# Reference Frameworks for Information Systems and Development: An Evaluation

**Vesa Savolainen**

Department of Computer Science and Information Systems
University of Jyväskylä
SF-40351 Jyväskylä
FINLAND

**Abstract:** This study presents the main results of an evaluatory comparison between the most remarkable existing reference frameworks for information systems and their development. HECTOR's Framework of Reference (FoR) and an IFIP WG 8.1 Task Group's Framework for Understanding (FfU) are frameworks for both ISs and ISDs and Framework of Information Systems Concepts (FRISCO), produced by another IFIP WG 8.1's Task Group, only for ISs. The analysis and comparison scheme is derived from a profound definition of the **ISD methodology and metamethodology** concepts. This rather detailed scheme is used to detect the strengths and weaknesses of these reference frameworks, but its interstructure can also be used the same way as the frameworks themselves, i.e. as a basis for comparing the ISD methodologies. Our analysis reveals that these three reference frameworks model the IS field from totally different standpoints, and therefore it is not sensible to set them into any preference order. They complete each other and it is therefore useful to apply them side by side.
**Keywords and phrases:** modelling, information system development, IS concepts, design methodology, reference framework, evaluation, comparison.

**Vesa Savolainen** was born in 1944. He received his M.Sc. in 1971, Ph.Lic. in 1973 and Ph.D in 1977 in Computer Science at the University of Tampere. He became Associate Professor in 1980. In 1982 he was Professor of Applied Mathematics at the University of Jyväskylä. During 1982-1994 he held the position of Professor of Computer Science at the same university. Now he is Acting Professor of Computer Science (Information Systems) and Head of the Department of Computer Science and Information Systems at the University of Jyväskylä.
He received Senior Scientists' Grants from the Finnish Academy of Sciences in 1983, 1987 and 1991-1992. He participated in the ESPRIT Project OSSAD, Office Support Systems Analysis and Design, in 1986-89, and ESPRIT II Project HECTOR, Harmonized European Concept and Tools for Organizational Information Systems, in 1989-90. Now he leads research projects on Executive Information Systems, Exception Handling, Multimedia Information Systems, and Measurement Theory. He is a Senior Member of IEEE, and a Member of ACM and IFIP WG 8.4 (Office Systems) and WG 8.3 (Decision Support Systems). He has authored over 100 scientific international papers, reviewed and referred papers on operations research and information systems, some of the latest on executive information systems, and also 2 books: "Information Resource Management in Organizations" and "Foundations and Algorithms of Graph Theory".

## 1. Introduction

The fact that both information systems research and the technology applied for the design and implementation of information systems have lately reached a certain level of maturity, has created a growing interest in forming a holistic picture of the whole information system (IS) concept and of the information system development (ISD) field, thus achieving a better understanding of them [6, 8, 12, 35]. The diversity of meanings of the rather loosely defined concepts in the ISD field is often felt frustrating among different IS research schools [19, 25, 27].

The only common denominator among researchers in this area seems to be their aim to support better ISD processes in practice while the ways to approaching this aim do not intersect with each other. Today, attempts at metamodelling IS and ISD areas by constructing their reference frameworks are current with many research groups [8]. However, no profound evaluation of the largest recent IS and ISD reference framework construction efforts, has been made in the IS literature. This paper presents a survey and an evaluatory comparison of the existing IS and ISD reference frameworks.

The two ISD reference frameworks discussed in this paper are basically more or less metamethodologies. HECTOR's Framework of Reference (FoR [27]) is one of them - and much more - while IFIP WG 8.1 Task Group's Framework for Understanding [FfU 25]) does not actually model methodologies but ISD approaches, views and aspects. By means of metamethodologies one can analyse, evaluate, compare, comprehend and choose methodologies, methods and tools. The metamethodologies are also constructed for clarifying the conceptual fuzziness and deficiencies which are common in information systems science. Especially the introduction of CASE systems has brought along the use of methodologies and interests in comparing different methodologies by using such frameworks as those analysed in this paper. Several major ISD concepts are, in one way or another, hierarchically interrelated [17]: a metamethodology can describe several methodologies, a methodology suggests the use of several methods, a method is supported by several tools, techniques and description languages.

The main concern of the paper is to evaluate the two ISD reference frameworks by considering their concepts and the entities defined in them. In addition to that, FRISCO (Framework of Information Systems Concepts [9]) is included in this study though it does not model the ISD process but the IS itself, and its analysis is here presented in a sub-area of the comparison of the ISD reference frameworks. As the reference frameworks have been constructed by researchers who represent different schools of IS research, it is of great interest to see what kinds of ISD approaches and principles, design functions, modelling scopes and levels, and areas of design (such as organisational, social and functional design, user interface design and adp-technical design) have been included in their frameworks. The intrastructure of our evaluation scheme tries to detect the properties of methodologies that are covered and modelled by metamethodologies. The intrastructure then helps compare the outer frames of the reference frameworks as to their purposes, general features and usability.

In Section 2 the main definitions of the ISD methodology and metamethodology concepts are given. In the next Section the three IS and ISD reference frameworks are generally described. Section 4 outlines a comparison and evaluation scheme, and Section 5 applies the scheme to the frameworks. Finally, Section 6 draws the conclusions.

## 2. Definitions of Basic Concepts

The analysis and the definition of the basic concepts are only meant here for laying a solid foundation to constructing a comparison scheme for the reference frameworks.

**Definition:** The *methodology for information systems development* is an organized set of methods, procedures, techniques, design tools and documentation tools to support effective development and introduction of new or improved information systems [17]. The ISD project aims at revealing, causing, evaluating, controlling and operating changes in organisational, social and technical systems [1, 15]. The methodology consists of functions or phases which can in turn hierarchically consist of subfunctions or activities or design tasks. These guide the designers in opting for the right methods, techniques and tools for every function, activity or task, while supporting them in the design, management, control and evaluation of ISD projects [29, 32]. The methodologies are usually based on some design principles and philosophic paradigms and, according to them, emphasize different human, scientific, pragmatic or other approaches [1, 5, 20]. In general, a methodology consists of an ISD function model, activity and task structure, design and modelling methods, description languages, tools and techniques, organization and decision rules, and application hints at increasing its usability.

A methodology should improve the effectiveness of the ISD process and the performance of the improved information system [24]. It should also be in such a form that it can be taught, learnt and transferred from one application environment to another. It should be comprehensible and socially acceptable in the application environment.

**Definition:** In information system sciences, a *metamethodology* means a model which aims at describing ISD methodologies, their structure, scopes, paradigms, approaches, principles, emphases, usability, coverage, flexibility and general applicability.

These definitions will be carefully analysed later in Section 4 when we construct our evaluation and comparison scheme. Before that we briefly review the existing reference frameworks.

## 3. Short Description of Three Reference Frameworks

There exist two important ISD reference frameworks, i.e. those constructed by a Task Group in IFIP WG 8.1 /OHM88/ and ESPRIT Project HECTOR [27]. The FRISCO Task Group in IFIP WG 8.1 constructed an IS concepts model [19]. In addition, ISO has started work on constructing an international standard for the same area but in this phase - according to its policy - we have not been able to review it. The reasons for starting those works and the destination of the resulting frameworks have been different. In the following we briefly describe the purposes and structures of the three frameworks.

## 3.1. Framework for Understanding

The Framework for Understanding (FfU) has been constructed to serve as a tutorial on information systems methodologies. It provides an understanding rather than prescribes alternative ways of tackling information systems planning, business analysis and systems design. The term *information systems methodology* is used in the sense of "a methodical approach to information systems planning, analysis and design". The term *system development* has been avoided in this context because FfU "limits consideration to that part of the information system life cycle leading up to the point at which a builder can start to construct a system" [25], and thus the system construction has mainly been left outside of FfU. Even the construction design, which usually means a very method- and tool-dependent system specification, is mainly left outside of FfU. Table 1 illustrates the areas covered by the three reference frameworks.

