Spring 3-27-2012

Meta Meta Modelling, Carnap, and Innovating Information Systems

Geoffrey Darnton
Requirements Analytics, gdarnton@darnton.info

Meta Meta Modelling, Carnap, and Information Systems

Geoffrey Darnton
Requirements Analytics, Bournemouth
Email: gdarnton@darnton.info

Abstract

In general, the field of information systems is misconceived: firstly, the term 'information system' is in itself an oxymoron, or extremely close to that; secondly, the traditional scope of books and courses about information systems is too narrow. This is illustrated using definitions or concepts of IS from numerous books. There is an exploration of course materials used for a much wider view of IS demonstrating the feasibility of innovating IS. An examination of the early evolution of the worlds of analysis and design for computer-based information systems comes to the conclusion that the most likely source is the philosopher Rudolf Carnap. Carnap's approach to languages and metalanguage is explored, with the conclusion that the same approach is extensible to a much wider concept of IS. Therefore, innovating the IS concept can be supported by extending the same analysis foundations currently in use. This paper shows how a meta meta modelling approach derived from Carnap's work is perfectly able to model a broad concept of 'information system'. The paper demonstrates how all the systems analysis techniques used in the computer world for the last half-century can be derived and consolidated using a meta language created with Carnap's constructor principles.

Keywords: Information System Innovation, metametamodel, metametalanguage, metametametalanguage, Carnap

1.0 Background and Introduction

The theme of this year's conference is 'Innovating IS'. This is a welcome and necessary debate. The thrust of this paper is that the field of IS needs a radical overhaul. 'IS' has lost its way - if ever it had a well-defined way.

This paper takes a wide journey. There is a brief review of several books that various people consider as key textbooks in the IS field. The broad conclusion is that in general, the IS field, as portrayed in many textbooks, is fundamentally misconceived for two essential reasons: (1) the term 'information system', when unpacked, is bordering on being an oxymoron, hence the terms 'Information System' and 'IS' can really only be considered as symbols of an as yet uncertain field of study; (2) the field is primarily focused on computers and their uses, and this does not constitute anything other than a small proportion of the total amount of information work going on (Darnton and Giacoletto, 1992; Machlup, 1962; Vincent, 1990). The IS field, as it is seen today by many authors, teachers and practitioners, does not even represent the scope indicated by early work (Machlup, 1962) that attempted to answer the question "if we are in an information era, or an information economy, how big is that in relation to the whole economy?"

The paper then provides examples of courses that have been used with several hundreds of master's students and which have resulted in a substantial broadening of the scope of IS well beyond ICT. As far as ICT is concerned, the section discussing broadening the scope of IS teaching concludes with a proposal of a number of generations of ICT going back into antiquity. In a pure IS sense, there is no fundamental problem with the idea that someone who is illiterate could also be highly educated.

This is followed by a discussion of approaches to modelling which have evolved in the IS and computer worlds for the analysis of systems and system requirements. The paper demonstrates that the most likely origin of much thinking in those worlds is the philosopher Rudolf Carnap, who's work pre-dates the modern computer era. Not only does it pre-date the modern computer era but also it has been used as major foundation stones for the computer world, but it is also eminently suitable for supporting analysis in a substantially innovated IS field.

There is nothing inherent in current approaches to modeling computer-based information systems which make them unsuitable for modelling the wider systems required by an innovated wider concept of IS.

2.0 Is 'Information System' an Oxymoron?

Information alone cannot be a system. A system will have many more components than information. Of course, this problem is central to the discussion by Checkland and Holwell (1998), although they basically admit that much more is needed for a concept of 'information system': "...the nature of an information system is that it is a function supporting people taking purposeful action..." (p110). So an information system is a function?

Therefore, it is necessary to look wider than information to understand information system.

3.0 IS Textbooks

Several textbooks with 'information systems' in the title have been selected to look at definitions (or concepts) of information system. There was nothing systematic about the choice. The selection is shown in Table 1.

Book/ Source	**Pp**	**Definition**
Aktas (1987)	10	The information system of an organisation may be defined as a system that serves to provide information in the organisation when and where it is needed at any managerial level. Information in an information system must be understood by its recipient in the proper frame of reference; it must be relevant to current decision-making process, it must have surprised value, and it must lead to user interaction. Effective processing and management of information, flexibility, and user satisfaction are some of the desirable attributes of an information system. Aktas also quotes Bryce (1983) "equating Information Systems to computer systems is a misconception born of decades of preoccupation with technology".
Alter (1999)	4	....an information system is a work system that uses information technology to capture, transmit, store, retrieve, and print, or display information, thereby supporting the work systems.
Benson and Standing (2005)	6	The definition of an information system became broader, and the components of an information system were generally accepted as: people, data/information, procedures, software, hardware, communications. Having said that, the authors proceed to make a very interesting statement: "Information Systems have been round 6,000 years."
Boddy, Boonstra, and Kennedy (2002)	6	An information system is a set of people, procedures and resources that connects and transforms data into information and disseminate disinformation. The authors indicate that data transformation can be performed through people, procedures, hardware, software, paper etc.
Chaffey and Wood (2005)	21	Information system-"a computerised or manual system to capture data and transform them into information and/or knowledge. These authors go on to quote the UKAIS definition of information system (P 43) but add a comment "of course, Information Systems do not have to use technology. There are still occasions where can sections are recorded and processed

