# University Information Systems
# Design and Implementation Aspects

Alexandru Lelutiu

Technical University of Cluj-Napoca
Department of Computer Science
Baritiu 28, RO-3400 Cluj-Napoca, Romania
lelutiu@cs.utcluj.ro

**Abstract:** This paper addresses two key issues in the design and implementation of University Information System (UIS): the importance of the initial approach and the impact of the implementation strategy on the management level of the UIS. Firstly, we discuss the necessity of a high level UIS conception and its influence on the final UIS quality. Secondly, we focus on the priority of computer data sources against the user interfaces, driven by a sound identification of natural user data sources which guarantees data quality in the sense of information uniqueness and actuality. The subject is strictly related to the correct definition of the main data collections by viewing their owners not only as clients of these resources but also responsible for their quality. Finally, a UIS prototype is shortly presented, which has been developed based on that conception.

## 1 Introduction

This presentation reports results of a four-year period of activity in University Information System (UIS) development. The project focussed on both, a particular academic approach and a practical experience in the implementation of an UIS at the Technical University in Cluj-Napoca. Our university structure includes eight faculties and a college which together consist of nearly 50 departments. The project is financially supported by the Romanian Government and by the World Bank in the framework of an international grant.

The wide spreading preoccupations involved in such project determine not only design but also research activities, especially in the domains of Database and Information Systems as well as Software Engineering w.r.t. the enhancement of the level of conceptual system design and the development of a productive design environment. A didactic activity is also related to our UIS, because a lot of students were being implied in module design and implementation during their diploma work preparation. High technology topics focused our attention like active and temporal databases, abstract object roles implementation, data model construction, UFI prototypes development. Moreover, concrete aspects related to the implementation of complex systems like computer and user data sources definition based on standards in software development and the quality of database information systems were studied. Some important ideas related to these topics

will be discussed within this paper, grouped in two parts, design and implementation aspects. Starting with the priority of data structures design the paper presents the persistence of the interest related to the data sources from the conceptual to the implementation phase. Two different points of view concerning the data sources are defined. One related to the design of computer data sources and the second to the identification and implementation of the actual and concrete user data sources.

## 1. UIS development Technology

One of the first questions in our approach was to decide if the UIS should be designed and implemented as

- a unique database system (centralized or distributed) viewed as a main computer data source or
- a couple of multiple application systems which communicate using appropriate data transfer services.

We decided to adopt the first approach. The reason for that is that a university education system has to be viewed as a production system. "Production" is used here in a rather general sense as has been claimed already in 1968 by Elmaghraby [El66], like "any activity that increase the utility of an object or a service". From a such starting point it follows that various aspects of the education process have to be correlated continuously. In particular, it must be possible at any time during an education activity to access and to assess the balance between the teaching task and the involved cost. The computer data source that provides the needed information for such comparisons must dispose a high degree of integration to assure a permanent on about all education processes deployed at different levels like departments, faculties and university. In this vision the UIS is able to offer the possibility of tasks and costs computation not only at the organizational units level, but also w.r.t. each member of the teaching staff. In contrast to that, individual application modules implemented in some special activity fields (personal administration, school-status administration, financial administration etc.) would make it more difficult to gain a global perspective on the education process, without considering the difficulties that arise from the necessary data transformation services.

Such a computer implemented data source in form of a database must satisfy a set of constraints related to

- the data structure capability (normalization, redundancy, consistency, computability etc.),
- the application business logic that originates in the end user space and must be implemented w.r.t. the data model.

All these requirements force the idea to build an integrated computer data source together with appropriate productive user interfaces. To shift the priority between the computer data sources design and the user interfaces construction is at this time a very dangerous attraction. Without a sound data collection, all the interfaces for data accessing, despite of their very attractive presentation forms and remote commodity, remain interesting joy-tools, powerless in that data consistence, accuracy and information generic richness is concerned.