FfU uses the term **information systems planning** to refer to the strategic planning of computerized systems [25]. The planners identify which information systems are needed rather than design in detail any specific system, how these systems should interact and how the right sort of phasing can be guessed from the existing situation to the future required set of systems. The FfU term **business analysis** is used in preference to the term 'system analysis' [25 ]. The reason for this choice is to emphasize that the business or enterprise is the subject of the analysis and not any kind of system. The area for business analysis is more constrained than that for information systems planning. This is a descriptive study of what is done in an enterprise and of what needs be done given the support of more advanced information systems.

**System design** starts without clear boundaries from where analysis ends. The results of analysis and design form together the design product, i.e. the design specifications. FfU is a framework within which information systems methodologies may be viewed. It also views information systems methodologies as a framework for the preparation of a design product.

FfU offers a conceptual view of information systems methodology. The framework has an essentially data oriented perspective, and has been defined as lists of 127 entity types, referred to as **components** of design products [25 ]. FfU gives an exact definition of each component and shows different ways of creating, presenting and documenting components. Each component of any design product is preferably regarded as having one of the following perspectives:

**1. Data oriented perspective** reflects the database aspects of ISs and thus reveals a complete and thorough analysis of data and their relationships, and attributes [25 ]. Here the important aspects are the following:

(1) business data produced during the business analysis stage,

(2) prescriptive data are construction and performance independent and often result from the logical database design in the system design stage,

(3) construction tool dependent prescriptive data coming out from the construction design stage, and

(4) stored representation of data which include indices and access paths and also come out from the construction design stage.

**2. Process oriented perspective** reflects the functions aspects of IS's [25 ]. There are four important aspects in this perspective:

(1) business activities which relate to the business analysis and system design stages,

(2) user perceivable tasks supported by a computerised IS (which also relate to the business analysis and system design stages),

(3) computerisable processes which relate to the construction design stage, and

(4) compilable units of programming also relating to the construction design stage.

**Table 1. The Stages in an Information System Life Cycle Covered by the ISD Reference Frameworks FfU and FoR; FRISCO covers only the system theoretical and philosophical study of the essence of ISs.**

| IS Life Cycle Stages | IS and ISD Reference Frameworks | | |
|---|---|---|---|
| 1. strategic study | | (FoR) | |
| 2. information systems planning | FfU | (FoR) | |
| 3. business analysis | FfU | (FoR) | |
| 4. analysis of IS and its essence | FfU | FoR | FRISCO |
| 5. system design | FfU | FoR | |
| 6. construction design | (FfU) | FoR | |
| 7. construction and workbench test | | FoR | |
| 8. installation | | FoR | |
| 9. test of installed system | | FoR | |
| 10. operation | | FoR | |
| 11. extension and maintenance | | (FoR) | |
| 12. phase out | | | |
| 13. postmortem | | | |

**Table 2. The List of Key Concepts in FfU**

| Key Concepts | Cross-references |
|---|---|
| design product | component in perspective |
| design process | component of design product |
| component | step used in stage |
| step category | step produces component |
| step | design process uses step |
| graphic representation | component in graphic representation |
| technique | step uses technique |
| scenario factor | option in combination |
| scenario factor option | design process valid for scenario |
| scenario | |
| perspective | |
| stage | |

**3. Behaviour oriented perspective** reflects dynamic, temporal aspects of IS's [25 ] and it can be characterized by such examples as business events, process control and triggers of actions.

FfU also presents cross-referencing components which relate components with one perspective to components with another perspective. The perspectives are included in the framework with a view at providing a facility for analysing different IS methodologies which frequently emphasize only one perspective and avoid the others.

Table 2 includes the list of FfU key concepts.

FfU was been constructed during the years 1985-87. Its builders had attempted to consolidate and merge the various views on information systems methodologies into a unified framework. Many of its foundations laid on the analyses of old well-known ISD methodologies, such as CIAM, IDA, ISAC, NIAM, SADT, SASS and SSA. The final FfU document [25] illustrates important aspects of human roles, modelling, representation and documentation in the use of information system methodologies. It analyses the components in business analysis, system design and information systems planning, and gives tested examples and case studies on them. And it also offers a summary of 34 existing methodologies.

The builders of FfU recognized as an unfortunate omission of their work that their framework did not cover the construction design stage [25]. One reason for the omission was that the construction design was felt to be tool-dependent - and all the tools are under significant development. Also such a component analysis would have resulted in an as long list of components as that appearing in the whole FfU.

## 3.2. HECTOR Framework of Reference

The construction of the HECTOR Framework of Reference (FoR [27]) originally aimed at providing a solid basis for establishing standards on the methodologies, methods and tools to be used in the development, improvement and evaluation of organizational information systems (OIS). The characterization and evaluation of the methodologies, methods and tools can be done in terms of their performance under certain environments. In spite of the fact that FfU and FoR have been developed independently of each other, FoR is an answer to several calls for completion revealed in FfU ( some of them also mentioned in the concluding remarks of the FfU document [25]). These are for example:

- need for formulating a framework also for *the construction design stage* which is missing in FfU (see Table 1),
- need for expressing *dynamic characteristics of ISD* and for emphasizing the *role and performance of the methods and tools in ISD projects*, and
- need for expressing the *framework in computerised forms*. FoR strongly points out the environmental (ISD contingency) aspects and it covers organisational, human and technical aspects of ISD.

There have been three goals in the development of FoR:

(1) **to identify the different aspects and gaps** of existing IS development and implementation methodologies, methods and tools in different organizational environments and applications,

(2) **to determine the performance criteria** that methodologies, methods and tools should satisfy with respect to the given task, application and environment, and

(3) **to define what is required for selecting effective and acceptable methodologies, methods and tools** for the analysis, design, implementation and evaluation of integrated organizational and office information systems under the given circumstances.

A general framework was established, partly based on the previous results of ESPRIT projects, especially those of OSSAD Project [11]. The entity- relationship representation has been chosen for the **static part** which models the development environment, OIS, methodologies, methods and tools, and performance criteria. About 450 entities in the HECTOR Framework of Reference have been identified and defined. This work has produced a HECTOR Glossary of the

**Table 3. Examples of Entity Categories, Classes and Subclasses in FoR**

| Category | Class | Subclass |
|---|---|---|
| Environment | Environment - Initial State | |
| | * External Environment | |
| | * Internal Environment | |
| | ** Organization Studied | |
| | ** Work Structure | |
| | ** Org. Policies and Regulation | |
| | ** Motives for Syst. Development | |
| | Changed Environment - Later Stage | |
| Organizational Inf. System | Org. Inf. System - Initial State | |
| | * Information Technology | |
| | * Organizational Structure | |
| | * Information Content and Network | |
| | * Performance Measurement - Static | |
| | * Performance Measurement - Dynamic | |
| | Org. Inf. System - Later State | |
| Methods and Tools | Methodologies | |
| | * Identification | |
| | * Conceptual Framework | |
| | * Coverage | |
| | * Application Aspects | |
| | * User Support | |
| | * Performance Measures | |
| | Tools | |
| OIS Improvement Project | OIS Improvement Process | Major Phases: |
| | | Set Business Strategy |
| | | Set OIS Strategy |
| | | Prepare OIS Improv.Project |
| | | Execute OIS Improv.Project |
| | | Operate and Maintain Improved OIS |
| | | Monitor OIS and Suggest More Improvements |
| | | Main Activities within Phase |
| | Environment of OIS Impr. Process | |
| | * Initial State before Project | Motives for Syst. Developm |
| | | Type of OIS Change Envisa |
| | | Initial Techn. Devel. Resour |
| | | Performance Measurement before Project |
| | * Later Stages during and after Project | Support Team for OIS Impr |
| | | Improved Techn. Dev. Res. |
| | | Performance Measurement during and after Project |
| | | Methods and Tools Selected for the Use of Project |

most important definitions of concepts connected to OIS development. Also measures have been set out for most entities, and the conceptual relationships have been shown in the ER-type models.