		on paper - this is a manual information system.
Curtis and Cobham (2005)	39	Management Information Systems are computerised systems that provide information for managerial decision-making. These systems rely on extracting and processing data from a commonly shared corporate database that stores results of transaction processing.
Benyon-Davies (2002)	4	An information system is a system of communication between people. Information Systems are systems involved in the gathering, processing, distribution and use of information.
Fortune and Peters, 2005	xi-xii	This book adopts a broader view that goes well beyond the integration of hardware and software. It considers an information system to be any system that has collection, processing, dissemination and use of information as a major component in terms of its purpose and the activities it carries out. Most modern information systems with any degree of complexity will, in practice, almost always incorporate ICT, but the technology is not the defining aspect. The significant issues are the generation, processing and use of information,
Gray (2006)	2	An information system is a combination of technology, people, and processes to capture, transmit, store, retrieve, manipulate, and display information. This is accompanied by a footnote which states: "this book is concerned with information systems that incorporate technology, particularly computer technology. Some information systems do not incorporate technology. Many of the same principles, however, apply.
Jessop and Valacich (2006)	5	Information Systems are combinations of hardware, software, and telecommunications networks which people build and use to collect, create, and distribute useful data, typically in organisational settings
Kroenke (2007)	4	An information system is a group of components that interact to produce information. That sentence, although true, raises another question: what are these components that interact to produce information? The author asserts these to be: computer hardware, software, data, procedures, and people.
Laudon and Laudon (2002)	7	An information system can be defined technically as a set of interrelated components to collect (or retrieve), process, store, and distribute information to support decision-making, coordination, and controlling organisation. In addition to supporting decision-making, coordination, and control, Information Systems feels managers and workers analyse problems, visualise complex subject, and create new products. The authors go on to distinguish computer-based Information Systems which are "information systems that rely on computer hardware and software for processing and disseminating information".

Table 1 - Information System Definitions and Concepts

For many texts with the term ‘Information System’ in the title, it is often not necessary to go beyond page 3 to conclude that it is focused primarily on computers and their applications.

In an undated web page document (UKAIS, 2012), the UKAIS has a definition of IS: "Information systems are the means by which people and organisations [sic], utilising [sic] technologies, gather, process, store, use and disseminate information".

Hence, it is clear that a key difference in concepts concerns whether or not there is a need for an information system to include technology. Many definitions do not require that. Hence it is reasonable to conclude that the current UKAIS definition is overly narrow. Many books accept there is no technology imperative for an IS. However, the use of information without technology is very under-developed in the literature.

4.0 Innovative IS Courses

Effort to innovate IS courses arose from the experience that so many students were arriving to study for a master's degree somewhat 'brain-damaged' by a belief that:

Information Systems ≈ Information and Communications Technology ≈ Computers

There were two contexts for the master's students. (1) All students who were enrolled for master's degrees in a Masters in Management program were required to take one module entitled Managing Information; (2) Students enrolled in the MA in Information Systems Management program were required to take an introductory module entitled Information Systems Fundamentals.

Both groups of students had two essays in common. The first essay required of students was to write about Information Systems that used zero technology. The second essay required students to write about ICT in use before 1900.

The students studying for an MA in Information Systems Management had two further essays to write requiring far greater depth of study and reflection in the subject matter. They were required to write two further essays. The third required them to write about music as an information system. That essay was inspired by musical instruments being defined my Machlup in his classification of information machines (Machlup, 1962). The fourth required them to write about culture as an information system. That was inspired by the realization that different cultures arise because of different information environments of the populations.

It is not necessary for this paper to attempt any empirical study of the topics presented by the students. What is much more important is understanding the key issues encountered by the students while attempting the essays. The first two essays have been attempted by hundreds of students and the other two essays have been attempted by approximately 150 students. Notwithstanding these high numbers of students, almost all students completed successfully.

The first essay required students to discuss Information Systems that use zero technology. This required students to get to grips with what constitutes technology. Does the expression ICT refer primarily to computer and communications artefacts that are used to support computer-based applications? If ICT has a broader meaning in terms of any artefacts used to support information work or play of any kind then what may constitute information technology takes on a different perspective. For such a broader meaning for ICT, artefacts such as books, pens and pencils, overhead projectors, chalkboards, and such like must be considered as examples of IT. Therefore, students had to learn very quickly that whether or not their proposed solution to the problem of talking about Information Systems that use zero technology is completely dependent on their definition of technology.

When students got to grips with articulating how they were using the term technology, they were able to proceed to describe and analyze information systems that used zero technology. As almost all students were able to do this task, the deeper IS subject issue arises as to the appropriateness of linking any definition of IS to technology. This approach taken by the students was also very effective in providing a solid foundation for the students to demonstrate critical thinking about IS and IT.

Thus the students were now able to think of Information Systems and technology as independent topics.

The second essay required students to consider ICT in use before 1900. Obviously this essay is not about computers! An unexpected consequence of setting this essay for students coming from a large number of different countries was that many students reverted to their own cultural background in order to identify examples of ICT in use before 1900. Given the hundreds of students involved, this has resulted in a rich archive of information which will be the subject of a different paper, or a focused book chapter. It is beyond the scope of this conference paper to enumerate the examples further. However, by way of summary, Table 2 proposes a set of generations of ICT based in part on the work done by these students.

