresses the question of conceptual modelling at a very detailed level since domain and attribute specifications are incorporated. The approach focuses on a detailed analysis in the early design stages which might lead to an 'over-specification'.

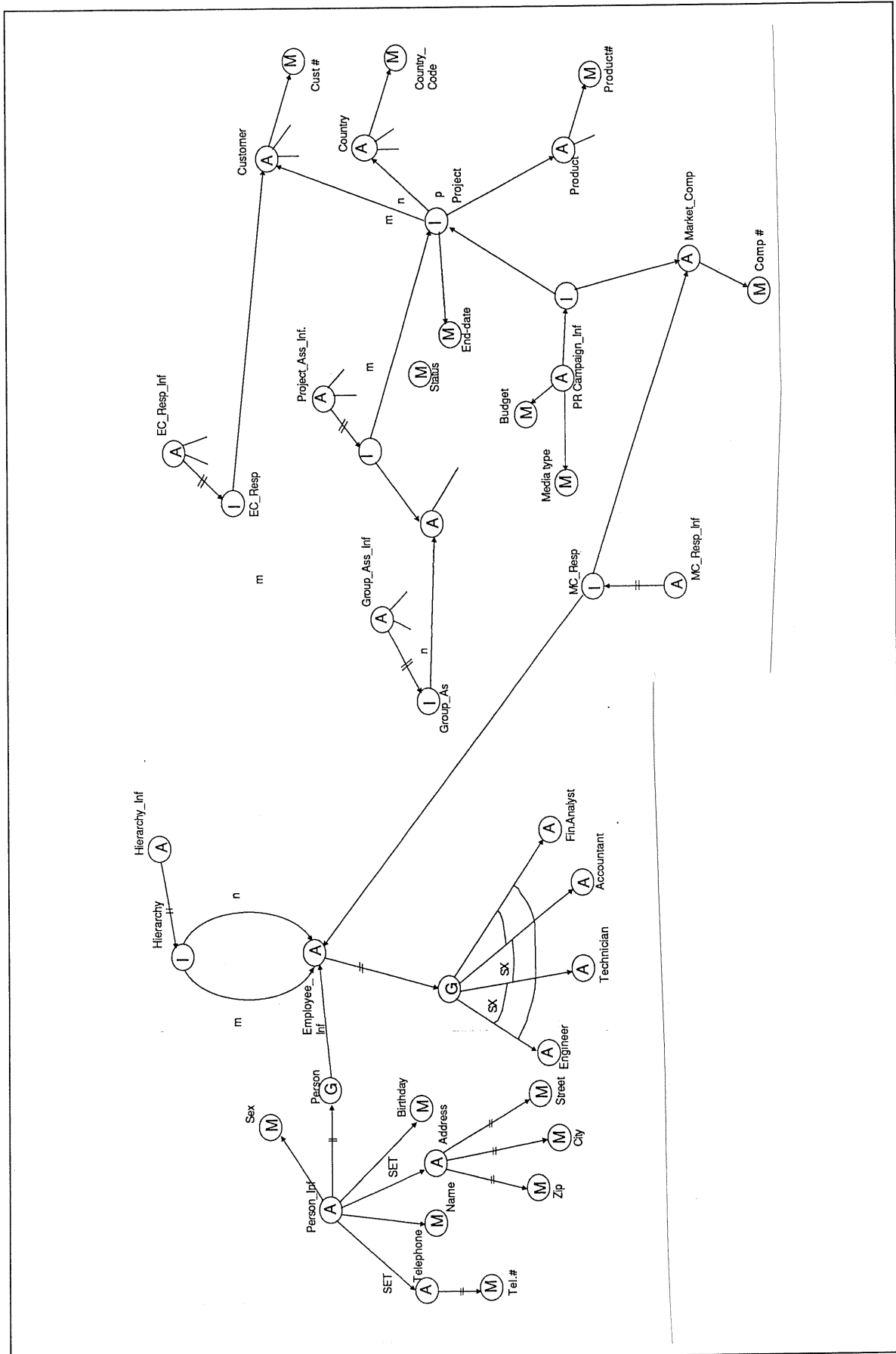


Fig. 31: The modelling example in SAM*

2.15 SERM

SERM⁴⁴ is a data modelling method focusing on the visualization of existence dependencies within the logical data structures. Therefore the SER schemes are graphically presented as diagrams with semi-hierarchical structure. The original idea of SERM was to combine the ERM with the concept of a semi-hierarchical graph. Thus the ERM approach is the methodological basis of SERM and consistent ER diagrams can be transformed into an equivalent SER diagram.

