Conceptual modelling of Work Systems using ABC notation

Mark Roger Gregory
Associate Lecturer, University of Hull, mark@teamkim.org

Conceptual modelling of Work Systems using ABC notation

Mark Gregory
Associate Lecturer, University of Hull, Cottingham Rd, Hull HU6 7RX, United Kingdom
Email: mark@teamkim.org

Abstract

This paper seeks to extend the Work System Method WSM of Steven Alter by means of a modelling notation and mechanism called ABC. This suggestion is made because it is often necessary to know (or at least to conjecture) how things, processes and events within a work system interrelate. Our contention, which we support by literature primarily derived from cybernetics, is that we must discern conceptual models, so as to understand and, potentially, to improve by design, active models – specifically, the information systems which support work systems. We do this in order to regulate work systems, whether actively by explicit control or implicitly by aiding learning, understanding and self-control by modellers and participants. This paper is not a definitive statement concerning ABC. Instead, it sufficiently introduces the modelling approach to enable the reader to understand some examples of the approach as applied to work systems. It can also serve in a tutorial approach to the ABC modelling of work systems.

Abstract: 160 words; paper : 10333 words excluding references

Keywords: work systems, visual knowledge modelling, model-based reasoning, conceptual modelling, concept maps, business process modelling

1. Introduction: how might we better model Work Systems?

The principal purpose of this paper is to argue that the work system method WSM of Steven Alter correctly identifies the primacy of the Work System notion as the proper focus for a systematic study of information-focused action, but that WSM often needs to be augmented by appropriate visual knowledge modelling or conceptual modelling. This is because knowing what is important and why in any given problem scenario is primordial, but it is necessary also to know (or at least to conjecture) ***how things, processes and events inter-relate***.

The paper makes use of a knowledge modelling language and approach, referred to here as ABC, which is very deliberately not the main subject of the paper and cannot yet be properly referenced for reasons connected to blind reviewing of other planned articles.

Instead, the focus is on the importance of the conceptual modelling of work systems in achieving individual and collective understanding of value both to modellers and model users. The paper also illustrates the usefulness of ABC in conceptual knowledge modelling and how model-based reasoning has the potential to improve our understanding of what is and of what might be.

1.1. Why this paper has been written, what it seeks to achieve and how it is structured

This paper follows a problem-solving methodology in which the problem or research gap to be addressed is how better to model Steven Alter's ***work systems*** (Alter 2011) and the principal mechanism is a tutorial on a novel approach to the ***conceptual modelling*** of work systems.

The paper firstly set out its understanding of what a modellable system is. It accepts that complex systems cannot be conceptually modelled. However, we note here that most engineered systems – (Simon [1970] 1996)'s artificial systems – are based on acts of design and that such systems can usefully be conceptually modelled. It extends the scope of conceptual modelling beyond the understanding of information systems requirements into the more general area of knowledge modelling.

Based on existing literature, it distinguishes work systems from information systems. The paper embraces the contention made by (Alter 2015) that the IS field should move away from arguments over definitions of obscure notions in favour of a concentration on what work systems are and how they are used:

> "[To] focus more directly on achieving the IS discipline's espoused goals of rigour, relevance and influence in the real world." (Alter 2015, p.2)

The paper uses a somewhat novel conceptual knowledge modelling approach called ABC to suggest a way in which work systems can be modelled, adding more ***how?*** to the ***what and why?*** previously addressed by Alter.

1.2. Argument and Methodology

The paper argues that a focus on work systems is pragmatically extremely useful, but that we need also to recall the IS field's origins in the practice of analysis and modelling of requirements for systems and in systems design. A practical skill that students of information systems require is that of appropriate conceptual modelling, specifically of existing or planned work systems. ***The method adopted in this paper is to introduce a somewhat novel approach to the modelling of work systems and then to use tutorial examples that should help readers to learn the approach and apply it.***

2. Background and insights from literature

What is a system? Figure 1 presents a basic view.

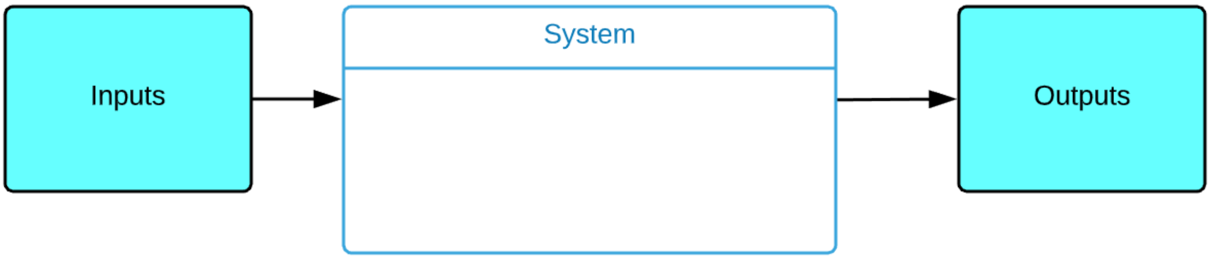


Figure 1 A basic view of a system

But what is the purpose of the system? Every system has a purpose; each is different. An example of a system is a university-level school.

What then is the purpose of a school? Figure 2 is a grey-box or outline model, which shows the externally visible characteristics of a school without showing any detail as to how it functions.

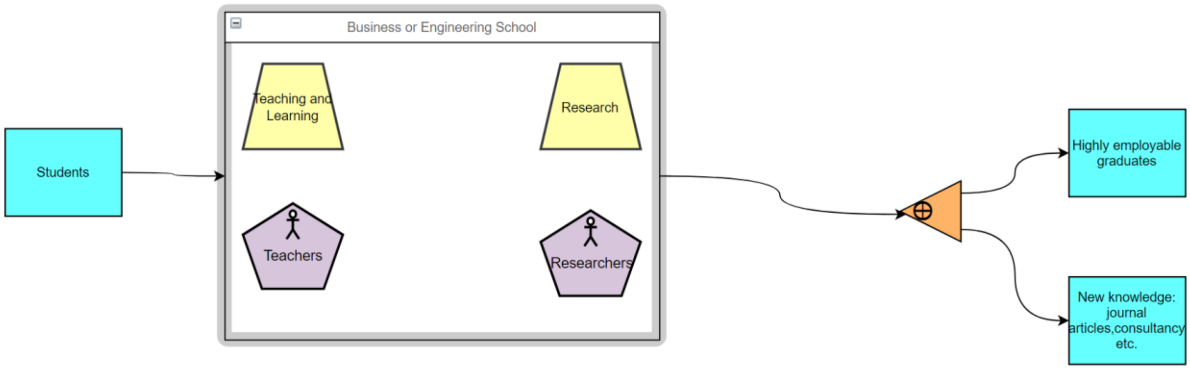


Figure 2 Grey box model of a school

2.1. Systems that can be modelled and those that cannot

(Snowden and Boone 2007) distinguish complex adaptive systems from systems that can be straightforwardly managed. As shown in Figure 3, they differentiate between obvious (or simple), complicated, complex and chaotic situations. They hold that “simple and complicated contexts assume an ordered universe, where cause and effect relationships are perceptible, and right answers can be determined based on the facts”. On that basis, we suggest also in Figure 3 that there will therefore be some systems that can readily be modelled conceptually, and some that will not. We suggest that the distinction will similarly apply to systems which are the result of deliberate design on the one hand and those that are only discernible or emergent on the other.[1]

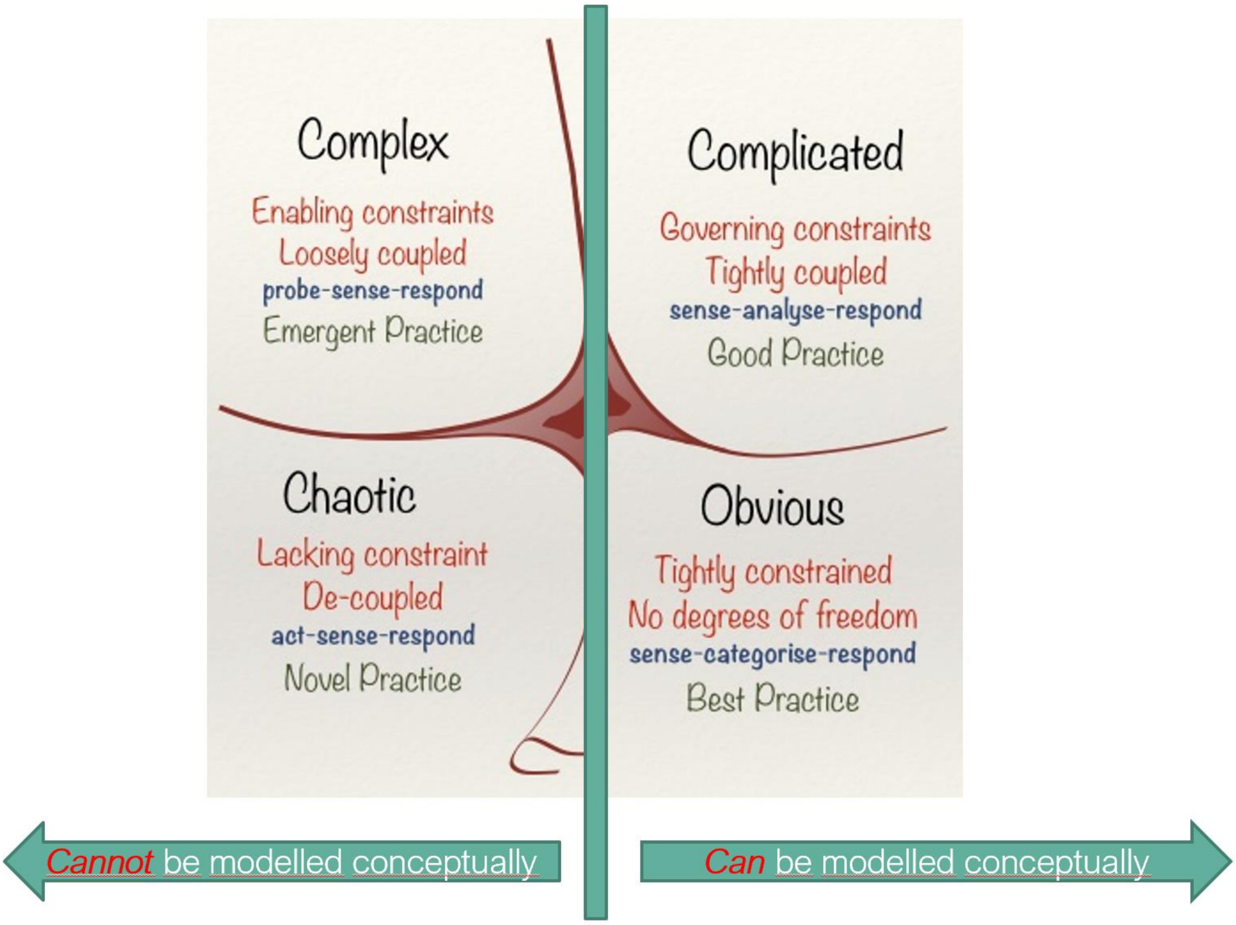


Figure 3 Distinguishing between systems that can and cannot be modelled conceptually
Based on (Snowden and Boone 2007).