When all students had finished successfully these first two essays they were able to understand easily that an information system may or may not use technology, and if it uses technology, the technology may or may not be a computer. Thus any 'brain damage' evident on arrival by the approximation set out above, was fixed and students were able to remove the implied 'approximately equal 'and treat the terms in that approximation equation as independent.

The third and fourth essays considering music and culture as information systems yielded very interesting assignments by the students. All students who attempted these completed them successfully. The results will be the subject of a separate paper. However, success in these illustrates a real need for a much broader concept of information system.

ICT Generation	**Example Artefacts (cumulative)**	**Approximate Time Period**
First	Paintings, Microliths, Megaliths, Symbols, Ornaments, Signals	Pre-History
Second	Tokens, Accounting tablets, Monumental inscriptions, Proto-paper	10,000 – 2,000 BP
Third	Writing implements, Printing, Paper, parchment, Semaphores	From 2,000BP
Fourth	Printing machines, Paper mills, Binding equipment, Flags, Instrumentation	From 15th century
Fifth	Calculating machines, Machine programming, Telegraph, Telephone, Wireless, Recording machines	From 19th century

Sixth	Tabulating machines, Television, Calculating machines, Computer hardware, Computer software, Computer networks, Multi-media machines, Robots	20^{th} century
Seventh	Massive technology convergence	21^{st} century

Table 2 - ICT Generations

All of this has profound implications for the scope of the subject Information Systems. Of course, the UK AIS definition of information system is perfectly adaptable to the situation 35,000 years ago. What remains problematic with the definition is the required link with technology, and of course any problems inherent in the term Information Systems itself.

5.0 Carnap and Fundamental IS/IT Concepts

For more than 50 years, there has been an evolution in approaches to systems analysis and in parallel there has been an evolution in the emergence of relational databases and object-oriented methods of system development. Examining the early stages of that evolution, the intellectual origin of all key terms used in the fields of requirements modelling, relational databases and object-oriented approaches to development is the philosopher Rudolf Carnap; these origins pre-date the computer era.

In order to make effective use of a digital computer there are two essential problems to be solved: (1) deciding what the computer should do; (2) creating a set of precise instructions to the computer to do what is desired (assuming the computer has been constructed correctly in the first place). The first problem usually arises before the second.

Many different approaches have been taken to those two essential problems, in parallel with the evolution of the computers themselves. So far, there are partial solutions to those two problems, but not full solutions. There is also the matter of a 'flow' from solving the first problem, to solving the second problem. Various attempts have also been made over the years to provide some automation assistance to that flow. This paper is focused on the first essential problem, so comments will be brief about the second. Also, as the concern of this paper is to broaden the concept of information system to include systems that may not use technology, or computers, the agenda for an innovated concept of information system still requires the creation of a human-made purposeful system that presents a solution to a problem, or a requirement.

In the 1950s and 1960s there were several published papers and articles about the term 'problem statements'. At one level, a natural language understanding of that term can be at a simple very general level such as 'the problem is to automate the system for shipping goods and invoicing customers, as far as is feasible'. That is undoubtedly a useful starting point, but it is not what key writers about problem statements had in mind; they envisaged a problem statement as a collection of inter-related, formally constructed, information, in some detail, setting out what the problem is. There should be enough information in the problem statement(s) for a system designer to start making progress with the design of a computer-assisted solution, or even look only at the computer-assistance and leave more general system issues to others.

Perhaps the earliest substantive paper referencing 'problem statement' is Young and Kent (1958) following the very strong hint at such a term in Canning (1956). They distinguish an abstract statement of a problem from a verbal statement.

> "An abstract statement of the problem can be made by preparing two lists: one of the information sets and the other of the documents... Thus the output documents are completely specified in terms of the transformations which are applied to the inputs. At the same time, the relationships among the information sets enable logical substitutions to be made in the input to achieve the same outputs." (p472).

They go on to provide an example verbal statement of a problem. It is unnecessary to quote that here, but suffice it to say that it is a set of precise, well constructed, natural language statements about the problem for which computer systems is required.

Young and Kent proceed to set out a formal notation to define symbolic representations of all items in the verbal description. Equations are presented to show relationships between inputs and outputs.

Thus, the approach proposed by Young and Kent involves problem statements which are detailed and precise articulations of the requirements of the proposed system. This is a far cry from a simplistic high-level statement of a problem to be solved.

Around 1961, there was work on an Information Algebra (IA) (Bosak et al., 1962). IA was not developed or used much. However, of particular interest in a discussion of correspondence between methods of system analysis and Carnap is the IA concept of a property space.

A property space is a set of all possible points that are a combination of possible values on a set of properties expressed as an n-dimensional space for n properties. McGill (see below) gives a simple example of a 3-dimensional property space: employee number, employee age, and employee sex. If there are 10,000 possible employee numbers, 45 possible ages of employees, and 2 possible sexes, the 3-dimensional space has 900,000 possible points in its property space. Is this a reflection of what is described by Carnap (1937) as an example of a coordinate-language that sets out coordinates whether or not objects exist at each coordinate?

Another influential paper from the same period is that of McGee (1963), continuing the theme of data processing problem formulation ("The principal function of data processing is to create output files from input files" - p42). He spends much of his paper discussing the emergence of a 'Data Processing Theory' and the Information Algebra that emerged from some CODASYL work. It is worth noting that McGee identifies some key high-level objects: entity, property, and measure (with associated datum, unit record, and file).