SERM forms the methodological basis for the SOM⁴⁵ (semantic object model) method which is an object-oriented approach for information modelling. The SERM method does not explicitly determine a life cycle concept or modelling process description but there are concrete rules how to model a SER diagram especially concerning the graphical representation of the object types and their inter-relationships. Compared to the basic ER approach and extended ER approaches the SERM introduces a new kind of construct - the so-called entity-relationship type (ER-type). This ER-type is characterized as an entity type which has a minimum and a maximum cardinality of one (1,1) within a relationship. Re-interpreted (aggregated) relationship types of extended ER approaches are transformed into an ER-type. SERM also contains the elements entity type (E type) and relationship type (R type). The SER diagram consists of nodes and arcs. Possible nodes are entity types, relationship types or entity-relationship types. The arcs describe four different min/max-cardinalities which are distinguished by different graphical representation. SERM supports (0,1), (0,n), (1,n) and (1,1) cardinalities, as can be seen in figure 32.

There are rules how to locate the different object types of a SER diagram. Thus, a dependent information object is always located on the right side of its superior information object. The SERM introduces two kinds of generalization/specialization hierarchies. It differentiates between subclass hierarchy and is-a hierarchy by considering the completeness and disjointness of the subclasses. SERM diagrams do not contain the representation of attributes.

SERM's focus on existence dependencies supports the transformability into underlying modelling levels e.g. the transformation into the relational model. Furthermore, SERM's explicit focus on existence dependencies defines certain aspects which are relevant for the transaction management of the underlying database system (e.g. delete-operations). Therefore, SERM is closer to the technical implementation than the original ER approach.