[1] This distinction between simple, complicated, complex and chaotic domains is criticised by (Boulton, Allen, and Bowman 2015, p.192). In particular, they note that a so-called simple intervention may have ***outputs*** which are easy to define and to measure – but this is not to say that their ***impact*** is simple.

2.2. Work Systems, Information Systems, and E-Business

From (Alter 2002a), we find that E-business is the practice of performing & coordinating business processes through the extensive use of ***information technology (IT)*** (information and communications technology ICT in UK usage).

Steven Alter's Work Systems Method (WSM) assumes that the topic of analysis is a ***Work System***, a system in which people and/or machines perform a ***business process*** using resources (e.g., information, technology, raw materials) to create products/services for ***internal or external customers*** (Alter 2002b).

Methods and tools that emphasize business viewpoints and issues should view such a system as a sociotechnical system and should focus on how to improve that system's performance and sustainability.

We follow Alter's contentions that:

- Current work systems are ICT-reliant. They rely on IT but are not IT systems.
- A work system's goal is to provide value for its customers, not just to operate consistently with its own specifications.
- Requirements are assumed to evolve over time and thus work systems must also evolve.

2.3. Work Systems and Information Systems

The concept of Work System can be used to describe situations ranging from the work of filling out simple computerised forms through to the complex work of producing an aircraft.

In Steve Alter's Work Systems Framework WSF (Alter 2011), information systems are presented as special cases of work systems in which all of the processes and activities are devoted to processing information.

2.4. Using information to inform action

Figure 4 is a model, a knowledge map (Basque et al. 2008; Okada, Buckingham Shum, and Sherborne [2008] 2014) made in a visual knowledge modelling language called ABC, of the actors, concepts, processes, events and decisions which are involved when deciding whether a small item requisition (for example, a pencil sharpener) can succeed and how.

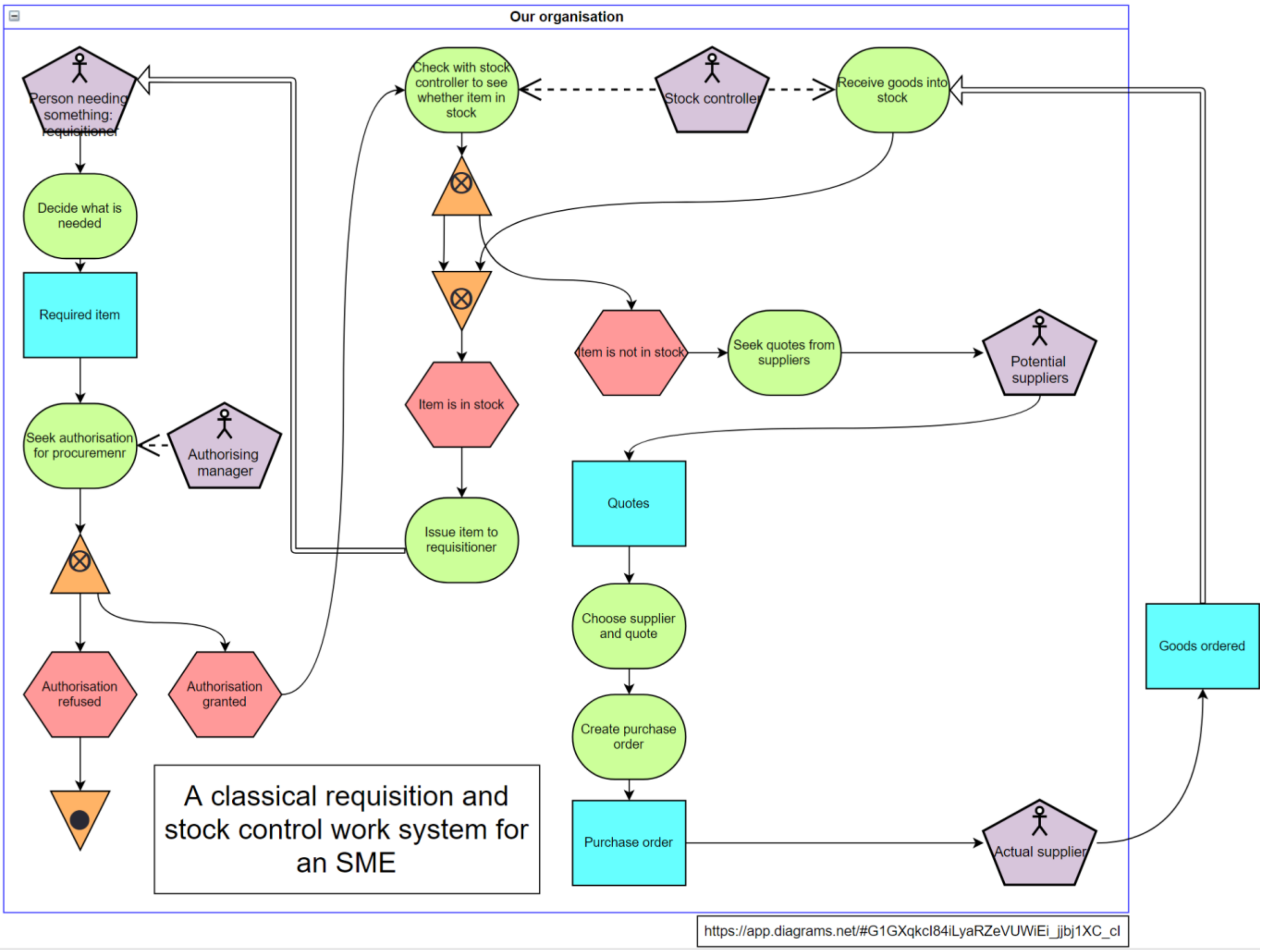


Figure 4 A classical purchase requisition and stock control work system.

Source: author

Note that this particular model of a work system includes one physical flow and has no modelled dependence upon ICT. Such dependence, although very common, is not essential to the nature of a work system. We will gradually introduce the notation used in this diagram, which for now we note distinguishes actors, procedures, concepts, events and operators.

2.5.Why make models of systems?

In this paper, we will frequently show representational models, in the form of knowledge maps. Figure 4 is an example of such a model.

Why model?

We model:

- To understand – representation (e.g. rich picture, presentation diagram, knowledge map…). The process of modelling helps the modeller(s) of a systemic situation to understand that situation better.

- To communicate. The models so created may also help participants in a modelled system towards better understanding of that situation and of their roles within it. Thus models are also used for communication purposes; this may be their main function, especially in communication within an organisation and for checking conformance to process standards.
- To control – regulation. This may be by means of an active regulator, as in the case of an aircraft flight control system. But in addition, a human actor may "obey" or learn from information presented by an apparently passive model.

2.6.Multiple models, multiple notions and multiple relationships

We observe that characterising a complex notion, such as an instance of an information system, cannot be reduced to a single model type because each model, which is conceptual, is a partial response to a specific topic question asked by a specific class of actor. For example, what an engineer would characterise as an information system will not be the same thing as the rich but often rather nebulous concept discussed in the academic IS community.

The engineer's conception is perhaps best described as an assemblage of ICT components. Figure 5 suggests that a dynamic website might enable users who are sources of data to store that data in a relational database so that users – in some cases the same people – can ask for information, the outcome of displaying SQL query results.

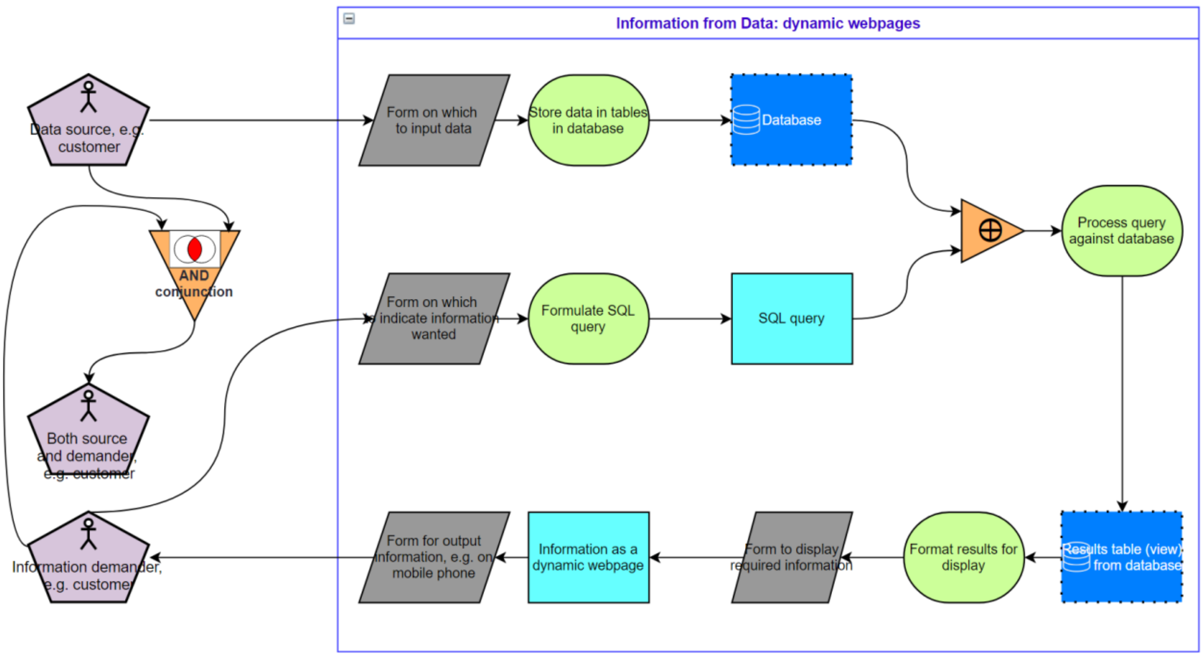


Figure 5 An engineering conception of an information system

https://app.diagrams.net/#G1GgF3Hq5zipnqoI1Kuku_3aLnYSINxNsO

The conception in the information system IS field (Hassan 2011) will look rather for evidence of systemicity (Bunge 2000), and in particular of emergence (Bunge 2003; Checkland 2012), in a broader sociotechnical view of what an information system is.