## 2. A High Level Conceptual Approach

Derived from the system complexity correlated with the portability demands, the conceptual approach must imply the construction of a data model, with the entire arsenal of capabilities (Structure, Restriction, and Operation definition). The use of an appropriate CASE tool is for that reason essential. The core of the generated relational structure will imply hundreds of tables so that defining the skeleton of the main entity classes together with their relationships becomes a very critical task. In general for large data collections, that are expected to be shared by different users, the owner remains unknown. Requirements definition, together with user data source identification and implementation, needs experience in data structure design. The guidelines for constructing a data model for UIS (figure 1) are briefly the following:

- **Multi-layer conceptual design**
  For implementing complex systems like UIS, we consider the use of relational DBMS as being evident. However, for conceptual design, an extension of the pure relational view on the data structure is appropriate. We propose for that purpose the constellation concept, used as a high-level layer of conceptual design, and the abstract object model concept a good method for refining of complex structures like constellations.
  - The constellation concept
    A UIS system core, even if generic structures (see the UNIT structure in figure 1) are used, implies hundreds of tables. To be able to control such a multitude of data collections w.r.t. the entity set definitions, the associated characteristics and natural relationships imply a permanent balance between the main points of the data structure skeleton and the accuracy of the details description. This can be achieved by using a high level data structure design using so-called "constellations", i.e. "*data groups around several domains of semantic interest"* [Le01]. Figure 1 presents the constellation structure proposed for our UIS. All the entity sets and the relationships together with the integrity rules in the data model are derived from this constellation structure. The constellations are grouped around the main aspects suggested by the basic requirements imposed to the UIS. The defined constellations provide also directives to the main sources of information in the data structure. The basic content of information for each constellation is suggested in figure 2. This content must be further refined by establishing an abstract object model.
  - The abstract object model concept
    The principles of that concept are the following:

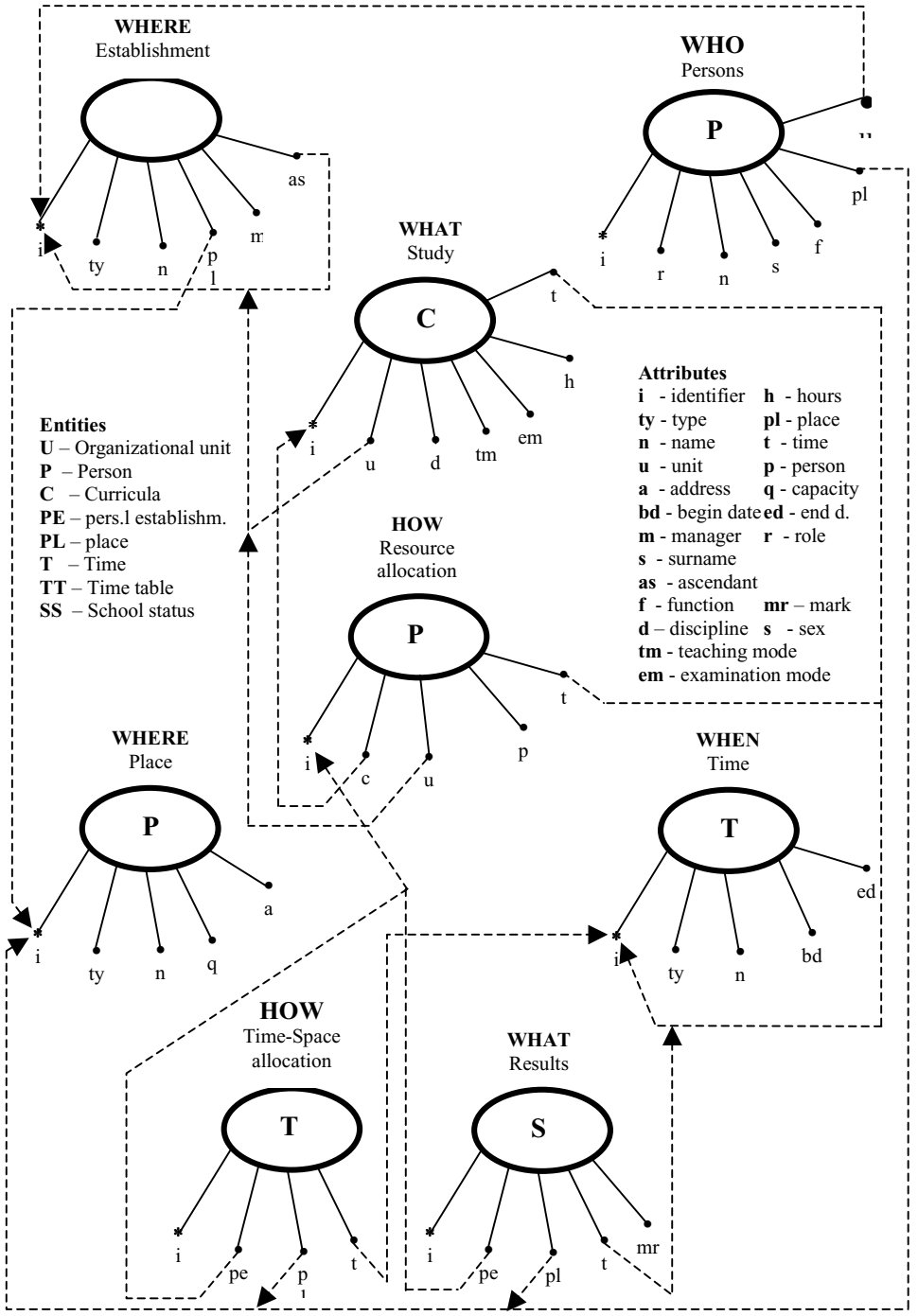


Figure 1: Constellation view of data collections in a HIS

a) Implement all relationships between two entity sets as two linkages to one table (the linkage table). This method permits to construct the two basic relationships (*one to many* and *many to many*) in the same manner, with a significant result in building the allocation process as a unique insert of a tuple into the linkage table. Associated insert triggers will describe the appropriate insert rules (the uniqueness of the parent). This procedure implements the basic abstraction concept of aggregation.

b) Implement the second basic abstraction concept, the generalization, each time an entity set must be classified. This solution avoids the procedural treatment of codes thus gaining independence of procedures against the data structures and also in data collection partition by using *Select List* objects in user interfaces.

Using these ideas the designer may describe the inner structure of a constellation with the abstract syntax from the *Abstract Object Model*.

| Constellation | Information Contents | Constellation | Information Contents | Constellation | Information Contents |
|---|---|---|---|---|---|
| ***Organizational Units*** | ***Code<br>Type<br>Name<br>Ascendant<br>Manager*** | ***Place*** | ***Building<br>Class*** | ***School-status*** | ***Student<br>Promotion<br>Taxes*** |
| ***Personal*** | ***Name<br>Surname<br>Birth Date<br>Birth Place*** | ***Time*** | ***Year<br>Semester<br>Day<br>Hour*** | | |
| ***Curricula*** | ***Discipline<br>Teaching Mode<br>Student Group<br>Hour Number*** | ***Time Table*** | ***Discipline<br>Professor<br>Student Group<br>Class<br>Hour*** | | |
| ***Personal Establishment*** | ***Discipline<br>Teaching Mode<br>Professor<br>Student Group*** | ***Exams*** | ***Discipline<br>Examination Mode<br>Professor<br>Student<br>Mark*** | | |

Figure 2: The main information content of the UIS Constellations

- **A multi-face object view**

  The *Role* concept is also drawn from the abstract object model. It is used to expand the multiple appearance of the same entity set instance in the real world.

  *Example*: The same PERSON (figure 1) can play the role of *TEACHER*, *STUDENT*, *EMPLOYEE*, *CANDIDATE* etc. The theoretical solution from the abstract object model, as the implementation of the role object depending on a previous defined relationship gave us full satisfaction.

- **A chronological object view**

  The *Time Stamp* concept - We implement an own chronological control of objects since the data in a UIS has a well-marked periodical character, this means the information in the entity sets are related to different time intervals (*set of years,*

*years, semester, months, weeks* etc.). Moreover, the user interest on the information may differ from one group to another.

Before using this concept the designer must know witch entity class and relationship in the data model has a chronological behavior and to divide the objects in *chronological* and *non-chronological*. Each chronological object is marked with a time stamp.

The Time Stamp is constructed as

- Two attributes (*Begin-Date* and *End-Date*) implemented in each chronological object,
- A set of user-controlled *Operation Parameters* that permits to declare the period of interest,
- A couple of appropriate *user-defined Stored Procedures* that assure the handling of time stamps (initialization, update, filter etc.).