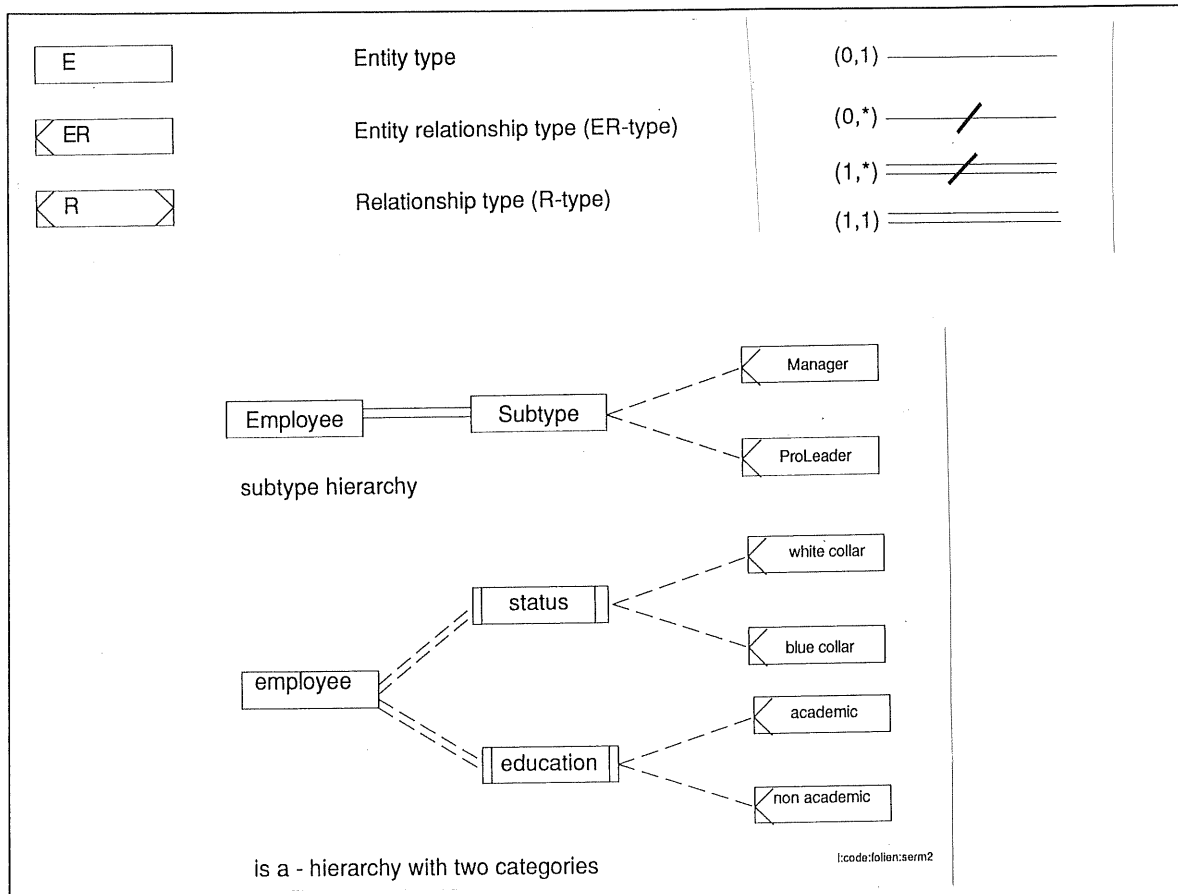


Fig. 32: SERM constructs and relationships

The SERM is part of a data modelling method which comprises the modelling of conceptual SER schemes, the generation of database schema for different database managements systems, the specification and modelling of external schemes and the program generation of data modules in different higher level programming languages. This method is supported by a knowledge-based development tool which also generates the SERM diagrams.

In the example depicted in figure 33, the attributes describing the different object types cannot be graphically represented. Additionally, a min/max-cardinality of the kind (3,5) cannot be expressed.

One advantage of the SER-method was that the dependencies between the different information objects of the example can be recognized from the diagram. The effort to graphically arrange the information objects of the SER-diagram is higher than the modelling effort using other methods. Using the SER modelling technique information can be lost. E.g. it cannot be seen from the SER-diagram if an E/R type originally was an aggregated relationship or only an entity type with a special cardinality.