A model may represent and ideally encapsulate a conceptualisation of a system which is often context-dependent and essentially socio-technical. It is however possible to attempt the definition and modelling of the separate but linked notions of work system and information system for the IS community with which the author identifies.

Specifically, this paper reflects on what a work system is, what an information system is, and the various relationships that exist between work systems and information systems.

2.7. Why model? A restated justification for an insistence on modelling

Our only understanding of reality is models: initially, mental models (Greca and Moreira 2000).

Once we have a model, we can:

- Reason
- Act
- Improve the model – model-based reasoning and model-based action

Information, we might hope, ***informs action***.

Modelling cannot be just an optional extra in situations where regulation is required – it is essential even when not explicit. This is a consequence of the Good Regulator Theorem GRT of (Conant and Ashby 1970), which is discussed in section 2.9 below.

Modelling is highly desirable in any situation in which analytical understanding is sought, for example of work systems

2.8. A suggested new approach to modelling work systems: ABC

ABC, which is introduced, described and exemplified in this paper, is a language and method for explicating and modelling aspects of explicit knowledge in a visual form. ABC itself is not the subject of this paper; ABC will be further described, positioned and justified in subsequent papers. The advantages initially claimed for the ABC approach are summarised towards the end of this paper, in section 8.3.

2.9.Models: some theory derived from cybernetics

Ashby's law of requisite variety - (Ashby 1956) – states that:

> "Variety absorbs variety, defines the minimum number of states necessary for a controller to control a system of a given number of states".

Restated, it holds that the variety in a controller must exceed the variety in the situation to be controlled if effective control is to be achieved.

Conant and Ashby's good regulator theorem - (Conant and Ashby 1970) states that:

> "Every good regulator of a system must be a model of that system...
>
> "The design of a complex regulator includes the making or maintenance of a model of the system to be regulated.
>
> "The theorem shows that any regulator that is maximally both successful and simple must be isomorphic with the system being regulated."

2.10. What does the Law of Requisite Variety teach us?

We need to manage complexity. Ashby suggests variety: the number of possible states of the system, as a measure of complexity.

To control a situation, it is necessary:

- EITHER to amplify signals that measure and control a situation
- OR to attenuate the variety arising from the controlled situation

These are essential mechanisms in rendering a system controllable.

2.11. Amplifiers and filters

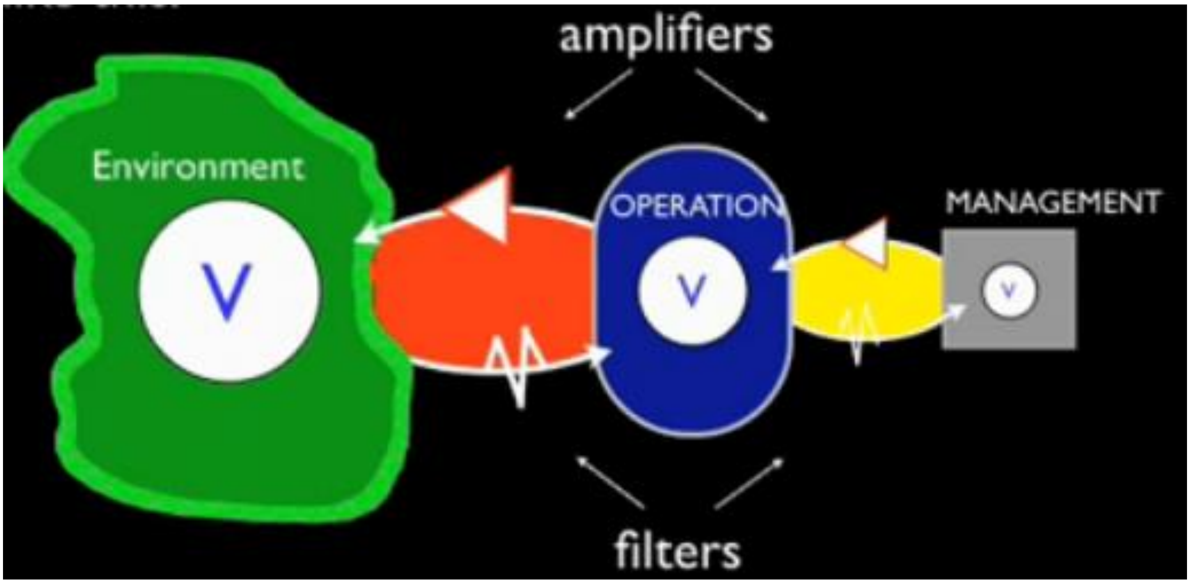


Figure 6 Variety mismatches: cf. (Espejo and Reyes 2011)

In general, the environment of an organisation is complex and hostile. For example, in the operation of a main business process, such as communicating with the public – advertising, public relations, customer service, webmaster: the organisation will not fully succeed in controlling the messages received about the organisation by its publics. However, by the use of ***filters and amplifiers in the form of information systems,*** the mismatch between external variety and requisite, useful, internal variety, can at least be reduced. Cf. Figure 6 and (Espejo and Reyes 2011).

2.12. Some practical implications of cybernetic principles

A management aim is to design and facilitate a process such that it can ***regulate itself*** – notably this implies empowering the people who work within the process and ***not*** reducing good variety by inappropriate attempts at excessive management control. Some models actively control themselves. Some models passively represent a situation; these models may nevertheless effectively become active as they influence the behaviour of active agents.

- Good information systems will attenuate inappropriate variety and amplify effective control.
- They should assist the manager in applying her knowledge to amplify effective control by giving appropriate information to the process being controlled and more generally by informing and thus empowering users.

2.13. Feedback and feedforward

In a feedback control system, a comparison is made between the actual output of a system and the desired or reference behaviour. The difference between the two is fed back as an error control signal to some system element which will change the input, an actuator. The effect which is sought is that the output more closely tracks the desired behaviour. This feedback is typically negative, acting against the tendency of the system to move outside its desired performance envelope.

In a feedforward control system, an attempt is made to forecast the future behaviour of a system and on the basis of this anticipated behaviour, again to derive an error control signal or other change in system parameters.

In both feedback and feedforward systems, lags in the detection or enactment of the error signal may cause system instability in the form of a tendency towards ringing or oscillation. An example of such instability is acoustic feedback in an audio system.

Early cyberneticians worked to understand negative and positive feedback and stability in closed and open systems. Analytical modelling of system stability is possible in certain forms of engineered system, but not in sociotechnical systems.

2.14. Good Regulator theorem

The good regulator theorem (Conant and Ashby 1970; see section 2.9 above) mandates the modelling of a system to be controlled within the controller. Conant and Ashby held that the developing knowledge of regulation, information processing and control is building criteria for measuring the effectiveness of the brain, and specifically, the extent to which our mental model of a situation is isomorphic with its reality.

2.15. Model based reasoning is a necessity

Modelling is NOT optional – and we do it all the time. It is essential to learning in the individual. It is also essential to the early stages of the creation of new knowledge. Deduction and induction confirm knowledge, but they cannot create it – that is the province of risky, but potentially productive, abduction.

Model based reasoning is intricately linked to abduction and uses abductive thinking:

(Gabbay and Woods 2005). It is discussed by (Nersessian 1999; Magnani, Nersessian, and

Thagard [1999] 2012). Nersessian argues that specific modelling practices employed by scientists are productive methods of conceptual change in science. To embrace modelling practices as methods of conceptual change is to challenge the traditional dominance of deductive and inductive logics of discovery. It also risks introducing unsoundness into the conclusions drawn. "Some non-traditional philosophical accounts have allowed for the possibility of "abductive" inference, but these accounts leave mysterious the nature of the reasoning processes underlying abductive inference and hypothesis generation. Analyzing modeling practices provides a way of specifying the nature of some abductive reasoning processes." (Nersessian 1999, 8).

Since Nersessian wrote, there has been a great deal more attention paid to the role of abduction in hypothesis generation, e.g. in (Psillos 2009)' account of Peircean abduction and in the work of Lorenzo Magnani (e.g. Magnani 2009). Here, I contend that ***the visual modelling of a putative or improved work system constitutes an extended abductive hypothesis on what that system should or does constitute.*** It is inherently risky in that the system so postulated and then implemented (with or without ICT support) will almost certainly evidence both desirable emergences and undesirable disturbances to the environment with which it interacts. But some design process must be followed and some model be constituted, if a controller is to be devised to regulate a system under design. The form of that regulator may initially only be a conception, as exemplified by Steve Whitla at http://meaning.guide/index.php/2017/12/04/reason-work-can-seem-meaningless/ accessed 08/02/2021 reproduced as Figure 7:

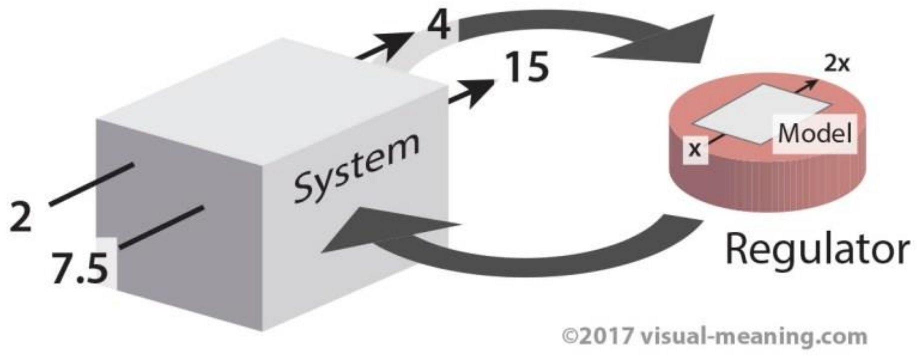


Figure 7 Model as part of regulator

2.16. Further implications of cybernetic thinking

A business is a system. Its sub-systems are ***work systems*** (Alter 2011) – each having clients (external customers, internal clients).