The entities have been grouped as follows [27 ]:

**Environment:** These entities are circumstances under which the information system is to be designed and implemented. They can be both internal and external to the organization, and they can cover both such parameters as the organization location, financial regulations and organization size and also the existing OIS, its past performance and technological constraints.

**Methodologies, methods and tools:** These descriptive entities, characteristic of methodologies, methods and tools, are used to develop and implement a new IS, for example name, life cycle phases supported, modelling capabilities and training costs. The HECTOR FoR focuses on these entities.

**Organizational information system:** The execution of the OIS improvement project aims at changing an old OIS with poor performance by a new OIS with better performances. The OIS includes both a computerised data processing system and an organisational system running it. Examples of such entities are worker autonomy, source of information and basis of departmentation.

**System Performance:** This is measured by considering the effects of the OIS and comparing what has been gained by the changes from the old OIS to a new one. Examples of such entities are maintainability, strengthening of customer/client ties and completion on time.

The Petri net representation was chosen for the **process model** to describe the dynamic system development activities and tasks. The process model is presented at four hierarchical levels which include processes and information flows between them. The first of the levels includes the processes Global strategy setting, Local and OIS strategy setting, Project preparation, Execution of OIS improvement project, Operation and maintenance of improved OIS, and OIS monitoring and suggestion of changes. The second level of Execution of OIS improvement project includes the processes Selection of method, Measures setting, Project management, Development of improved OIS, Evaluation of improved OIS, and Project assessment. For the illustration, see Table 3.

The HECTOR FoR can be used for

- analysing what methodologies, methods and tools are the most appropriate for a given project,
- comparing methodologies, methods and tools in terms of their characteristics, and
- identifying gaps among the existing methodologies, methods and tools.

In principle, a simple way of doing this is to check if there is any required entity which is not covered by the methodology, method or tool under study. In practice, however, this type of an applier often gets handicapped in measuring, interpreting and evaluating the factors which are matters of different opinions, beliefs, attitudes, views, worlds of values, etc.

There were certain similarities in HECTOR approach and that of Olle et al to FfU, but not too many. Olle's working group seems to have stuck to data analysis views, old methodologies and the ways of thinking appearing in them. The HECTOR work was based on newer research results of several ESPRIT projects of the last five years, on newer ways of thinking and working, e.g. better taking into account participation, prototyping, new CASE tools, and organisational and human needs, and on new methodologies, such as OSSAD. FoR was developed mainly in 1989 and its refinement continued in 1990.

## 3.3. FRISCO Framework

The FRISCO Task Group in IFIP WG 8.1 was set up in 1988 with a deep concern with the use of too many ill-defined or fuzzy concepts and the lack of a commonly accepted conceptual reference and terminology in the IS area [19]. Even the term 'information system' itself was found as being used in at least four different acceptances. However, the FRISCO Task Group could not - just as other such attempts had not been able to do either - form a

**Table 4. Key Concepts of the FRISCO Approach (cf.[19] pp. 16, 23, 29 and 30; this is not an exhaustive list by no means)**

| | |
|---|---|
| Kinds of system | **Examples of is-a relationships** |
| System | organizational specialization |
| Open active dynamic system | |
| Organizational system | **Abstraction relationships (is-aspect-of)** |
| Formalized organization system | |
| Knowledge realization system | |
| Information realization system | application abstraction |
| Information system | formalization |
| Data handling system | semiotic abstraction |
| Generalized information system | infological abstraction |
| Generalized data handling system | datalogical abstraction |
| Closed passive static system | |
| Open passive dynamic system | **Other relationships** |
| Closed active dynamic system | |
| Reactive system | is conception of |
| Responsive system | represents |
| Autonomous system | |
| Dataprocessing system | **Semiotic concepts** |
| **System theoretic concepts** | |
| | Knowledge |
| | Knowledge representation |
| System view | Information |
| Environment | Information representation (Data) |
| System domain | Empation |
| Domain of environment | Empation representation |
| systemic property | |
| system viewer | |
| actor | |
| activity | |
| operand | |
| process | |
| behaviour | |
| event | |
| trigger | |
| point of time | |
| entity | |
| relationship | |
| concept | |
| phenomenon | |
| operation of data | |
| data handler | |
| data processor | |
| language | |
| model | |

single unified vocabulary for the whole IS domain, but it provided some insight into the diversity of its subdomains. Some of these subdomains could be covered with an in-depth analysis but not at the level of establishing standards.

The outsider reader feels that the resulted FRISCO framework is a philosophic, unstructured essay about the essence of information systems and their paradigms. It focuses on systems' foundations and theories referring to such theorists as Langefors, Ackoff and Checkland [19 ], and not at all on IS development (see Table 1). It aims at "providing bridges between general systems theory, organizational systems, information systems and data handling systems. These bridges are provided by various kinds of specializations and other abstractions leading to the notion of systems being specific aspects of larger systems, i.e. subsystems of them. In this way various is-a relationships, is-aspect-of relationships and other specific relationships can be specified between these systems. The successive application of these specializations and abstractions leads to a hierarchy of systems where the root is systems in general and the interjacent nodes and leaves are systems more or less well-known in our area".[19].

Table 4 illustrates the key concepts of the FRISCO approach.

The members of FRISCO Task Group were nine European IS experts from six countries with more or less academic background and they represented a diversity of IS domains "but by no means all" as they admitted [19] . The following citation from [19 ] illustrates this:

"Communication has six distinct layers of problems:

| | |
|---|---|
| Physical | the media and amount of contact available |
| Empiric | the entropy, variety, equivocation encountered |
| Syntactic | the language, the structure, the logic used |
| Semantic | the meanings and the validity of what is expressed |
| Pragmatic | the intentions and significances behind the messages |
| Social | the interests, beliefs and commitments shared as a result" |

The basic model in FRISCO expresses the notion of system and specifies it formally:

$FW_s = <A,SV>$

where A is an arbitrary area, and SV is the **system view** expressed as:

$SV = <S,E,SD,ED,SP>$

where

- S is **a system**
- E is **the environment** of the system
- SD is **the system domain**, i.e. the area conceived as a system
- ED is **the domain of the environment**, i.e. the area conceived as the environment
- SP are **the systemic properties**, relationships between the system and the environment.

Behind the system view is always a person - **the system viewer** - who interprets the world in the above- mentioned way, i.e. by applying $FW_s$.

The FRISCO Framework classifies different systems and defines **static and dynamic, open and closed, active and passive, as well as reactive, responsive and autonomous systems**. It discusses semiotic concepts, such as **knowledge, information, empation, data, databases and their structures**. It also analyses organizational systems where **actors, activities, operands, events, triggers, points of time, entities and relationships** are of central interest. Finally, it defines basic concepts and structures of data handling systems: **operations of data, data handlers, data transfer systems, data processors and data processing systems**.

FRISCO introduces or uses about 265 entities or concepts. Its glossary includes 128 concepts out of which about 62 have been defined in Table 4. The FRISCO Task Group intended to apply the framework to a number of traditional IS approaches, but at present the FRISCO document [19] only comprises a few illustrations of the framework on the computer science/datalogical area.

If compared to FfU and FoR, FRISCO is only a first step towards a well-structured framework. However, it can even at this stage offer the reader philosophic grounds in information systems. But practically, it does not offer so much to an IS engineer or ISD project manager.

## 4. Scheme for Evaluating Metamethodologies

For an analysis and comparison of the two ISD reference frameworks and the FRISCO IS reference framework, there should be developed a scheme including evaluation and comparison criteria. Quite a few of these criteria can be similar to those selected in literature for comparing the ISD methodologies, because they can help measure the coverage as well as various views and emphases of the frameworks. The three frameworks represent at least three different views of the same universe of discourse, and our purpose is not to find any absolute ranking for them, but to show their differences and, in this way, support further development of the frameworks. As FRISCO does not cover ISD activities, we shall discuss it only when applicable.