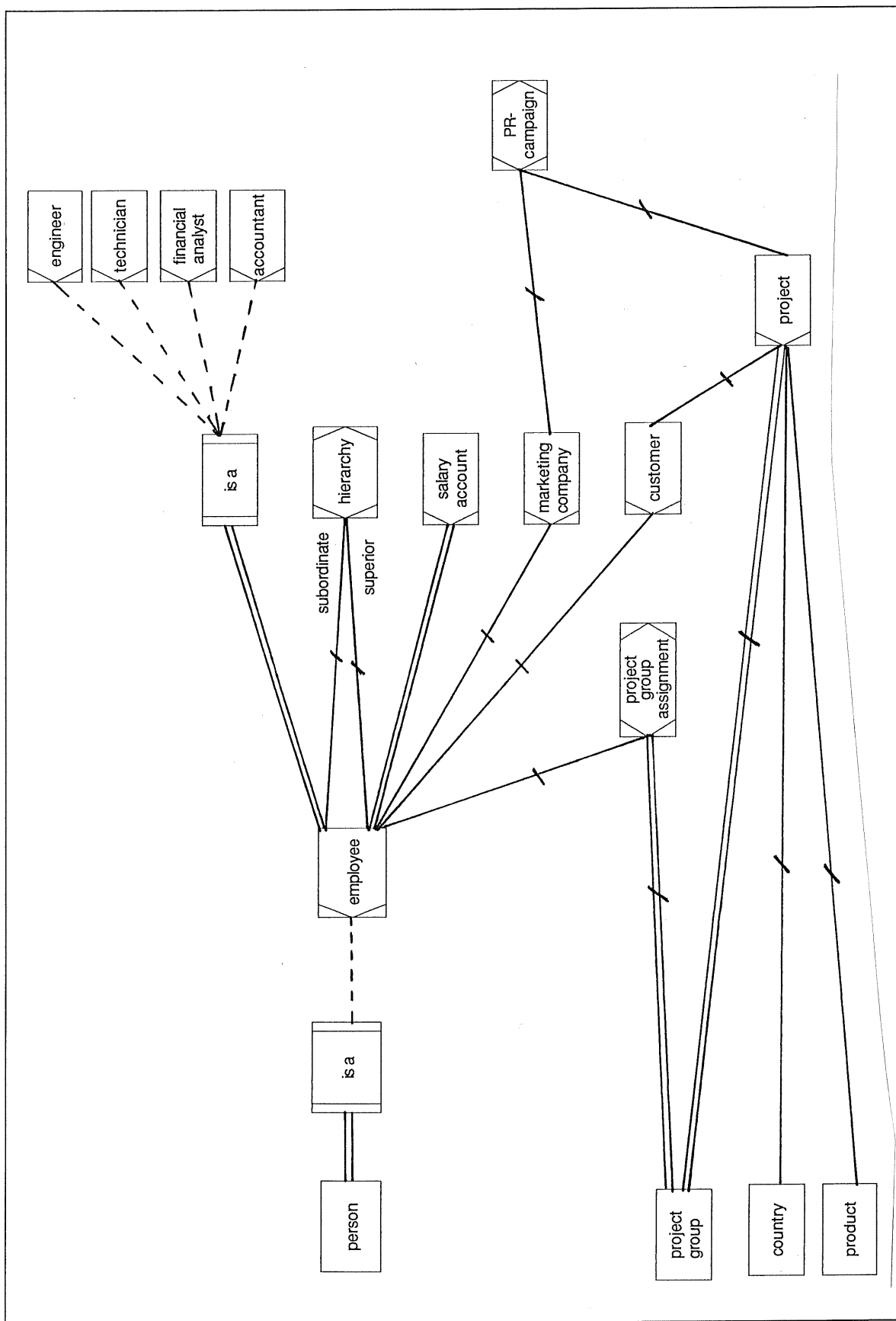


Fig. 33: The modelling example in SERM

3. CONCLUSION

Data modelling is a means both to communicate between humans who have to exchange views concerning the most important facts that govern an enterprise and to communicate between human and computer with the aim of efficiently designing the implementation structures of an enterprise's information system. It is a core application for the design and the maintenance of information systems. This can also be derived from the large number of data modelling methodologies that were available for this study. It has to be recognized that every modelling methodology is limited in its expression capability. From this, knowledge representation also draws its strengths because an unlimited expression capability would mean that computer support would scarcely be possible.

It is in the area of application where almost all of the methodologies display weaknesses. And there it is, where the future research has to focus upon. Two main streams can be discerned. The first is the definition of the modelling process. It has to contain a step-by-step definition of how to carry out the modelling case. This task is made more difficult, because modelling can be applied for different objectives and from persons with different modelling expertise. Therefore a set of modelling techniques has to be described which can be flexibly combined by the modeller and the description of the steps has to support possible modifications which an experienced modeller would like to make to the process. Shortcuts and parallel activities have to be allowed. The second main stream, the research work has to focus upon, is to make available the computer support for these modelling activities. This is not a mere question of implementation, it requires to build the modelling methodologies and the modelling techniques in a way fitting for human-computer interaction.

Literature:

¹ Scheer, A.-W.: Architektur Integrierter Informationssysteme, Berlin et al., 1991.

² Scheer, A.-W.: Wirtschaftsinformatik - Informationssysteme im Industriebetrieb. 3. edition, Berlin et al, 1990, p.7.

³ Chen P. : The Entity-Relationship Model- Toward a Unified View of Data

⁴ Major modelling methodologies incorporated into the CIM-OSA framework are the M* method described in: Di Leva, A.; Vernadat, F.; Bizier, D.J.M.: Information System Analysis and Conceptual Database Design in Production Environments with M*. In: Computer in Industry 9(1987), p. 183-217, and the SAM* method described in this paper.

⁵ For a detailed discussion see: Jorysz, H.R.; Vernadat, F.B.; CIM-OSA Part 1: total enterprise modelling and function view. In: International Journal of Computer Integrated Manufacturing, 3(1990)3/4, pp. 144-156.

⁶ Jorysz H.R., Vernadat F.B.: CIM-OSA Part 1: total enterprise modelling and function view. In: Int. Journal of Computer Integrated Manufacturing, 1990 Vol. 3. p.147.