The work systems are nearly always served by ***business information systems*** which are themselves models (practical, working ones, ideally isomorphic) of the work systems they serve.

Before we can build or buy a business information "system" or software, we need a good understanding of what the work systems are and of how they might be developed. The starting point may well be homomorphic mental models. We may then build upon these using ***appropriate*** diagrammatic models where that is necessary. These might be used, for example, to validate the applicability of a putative software design or of functionality offered by an existing package.

2.17. Why may we need several different models?

So if modelling is essential, in what ways can we make it more explicit and at the same time more approachable for students and professionals? ***Simplicity*** is pragmatically essential because business students and business professionals rightly resist unnecessary analytical complexity.

How can we ease the modelling task? Perhaps, following (Dijkstra 1982), by a ***separation of concerns:*** using different modelling techniques within the context of the business-oriented work systems framework. But we should arguably eschew the complexity and conflicting overlapping techniques of, for example, UML (Booch, Rumbaugh, and Jacobson 2005). UML's primary emphasis is in any event on design rather than analysis.

Who should make the models? Accuracy in the sense of fidelity of model to situation modelled is implicitly increased by the increased possibility for partnership and communication that arises when models are ***shared and jointly developed***.

2.18. Models as active regulators

See Figure 8, which illustrates an active regulator, the Boulton and Watt centrifugal governor on a steam engine.

Figure 8 A model as an active regulator

Source: https://commons.wikimedia.org/wiki/index.%20php?curid=9964214

We contend that the output of information systems, which at first glance may constitute passive regulators, such as lists of things to do, ***becomes active because of the agency of the user***. See Figure 9 for an illustration of part of a paper-based information system. Not shown is ***the person who maintains the list and is influenced in their actions by the list.***

Figure 9 Model as passive regulator: To Do list.

Source: *https://commons.wikimedia.org/wiki/File:To_Do_List_Scene_Vector.svg*

3. Modelling Work Systems

3.1.Models as representations

Figure 10 shows the firm as a system (Alter 2002a); this is suggested as an essential starting point in beginning to identify Work Systems.

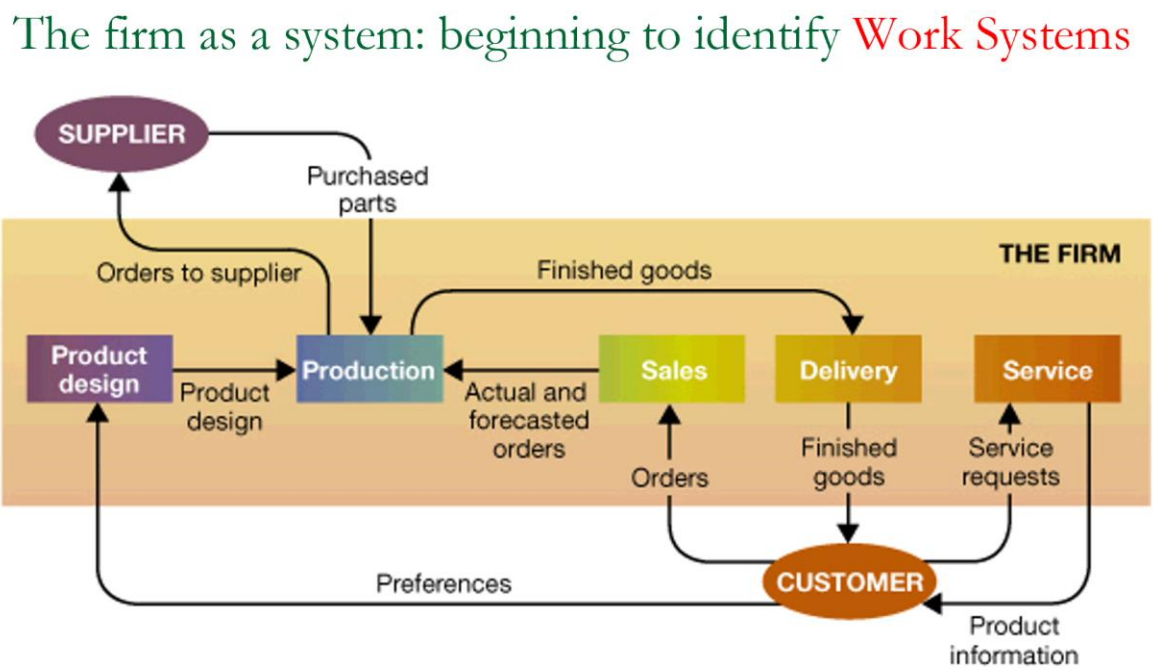


Figure 10 The firm as a system

Source: (Alter 2002a)

Steven Alter’s Work System Framework (WSF) (Alter 2002a) is shown diagrammatically in Figure 11.

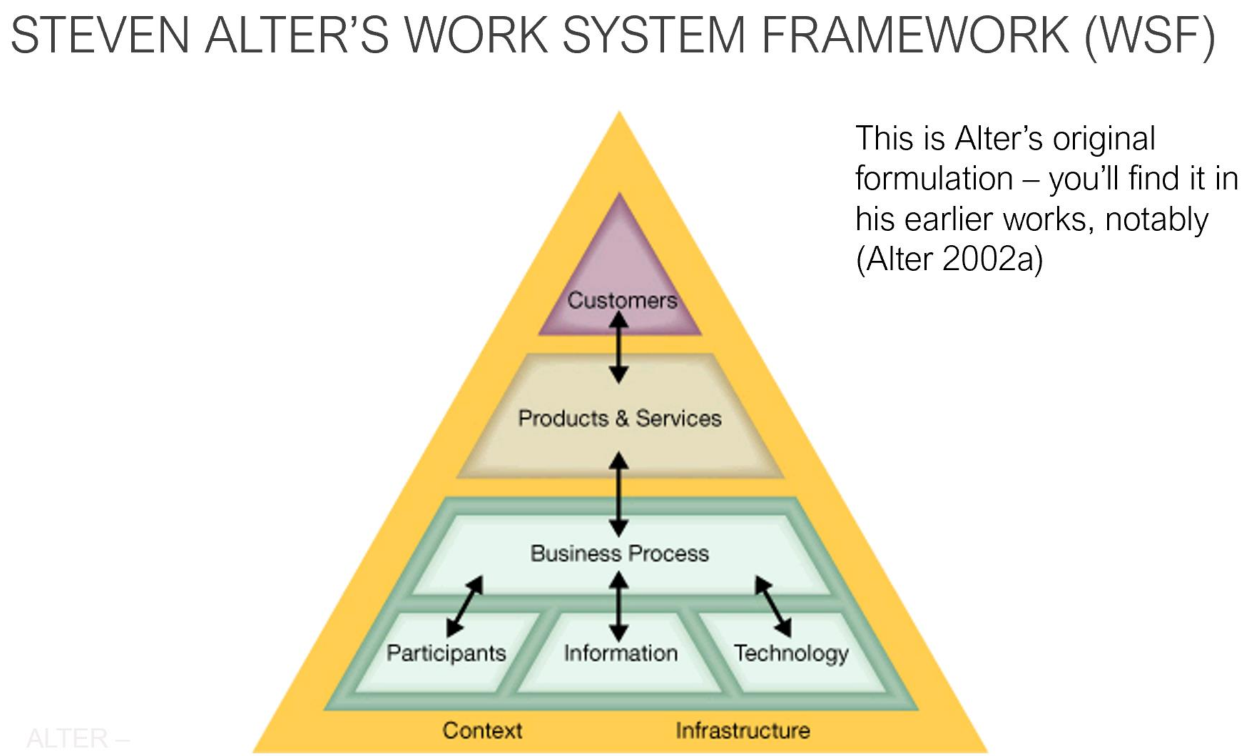


Figure 11 Early formulation of the Work System Framework

3.2. What is a Work System?

Recapping, we see that the Work Systems Method (WSM) (Alter 2006) assumes that the topic of analysis is a work system, a “system in which human participants and/or machines perform processes and activities using information and technology to produce products and services for internal and/or external customers”.

Methods and tools that emphasise business viewpoints and issues should view such a system as a sociotechnical system and should focus on how to improve that overall system’s performance and to minimise undesirable disturbances.

The concept of work system (Alter 2011) can be used to describe situations ranging from the work of filling out simple computerised forms through to the complex work of producing an aircraft.

Important particular cases of work systems include supply chains, self-service ecommerce systems, and projects.

Very many current work systems are ICT-reliant. They rely on ICT but are not ICT systems. A work system's goal is to provide value for its customers, not just to operate consistent with its own specifications. Requirements are assumed to evolve over time.

3.3. Work Systems and Information Systems

Information systems are, *inter alia*, special cases of work systems in which all of the processes and activities are devoted to processing information.

3.4. How to make practical use of the Work Systems Framework

Alter suggests that as a minimum step in the identification of work systems, it is necessary to create a WSF Snapshot to identify the scope of a work system. In any company, there will be a significant number of major work systems. One single information system rarely (never?) meets the needs of all the work systems. Therefore we should ***set and define the scope of a work system and perhaps of any supporting business information system***.

An example Work Systems Framework Snapshot appears as Figure 12.

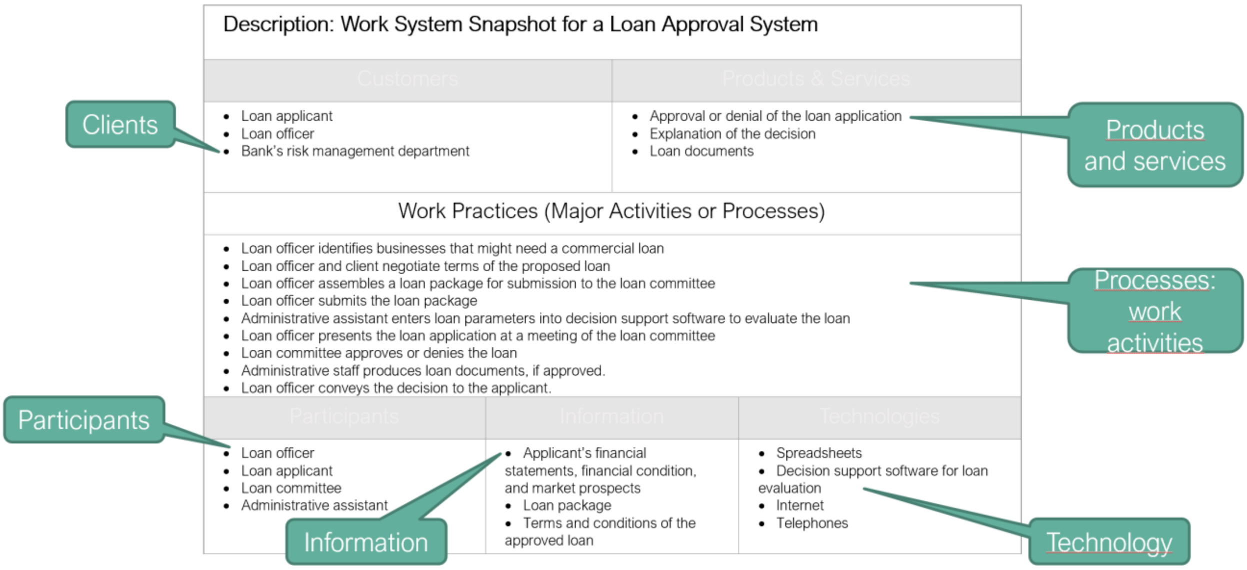


Figure 12 Example WSF snapshot with annotation

Source: (Alter 2006) with author's annotation.