  The time stamp attributes are not represented in figure 1 in order to keep the schema simple. We find between the main information structure the entity set TIME, but it has a different meaning as the chronological property of the entity sets. The difference consist of the following:
- the TIME entity set describes the *time placement* of processes, activities and events that occurs in a UIS
- the Time Stamp describes the *availability* of one entity set (including the TIME entity set) or relationship; it may be interpreted in two ways, strictly and not-strictly (figure 3)

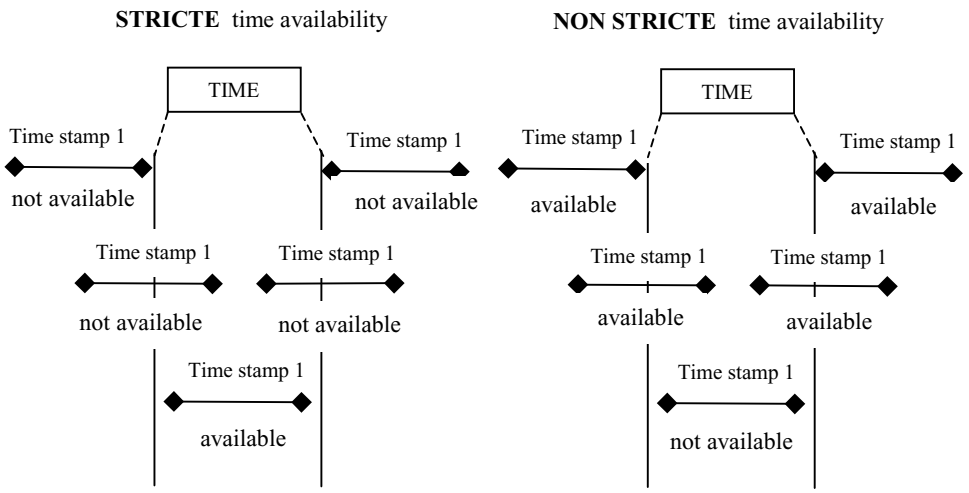


Figure 3: Time availability conferred by time stamps

- **A nonprocedural data processing view**

We see nonprocedural data processing as a sound way to assure the maximum degree of data independence against the procedures. For, taken that approach, proce-

dures migrate into the data model, thus allowing for thin user interfaces by making them free of business logic.

Concerning the materialization of implicit data, *pure materialization* based on stored procedure and *redundant data materialization* /by functionally dependent elements) have been used in order to avoid any kind of "off-line" procedures.

- The **Allocation** concept
  By using the two kernel processes *Resources Definition* and *Resources Allocation* supported by individual abstract operations [EL66] which are implemented as stored procedures, the user will receives a view on its application as a sequence of repeated, interactive and very simple processes, while he is face to face with the available resources (*ROLES*, *STUDENTS*, *TEACHERS*, *CLASSROOMS* etc.) and the allocation units (*PERSONS*, *STUDIING GROUPS*, *DISCIPLINES*, *STUDYING HOURS* etc.). Using appropriate interface objects, like *Allocation Lists*, the end user may do an individual or collective allocation and see immediately the effects on the available and allocated resources, and finally confirm or redo the initiated allocation. Combined with appropriate *Filter* and *Order* objects (*ORGANIZATIONAL UNITS*, *UNIT STRUCTURE TYPES*, *ROLES*, *TIME STAMPS* etc.) this non-procedural data processing can be used for very sophisticated procedure implementation like the *Candidate Repartition, Personal Establishment* and *Timetable* preparation. Adopting these fundamental system processes we can divide the main activities in UIS Data Sources:
  - Entity sets construction – as a set of simple operations dedicated to load an instance with attributes values,
  - Relationship construction – as an allocation of *MEMBER* entities to a given *OWNER* entity; in the case of *Complete Relationships* we can attach to this operation also an appropriate declaration of the special attributes that describe the intensity of the relationships (ex. *Number of MEMBER*, *Order Number of MEMBER* etc. implied in one relationship)

## 3. Ensuring Quality by using Standardized Tools

In order to ensure high quality, we restricted ourselves to the use of standardized tools for the implementation phase based on the Federal Information Processing Standards (FIPS) [FI93], with their components:

- *IDEF0 – the standard about Integration Definition for Function Modeling,*
- *IDEF1 – the standard about Information Modeling,*
- *IDEF1X – the standard about Integration Definition for Information Modeling,*
- *IDEF3 – the standard about Process Description Capture Method.*

For our UIS data model construction and process description all used design tools were aligned to these standards.

## 4. UIS User Data Sources

As presented above, a UIS needs large computer data sources, the implementation of which arises critical organizational problems related to the ownership and administration of the implied data collections.