- ⁷ Jorysz, H.R.; Vernadat, F.B.; CIM-OSA Part 2: Information view. In: International Journal of Computer Integrated Manufacturing, 3(1990)3/4, p. 157.
- ⁸ A more precise definition of a total respectively partial attribute can be found in chapter ??.
- ⁹ The meta model of the information view concepts can be found in: Jorysz, H.R.; Vernadat, F.B.; CIM-OSA Part 2: Information view. In: International Journal of Computer Integrated Manufacturing, 3(1990)3/4, p. 159.
- ¹⁰ Czejdo, B; Elmasri, R.; Rusinkiewicz, M.; Embley, D.W.; A Graphical Manipulation Language for an Extended Entity Relationship Model. In: IEEE Computer 23 (1990)3, p. 26-36
- ¹¹ Czejdo, B; Elmasri, R.; Rusinkiewicz, M.; Embley, D.W.; A Graphical Manipulation Language for an Extended Entity Relationship Model. In: IEEE Computer 23 (1990)3, pp. 30.
- ¹² Elmasri, R.; Weeldryer, J.; Hevner, A.: The category concept: An extension to the entity-relationship model. In Data & Knowledge Engineering 1 (1985), pp. 75-116.
- ¹³ Elmasri, R.; Weeldryer, J.; Hevner, A.: The category concept: An extension to the entity-relationship model. In Data & Knowledge Engineering 1 (1985), p.78-79.
- ¹⁴ Product Data Representation and Exchange - Part 11: The EXPRESS Language Reference Manual. International Organization for Standardization. Subcommittee 4. Industrial Data and Global Manufacturing Programming languages, p. 26.
- ¹⁵ See aforementioned reference, p. 52.
- ¹⁶ See aforementioned reference, p 133.
- ¹⁷ However, it may be marked by an asteriks (*) in the diagram, that a referencing rule in the EXPRESS language exists.
- ¹⁸ Molter, E.: Modellierung eines Fallbeispiels mit IDEF. Seminar paper ed.: Scheer, A.-W.; Saarbrücken, 1991 p.2.
- ¹⁹ EDV Studio Ploenzke (ed.): ISOTEC: ISA/FSA Manual Version 2. Wiesbaden.
- ²⁰ EDV Studio Ploenzke (ed.): ISOTEC: ISOTEC: SSF Manual Version 2. Wiesbaden.
- ²¹ EDV Studio Ploenzke (ed.): ISOTEC: ISA/FSA Manual Version 2. Wiesbaden, p.3.
- ²² Martin, J.: Information Engineering. Book I: Introduction. Englewood Cliffs, New Jersey 1990, p.57.
- ²³ Martin, J.: Information Engineering. Book II: Planning & Analysis. Englewood Cliffs, New Jersey 1990, pp. 13-196.
- ²⁴ Martin, J.: Information Engineering. Book II: Planning & Analysis. Englewood Cliffs, New Jersey 1990, pp. 197-294.
- ²⁵ Tardieu, H. u. Rochfeld, A.: La methode Merise. Bd. 1, Paris 1983. Tardieu, H. u. Rochfeld, A.: Merise: An Information system Design and Development Methodology. In: Information & Management, Nr. 6 (1983), pp. 143- 159. Rochfeld, A.: Merise, an Information System Design and Development Methodology. Proceedings of the Fith International Conference on Entity Relationship Approach, Hrsg.: S. Spaccapietra. Amsterdam 1987, pp. 489-528. Banos, D. u. Malbosc, D.: Merise pratique Bd.1 u. Bd.2. Paris 1988.
- ²⁶ Nijssen, G. M.; Halpin, T. A.: Conceptual Schema and Relational Database Design. A Fact Oriented Approach New York et al. 1989. Reusch, P.; Wintraecken, J.-J.: Systemanalyse und Systemspezifikation. Mannheim et al. 1990.
- ²⁷ Nijssen, G. M.; Halpin, T. A.: Conceptual Schema and Relational Database Design. A Fact Oriented Approach New York et al. 1989, p. 32.
- ²⁸ Date, C. J.: An introduction to Data Base Systems. Vol. 1. 5th. Ed. Reading, Mass. 1990, p. 284.
- ²⁹ For a detailed definition see: Date, C.J.: An Introduction to Data Base Systems, p. 249 ff.
- ³⁰ The definition of a relation in the mathematical sense, however, does allow attributes of which the domain itself is a relation.