3.5. Models, modelling and analysis for business students

We must discern conceptual models (Gemino and Wand 2004), so as to understand and, potentially, to improve by design, active models, in order to regulate, whether actively by explicit control or implicitly by aiding learning, understanding and self-control.

3.6.What no longer needs to be modelled in 2021?

Two trends have arguably eliminated the need to teach conventional, structured, systems analysis techniques, following for example (Yourdon and Constantine 1976).

One is the componentisation and packaging of information systems functionality. This suggests a Space Lego analogy, in which we raise the level of the bricks we select from the toy box and no longer have to analyse and design each brick.

The other is agile development (e.g. Abrahamsson, Conboy, and Wang 2009) which appears to eliminate the need for explicit modelling.

But the GRT shows that the need for models CANNOT be eliminated – without models we cannot regulate systems. If agile development has succeeded, it must therefore be because the model exists in the head of the developer(s), and is then faithfully translated into a working ICT-based system (with a concurrent revision of the mental model).

Instead, this paper now suggests that what is needful is to continue to emphasise the significance of modelling while ***raising the level at which we model***.

Specifically, we must learn how to model in business terms, using typed notions.

3.7.Extending the Work Systems approach

Alter has argued persuasively (e.g. Alter 2015) that long debates about the nature of the « IT artefact » demonstrate their own futility, at least in part because they seek a theoretical basis for what should be pragmatic efforts to understand systems in terms of what they do and of their demonstrable utility in the enterprise or the organisation.

However, in his published work Alter appears resistant to complex diagrammatic representations of work or information systems. Instead, he distinguishes between a static WSF, a dynamic Work Systems Life Cycle WSLC and a WSM Work System Method, seeking throughout to keep the level of discourse at that of business.

This is admirable up to a point. But as Alter himself notes, many work systems are ICT-dependent.

1. Therefore, at some point, it is necessary to have a clear statement of the actual functionality required of an ICT-based system.
2. Equally, it may be necessary to represent the functionality of a completed bespoke or package-based ICT-based system in user domain terms.

So, we can use the Work Systems Framework WSF to identify the scope of a business process(es). It is then incumbent to find out WHO uses a system to do WHAT process (and WHEN and HOW. To do this, it eventually becomes necessary to go beyond the WSF. In some cases, prototyping using low-code approaches (Vincent et al. 2019) can then immediately follow. In many cases, it is desirable to employ varied and specific additional diagrammatic techniques: such as employing ***usage diagrams*** (use case with explicit modelling of interactions / forms – see Table 1 below). Techniques which have worked in practice for modelling HOW the system must function include Event Process Chain (EPC) diagrams - (Scheer, Thomas, and Adam 2005). In the version espoused by SAP, these are extended with data elements. They then link events to processes and to data.

WHAT data is arguably more a design question than an analysis one. However, we – mainly successfully - taught entity relationship modelling with practical work in Microsoft Access to many hundred French business school students from before 2002 until I left that school in 2016. Such closing of a learning loop by building a database or an app has both pedagogical and pragmatic value. See also section 8.4 below.

3.8. Learning about systems

Systems are not the same thing as technology or even the use of technology. The work systems perspective emphasises that organisations, people, systems and technology are intimately interconnected. Managers have to understand systems (although perhaps less so, technology). The work systems framework and the work systems method can give them some of the key knowledge they need to be able to understand, embrace and profit from systems. We now suggest a further learning step.

4. ABC : A mechanism for further Work Systems analysis

Conceptual modelling in the literature primarily concerns itself with information systems (IS) requirements analysis.

ABC, a mechanism for conceptual knowledge mapping, goes well beyond any narrow focus on IS requirements analysis – but it does have value in what used to be called “systems analysis” and particularly in documenting work systems (Alter 2008; 2013) and the related analysis of requirements for information systems.

4.1.What is ABC ?

ABC is a fundamentally visual approach to knowledge mapping in which there are various ***basic (primitive) types of notion***. Primitive types include ***concepts*** and ***procedures***. These notions are related either structurally or in a procedural manner.

4.2.Example concept map for selling online using ABC notation

See Figure 13., which exemplifies the more structured form of concept map called an event process chain EPC diagram.

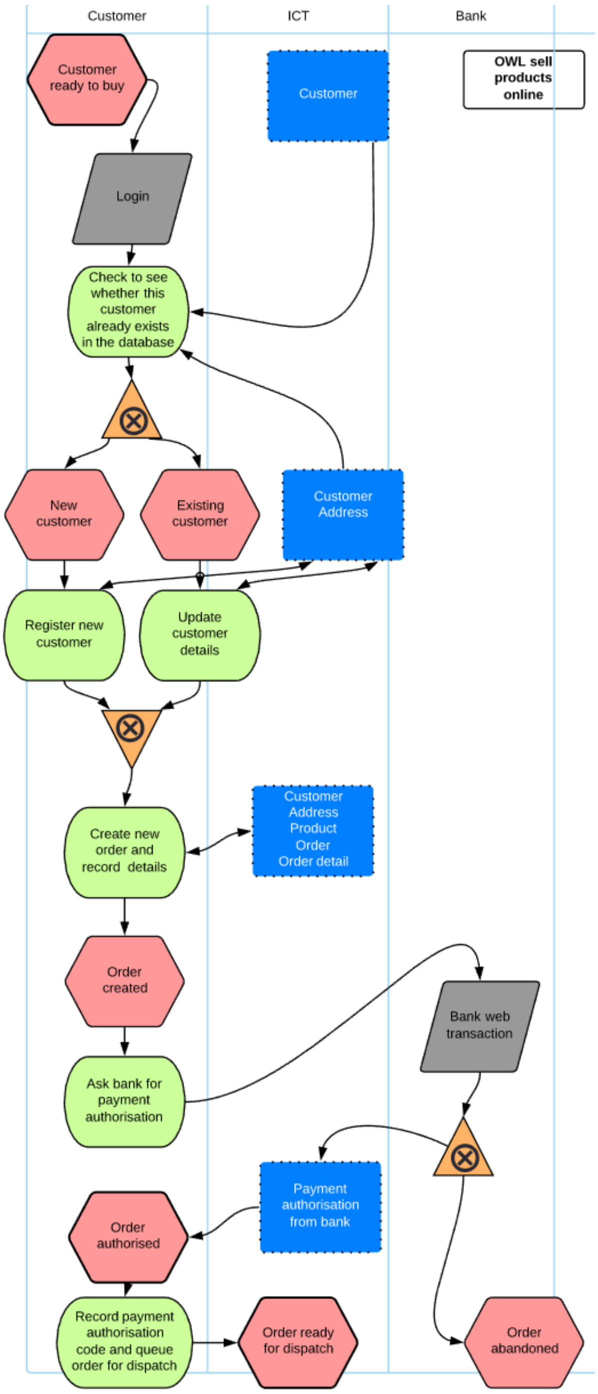


Figure 13 An event process chain model created using ABC. *Source: author.*

In this map of a computerised information system, there are various types of notion – concepts, events, forms, procedures and logical operators. In a map of an information system, concepts are shown to be reified (made real) as data views. The symbols are listed in Figure 14 below.

Actors are here identified as swimlanes, the vertical subdivisions on the map.

The only type of relationship in this map is called ***prompting*** – it shows successor notions as the consequences of predecessor notions. The relationship is a directed arrow.

4.3.Basic notion types

In ABC, notions and relationships are "typed", which means all the various symbols correspond to a single classification or "type" – this follows (Church 1940).

This ABC map shows examples of the major notion types:

- Relationship – here, prompts, the arrow.
- Actor
    - Here shown by swimlanes: Customer, ICT and Bank
- Concept
    - Here, only data views
- Procedure
- Form – an interaction, usually between an actor and a procedure
- Logical operator, split or join
- Event

The basic notion types are identified in Figure 14.