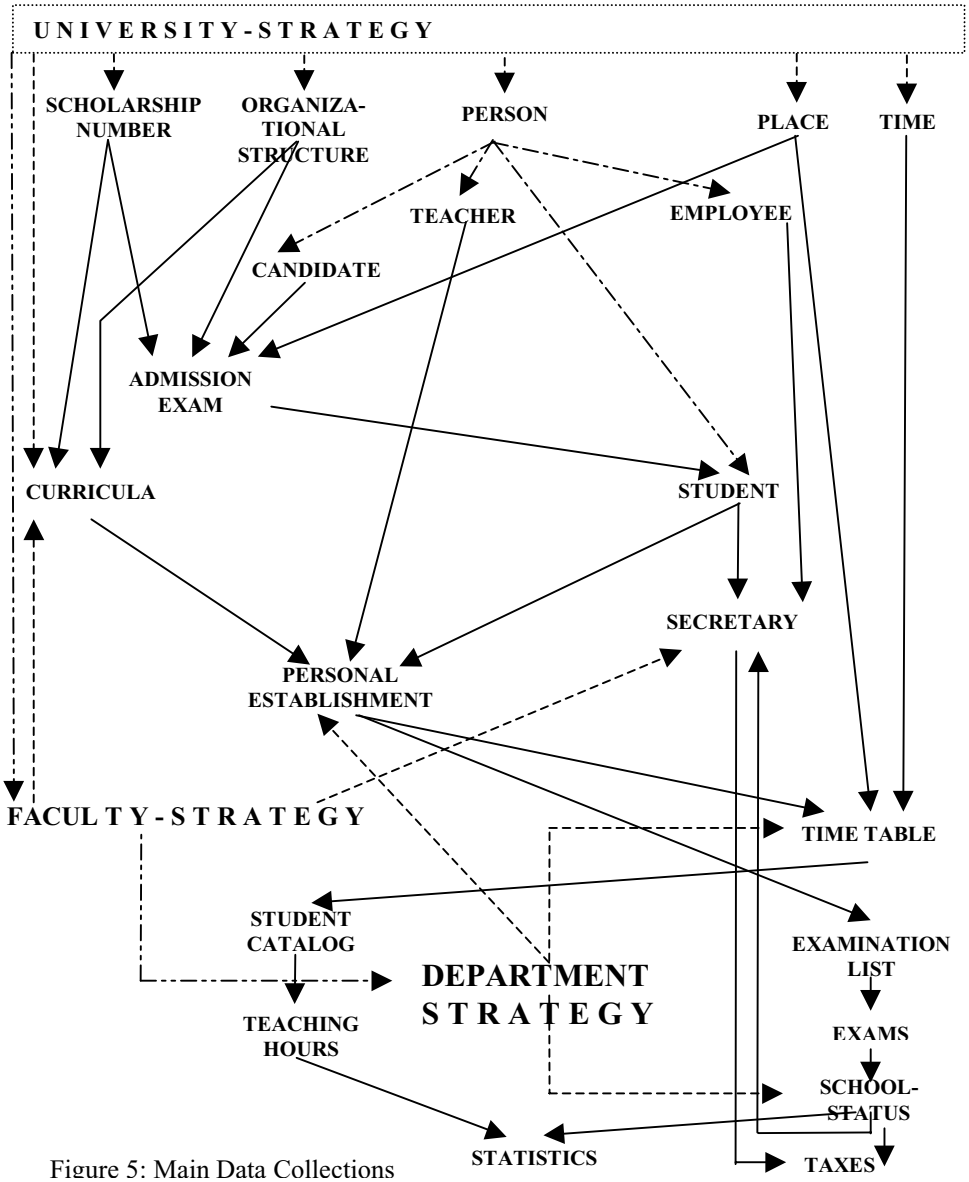


Figure 5: Main Data Collections

The first topic is related to the user data Sources that must be attached to an existent computer data source. The kernel of this preoccupation can concentrate in a lot of questions: WHO, WHERE and WHEN updates occur of the integrated data collections, which must remain unique and shared at any time. These questions find here a new perspective, because they imply new management concepts like *Levels of Management*, *Activities* related to *Data Collections* (figure 5).

Three level of management together with the appropriate strategies can be observed in the schema. All the main data collections arise around the principal objectives of the educational activity:

- establish the required CURRICULA,
- determine the SCHOLARSHIP NUMBER,
- prepare the PERSONAL ESTABLISHMENT.