- ³¹ Date refers to these integrity rules as so-called meta-rules, see Date, C. J.: An introduction to Data Base Systems. Vol.1. 5th. Ed. Reading, Mass. 1990, p. 288. Some authors define third meta-rule: The values for attributes must be drawn from the according domain.
- ³² An Algebra is defined as a system containing a non-empty set and a family of operations within this set, cf. Schlageter, G; Stucky, W.: Datenbanksysteme: Konzepte und Modelle. Stuttgart 1983, p.139.
- ³³ Codd, E. F.: Extending the Database Relational Model to Capture More Meaning. In: ACM Transaction on Database Systems 4(1979)4, see also Date, C. J.: An introduction to Data Base Systems. Vol.II. 2nd. ed. Reading, Mass. 1985, pp. 241.
- ³⁴ Date, C. J.: An introduction to Data Base Systems. Vol.I; 5th. ed. Reading, Mass. 1990, p. 595.
- ³⁵ Date, C. J.: An introduction to Data Base Systems. Vol.II. 2nd. ed. Reading, Mass. 1985, p. 254.
- ³⁶ Date, C. J.: An introduction to Data Base Systems. Vol.II. 2nd. ed. Reading, Mass. 1985, p. 242.
- ³⁷ Su, S.Y.W., Modelling integrated manufacturing data with SAM*. In: IEEE Computers, Jan.(1986), p. 34-48.
- ³⁸ Su, S.Y.W., Modelling integrated manufacturing data with SAM*. In: IEEE Computers, Jan.(1986), p. 37.
- ³⁹ Hull, R; King, R.; Semantic Database Modeling: Survey, Applications, and Research Issues. In: ACM Computing Surveys, Vol. 19, No. 3 September 1987, p. 237.
- ⁴⁰ Su, S.Y.W., Modelling integrated manufacturing data with SAM*. In: IEEE Computers, Jan.(1986), p. 38.
- ⁴¹ Su, S.Y.W., Modelling integrated manufacturing data with SAM*. In: IEEE Computers, Jan.(1986), p. 42.
- ⁴² Su, S.Y.W., Modelling integrated manufacturing data with SAM*. In: IEEE Computers, Jan.(1986), p. 45.
- ⁴³ for a further discussion see: Johnson, R.R.: Modelling Summary Data. In: Proceedings of the International Conference on Very Large Data Bases, 1984.
- ⁴⁴ Sinz, E.J.: Das Strukturierte Entity-Relationship-Modell (SER-Modell). In: Angewandte Informatik 5/1988, pp. 1991-202. Sinz, E.J.: Konzeptionelle Datenmodellierung im Strukturierten Entity-Relationship-Modell (SER-Modell). In: Müller-Ettrich, G.(Hrsg.): Effektives Datendesign, Verlag Rudolf Müller, Köln 1989, pp. 76-108. Sinz, E.J.: Das Entity-Relationship-Modell und seine Erweiterungen. In: Handwörter der Modernen Datenverarbeitung.152 (1990) pp. 17-29.
- ⁴⁵ Sinz, E.J.: Datenmodellierung betrieblicher Probleme und ihre Unterstützung durch ein wissensbasiertes Entwicklungssystem. Habilitationsschrift Universität Regensburg 1987. O. Ferstl u. E. Sinz: Objektmodellierung betrieblicher Informationsmodelle im Semantischen Objektmodell (SOM). In: Bamberger Beiträge zur Wirtschaftsinformatik Nr. 1/1990.

Die Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) im Institut für empirische Wirtschaftsforschung an der Universität des Saarlandes erscheinen in unregelmäßiger Folge.

* Die Hefte 1 - 31 werden nicht mehr verlegt.