The scheme has two levels. First, the intrastructure - reflecting the internal validity or applicability of the frameworks - includes the same factors as those valid for evaluating methodologies. Here the intrastructures of metamethodologies are inspected: what are the properties of methodologies that are covered and modelled by metamethodologies? These properties are represented in metamethodologies by functions, phases and entities. Secondly, the interstructure - reflecting the external validity of the frameworks - includes the outer properties of metamethodologies, such as their purposes, general features and usability. These properties are relevant when metamethodologies are used for their original purposes.

### 4.1. Internal Validity of Frameworks

The coverage of an ISD methodology consists of the following ten groups of evaluation criteria: areas of design, design functions or phases, design approaches, design principles, modelling scopes, modelling levels, working facilities, usability, consistency and control, and benefit measurement factors. All these criteria have been derived from the definition of a methodology and from its general user requirements. In this paragraph we shall inspect the contents and motivate the presence of each group.

*Areas of design*: Adp-technical design includes such tasks as the design of software, hardware, database, systems interconnections and technical quality control. However, the IS improvement process today means not only adp-technical improvements but also more and more social and organisational changes [14 ]. As this fact had not been admitted and appreciated enough earlier, the information systems did not always succeed. Failures of information systems are widely studied and demonstrated in information systems literature [32, 34]. Organisational and adp-technical parts of information systems are connected to each other by user interfaces, and their design is an essential third part - and often in practice the most important one - in the design activities [30, 31]. Therefore the user participation and acceptance is the clue of the success of the IS. The methodologies should thus cover the following areas of design:

- organisational, social and functional design,
- user interface design, and
- adp-technical design.

*Design functions or phases*: One of the most important criteria in selecting a suitable ISD methodology is how well it covers the functions, design activities and tasks throughout the information systems' life cycle (23, 11]. The functions presented here often appear as phases in most traditional methodologies:

- project definition,
- situation analysis,
- systems design,
- implementation of changes, and
- systems performance monitoring and maintenance.

*Design approaches*: The prototyping is used for small- size personal systems and for those that are planned to be further developed all the time. It is one form of making participative approaches. Waterfall approaches are traditional approaches to mainly constructing fairly large information systems. They usually include a division of design tasks into phases. Function based approaches, for example the OSSAD Methodology [11], place between the two other approaches; no phase division applies there, but the objectives and design procedures of each function are given and, for example, prototyping is suggested to be applied as much as possible. According to the size and pace of the ISD project and to its other contingencies, the project leader usually selects out of three main approaches, i.e.

- prototyping,
- waterfall approach and
- function based approach.

*Design principles*: The contingency principle is generally accepted among information systems theorists and practitioners. Its originality is mainly attributed to Davis [1, 13, 11 ]. The main message is that that there is no such ISD methodology that would be better than any other methodology under all circumstances and that, therefore, a methodology should always be tailored to its application area, environment and its other contingencies. The user participation has an enormous impact on the IS introduction, acceptance, effectiveness and success [3, 21]. An identified problem and the aim of solving it are generally the reasons why launching a project and they should guide the ISD work. Traditionally , iteration is a basic design principle, and the feedback to earlier design tasks should exist [2, 11 ]. Techniques like pilot testing, simulation or prototyping are often used to provide the necessary feedback before the implementations of the final IS version. A methodology can be classified and selected according to the degree of explicitly applying the design principles

- contingency and flexibility,
- user participation,
- problem orientation,
- iteration, and
- experimentation.

*Modelling scopes*: The chosen view of an information system affects the IS modelling scopes, and the same IS can present different scopes. The scope of data emphasizes the meaning of data in the system. The real nature of the IS can be realized and identified by the stored data, their nature and flows [16, 35]. The scope of processes yields a definition of the IS by the organizational processes (process models) and their functioning (functional models) [18, 37]. The scope of organization is based on the organizational structures and their interconnections. The modelling scopes are:

- data,
- process and
- organization.

*Modelling levels*: The levels of modelling details in the present and future IS, are called modelling levels. Progress from a rougher level to a more detailed one also corresponds to progress among the design functions towards IS implementation [11]. The abstract model (often called the conceptual model in the database environment) describes the essentials of an organization, such as objectives, strategies, business functions and main data flows in the organization. The descriptive model (often the logical model) can describe both present and future systems, and even several suggested versions of the future IS. It presents at the level of such details as tasks, procedures, operations, organizational units, roles, actors and resources. The specification model (or the physical model) describes the technical and organizational details of the IS alternative. The user interface is the interconnection between the organizational system and its technical support system [11, 31, 36 ]. Thus, the modelling levels are:

- abstract models,
- descriptive models and
- specification models.

*Working facilities*: The last few years have been a period of fast progress in the number and significance of various design facilities. Certain methodologies are supported by certain methods and techniques and all are supported by several design tools and description languages. The design tools can be computer assisted (e.g. CASE tools) or manual. The choice of the most suitable design facilities is a central problem of the ISD leader [2, 24]. The design facilities are

- methods,
- techniques,
- tools and
- languages.

*Usability*: One of the most important standards in assessing the suitability of a methodology is its usability. Poor methodology usability creates loss of time resources and other problems [17]. The usability has several components out of which only the most important ones will be taken into account here. Vendor support may be important to the methodology user in the introduction phase. Today the user often wishes to know whether the methodology is computer-aided or not, or if it is easy for the designer or for the end-user to comprehend and learn the application of the methodology. Does the methodology offer any teamwork support? Some methodologies explicitly specify the roles of participation in each ISD project activity. And finally, the extent of the methodology use in different types of projects and in different business areas naturally interests the user of the methodology. Thus, the components of the methodology usability are:

- vendor support,
- computer support
- comprehensiveness,
- teamwork support,
- user roles and
- extent of use.

*Consistency and control*: Consistent philosophy and complexity control strategies are important aspects in the methodology evaluation. Cohesion tells us how well the functions, methods, techniques and tools are connected to each other, e.g. through a philosophy of the methodology [5 ]. Structuredness is a measure of the extent to which the principle of structuring and stratifying is applied to the design processes, to the description languages , etc. (see [9, 26]). Formality expresses the amount of formal methods, logics, structures, foundations and techniques applied by the methodology. Performance measurement of the IS is often explicitly included in the different phases and levels of the IS design. And finally, criteria and techniques for the design and manufacturing quality should be defined for all models and all steps. The components of consistency and control are:

- cohesion,
- structuredness,
- formality,
- performance measurement and
- quality control.

*Benefit measurement*: The use costs and the cost/profit ratio are generally taken into consideration when planning the use of a methodology or a new tool. Organizations are also interested in the investment costs of a new methodology [4, 6, 7, 11, 17, 28, 32]. Often a methodology, that has been highly evaluated , is not cost-effective in other respects. Such factors are:

- use costs and
- cost-effectiveness.

## 4.2. External Validity of Frameworks

The inspection of the interstructure of frameworks will here be based on the definition of a metamethodology. The definition presented above is largely oriented by the use of a metamethodology. It should first describe an ISD methodology, its structure, scopes, paradigms, approaches, principles, emphases, usability, coverage, flexibility and general applicability. These requirements are moderately well- satisfied in our scheme intrastructure. And secondly, it should assist the user to analyse, evaluate, compare, comprehend and choose the ISD methodologies; it should help in clarifying the conceptual fuzziness and deficiencies of the field. It is difficult to evaluate how useful a metamethodology is for its purpose, but this can at least partly be done by studying the factors grouped in the following subtitles 'framework foundations' and 'usability'.

*Framework foundations*: Differences and similarities of the framework foundations can be revealed by studying the purposes of the frameworks, their object groups [6], methods for complexity control [5 ] and general structures [23]. Hierarchy and stratification structures of the frameworks [17] can be used for the comparison of their structures. Various opportunities offered by the frameworks are also noteworthy. Hence, the factors are

- purpose of framework,
- object group of framework,
- complexity control methods,
- general framework structure,
- opportunities offered by framework,and
- objects described by entities.