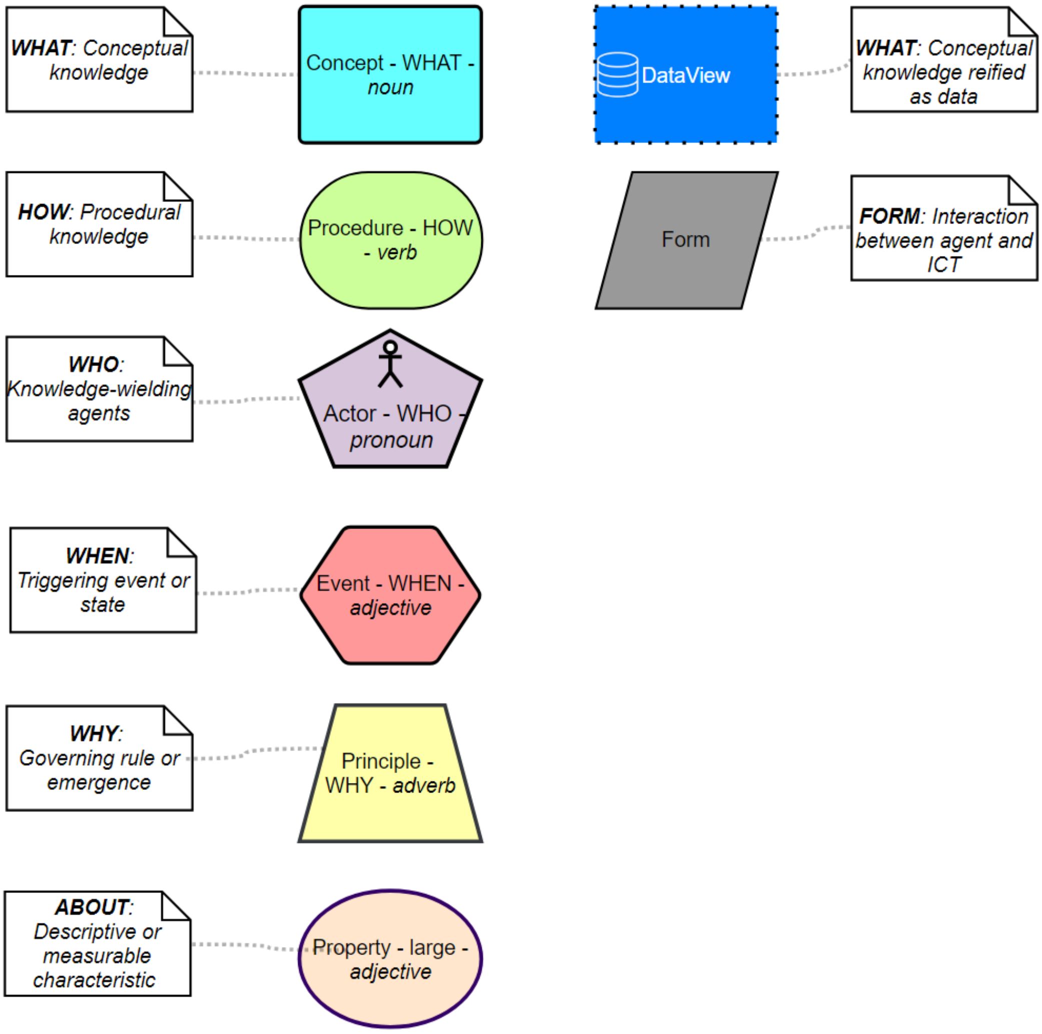


Figure 14 Basic notion types in ABC

4.4.Relationships as structure

Notions, even typed notions, are not enough. Just as in a natural language it is necessary to have fundamental types of speech such as nouns and verbs, so also in a modelling language structure is provided by linking the fundamental notions by means of parts of speech which go beyond nouns.

Relationships are therefore themselves also inherently typed. However, in simple ABC modelling, there are only three essential relationship types, which are association, prompting and regulation.

See Table 3 below for a fuller list of the relationship types in ABC.

4.5.Representing simple ABC relationships and operators

See Figure 15 for the relationship types and Figure 16 for the split and join versions of an operator, AND.

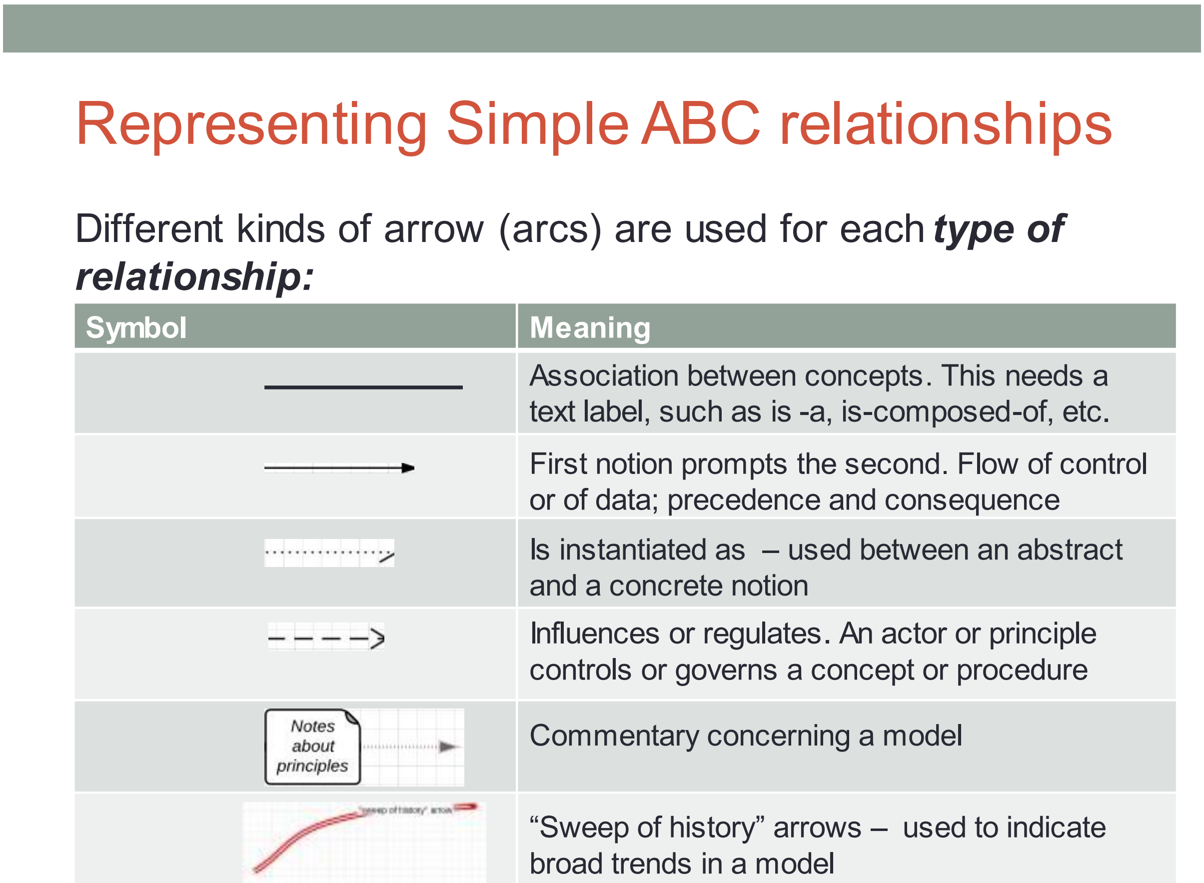


Figure 15 Representing Simple ABC relationships

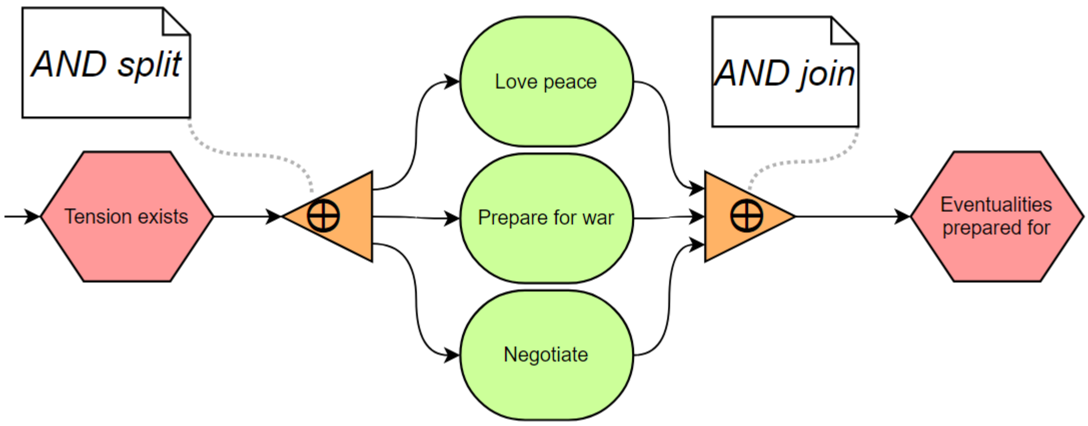


Figure 16 An example of AND split and join operators

4.6.How to build a model

This is a very brief introduction to how to create an ABC model.

- What is the question or topic area that you are addressing?
- What are the top five or so concepts?
- Are there any direct relationships (associations) between these concepts?
    - E.g.: is-a-kind-of, consists-of…
- Otherwise: what procedures (processes) link or transform the concepts?
- Make lists of likely concepts and procedures
    - Perhaps keep these lists in a formal or informal dictionary?
- Identify and make lists of concepts and their "obvious" structural links / associations.
    - Example: Kat is-instance-of cat; beech is-a-kind-of tree
- Identify procedures or processes which link concepts where one needs to be changed or transformed in some way which goes beyond a structural association.
    - farmer (actor) buys (procedure) bull (concept)
    - cow gives-birth-to calf (but here, alternative and perhaps better – structural - models are possible, expressed in terms of parent and child.)
- Sketch out an initial model – on a large sheet of paper or on a whiteboard; perhaps preserve this using a smartphone picture
- Include rich picture elements as images on the map if they will aid comprehension

4.7.ABC Profiles

A ***Profile*** is a named usage of ABC intended for use by a defined group of model writers and readers.

The two profiles intended for general use are ***Starter*** for beginners or for unaccompanied use; and ***General***, for more expert or for mentored users. The general or full profile notably includes further fully-typed relationships.

Because ABC can be used both for work system modelling and for IS requirements analysis, a small number of targeted extensions, called ***profiles***, extend ABC basic notion support to ***use case analysis as usage diagrams***, ***event process chain*** models, ***entity-relationship modelling*** and even "flowcharting" (Nassi and Shneiderman 1973) – which it modernises and improves as ***process – operator – form – view*** diagrams. Similarly, certain IS-specific notions (form, view, entity, attribute) have been introduced into ABC. A major advantage claimed for this approach is that similar notions in different profiles share the same representation, which reduces the learning workload for people who use more than one profile. This in turn eases the learning and implementation times required for multiple but focussed models. See Table 1 for a list of the profiles.