Two basic data collections must be available for this goal:

- the Organizational Structure – implemented as a set of organizational UNITS of different type (*university, faculty, department, student group, financial department* etc.); this is a complex structure because it contains many overlapping hierarchical structure (*EDUCATION* structure, *ADMINISTRATIVE* structure etc.) that implies several *Unit Structure Types.*
- the PERSONS – we recognize in this structure the needs for using *ROLES* (*TEACHER*, *STUDENT*, *CANDIDATE*, *EMPLOYEE* etc.) in PERSONS refinement

Another two referential data collections must be available in any UIS:

- TIME – describing the location in time of each event and activity (*Study Year, Semester, Week, Hour* etc.)
- PLACE – describing the location in space of each event and activity (*Building, Classroom, Meeting* etc.)

We find also a couple of derived data collections (*TIMETABLE, STUDENT CATALOG, EXAMINATION LISTS* etc.), all dependent on the previous described information.

On the same schema we can see the data flow between different data collections together with the relationships between the principal education and administrative activities (teaching, examination and secretary).

Associated with each data Collection we can observe the presence of several properties:

- The *Time of First Entry* in the University Educational System (UES). Since the same data collection may be present at different places in a UES, we can speak about different instances of the same collection. These instances may become unsynchronized in time, as a result of different updates. The property is useful to establish the priority of one data collection instance (the first entry) against the another, in order to guarantee the data consistence.
- The *Place of First Entry* in the UES. This property is useful to establish the *natural owner* of a data collection (the department where the first entry occurs).
- The *Responsible* of the data collection, i.e. its the natural owner.

- The *Legality* conferred to the information content of a data dollection – each Data collection must be generated by using a legal document.
- The actual *Role* associated with a data collection – a unique subset of a data collection, defined as specialization partition by means of an auxiliary property, must be used at a time.
- The *Accuracy* of a data collection – since a lot of older instances will be grouped in one unique data source (possibly replicated) accuracy must contain the whole information associated with the data collection.
- The *Fidelity* of data collection – the time-varying data collection must be synchronized with the real events that change their information content.

On the basis of the above-mentioned properties we can define an independent user data source as a source of a unique Data Collection, shared by all interested users, but having a unique natural owner charged with the responsibility of its legality, accuracy and fidelity.

As an example of the above-described independent user data sources we present the main structural units our UIS project in figure 6.

There are two kinds of user data sources:
- *unique user data sources* – data sources that issue information that may be shared by another user but have a *unique owner*, represented by an unique structural unit which can assure the full control of the information updates. We name the associated data collection as defined owner data collection. (Examples: Faculty Department for PERSONAL ESTABLISHMENT).
- *distributed user data sources* – multiple distributed data sources that issue information in the same data collection together with another user. The corresponding data collection can be viewed as an undefined owner data collection. We find here distributed structural units that can not alone assure the full control of the information updates. (Examples: The UDS – Faculty, Department, Personal Administration - for the STRUCTURAL UNITS data collection.)