*Usability*: The usability of the frameworks can be studied, as in the intrastructure, from several viewpoints, but we dwell on the following: ease of use implies the existence of social factors and human roles in metamethodologies. For example, can people at different organisational levels and with different interests (e. g. top managers, adp specialists, operational leaders and end-users) communicate by means of the frameworks [21, 17]? Practicability can, for example, be measured by the description methods, techniques and concepts used in the frameworks as they influence comprehensibility. Also, it is of great interest today to find out whether the frameworks are computer-aided or not [17]. The flexibility of metamethodologies means , let's say, ease to add new parts or details and to update them [17]. From the point of view of marketing the frameworks and increasing their usability, it is important to know the background organizations, sponsors and other support institutions which have been developing and supporting the activities of each framework. Considering this information , it is possible that the organizations that could offer more application knowledge about the frameworks, should be identified [6]. The usability factors are

- ease of use,
- practicability,
- flexibility and
- sponsors.

## 4.3. Coverage of Frameworks

A good practical means of analysing and comparing the three frameworks of reference would be to empirically try to analyse real ISD methodologies with them. This approach, however would not reveal all the characteristics presented above. Nevertheless, our approach- although at first sight presumably theoretical- is rather practical because of its content including both largely theoretic-scientific knowledge and versatile practical views and experiments of the last two decades [2,5,35,36]. This has determined an extensive analysis and comparison scheme. It could have included even more different views and details, if our work did not prove that the defined scheme well covered the universe of discourse and was able to indicate enough similarities and differences of the frameworks.

As a summary of the comparison scheme and as a starting point of our evaluation, see Tables 5 and 6.

# 5. Comparison Results

Based on our evaluating scheme constructed in Section 4, we are now going to analyse how each criterion has been followed in the three reference frameworks. We would try to study the influence the possible differences might have on the use of the ISD methodologies.

*In this analysis, the entities and concepts of the frameworks have been selected to represent the frameworks*. We wish to remind the reader of the main different natures of the entities: FfU dwells on the components of design products, FoR describes the ISD process in more detail (in the process model) but also the IS itself, its environment, ISD methodologies, methods and tools, and system performance, while FRISCO focuses on the philosophy of the IS. *In order to make a comparison of the coverage of these frameworks, we have had to make interpretations and inferences from the defined concepts and entities that are not of the same nature and from definitions which do not correspond one-to-one*. It is apparent, for example, that as to the design process we can infer something from the design

**Table5. Internal Validity of Frameworks. This table includes the comparison factors of ISD methodologies described by the frameworks. Value 2 corresponds to a clear inclusion, value 1 a weak inclusion and 0 a missing inclusion of the factor.**

| Comparison Factors | FfU | FoR | FRISCO |
|---|---|---|---|
| *Areas of design* | | | |
| – organisational, social and functional design | 2 | 2 | 2 |
| – user interface design | 2 | 1 | 1 |
| – adp-technical design | 0 | 2 | 1 |
| *Design functions or phases* | | | |
| – project definition | 1 | 2 | 0 |
| – situation analysis | 2 | 2 | 1 |
| – systems design | 1 | 2 | 1 |
| – implementation of changes | 0 | 2 | 0 |
| – systems performance monitoring and maintenance | 0 | 2 | 0 |
| *Design approaches* | | | |
| – prototyping | 0 | 2 | 0 |
| – waterfall approach | 2 | 2 | 0 |
| – function based approach | 2 | 2 | 0 |
| *Design principles* | | | |
| – contingency and flexibility | 1 | 2 | 1 |
| – user participation | 1 | 2 | 1 |
| – problem orientation | 2 | 2 | 0 |
| – iteration | 0 | 1 | 0 |
| – experimentation | 0 | 1 | 0 |
| *Modelling scopes* | | | |
| – data | 2 | 2 | 2 |
| – process | 2 | 2 | 2 |
| – organization | 2 | 2 | 2 |
| *Modelling levels* | | | |
| – abstract models | 2 | 2 | 2 |
| – descriptive models | 1 | 2 | 1 |
| – specification models | 0 | 2 | 1 |
| *Working facilities* | | | |
| – methods | 1 | 2 | 0 |
| – techniques | 1 | 2 | 0 |
| – tools | 0 | 2 | 0 |
| – languages | 2 | 2 | 2 |
| *Usability* | | | |
| – vendor support | 0 | 2 | 0 |
| – computer support | 0 | 2 | 0 |
| – comprehensiveness | 0 | 2 | 0 |
| – group work support | 0 | 1 | 0 |
| – user roles | 2 | 2 | 0 |
| – use extent | 0 | 2 | 0 |
| *Consistency and control* | | | |
| – cohesion | 0 | 2 | 0 |
| – structuredness | 0 | 2 | 0 |
| – formality | 0 | 2 | 0 |
| – performance measurement | 2 | 2 | 0 |
| – quality control | 1 | 2 | 0 |
| *Benefit measurement* | | | |
| – use costs | 1 | 2 | 0 |
| – cost-effectiveness | 2 | 2 | 0 |

**Table 6. External Validity of Frameworks. This table includes the comparison factors of the external characteristics of the frameworks.**

**Framework Foundations**

- *purpose of framework*

  FfU: to evaluate, teach, study and understand IS methodologies, and to present the techniques used in IS methodologies in a way that shows how the various techniques interact with each other and how such techniques may be integrated FoR: to identify the different aspects and gaps of existing ISD methodologies, methods and tools, to determine the performance criteria that the methods and tools should satisfy, and to define what is required to ensure the selection of effective and acceptable ISD tools, methods and methodologies under the given circumstances FRISCO: to reach a better understanding of concepts in the information system area, to describe this conceptual understanding and to propose a reasonable terminology for the IS profession

- *object group of framework*

  FfU: students and teachers, practitioners and computer professionals and researchers

  FoR: IS decision makers at the user department level, ISD project leaders, IS consultants and IT professionals, and management and IS students at Master level

  FRISCO: the scientific IS community

- *complexity control methods*

  FfU: top-down principle, EAR modelling techniques

  FoR: top-down principle, applied EAR models, structured Petrinet explosion charts

  FRISCO: unstructured analytical essay, applied EAR models

- *general framework structure*

  FfU: information system planning divided into business plan, organization and mental model related, and information systems plan and cross-references; and both business analysis and system design divided into data, process and behaviour perspectives and cross-references between perspectives; definition of concepts

  FoR: internal and external environments, IS itself, its static and dynamic performance measurement, methodologies, methods and tools; relationships between entities in the EAR models; dynamic characteristics of information systems in process model; definition of concepts

  FRISCO: collection of viewpoints on systems classification, organizational systems, abstractions, and descriptive concepts

- *opportunities offered by framework*

  FfU and FRISCO: a tool for helping us understand certain aspects of ISD and IS functions, activities, design products and relationships of concepts

  FoR: a harmonizing framework to help us bring together decision makers, researchers, consultants and IT suppliers to apply common languages and common approaches and to identify the gaps in method and tool markets

ctd...

**Table 6** (continued):

- *objects described by entities*

  FfU: components of design products in information systems planning, business analysis, and system design

  FoR: internal and external environments, IS itself, its static and dynamic performance measurement, methodologies, methods and tools

  FRISCO: system theoretical and philosophical essence of ISs

**Usability**

- *ease to use and practicability of frameworks*:

  FfU: serves well EAR - and database oriented users who respect the data analysis concepts and approaches

  FoR: includes a computerized ISD process model and a DSS prototype to help decision makers in methods and tools selection; gap analysis of a methodology is possible on the basis of analysis of entity coverage

  FRISCO: offers a semantic means in a communication process within the scientific IS community

- *flexibility of frameworks*

  FfU, FoR and FRISCO: they all can be enlarged and completed without difficulties; level of details can freely be selected; because the domain of FRISCO is fairly wide in the philosophical sense, it needs a lot of additional work to fill the gaps in its certain subdomains

- *sponsors*

  FfU: IFIP and 7 research institutions and IT industrial companies and consultancies from 5 countries representing a wide background and potential usage basis

  FoR: ESPRIT and 9 research institutes and IT industrial companies and consultancies from 5 countries representing a wide background and potential usage basis

  FRISCO: IFIP and 9 academic or near-academic research institutions from 6 countries representing a wide background and potential usage basis products, but the difficulty lies in how far we can go this way.