This earlier work about problem formulation and problem statements was followed in 1967 by the start of what is probably still the world's most extensive applied research project into requirements modelling and analysis. At Case Western Reserve University, there was a proposal for "A Research Project to Develop Methodology for the Automatic Design and Construction of Information Processing Systems" by Teichroew and Stieger (1967). This was the birth of the ISDOS (Information System Design and Optimization System) project. A key element in that proposal was for a Problem Statement Language and a Problem Statement Analyzer, which became the very widely known PSL/PSA. The ISDOS Project was born at Case Western Reserve University, but moved very quickly in 1968 to the University of Michigan (UoM), where almost all of its work was done.

The choice of the term 'Problem Statement Language' should be no surprise when looking at the background. Teichroew (and hence ISDOS) had been influenced strongly in his early years by Young and Kent (1958), and Canning (1956). He had worked with Young and Kent at the National Cash Register Company; he had worked with Canning at the University of California. McGee's work (1963) figured in the original ISDOS proposal. Teichroew already had students working in classes in 1966 on problems based around Young and Kent's work, prior to the 1967 ISDOS proposal.

A summary of "Problem Statement Languages in MIS" was produced by Teichroew (1971). That summary is a development of the material used for the ISDOS (information System Design and Optimization System) project proposal in 1967. The various problem statement languages were compared with respect to outputs and inputs, data definitions, processing requirements, other data, and whether or not the language had associated with it any other aids that help the design process. They are also compared with respect to life cycle phase coverage (needless to say, they are all focused at the early phases, some creeping into design, and only one going as far as construction).

So, where does Carnap fit into that picture of problem statements and formulation?

ISDOS started life as an industry sponsored university research project. The initial success of the project was based on a very careful attention to the details of a balance between academic rigor, extensive engagement with, and involvement of, industry and government, combined with the development of a computer infrastructure to deal with practical computer issues. Right from the beginning, the project was underpinned by multiple PhD projects thus providing important research underpinnings. Several PhDs were started at Case Western Reserve University (including McCluskey (1969), Nunamaker (1969) and Tremblay (1969)) and continued with Teichroew's supervision after the move of ISDOS to the University of Michigan (UoM). After the move to UoM there were further PhDs (e.g. Winters (1981), Yamamoto (1981), and many more). The whole series of PhDs relevant to ISDOS related issues ran from 1967 into the first decade of the 21st century.

The Carnap 'smoking gun' is explicit and detailed. In his PhD, Winters (1981) has a detailed discussion of Carnap's constructor principle. That PhD was done while working with Yamamoto (1981), one of the key architects of PSL/PSA, and supervised principally by Teichroew. Several key concepts from Carnap had a profound impact on the development of PSL/PSA.

All the discussions in the key approaches to problem statements can be reduced to a small set of distinct abstract objects and relations between them.

As far as the ISDOS project was concerned, problem statements and requirements are completely inter-linked:

> "The Problem Definer (PD) states his data processing requirements in a problem statement (PS) according to a Problem Statement Language (PSL). (The term "problem definer" is used here to mean the person or persons who define requirements.)" (Teichroew and Stieger, 1967).

In other words, requirements are expressed in terms of a set of problem statements. It is useful to set out several key developments around this time:

- 1967 - initial ISDOS proposal (Teichroew and Stieger, 1967)
- 1969 - Bachman (1969)
- 1971 - Teichroew and Sayani (1971), and, PSL v2.0
- 1976 - Chen (1976)
- 1977 - Teichroew et al. - most widely cited paper about PSL/PSA (1977)
- 1979 - Yourdon (1979)
- 1979 - Gane and Sarson (1979)

The very close links between deMarco, Yourdon, Gane and Sarson are well known. In terms of data flow diagrams, the differences are primarily notation. Yourdon provides another 'smoking gun' in terms of links to Carnap. Yourdon and Constantine (1979) are dependent on Martin and Estrin (1967) in their discussion of graphs. Martin and Estrin refer explicitly to Carnap. Therefore, Yourdon and Constantine are acknowledging, albeit indirectly, an influence by Carnap on their approach to graphs.

Bachman (1969) and Chen (1976) are focused on graphical methods of presenting a very small number of abstract objects and relations between those. These objects are about data.

Gane, Sarson and Yourdon are similarly focused on a very small number of abstract objects and relations between them. These objects are about processes, data flows, and data stores.

Teichroew and PSL/PSA use a wider set of abstract objects than the other writers, but it is a set of objects of which the others are clear subsets. PSL covers system aspects including: system dynamics; data structure, derivation and flows; requirements traceability, and quantification and resource usage. The significance to the discussion about Carnap below, is that all these approaches to requirements modelling can be reduced to a small set of distinct abstract objects and relations between them. Several of the authors mentioned directly or indirectly, occasionally address the matter of properties (of either objects or relations).

The approaches to modelling requirements mentioned so far (and the others included in the surveys mentioned, but not included here) differ in their methods of presentation: there are linguistic methods, graphical methods, and some intermediate methods such as tables.

The core concepts of relational databases are simple. They were set out in Codd's often cited paper (1970). That is a very muddled paper in its confusion between methods of storage in use at the time, and logical structures of information. In some respects, it also contains some serious errors. The translation of a relational model of data into a network (CODASYL) model, and vice versa, is an algorithmic problem and not conditioned by any decisions about actual storage. It would have been better had Codd called for separation between logical and physical approaches to storing data rather than confusing those in his discussion.