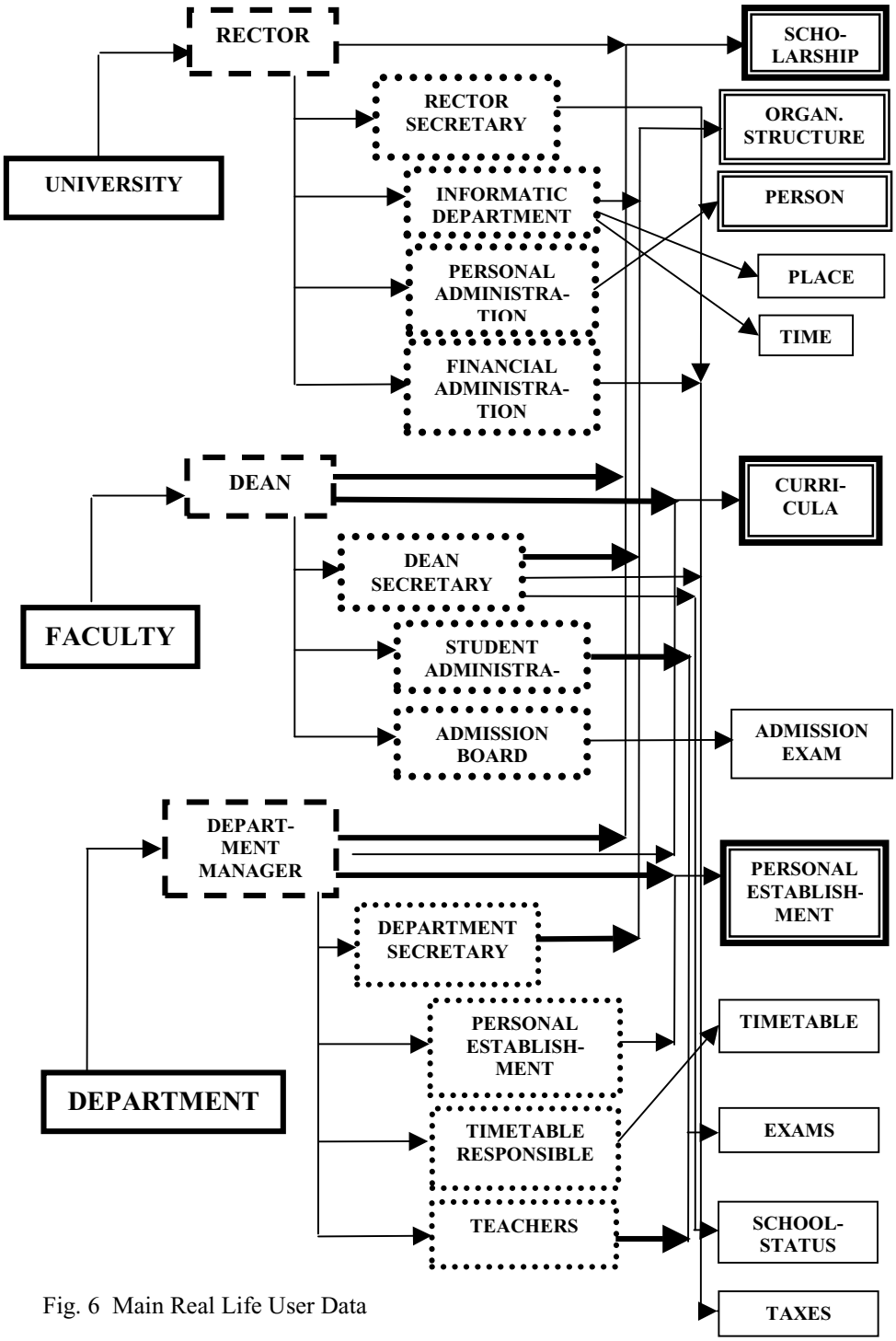


Fig. 6 Main Real Life User Data

## Summary

The approach to UIS presented here was developed in a three-year research and design activity by a working team consisting of 1 researcher, 2 designers and 2 implementers. The team was be enlarged by student implementers being in the last year of their studies. Summing up the characteristic features of that approach we can state that it leads to UIS that are

- management information systems,
- designed for client/server architectures and different software platforms,
- oriented at the IDEF standard,
- providing an own advanced user interface that supports
  - user friendliness,
  - a productive programming environment.

Last but not least the conceptual bases, the soundness proof of our theoretical concepts and the very good performances in implementation activities give us hope for regarding this UIS approach as a generic one which could be easily adopted to other University Education Environments.

## References


[El66] Elmagraby, S. E.: The design of production systems. Reynhold Publishing Corporation, New York, 1966

[FI93] FIPS: Integration definition for information modeling (IDEF1X). Federal Standard, December 1993

[GS96] Grauer, M; Schmidt H.: Ansätze zur Modellierung und Bewertung von Leistungsprozessen an einer Universität". Proc. Workshop "Unternehmen Hochschule", Informatik'96, Klagenfurt, 1996

[Le01] Lelutiu, A.: A new Prototype for the User interfaces in a Client-Server environment. Proc. Scuola superiore G. Reiss Romoli (SSGRR-2001) Conference, L'Aquilla Italia, August 2001

[LH00] Lelutiu, A.; Haller, P.: Using Standardization in Software Engineering. Editura "Casa Cartii de Stiinta", 2000

[SK99] Schmied, J.; Küng, J.: Die KeplerCard – Einsatz einer Chipkarte als multifunktionaler Studentenausweis an der Johannes Kepler Universität Linz. Proc. Workshop "Unternehmen Hochschule", Informatik'96, Klagenfurt, 1996