There is no doubt that the entity based approach in the analysis of frameworks would be too narrow-sighted, and therefore other properties presented in the manuals on frameworks, have also been considered in our analysis.

## 5.1.Internal Validity of Frameworks

Areas of design:The IFIP FfU takes partly defectively into consideration the design areas: *Organizational design* is well included in the **business planning of information systems planning** and *functional design* has been reckoned **in process views** of both **business analysis** and **systems design**, but no attention has been paid to the features of *social design* or *adp-technical design*. *Human-computer interaction design* can be seen as **display option, menu hierarchy, menu** and **task in menu** of **system design components**, a rather restricted approach. See Table 5.

The HECTOR FoR has paid more attention to all these areas: *organizational design* can be found in **environment, organizational structures, performance measurement** and in **dynamic** and **static organizational measures in OIS**, and **application aspects in methodologies**. *Social design* has been considered as **social measures in OIS performance measurement**. *Functional design* appears **in environments work structures, in OIS organizational structures, in methodologies' application aspects as structural alternatives** and as **decomposition/aggregation** of **work processes**. *Human-computer interaction design* can be seen only in the process model and as **user interface in tools** properties, not too much after all. *Adp-technical design* is well taken into account in **OIS technology** and in the technical properties of **tools**.

Thus FfU emphasizes functional design , while other important areas are not supported. FoR covers well most of design areas. FRISCO discusses *organizational design* (e.g. **organizational formalization, organizational abstraction, organizational subsystems**), *social design* (e.g. **social communication, interests, beliefs, commitments**), *functional design* (e.g. **actor, activity, event, internal and external function, function domain, process, reactive systems**), not so much about *human-computer interaction design* (e.g. **the system viewer, communication systems**), and quite a lot about *adp-technical design* (e.g. **media, database, datalogical abstraction, data handling system, information structure, computation, data processors**).

*Design functions or phases*: *Project definition, situation analysis, systems design, implementation of changes and systems performance monitoring and maintenance* cannot be found as design functions in any form in FfU (where **project plans** can be found in **information systems planning and situation analysis in business analysis**, but it is not what we mean here) or in FRISCO (where **infological and datalogical abstractions** are mentioned at the level of the names of functions intended for ISD). FoR includes *all* these main functions in a separate **process model**. For example, project definition is given accordingly both in the **definition of OIS strategies** and later on in a local project launching.

*Design approaches*: *Prototyping, waterfall approach and function based approach* are all taken into account in FoR in several ways. Of all these approaches, FfU covers well only the *function based approach* and it can be seen as **process perspectives and behaviour perspectives**. FRISCO does not discuss ISD approaches.

*Design principles*: In FfU *contingency and flexibility* can be interpreted as going with **information systems planning** only. In FoR the contingency principle and the need for **flexibility in methodologies** as well as the **customization capabilities** have been twice expressed explicitly; environmental and organizational circumstances and also social factors are detectable in all functions. FfU expresses *user participation* only in manual text in connection with user roles while in FoR it can be seen in **internal environment**, in **OIS organizational structures**, in **performance measurement** of **organizational climate** and in **methodologies conceptual framework**. Both FfU and FoR take *problem orientation* well into account. *Iteration* cannot be seen in FfU but it is somehow included in FoR's **process model**.

*Experimentation* is not explicitly included in the entity lists of these frameworks. However, FoR has many connections to it by sustained user participation and use of methods and tools, and also the FoR's **process model** includes it.

FRISCO does not cover ISD principles but several *contingency* and *flexibility* aspects are perceivable e.g. in **interest groups, IT user, actor and environment.**

Modelling scopes: The *data, process and organization* scopes (cf. perspectives in FfU, with slight differences) are all well- indicated in these frameworks. The data and process scopes are emphasized in FfU while the organization scope can be seen only in **business analysis** and **planning,** not as part of the system design. The organization scope is emphasized in all the components of FoR: in **OIS, methodologies** and **tools.** FRISCO discusses all these scopes from system's theoretical viewpoints.

Modelling levels: This group of evaluation criteria is rather difficult to analyse. FfU moves well in the *abstract and descriptive model* levels in **business planning** and in the *specification model* level in certain areas (i.e. **data, process and behaviour**) of **system design.** Several aspects of IT technical specification, for example, are not considered. FoR rightly evaluates all these modelling levels , and even performance and quality control aspects of organizational information system (a wide range of technical aspects of **information technology** and **organizational design** specification) which mostly belong to specification models. FRISCO discusses these levels when it defines **abstraction levels, models,** and **linguistic** and **descriptive concepts.**

Working facilities: In the FfU component structure, *methods, techniques and tools* play no role. In general, the FfU manual includes a lot about methods and techniques, and about *languages*, it includes a whole chapter titled 'Representation and documentation'. These criteria are extremely well-satisfied in FoR which emphasizes the working facilities and methodologies, especially in OIS improvement projects. Techniques and tools are connected to methods for supporting them. For example: *methods* are handled with **methodologies from many points of view;** *techniques* are present in **application aspects** of methodologies as several **information acquisition techniques, representation techniques** and **requirements for computer support;** *tools* are specified versatilely from the points of view of **technical, usage** and **application aspects** and **performance evaluation.** *Languages* and especially description languages are included in **methodologies, methods, techniques** and **tools** in FoR. FRISCO does not cover ISD *methods, techniques* and *tools* but it largely discusses *languages*.

Usability: FfU entity model does not include any aspects of methodology usability. Only in the FfU manual text (OHM88 pp. 189-190) would the system design **roles** appear. In FoR these usability criteria (*vendor support, computer support, team work support, comprehensiveness, user roles, extent of use*) are explicitly and repeatedly met in connection with **methodologies** and **different working facilities,** most of them in several components (e.g. as **application aspects**) and from several viewpoints. FRISCO does not cover *usability* aspects of ISD methodologies.

Consistency and control: The FfU entity structure does not include many consistency and control aspects except for those of *performance measurement and quality control.* In FoR, the integration of methods, techniques, tools and description languages (i.e. *cohesion*) is achieved, e.g. in **conceptual framework of methodologies** as **subordinate method,** as **compatibility with organizational procedures** and as **compatibility with other methods,** as a set of suggested **techniques, languages** and **tools,** as different aspects of **computer support,** and as **application aspects** of **methods** and **tools.** *Structuredness* is a general property of the HECTOR **process model** and it has been expressed in many ways, e.g. as **structuring capabilities of application aspects,** as **decomposition diagrams** and as **hierarchical business modelling** in **representation.** *Formality* appears as **formalisms** and, to some extent, as **modelling capabilities.** *Performance measurement* has a special, strong position in HECTOR FoR and it is included there versatilely. *Quality control* appears e.g. in **methodologies** as **quality assurance** and in several aspects of **performance measures,** e.g. **quality improvement.** Consistency and control aspects are not explicitly covered by the FRISCO model.

Benefit measurement: The benefit measurement factors have been treated by FfU from the point of view of the OIS development project and its possible economic advantages to the organization. FoR investigates the gains of methodology usage for the project and the organization (**effectiveness, efficiency: training costs, other initial costs, project labour costs and other on-going costs** in entity lists and **user needs satisfaction** in process **model**). The profit factors have been considered in connection with methodology performance measurement (**profit increase, cost reduction, productivity increase**). So, both the *use costs* and *cost-effectiveness* aspects have been well taken into account in FoR. FRISCO does not cover these aspects.