However, notwithstanding weaknesses and errors in that paper, the core concepts of relational databases are very simple. The basis is relational theory. There are additional terms introduced of tuple and domain. For 'domain' simply read property, or attribute. For tuple, simply read record, or instance.

There is little to add in respect of object-oriented approaches. They are focused very clearly on objects, relations between those objects, and de-composing objects into sets of components (with relations between those components).

Object-oriented approaches attempt to overcome the problems arising from a separation of data and process analysis techniques. PSL had never made that mistake, right from its first version in 1968.

It is interesting that Yourdon was extolling the virtues of structured analysis techniques as late as 1989 (Yourdon, 1989), but only one year later, he was putting his name to a tirade against structured analysis, and climbing on board a new emerging object-oriented fashion (Coad and Yourdon, 1990): "Repeatedly, in practice, separate notations and strategies for different process and data models have kept the two forever apart". Does this really mean that there had been no prior empirical work on the effectiveness of pursuing data and process modelling separately, or indeed into the effectiveness of the associated graphical notations? Coad and Yourdon attribute their movement into object-oriented analysis in terms of a very small number of anecdotes, not substantive research.

In 1928, Carnap had laid down, albeit in German, many of the fundamental building blocks used today for various purposes related to systems analysis.
Carnap's starting point is to the effect that there are certain fundamental concepts that underpin all science. His starting point is to use the term 'object':

> "The present investigations aim to establish a "constructional system", that is, an epistemic-logical system of objects or concepts. the word "object" is here always used in its widest sense, namely, for anything about which a statement can be made. Thus, among objects we count not only things, but also properties and classes, relations in extension and intension, states and events, what is actual as well as what is not...the present study is an attempt *to apply the theory of relations to the task of analyzing reality*" (Carnap, 1967:5-7).

If there is any doubt about Carnap's use of the terms mentioned above because his 1928 book did not appear in English until 1967, that doubt can be dispelled because he also discusses the terms object, property, relation, class, extension, intension, and relation (that may be one-one, one-many, many-one) in other works prior to 1967, in English (Carnap 1936, 1958). Carnap discusses relational theory (Carnap, 1936) and various kinds of relations (Carnap, 1958). Systems and database analysts would recognize many core concepts in those discussions, and indeed would probably find Carnap's treatment much more rigorous than normally found in today's practice (and literature).

Thus, this paper proffers the observation that Carnap articulated object-property-relationship (OPR) modelling, and that was followed by entity-relationship-attribute (ERA) where objects are called entities (Bachman (1969) does this explicitly). and properties are called attributes. Object oriented analysis can also be derived from Carnap who has thorough discussion of objects and classes, as can the underpinnings of relational database theory by his explicit discussion of relational theory and use of so many terms in use today for database analysis and design.

Carnap's meta-meta-modelling (he does not use that term *per* se) of the world commences with a very general concept of 'object' which is instantiated as a set of objects which can be used in combination to provide lower levels of model. Thus he talks about objects not only as things, but other objects as set out in the quotation above, including properties, things, relations, classes, and so forth.

A key point of this paper is that many of the terms used by systems analysts, relational database designers, and object-oriented developers are used by Carnap, and prior to the modern computer world. Hence, Carnap is proposed as the most likely origin for all these key concepts.

Table 2 sets out an illustrative list of concepts and where they can be found in Carnap. This list is not intended to be exhaustive or to present all occurrences in Carnap's works.

Concept	**Carnap Source**
Class	(1937) p134ff; (1958) p 77, 109; (1967) p57, pp68ff
Domain	(1958) p34; (1967) p 29, 59ff
Extension	(1937) pp240ff; (1958) pp40ff (1967) p56. 59
Intention	(1937) pp245ff; (1958) pp40ff; (1967) pp72ff, 261
Many-to-One	(1937) p222; (1958) p75;
Object	(1937) 13; (1967) pp31ff, 43ff
One-to-Many	(1958) p75
One-to-One	(1937) p223; (1958) p75;
Property	(1937) 13; (1958) p40, 109; (1967) p51
Property Space (as coordinate language?)	(1937) 141;
[Ir]Reflexive	(1937) p261; (1958) p120; (1967) pp166ff
Relation	(1937) p13; (1958) p40; (1967) pp21ff
Relational Theory	(1937) pp 260ff; (1967) pp59ff
Thing*	(1958) pp157ff; (1967) p32
[In]Transitive	(1937) p261; (1958) p119; (1967) pp166ff
Tuple	(1958) p75

*Carnap does not use the term 'entity'; he talks about 'thing' and a thing-language. The earliest formal and explicit definition of Entity as a word for Thing the author can find is in the definition of Information Algebra (Bosak et al., 1962:191)

Table 2 - Key Terms in Carnap's Works

The key significance of Carnap's work with respect to the various definitions of 'information system' used by the authors referred to (and others) is that they all come together when abstracted to a meta model level. Thus either PSL/PSA can be used to model information systems according to any of those definitions. If there are any weaknesses in using PSL/PSA, PSL could be modified using the meta meta language described in Teichroew et al. (1980).