- Heft 32: A.-W. Scheer: Einfluß neuer Informationstechnologien auf Methoden und Konzepte der Unternehmensplanung, März 1982, Vortrag anläßlich des Anwendergespräches "Unternehmensplanung und Steuerung in den 80er Jahren in Hamburg vom 24. - 25.11.1981
- Heft 33: A.-W. Scheer: Dispositio- und Bestellwesen als Baustein zu integrierten Warenwirtschaftssystemen, März 1982, Vortrag anläßlich des gdi-Seminars "Integrierte Warenwirtschafts-Systeme" in Zürich vom 10. - 12. Dezember 1981
- Heft 34: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS - Ein Ansatz zur Entwicklung prüfungsgerechter Software-Systeme, Mai 1982
- Heft 35: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Konzept einer computergestützten Prüfungsumgebung, Juli 1982
- Heft 36: A.-W. Scheer: Rationalisierungserfolge durch Einsatz der EDV - Ziel und Wirklichkeit, August 1982, Vortrag anläßlich der 3. Saarbrücker Arbeitstagung "Rationalisierung" in Saarbrücken vom 04. - 06. 10.1982
- Heft 37: A.-W. Scheer: DV-gestützte Planungs- und Informationssysteme im Produktionsbereich, September 1982
- Heft 38: A.-W. Scheer: Interaktive Methodenbanken: Benutzerfreundliche Datenanalyse in der Marktforschung, Mai 1983
- Heft 39: A.-W. Scheer: Personal Computing - EDV-Einsatz in Fachabteilungen, Juni 1983
- Heft 40: A.-W. Scheer: Strategische Entscheidungen bei der Gestaltung EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 41: H. Krcmar: Schnittstellenprobleme EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 42: A.-W. Scheer: Factory of the Future, Vorträge im Fachausschuß "Informatik in Produktion und Materialwirtschaft" der Gesellschaft für Informatik e. V., Dezember 1983
- Heft 43: A.-W. Scheer: Einführungsstrategie für ein betriebliches Personal-Computer-Konzept, März 1984
- Heft 44: A.-W. Scheer: Schnittstellen zwischen betriebswirtschaftlicher und technische Datenverarbeitung in der Fabrik der Zukunft, Juli 1984
- Heft 45: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Ein Werkzeug zur Messung der Qualität von Software-Systemen, August 1984
- Heft 46: H. Krcmar: Die Gestaltung von Computer am-Arbeitsplatz-Systemen - ablauforientierte Planung durch Simulation, August 1984
- Heft 47: A.-W. Scheer: Integration des Personal Computers in EDV-Systeme zur Kostenrechnung, August 1984

- Heft 48: A.-W. Scheer: Kriterien für die Aufgabenverteilung in Mikro-Mainframe Anwendungssystemen, April 1985
- Heft 49: A.-W. Scheer: Wirtschaftlichkeitsfaktoren EDV-orientierter betriebswirtschaftlicher Problemlösungen, Juni 1985
- Heft 50: A.-W. Scheer: Konstruktionsbegleitende Kalkulation in CIM-Systemen, August 1985
- Heft 51: A.-W. Scheer: Strategie zur Entwicklung eines CIM-Konzeptes - Organisatorische Entscheidungen bei der CIM-Implementierung, Mai 1986
- Heft 52: P. Loos, T. Ruffing: Verteilte Produktionsplanung und -steuerung unter Einsatz von Mikrocomputern, Juni 1986
- Heft 53: A.-W. Scheer: Neue Architektur für EDV-Systeme zur Produktionsplanung und -steuerung, Juli 1986
- Heft 54: U. Leismann, E. Sick: Konzeption eines Bildschirmtext-gestützten Warenwirtschaftssystems zur Kommunikation in verzweigten Handelsunternehmungen, August 1986
- Heft 55: D. Steinmann: Expertensysteme (ES) in der Produktionsplanung und -steuerung (PPS) unter CIM-Aspekten, November 1987, Vortrag anlässlich der Fachtagung "Expertensysteme in der Produktion" am 16. und 17.11.1987 in München
- Heft 56: A.-W. Scheer: Enterprise wide Data Model (EDM) as a Basis for Integrated Information Systems, Juli 1988
- Heft 57: A.-W. Scheer: Present Trends of the CIM Implementation (A qualitative Survey) Juli 1988
- Heft 58: A.-W. Scheer: CIM in den USA - Stand der Forschung, Entwicklung und Anwendung, November 1988
- Heft 59: R. Herterich, M. Zell: Interaktive Fertigungssteuerung teilautonomer Bereiche, November 1988
- Heft 60: A.-W. Scheer, W. Kraemer: Konzeption und Realisierung eines Expertenunterstützungssystems im Controlling, Januar 1989
- Heft 61: A.-W. Scheer, G. Keller, R. Bartels: Organisatorische Konsequenzen des Einsatzes von Computer Aided Design (CAD) im Rahmen von CIM, Januar 1989
- Heft 62: M. Zell, A.-W. Scheer: Simulation als Entscheidungsunterstützungsinstrument in CIM, September 1989
- Heft 63: A.-W. Scheer: Unternehmens-Datenbanken - Der Weg zu bereichsübergreifenden Datenstrukturen, September 1989
- Heft 64: C. Berkau, W. Kraemer, A.-W. Scheer: Strategische CIM-Konzeption durch Eigenentwicklung von CIM-Modulen und Einsatz von Standardsoftware, Dezember 1989
- Heft 65: A. Hars, A.-W. Scheer: Entwicklungsstand von Leitständen^[1], Dezember 1989
- Heft 66: W. Jost, G. Keller, A.-W. Scheer: CIMAN - Konzeption eines DV-Tools zur Gestaltung einer CIM-orientierten Unternehmensarchitektur, März 1990
- Heft 67: A.-W. Scheer: Modellierung betriebswirtschaftlicher Informationssysteme (Teil 1: Logisches Informationsmodell), März 1990