As a general impression of the interstructures of these frameworks, we can say that FfU emphasizes the aspects *what* IS components, i.e. design products, are described in the methodologies or which entities are essential to each component. FoR emphasizes both the selection of the objects and components, i.e. *what* is described about the system, organization and methodologies, and also *how* and *based on which methods and tools* this is done. FRISCO reveals the philosophic and theoretical aspects of IS but says little of ISD.

## 5.2. External Validity of Frameworks

In this paragraph we analyse the aspects of employing the frameworks to find out how different their foundations and purposes actually are, and how practical they might be. See Table 6.

*Framework Foundations*

The major differences between these frameworks are explained by their different usage *purposes*. FfU has, according to its definition, been developed to be a tool for evaluating, teaching, studying and understanding the information system methodologies. The emphasis is on strategic information systems planning and business analysis, not so much on system design (i.e. logical ISD) and not at all on construction design (i.e. physical ISD specification) and implementation. There have been three goals in the development of FoR: to identify the different aspects and gaps of the existing ISD methodologies, methods and tools, to determine the performance criteria that methods and tools should satisfy, and to define what is required to make sure that there would be selected effective and acceptable tools, methods and methodologies for the analysis, design, implementation and evaluation of integrated organizational and office information systems under the given circumstances. FRISCO has been created "for reaching a better understanding of concepts in the information system area, for describing this conceptual understanding and for proposing a reasonable terminology for our profession".

Also the *object groups of the frameworks* differ. FfU has been developed rather generally for students and teachers, practitioners and computer professionals and researchers. FoR's object groups are IS decision-makers at the user department level, ISD project leaders, IS consultants and IT professionals, and management and IS students at Master level. It seems to a reader of [19] that FRISCO is directed to the scientific IS community, not to people in ISD projects. At least emphasis differences can be seen in these groups: though all three frameworks are theoretical and analytical, FfU and FRISCO aim at explaining things while FoR is more practice-oriented and user-centred taking the ISD and environmental contingencies better into account.

The *complexity control methods* of the ISD frameworks are rather similar at first sight: both FfU and FoR apply the same top-down principle proceeding on from larger concepts to smaller ones while FRISCO is mainly in the form of an unstructured essay. FfU applies the EAR modelling techniques to connecting the entities together through the framework. And independently of the IFIP WG 8.1 Task Groups' work the HECTOR Project drew on the applied EAR models to serve its description needs. In addition, the HECTOR FoR offers Petri net-based explosion charts of the ISD process model, which clearly illustrate the different level connections of the ISD activities, tasks and results. FRISCO applies EAR in some cases.

The *general framework structures* are different. FfU consists of various life cycle stages: **information systems planning** is divided into **business plan, organization** and **mental model related**, and **information systems plan** and **cross-references**; and both **business analysis** and **system design** are divided into **data, process** and **behaviour perspectives** and **cross-references between perspectives**. FoR consists of **internal** and **external environments, OIS** itself, its **static and dynamic performance measurement, methodologies, methods** and **tools**. FoR also includes cross-references in the EAR models. The **process model** of FoR was constructed to describe the dynamic characteristics of information systems. Both frameworks define their concepts carefully. These different basic structures can be regarded as the main aspects to help us understand the differences between the two ISD frameworks.

It is difficult to find the skeleton of FRISCO structure. It is a collection of viewpoints on **systems classifications, organizational systems, abstractions, and descriptive concepts.**

The *opportunities offered by frameworks* can be derived from their usage purposes, contents and structures. FfU and FRISCO are tools for helping us understand certain aspects of IS and ISD functions, activities and relationships of concepts. FoR is a harmonizing framework helping us bring together decision-makers, researchers, consultants and IT suppliers, who will apply common languages and common approaches and identify the gaps on method and tool markets.

The *objects described by entities* have a totally different nature in these three frameworks. FfU describes the components of design products from information systems planning, business analysis, and system design. The entities in FoR describe the internal and external environments, OIS itself and its development process, the static and dynamic performance measurement, and the methodologies, methods and tools. FRISCO concepts describe the system theoretical and philosophical essence of IS.

*Usability*

The questions of *ease of use* and *practicability* could be handled in several ways. As the frameworks are supposed to act as a tool- like facility in supporting their users, we here search for the features of the frameworks which might assist the user in quickly adopting the main messages of the frameworks. As regards the object groups and usage purposes of the frameworks we can say that *they serve well their purposes in their own fields and user groups*. FfU is very strongly EAR- and database-thinking oriented, and it serves well the practitioners and teachers who know and accept data analysis concepts and approaches. FoR is directed towards development process, methods and tools. HECTOR has made a computerized process model of the main part of the framework and a software prototype OISEAU to help decision makers in the selection of methods and tools. FoR offers a more holistic picture of ISD activities and it properly takes into consideration social and organizational aspects. FRISCO offers a semantic means for the communication process of the scientific IS and ISD communities. All these frameworks have been published only in English.

The *flexibility of the frameworks* can be evaluated e.g.in how easily they can be enlarged or completed with details. All these frameworks offer a lot of opportunities in this sense. The HECTOR's process model in FoR offers 4 levels of details so that the user should choose the most suitable one for her/his purposes. The aspects and entities can easily be added to both frameworks.

In information systems science most theories and models, even tools and systems have originally been developed in universities and research institutes. The gap between theory and practice takes often several years to bridge. Often the business aspects, social factors and user roles and aspects have been omitted and the research results have not been able to reach the business users. The frameworks under analysis here represent a new type of results. Besides academic researchers, there have been several practitioners in developing these frameworks and the background *sponsors* (IFIP and ESPRIT) are application-oriented. However, the majority of developers of FfU and FRISCO have been university and research institutes' people while FoR has been more strongly supported by practitioners and consultants.The developers of FfU represent 5 countries and 7 institutions, the developers of FoR 5 countries and 9 institutions, and the developers of FRISCO 6 countries and 9 institutions, which means, considering their sponsorships, opportunities for wide spread.

## 6. Conclusions

In this paper we have constructed an analysis scheme and presented the main results of an evaluatory comparison between the most remarkable existing reference frameworks for information systems and their development. The differences of these reference frameworks can be derived for example from the differences in the chosen

- paradigms and worlds of values,
- emphasis put on along the IS life cycles,
- description techniques,
- levels of details and
- views and approaches to social, organizational and technical systems.

The choice of concept names for the frameworks has been difficult as noted in [25 ]. The different backgrounds of the framework builders have led to great differences in the definitions of these three frameworks. The reader of the manuals of these frameworks can hardly avoid the suspicion that it is perhaps not possible to make any strict or even fair comparison of them:

1. FfU focuses more on business analysis and strategic information systems planning and also, as its main approach, on the data-oriented perspective and on conceptual modelling. FoR analyses the ISD process, the factors of the use of methodologies, methods and tools in ISD projects and the performance measurement under different circumstances. FRISCO points to the essence of IS concepts.

2. The FfU entities focus on the components of design products, the FoR entities more on the ISD process (in the process model), but also the IS itself, its environment, ISD methodologies, methods and tools, and system performance. FoR takes the design products into account somehow but not in any case at the same level of details as FfU does. FRISCO approaches the IS concepts mostly from the system theoretical and philosophical viewpoints.

The summaries in paragraphs 1 and 2 also describe the strengths and weaknesses of these frameworks.

3. There is no one-to-one correspondence in the views, definition of concepts and entities, and the purposes of the frameworks. That is a reason, for example, why *it is not possible to apply one framework to another in order to reveal their possible gaps*.

4. What is then the influence of the differences of these reference frameworks on the ISD methodologies? In the short run not much, because there are not many methodology developers to have been acquainted with all the three documents [19, 25, 27]. Some IT experts with different backgrounds might not accept all these frameworks with various paradigms while hopefully many will find a lot of good harmonizing work done in them. We can say that all these frameworks are useful for their own purposes and somehow they complete each other.

Our evaluation and comparison of the three IS and ISD reference frameworks were possible by inferring their coverages from their concepts and entities of different nature. This was, no doubt, a little bit risky. In spite of the danger of concluding wrongly in some cases, we have tried to offer a deeper understanding of the natures of these reference frameworks.