6.0 Models, Metamodels, Metametamodels, and Metametametamodels

The concept of a metametamodel (written variously as 'meta meta model', 'meta-meta-model', 'meta metamodel', 'meta-metamodel', and with a variety of capitalization of the 'm's in whichever variation it is written; in this article, the variant 'metametamodel' is used) is normally associated with the concepts of metamodel, model, and subject (of the modelling). The terms 'metamodel' and 'metametamodel' are also closely associated with the terms 'metalanguage' and 'metametalanguage'.

The origin of the terms 'metamodel' and 'metametamodel' is not clear. Neither the Oxford English Dictionary (OED, 1991) nor Webster's International Dictionary (Gove, 1986) has an entry for either term. Therefore those terms must be considered recent. The OED does have meta-metalanguage, but Webster does not. The terms subject, model, metamodel and metametamodel are usually depicted as a hierarchy with metametamodel at the top and subject at the bottom, as illustrated in Figure 1 ('real world' corresponds to 'subject' or 'object' as used variously in the literature).

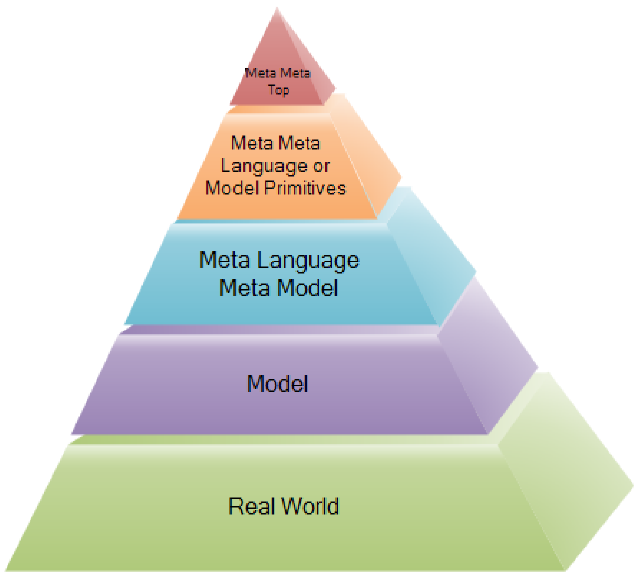

Figure 1. Levels of Language and Model used in IS/IT literature

The term metametamodel (expressed as meta-metamodel) appears as an essential part of the Unified Modelling Language (UML): [a metametamodel is] "A model that defines the language for expressing a metamodel. The relationship between a meta-metamodel and a metamodel is analogous to the relationship between a metamodel and a model. This level of indirection is usually relevant only to tool builders, database builders, and the like. UML is defined in terms of a meta-metamodel, called the Meta-Object Facility (MOF)" (Rumbaugh et al. 1999 p340)). The MOF has been incorporated into the international standard ISO/IEC 19502:2005 Information technology -- Meta Object Facility (MOF). MOF has 4 levels, M3, M2, M1, and M0 corresponding to metametamodel, metamodel, model and subject (some people use the term 'object' for this), respectively.

The idea of a metametamodel appears also in the effort to standardize the exchange of model data between different CASE (computer assisted software engineering) tools resulting in an EIA/CDIF (Electronic Industries Alliance/CASE Data Interchange Format) architecture framework "...which basically consisted of the meta-metamodel, which defines the concepts available for creating EIA/CDIF metamodels" (Flatscher, 2002). Various ISO standards have evolved from the EIA/CDIF framework (ibid).

The terms metametamodel and metamodel have had use in multiple independent and substantial projects (for example, Frank, 1998, Pasetti and Rohlik, 2005). The provenance of the terms is not clear, but around 1997. The explicit links between metamodel and metalanguage in all these works using the term metamodel makes it much easier to trace back to origins of the key underlying concepts. Earlier uses of the term 'metalanguage' are much clearer and two key English language dictionaries (OED and Webster) have one or more entries for that term. The OED explains metalanguage as "A language which supplies terms for the analysis of an ‘object’ language; a system of propositions about other propositions".

The works and projects referred to are all concerned with using metametamodels to derive metamodels which in turn are used to create models of target subjects (or objects), all from around 1997 onwards. However, these are not the first examples of projects to do that.

The ISDOS project has already been introduced. Teichroew did not use the term 'metametamodel'; he was primarily concerned about languages (even though 'model' is in his title). Much of the work done in the ISDOS project that was concerned with metametalanguages (or metametamodelling) remains unpublished (as at the beginning of 2012) notwithstanding research and development from the 1970s. There is a published paper that gives very strong hints about what was going on in this regard in the ISDOS project (Teichroew et al., 1980). That paper is concerned with languages, much more than models. Indeed, the ISDOS project produced an internal tool known as LDM (Language Definition Manager) referred to in the Teichroew et al. paper as an ISLDS (Information System Language Definition System) Architecture (ibid p19). What is not so well known is that LDM was used to define and generate the well known toolkit known as Problem Statement Language/Problem Statement Analyzer (PSL/PSA). LDM has been used subsequently as a metametamodelling language to create other languages that are very different from PSL (for example, languages for information architecture and competence) (Darnton, 2002).

As metamodel has evolved from metalanguage, it is much easier to identify the origins. The OED's discussion of metalanguage includes the observation " Hence ˈmeta-ˈmetaˌlanguage, a language used in the description of another language which is itself a meta-language; the universal linguistic or symbolic system from which a particular metalanguage derives", and proceeds with references to philosophers who have used the term metametalanguage. The OED entry for meta-metalanguage redirects to metalanguage.