Table 1 Profiles in ABC

Name	Type of model	Explanation / usage	Acronym
Starter	Simple, intended for general knowledge mapping	Beginners' profile - simple concept mapping for beginners and for unaided use by domain specialists or learners (students). The only type of relationship between concepts is an association.	Concept Image Association Operator Procedure Event Actor Property
General	The complete profile, intended for general and advanced knowledge mapping and action modelling. Requires much fuller understanding than the simple profile, principally because there are a range of relationship types. Typical uses include representing knowledge as-is and as-ought, demonstrating understanding, documenting a body of knowledge, design of teaching, research design, advanced self-observation, advanced learning and evaluation.	ABC expert; domain specialist aided by ABC expert.	Typed-Relationship Operator Principle Image Concept Procedure Event Actor Property
Flowchart	Process modelling, e.g. simple flowchart or process-data model	Teaching and learning computer programming, as in introducing algorithmic thinking.	POFV: Procedure, logical Operator, Form, concept as data View
Usage	Usage modelling. ABC use case diagrams are generally similar to UML UCDs (Cockburn 2001) but they are extended to show the interaction between an actor and a use case as a specific interaction element. This is done because such interactions normally need to be implemented, sometimes as form and subform hierarchies, sometimes as webpage hierarchies.	Enhanced use case model, with interactions explicit as forms. Forms can subsequently be implemented as, for example, webpages.	Form Relationships Actors Procedures FRAP
Event-Process Data	Event Process Data modelling. ABC event process chain diagrams are generally similar to ARIS EPC diagrams (Scheer 2000), but they are optionally extended by incorporating a specific Data swimlane. The data swimlane is populated by concepts, which may subsequently be implemented as data tables, data views, specific file-types or by webpages. The value of the data swimlane is that interactions between it and other (non-data) swimlanes	Used to model overall processes, as in business process modelling or work system modelling.	Events Procedure as process Actors as swimlanes or org units Views Operators EPAVO

Name	Type of model	Explanation / usage	Acronym
	enable the modelling of data flows (dataflows) without a separate profile.		
Entity-Relationship	Data modelling, e.g. entity relationship attribute modelling. ABC Entity / Relationship diagrams broadly follow the conventions established by (Chen 1976) and subsequent work. However, ordinality, cardinality and multiplicity are shown in the ABC / UML class diagram style because this is more expressive than Chen's notation.	Trained unaided domain specialist; domain specialist aided by ABC expert. Data models can direct subsequent database implementation.	eNtity Relationship Attribute View NRAV

5. ABC in the service of work systems thinking

ABC is a visual and textual language and toolset intended for capturing, expressing, communicating and co-creating models of topic areas of domain knowledge by domain experts or learners.

An expert or a learner in a given knowledge work domain, decides the vocabulary: the basic notions of interest.

The user follows what are initially quite simple grammar rules as she builds a visual model of her understanding of a topic within their domain of interest.

The user might be a teacher, student, action researcher, autoethnographer… indeed, almost any knowledge worker.

Ideally, they should be mentored in their work by someone with wider experience of the approach.

In this section 5, we present more details concerning ABC.

5.1. Example knowledge representation KR in ABC

Figure 17 shows an ABC map of "Planning and doing the shopping"

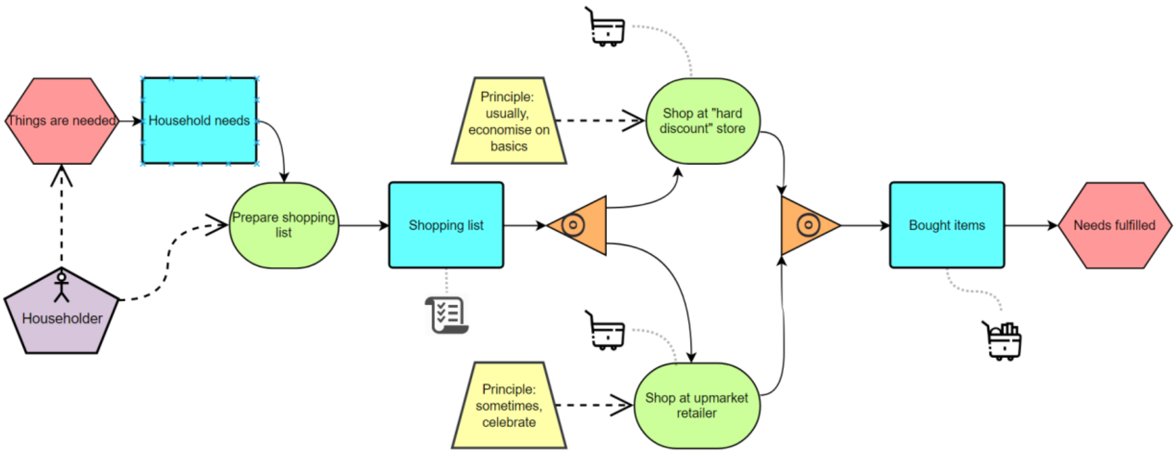


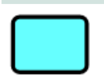
Figure 17 Doing the shopping

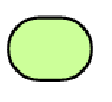
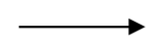
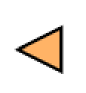
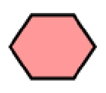
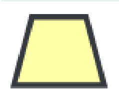
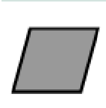
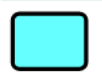
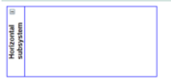
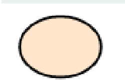
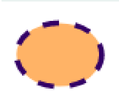
My experience as a teacher is that the creation of such a model is within the reach of almost any business student or practitioner (especially if mentored).

5.2.Notion Types

See Table 2. In ABC, notions and relationships are typed, that is classified. In a manner analogous to the use of parts of speech in natural language, this basic ontological classification assists in the accurate expression of ideas and how they link. ABC extends the Bunge-Wand-Weber BWW ontology (Wand and Weber 1990); Rosemann and Green 2002) beyond its origins in the conceptual modelling of information systems into more general knowledge modelling, specifically, that of work systems. ABC is also indebted to the LICEF G-MOT tradition described notably by (Paquette 2010) and (Basque et al. 2008). The antecedents of ABC are discussed further in section 6.3.

Table 2 Notion types in ABC

Notion	Description	Shape & colour
Concepts	Things, ideas, etc.; these are usable and (perhaps) decidable classes of explicit knowledge or data. Subtypes exist for IS requirements analysis and for social ontology.	
Actors	People, organisations, external systems: types of agent.	

Notion	Description	Shape & colour
Procedures	The means of enacting knowledge in the form of specific activities, repeatable actions and processes – the latter being templates for repeated actions.	
Typed Relationships	Concepts are related by relationships or relationship instances (links). In ABC the only available types of relationship between notions are prompts, regulates and association (which should be given a name). In THE GENERAL PROFILE, relationships may have specific types such as generalisation, aggregation, composition…	
Images	Images illustrate concepts (or any other notion).	
Operators	Logical operators, notably XOR, OR, AND or NOT.	
Events	EITHER occurrences in time that change the state of a class of objects OR named states of a class of objects.	
Principles	Constraints, rules or complex conditions; also, software. Principles may either be pre-existent axioms (input) or emergences at one point in a model of a system (outcome). Principles may regulate or control procedures or concepts.	
Forms	The means of interaction between a system user and an information system; this is typically a webpage or a form in an app. IS-specific, but also used in usage diagrams.	
Entities	Entities are normalised classes of data. They are a specialisation of concepts.	
Views	A view is the computer representation as data of a concept. Frequently it is an unnormalised and updatable query across one or more database entities. IS-specific.	
Systems	Systems and sub-systems can be identified and bounded.	
Properties	Additional concepts about a primary notion, e.g. name (property) of a firm (concept)	
Attributes	An atomic property of an entity.	

5.3. Types of relationship

Different types of links (relationships) exist. They are summarised in Table 3.

Table 3 Relationship types in full ABC

Sign	Relationship name	Profile	Meaning
	associates	Simple	is-associated-with
	prompts	Simple	comes-after, comes-before, leads-to, results-in, causes, is-input-to
	regulates	Simple	controls, directs, influences
	instantiates	Simple	is-an-instance-of
	commentates	Simple	commentates, annotates
	specialises (inherits) / generalises	Full	kind-of
	composes	Full	is-a, is-made-of dependent parts
	aggregates	Full	is-a, is-made-of independent parts
	direction	Full	trend, sweep-of-history
	physical flow	Simple	physical-flow

5.4.Grammar Rules

Grammar rules govern the valid types of links that may join the knowledge types - cf. (Paquette 2010). However, the diagrams.net implementation of ABC does not enforce these rules.

6. The significance of ABC, current and potential

ABC is in part based on original research by LICEF Montréal reported in (Paquette 2010).

6.1.ABC: Why – the motivations

ABC is a visual and textual ***language*** and ***toolset*** intended for capturing, expressing, communicating and co-creating ***models of topic areas*** of domain ***knowledge*** by domain ***experts*** or ***learners***

Whether as an expert or a learner in a given domain, the modeller identifies or decides the ***vocabulary*** in that domain. She then follows what are initially quite simple grammar rules as she builds a ***visual model*** of her understanding of a topic within her domain of interest.

1. I as a ***teacher*** wish to:

   Provide learners with ***signposts to*** and ***syntheses of*** course material – (Ausubel 1963) and (Ausubel 2000)'s "advance organisers".

   Stimulate and assist ***student learning*** as students themselves ***create their own concept maps.***

   Evaluate and ***enhance*** student learning

Both students and I can use ABC maps for these purposes.

2. I as a ***researcher*** need to ***map concepts and their relationships*** in order to:

   Model ***personal work systems*** – the subject of my research; these PWS might belong to me or to research volunteers.

   Clarify and record my ***understanding of complex issues*** and sometimes of complex ***articles*** or ***working documents*** that I read or write.

and

   To ***communicate*** that understanding to others.

ABC therefore aims to be a simple, relevant, easily applicable way to represent, manage and facilitate the communication of ***personal knowledge.***

3. I as a ***researcher of personal information management*** have chosen as one of my research methods ***auto-ethnography*** (Alvesson and Sköldberg 2009); (Jones 2005) which can be characterised by ***structured self-observation*** (Rodriguez and Ryave 2002)**.**

One way of structuring that self-observation is to create ***concept maps*** or ***ABC maps;*** precisely in order to ***give structure to my self-observation and to that of other collaborators.***

4. ***People in commerce and industry often need to model business processes,*** including things, their states and how they move between states in response to events and as causes of events.

ABC has the potential to ***aid conceptualisation, investigation into existing processes and to the design of new.***

5. ABC as a ***means of analysing requirements for work systems and information systems,*** or ***of identifying the functionality offered by existing work systems or information systems.***

Conceptual modelling often concerns itself with information systems (IS) requirements analysis.

The historical roots of conceptual modelling arguably lie in the work of researchers such as (Yourdon and Constantine 1976) on structured program design and of (Chen 1976) on entity-relationship modelling.

ABC goes well beyond any narrow focus on requirements analysis – but it also has potential value in what used to be called “systems analysis”.

For this reason, certain IS-specific notions (form, view, entity, attribute) have been introduced into ABC. A small number of extensions to the ABC basic notation support use case analysis, event process chain models and even entity-relationship modelling.