## REFERENCES


1. AHITUV, N., **A Metamodel of Information Flow: A Tool To Support Information Systems Theory**, COMM. ACM, Vol. 30, No. 9,1987, pp. 781-791.
2. AVISON, D.E. and FITZGERALD, G., **Information Systems Development: Methodologies, Techniques and Tools**, BLACKWELL SCIENTIFIC PUBLICATIONS , London, 1988.
3. BAROUDI, I., OLSON, M. and IVES, B., **An Empirical Study of the Impact of User**

**Involvement on System Usage and Information Satisfaction**, COMM.ACM, Vol. 29, No. 3, 1986, pp. 232-238.

4. M.A. Bemelmans (Ed.) **Beyond Productivity: Information Systems Development for Organizational Effectiveness**, NORTH-HOLLAND, Amsterdam, 1983.
5. BLOKDIJK, A. and BLOKDIJK, P., **Planning and Design of Information Systems**. ACADEMIC PRESS INC. LTD, London, 1987, pp. 22,75.
6. BRANDT, I., **A Comparative Study of Information Systems Design Methodologies**, in T.W. Olle , H.G. Sol , C.J. Tully (Eds.) Information System Design Methodologies: A Feature Analysis. NORTH-HOLLAND,1983, pp. 9-36.
7. G. Bracchi and D. Tsichritzis (Eds.) **Office Systems: Methods and Tools**, NORTH-HOLLAND, Amsterdam, 1986.
8. BRINKKEMPER, S., **Applications of Metamodelling in Information Systems Research**, Paper presented at the First International Summer School on Metamodelling and Methodology Engineering, University of Jyväskylä, June 1992.
9. BULL, M., **Systems Development Using Structured Techniques**, CHAPMAN & HALL LTD, London, 1989.
10. CHECKLAND, P., **Systems Thinking, Systems Practice**, WILEY, 1981.
11. D. W. Conrath and P.J. Dumas (Eds.) **Office Support Systems Analysis and Design**, Final Report on Office Modelling, Language and OSSAD Methodology, ESPRIT Project No. 285, R&D Area 4.1, Office Systems Science and Human Factors, IOT, Munich, 1989, pp. 2,17-19.
12. CONRATH, D.W., **An Overview of HECTOR's Results:Methods and Tools for Improving Organizational Information Systems**, HECTOR Project, London, 1990.
13. DAVIS, G.B., **Strategies for Information Requirements Determination**, IBM SYST.J.,Vol. 21, No. 1, 1982, pp. 4-30.
14. DE MARCO, T. and LISTER, T., **Peopleware Productive Projects and Teams**, DORSET HOUSE PUBLISHING CO., New York, 1987, pp. 3-4.
15. EGGELSTON, R.G., **The Changing Nature of the Human-Machine Design Problem: Implications for Systems Design and Development**, in [29].
16. EPISKOPOU, D.M.and WOOD-HARPER, A.T., **Towards a Framework To Choose Appropriate IS Approaches**, THE COMPUTER JOURNAL, Vol. 29, No. 3, 1986, pp. 222-228.
17. HIRSCHHEIM, R., **Office Automation - A Social and Organizational Perspective**, JOHN WILEY & SONS, Chichester, 1985, pp. 146,148-149.
18. KUNG, C.H. and SÖLVBERG, A., **Activity Modelling and Behaviour Modelling**, OSV86, pp. 145-172.
19. P. Lindgreen (Ed.), **A Framework of Information Systems Concepts**, Interim Report, IFIP WG 8.1 Task Group FRISCO, University of Nijmegen, 1990, pp 8,9,13,21.
20. E. Mumford, R. Hirschheim, G. Fitzgerald and A.T. Wood- Harper (Eds.), **Research Methods in Information Systems**, NORTH-HOLLAND, Amsterdam, 1984.
21. MUMFORD, E., **Participative Systems Design: Structure and Method,in Systems, Objectives, Solutions I**, NORTH-HOLLAND, Amsterdam, 1981, pp.5-19.
22. OLLE, T.W., **Comparative Review of Information Systems Design Methodologies. Stage 1: Taking Stock**, in [23], pp. 1-14.
23. T.W. Olle, H.G. Sol and A.A. Verrijn-Stuart (Eds.) **Information Systems Design Methodologies: A Comparative Review**, NORTH- HOLLAND, Amsterdam, 1982.
24. T.W. Olle , H.G. Sol and A. A. Verrijn-Stuart (Eds.) **Information Systems Design Methodologies: Improving the Practice**, NORTH-HOLLAND, Amsterdam, 1986.
25. T.W. Olle , J. Hagelstein , I.G. Macdonald , C. Rolland ,H.G. Sol , F.J.M. Van Assche and A.A. Verrijn-Stuart (Eds.) **Information Systems Methodologies: A Framework for Understanding**, THE BATH PRESS, Avon, 1988, pp. 6,12-13,38,41-43,55-56.
26. POWERS, M.J., CHENEY, P.H., and CROW, G., **Structured Systems Development: Analysis, Design,**

**Implementation**, BOYD AND FRASER PUBL. CO. , Boston, MA, 1990.
27. PULST, E., WOLFRAM, G., LECLERC, A., PUERTA, M.C., MANFREDI, F., BLEECKE, J., KIEBACK, A., GÖRNER, C., KLEIN, B., NIEMEIER, J., BESLMÜLLER, E., FRESKO, M., CONRATH, D., HAWGOOD, J., LEPPÄNEN, M. and SAVOLAINEN, V., **HECTOR's Framework of Reference,.** Report of ESPRIT 2082 Project HECTOR: Harmonized European Concepts and Tools for Organizational Information Systems, Fraunhofer Institute for Industrial Engineering (IAO), Stuttgart, 1990.
28. ROSS, G., **HECTOR Market Report**, PA Consulting Group, London, 1990, pp 3-7.
29. W.B.Rouse and K.R. Boff (Eds.) **System Design, Behavioural Perspectives on Designers, Tools, and Organizations**, North-Holland Series in System Science and Engineering, NORTH-HOLLAND, New York, 1987.
30. SAVOLAINEN, V., **Enlarged Ossadic Approach to Organizational Design**, in Z. Kaltnekar , J. Gricar (Eds.) Proceedings of International Conference on Organization and Information Systems, School of Organization, Kranj, 1989, pp. 120-125.
31. SAVOLAINEN, V., **Technical Specification Model of An Information System**, Paper presented at the Tenth International Conference on Systems Science, Wroclaw, 1989.
32. SAVOLAINEN, V., **Definition of Favourable Atmosphere for Effective IT Decisions**, in H.G. Sol and J. Vecsenyi, (Eds.) Environments for Supporting Decision Processes, ELSEVIER SCIENCE PUBLISHERS B.V., NORTH HOLLAND, 1991, pp. 129-140.
33. SAVOLAINEN, V., **Comparison Analysis of Design Methodologies and Reference Frameworks for Information System Design**, in H.-J. Bullinger (Ed.) Human Aspects in Computing, ELSEVIER SCIENCE PUBLISHERS , Amsterdam, 1991, pp. 1135-1139.
34. WATERS, S.J., **Introduction to Computer Systems Design**, WILEY, New York, 1973.
35. WOOD-HARPER, A.T.C. and FITZGERALD, G., **A Taxonomy of Current Approaches to Systems Analysis**, THE COMPUTER JOURNAL, Vol. 25, No. 1, 1982, pp. 12-16.
36. WOOD-HARPER, A.T.C., **Comparison of Information Systems Definition Methodologies: An Action Research, Multiview Perspective**, Ph.D.Diss., University of East Anglia, 1989, pp 107-110.
37. ZACHMAN, J.A., **A Framework for Information Systems Architecture.** IBM SYSTEMS JOURNAL, Vol. 26. No. 3, 1987, pp. 276-292.