There are hints in the literature that increasing levels of abstraction from object to metametamodel via model and metamodel could be continued ad infinitum. However, the previous quotation from Carnap shows that although his metametalanguage consists of object, property, and relation, these are all instantiations of 'object'. Hence levels of abstraction by this scheme would stop at metametametalanguage, which consists of 'object'. In the metametamodelling literature, that is also, effectively, the conclusion of Frank (1998:6): "Similarly to any other model, the definition of meta-metamodel requires a set of higher order concepts - we could also speak of a meta-meta-metalanguage. In order to avoid a regression ad infinitum, the concepts used for a language specification can be formalized at some point. That would require to define a set of symbols, a precise syntax and a calculus that would allow to generate all valid expressions (models)". This is what was implemented by the ISDOS Language Definition Manager. In any event, it is unlikely that such abstraction could continue ad infinitum.

7.0 Conclusions

So many considerations and definitions of 'IS' take an overly narrow view, concentrating more on computers specifically, or technology generally.

There is also a very good case for incorporating knowledge into the scope of IS, notwithstanding that 'knowledge management' seems to have emerged as a separate field of study. One of the 'grandfathers' of the study of information and knowledge had this to say: "I prefer that we get rid of the duplication "knowledge and information" when we refer to what people know or are informed about" (Machlup 1980: p8). In producing his classification system for information machines, Machlup certainly didn't distinguish between knowledge and information; indeed he made the same point about the futility of such a distinction (Machlup, 1962). One important difference between IS and KM is that the field of IS has been hijacked by the computer, whereas the situation concerning KM is not yet that serious. There are, or course, many tortuous efforts to distinguish between information and knowledge, but we are still waiting for any generally agreed operational distinction.

The discussion in this paper supports some modified definitions:

- 'IS (information systems) is the branch of systems thinking focused on the roles of information and knowledge in systems.
- ICT (information and communications technology) is the study or science of the use of artifacts to support information or knowledge work or play.
- ICA (information or communications artifact) is an object used to support information or knowledge work or play.

The discussion of the origins of key analysis techniques and database models shows that there is no problem in deriving languages than can support definitions of information systems 'in the large' and not only 'in the small' (i.e. focused on contemporary computer-based systems). Hence, the analysis infrastructure is already in place to support an innovated concept of information system. Metalanguages for information systems that go well beyond technology based systems have already been developed, demonstrating feasibility.

References

Aktas, A.Z. (1987) *Structured Analysis & Design of Information Systems*, Prentice-Hall, Englewood Cliffs.

Alter, S. (1999) *Information Systems: a management perspective (3rd ed.)* Addison-Wesley, Reading, MA.

Bachman, C. W. (1969) "Data structure diagrams." *SIGMIS Database*, 1(2), 4-10.

Benson, S., and Standing, C. (2005) *Information Systems: a business approach*, John Wiley & Sons, Milton, Qld.

Benyon-Davies, P. (2002) *Information Systems: an introduction to informatics in organisations*, Palgrave, Basingstoke.

Boddy, D., Boonstra, A., and Kennedy, G. (2002) *Managing Information Systems: an organisational perspective*, Prentice-Hall, Harlow.

Bosak, R., Clippinger, R.F., Dobbs, C., Goldfinger, R., Jasper, R.B., Keating, W., Kendrick, G., and Sammet, J.E. (1962) "An information algebra: phase 1 report-language structure group of the CODASYL development committee". Communications of the ACM. Vol 5. No. 4. April 1962.

Canning, R. G. (1956) *Electronic Data Processing for Business and Industry*, New York: John Wiley & Sons.

Carnap, R. (1928) *Der Logische Aufbau der Welt*, Berlin-Schlachtensee: Weltkreis Verlag.

Carnap, R. (1937) *The Logical Syntax of Language*, London: Kegan Paul, Trench, Trubner & Co. Ltd.

Carnap, R. (1958) *Introduction to Symbolic Logic and its Applications*, (1st English ed.) New York: Dover Publications, Inc.

Carnap, R.(1967) *The Logical Structure of the World and Pseudoproblems in Philosophy*, (2nd (1st in German 1928)) London: Routledge & Kegan Paul.

Chaffey, D., and Wood, S. (2005) *Business Information Management: improving performance using information systems*, Prentice-Hall, Herlow.

Checkland, P. and Holwell, S. (1998) *Information, Systems and Information Systems - making sense of the field*, John Wiley & Sons, Chichester.

Chen, P. P.-S. (1976) "The Entity-Relationship Model - Toward a Unified View of Data." ACM Transactions on Database Systems, 1(1), pp9-36.

Coad, P., and Yourdon, E. (1990) *Object-Oriented Analysis*. Englewood Cliffs, NJ: Yourdon Press. 1990.

Codd, E.F. (1970) A Relational Model of Data for Large Shared Data Banks. Communications of the ACM. Vol. 13 No. 6 June 1970.

Curtis, G., and Cobham, D. (2005) *Business Information Systems: analysis, design and practice*, Prentice-Hall, Harlow.

Darnton, G. (2002) "Modelling Requirements and Architecting Large-Scale On-Line Competence-Based Learning Systems" in Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT 2002), Kazan, Russia. pp170-174. Available at: http://lttf.ieee.org/icalt2002/proceedings/t411_icalt119_End.pdf.