- Heft 68: W. Kraemer: Einsatzmöglichkeiten von Expertensystemen in betriebswirtschaftlichen Anwendungsgebieten, März 1990
- Heft 69: A.-W. Scheer, R. Bartels, G. Keller: Konzeption zur personalorientierten CIM-Einführung, April 1990
- Heft 70: St. Spang, K. Ibach: Zum Entwicklungsstand von Marketing-Informationssystemen in der Bundesrepublik Deutschland, September 1990
- Heft 71: D. Aue, M. Baresch, G. Keller: URME_L, Ein Unte_RnnehmensModE_Llierungsansatz, Oktober 1990
- Heft 72: M. Zell: Datenmanagement simulationsgestützter Entscheidungsprozesse am Beispiel der Fertigungssteuerung, November 1990
- Heft 73: A.-W. Scheer, M. Bock, R. Bock: Expertensystem zur konstruktionsbegleitenden Kalkulation, November 1990
- Heft 74: R. Bartels, A.-W. Scheer: Ein Gruppenkonzept zur CIM-Einführung, Januar 1991
- Heft 75: M. Nüttgens, St. Eichacker, A.-W. Scheer: CIM-Qualifizierungskonzept für Klein- und Mittelunternehmen (KMU), Januar 1991
- Heft 76: Ch. Houy, J. Klein: Die Vernetzungsstrategie des Instituts für Wirtschaftsinformatik - Migration vom PC-Netzwerk zum Wide Area Network (noch nicht veröffentlicht)
- Heft 77: W. Kraemer: Ausgewählte Aspekte zum Stand der EDV-Unterstützung für das Kostenmanagement: Modellierung benutzerindividueller Auswertungssichten in einem wissensbasierten Controlling-Leitstand, Mai 1991
- Heft 78: H. Heß: Vergleich von Methoden zum objektorientierten Design von Softwaresystemen, August 1991
- Heft 79: A.-W. Scheer: Konsequenzen für die Betriebswirtschaftslehre aus der Entwicklung der Informations- und Kommunikationstechnologien, Mai 1991
- Heft 80: G. Keller, J. Kirsch, M. Nüttgens, A.-W. Scheer: Informationsmodellierung in der Fertigungssteuerung, August 1991
- Heft 81: A.-W. Scheer: Papierlose Beratung - Werkzeugunterstützung bei der DV-Beratung, August 1991
- Heft 82: C. Berkau: VOKAL (System zur Vorgangskettendarstellung und -analyse) - Struktur der Modellierungsmethode - Juni 1991 (wird nicht verlegt)
- Heft 83: A. Hars, R. Heib, Ch. Kruse, J. Michely, A.-W. Scheer: Concepts of Current Data Modelling Methodologies - Theoretical Foundations - 1991
- Heft 84: A. Hars, R. Heib, Ch. Kruse, J. Michely, A.-W. Scheer: Concepts of Current Data Modelling Methodologies - A Survey - 1991
- Heft 85: W. Hoffmann, M. Nüttgens, A.-W. Scheer, St. Scholz: Das Integrationskonzept am CIM-TTZ Saarbrücken (Teil 1: Produktionsplanung), Oktober 1991
- Heft 86: A.-W. Scheer: Koordinierte Planungsinseln: Ein neuer Lösungsansatz für die Produktionsplanung, November 1991