Darnton, G., and Giacoletto, S. (1992) *Information in the Enterprise: it's more than technology*, Digital Press, Burlington, MA
Flatscher, Rony G., (2002), "Metamodeling in EIA/CDIF---meta-metamodel and metamodels" in ACM Transactions on Modeling and Computer Simulation Volume 12 Issue 4, October 2002 pp 322-342.
Fortune, J., and Peters, G. (2005) *Information Systems: achieving success by avoiding failure*, John Wiley & Sons, Chichester.Jessup, L., and Valacich, J., (2006) *Information Systems Today: why IS matters,* Pearson, Upper Saddle River, NJ.
Frank, Ulrich (1998), The Memo Meta-Metamodel. Koblenz: University of Koblenz-Landau. Available at: http://www.wi-inf.uni-duisburg-essen.de/FGFrank/documents/Arbeitsberichte_Koblenz/Nr9.pdf
Gane, C., and Sarson, T. Structured Systems Analysis: Tools and Techniques. Englewood Cliffs, NJ: Prentice-Hall. 1979.
Gove, P.B. (1986) Webster's Third New International Dictionary, Springfield MA: Merriam-Webster Inc.
Gray, P. (2006) *Manager's Guide to Making Decisions about Information Systems,* John Wiley & Sons, Chichester.
Kroenke, D.M. (2007) *Using MIS*, Pearson, Upper Saddle River, NJ.
Laudon, K.C., and Laudon, J.P. (2002) *Management Information Systems: managing the digital firm (7th ed.),* Prentice-Hall, Upper Saddle River, NJ.
Machlup, F. (1962) *The Production and Distribution of Knowledge in the United States*, Princeton University, Princeton, NJ.
Machlup, F. (1980) *Knowledge: It's Creation, Distribution, and Economic Significance Volume 1 Knowledge and Knowledge Production,* Princeton University, Princeton, NJ.
Martin, D., and Estrin, G. "Models of Computations and Systems-Evaluation of Vertex Probabilities in Graph Models of Computations". Journal of the ACM Vol. 14. No. 2. April 1967.
McCluskey, W. A. Toward the Automatic Design of Data Organization for Large-Scale Information Processing Systems, PhD Thesis, Cleveland, OH: Case Western Reserve University. 1969.
McGee, W. C. "The Formulation of Data Processing Problems for Computers", in F. L. Alt and M. Rubinoff, (eds.), Advances in Computers Vol IV. Academic Press, pp. 1-52. 1963.
Nunamaker, J. F. On the Design and Optimization of Information Processing Systems, PhD Thesis, Cleveland, OH: Case Western Reserve University. 1969.
OED (1991) The Compact Oxford English Dictionary (2nd ed.), Oxford: Clarendon Press.
Pasetti, A., and Rohlik, O. (2005) Technical Note on a Concept for the Xfeature Tool (Issue 1.3), Taegerwilen-Thurgau: P&P Software GmbH
Rumbaugh, J., Jacobson, I., and Booch, G. The Unified Modeling Language Reference Manual. Reading MA: Addison-Wesley 1999.
Teichroew, D. "Problem Statement Languages in MIS" in Grochla, E. (ed.) Management-Informations-systeme. Wiesbaden: Betriebswirtschaftlicher Verlag Dr. Th. Gabler. 1971.
Teichroew, D., and Hershey, E. A. "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems." IEEE Transactions on Software Engineering, 3(1), pp 6-15. 1977.
Teichroew, D., and Sayani, H. "Automation of System Building" Datamation. pp 25-20. 1971.

Teichroew, D., and Stieger, W. H. A Research Project to Develop Methodology for the Automatic Design and Construction of Information Processing Systems. ISDOS Working Paper, WP001, Cleveland, OH: ISDOS. 1967.

Teichroew , D., Macasovic, P., Hershey, E.A. III, Yamamoto, Y. (1980) "Application of the Entity-Relationship approach to Information Processing System Modeling" in Chen, P.P. (ed.) Entity-Relationship Approach to Systems Analysis and Design, Amsterdam: North-Holland. pp 15-38.

Tremblay, E. J. P. A Problem Statement Language Definition and its Syntactic Analysis, PhD Thesis. Cleveland, OH: Case Western Reserve University. 1969.

UKAIS (2012) [definition of information systems] web page at: http://www.ukais.org.uk/about/DefinitionIS.aspx.

Vincent, D.R. (1990) *The Information-Based Corporation: stakeholder economics and the tecjhnology investment,* Dow Jones-Irwin Homewood, Il.

Winters, E. W. On the Analysis and Synthesis of Human and Computer Subsystems for Information Processing Systems, PhD Thesis, Cincinnati, OH: Union of Experimental Colleges and Universities,. 1981.

Yamamoto, Y. An Approach to the Generation of Software Life Cycle Support Systems, PhD Thesis, Ann Arbor: .University of Michigan.1981.

Young, J. W., and Kent, H. K. "Abstract Formulation of Data Processing Problems." Journal of Industrial Engineering, IX(6), pp 471-479. 1958.

Yourdon, E. Modern Structured Analysis. Englewood Cliffs, NJ: Yourdon Press. 1989.

Yourdon, E., and Constantine, L. Structured Design. Englewood Cliffs, NJ: Prentice-Hall. 1979.