6.2. Why add yet another modelling language?

It can be argued that agile development (e.g. Abrahamsson, Conboy, and Wang 2009) appears to eliminate the need for explicit modelling before developing software. However, students in many business schools in continental Europe still learn business process modelling and data modelling. Those in engineering schools need to appreciate the importance of identifying existing system functionality and of systems integration. They may well encounter models of such functionality expressed in the form of, for example, ARIS event process chains describing ERP functionality (Scheer 2000).

The design principles that have driven the development of ABC have included:

- Designing a single notation which can be used across the various model types identified in Table 1. This is intended to reduce the cognitive workload required of people learning the language.
- Learning from existing usage and experience.
- A strong emphasis on "learnability".
- An appropriate amount of semantic structure based on a classification of a small number of primitive notion types, with no encouragement to invent new ones.
- Strong support for the basic knowledge modelling process, which we identify as the creation by a modeller of an initial model which is then refined (evolved and simplified)
    - Either alone by the original modeller
    - Or by means of co-modelling by the original modeller and other modellers who may be better "wielders" of ABC
    - Or by exchange between the modeller(s) and domain experts who are not (yet) ABC modellers
- A clear distinction between meta-concepts or ***notion types*** on the one hand, and ***structural relationships*** between notion types on the other.

- The aim is to permit the creation of ***well-expressed models*** which enhance both the ***understanding of the phenomenon*** being modelled and the understanding and ongoing ***learning*** of the modeller(s) and their audiences.

6.3. Antecedents and design influences

- ***Primary historical design influence:*** ABC is a concept mapping approach originally based on the work of Gilbert Paquette and his collaborators at LICEF, Université de Québec à Montréal UQAM.
    - ABC follows on from the LICEF MOT = ***Modélisation par Objets Typés*** and more recent G-MOT (Paquette 2010).
    - Since MOT is ***object-influenced*** (but not object-oriented), so is ABC.
    - However, neither MOT nor ABC are, strictly-speaking, object-**oriented** in the sense understood by (Booch, Rumbaugh, and Jacobson 2005) – in that neither implement full inheritance nor polymorphism; further, both **eschew** encapsulation and instead **explicitly separate concepts and procedures.**
- The LICEF work can be positioned within a tradition which distinguishes various types of knowledge. (Paquette 2010, 18) refers to the distinction made by (Romiszowski 1984) between facts, procedures, concepts and principles:
    - Factual knowledge about objects, events, people and procedures.
    - Concepts that allow us to recognise and categorise instances of a phenomenon.
    - The rules and principles that allow us to link concepts and facts in order to predict or explain phenomena.
    - Thus we can distinguish at least four types of knowledge: facts, procedures, concepts and principles.
- An alternative distinction between types of knowledge is also made by (Paquette 2010, 18):
    - Declarative knowledge is "knowing about" a domain
    - Procedural knowledge is "knowing how" to apply concepts, rules or principles
    - Contextual knowledge is "knowing when and why" to apply concepts, rules or principles
- The author chose to move away from the LICEF G-MOT approach because:
    - The work of LICEF now emphasises instructional design and semantic web modelling rather than general knowledge modelling.
    - The editor is apparently no longer under active development.

- The expression is not very visual, depending too much on textual elements and not on images and icons. It does not engage the right brain. The visual representation used in G-MOT is occasionally obscure, specifically in the areas of:
  - How the different types of relationship are displayed; they are signified by a character label rather than by a visual device.
  - The visual distinction between classes and object-instances (although this is better in G-MOT than in the earlier Mot+).
- It does not permit the clear expression of algorithms, in particular conditionality (if… then… else… endif) and repetition (do while…; repeat until…); both are much clearer in ABC.
- Cardinality and ordinality (together: multiplicity) are not made explicit in associations.
- The language does not encourage consideration of object state and/or events. Events are a separate notion type in ABC.

The author wished instead to emphasise different ontological issues. Specifically, ABC is firmly situated within the scientific realist conceptual modelling tradition in general and the Bunge-Wand-Weber BWW approach in particular: see (Wand and Weber 1990), (Bunge 1977), (Bunge 1979). However, BWW-based approaches tend to restrict themselves to applications associated with information systems analysis and design. ABC is intended for general ***knowledge*** modelling and does not restrict itself to, nor major on, IS application.

BWW itself is not without its critics, notably (Guizzardi and Halpin 2008) who dislike the dogmatic certainties of Bunge and his followers and argue instead for a trope-based approach. However, there are as yet few practical applications of their approach.

Other design influences on ABC include:

- ***Conceptual graphs*** (Sowa 2000); (Wille 1997) following the work of the philosopher Charles Sanders Peirce (early 20th C). However, the ABC approach is deliberately less formal so as to make it more approachable by non-mathematicians and people without formal training in logic or analytical philosophy.
- Certain aspects of the Unified Modelling Language ***UML*** (Booch, Rumbaugh, and Jacobson 2005), in particular use case models and class diagrams - as expounded (inter alia) by (Pooley and Stevens 1999); and certain relationship types and notations.
- Event-driven process chains – ***EPC*** models (Scheer, Thomas, and Adam 2005).
- ***Sequence / condition / iteration:*** the thinking behind the structured programming of (Boehm 1983) can have relevance to more general knowledge modelling.

- The ***rich pictures*** used by Peter Checkland in SSM - (Checkland 1981), (Checkland and Poulter 2006). These are particularly useful in the early stages of co-modelling.

Potential design influences which we have not followed include the ***CMap*** approach adopted by Joseph Novak and his collaborators at Florida Institute for Human and Machine Cognition IHMC, who themselves acknowledge the pioneering work of the cognitive scientist and Piaget-influenced educational psychologist David Ausubel (Inhelder and Piaget 1955), (Novak and Cañas 2008). The merits of this approach include simplicity – the target audiences include school children. Perhaps as a consequence, the modelling language is insufficiently semantically precise. Similarly, we have not followed the cognitive mapping approach typified by the work of Ackerman and Eden, e.g. (Ackermann, Eden, and Cropper 1992; Eden 1988). Again, we regard the modelling language as insufficiently semantically precise for our purposes.

We can summarise diagrammatically the design influences on ABC as in Figure 18:

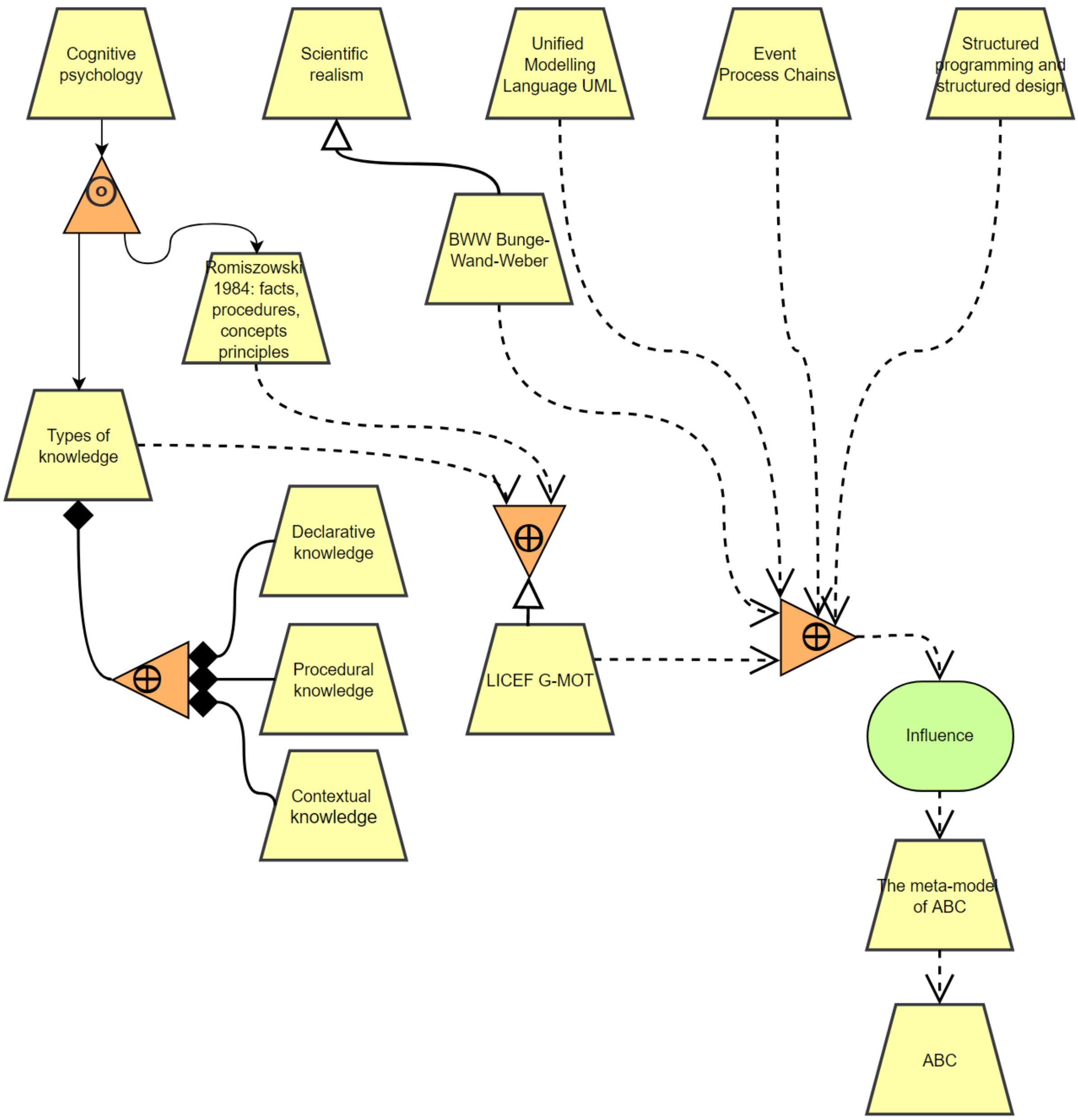


Figure 18 Design influences on ABC

https://app.diagrams.net/#G1leWLSI7iMtbn1c2iAg13i5wXxlMToDps

A meta-model for ABC exists but is not germane to this paper.

6.4. Summary of notion types and roles and analogy with parts of speech

There is a small number of basic (primitive) notion types, most of which are shown here in Figure 19. Each notion type has a symbol, a notion type, a keyword, accompanying notes and a mini-symbol. The keywords correspond fairly closely to parts of speech in a natural language.

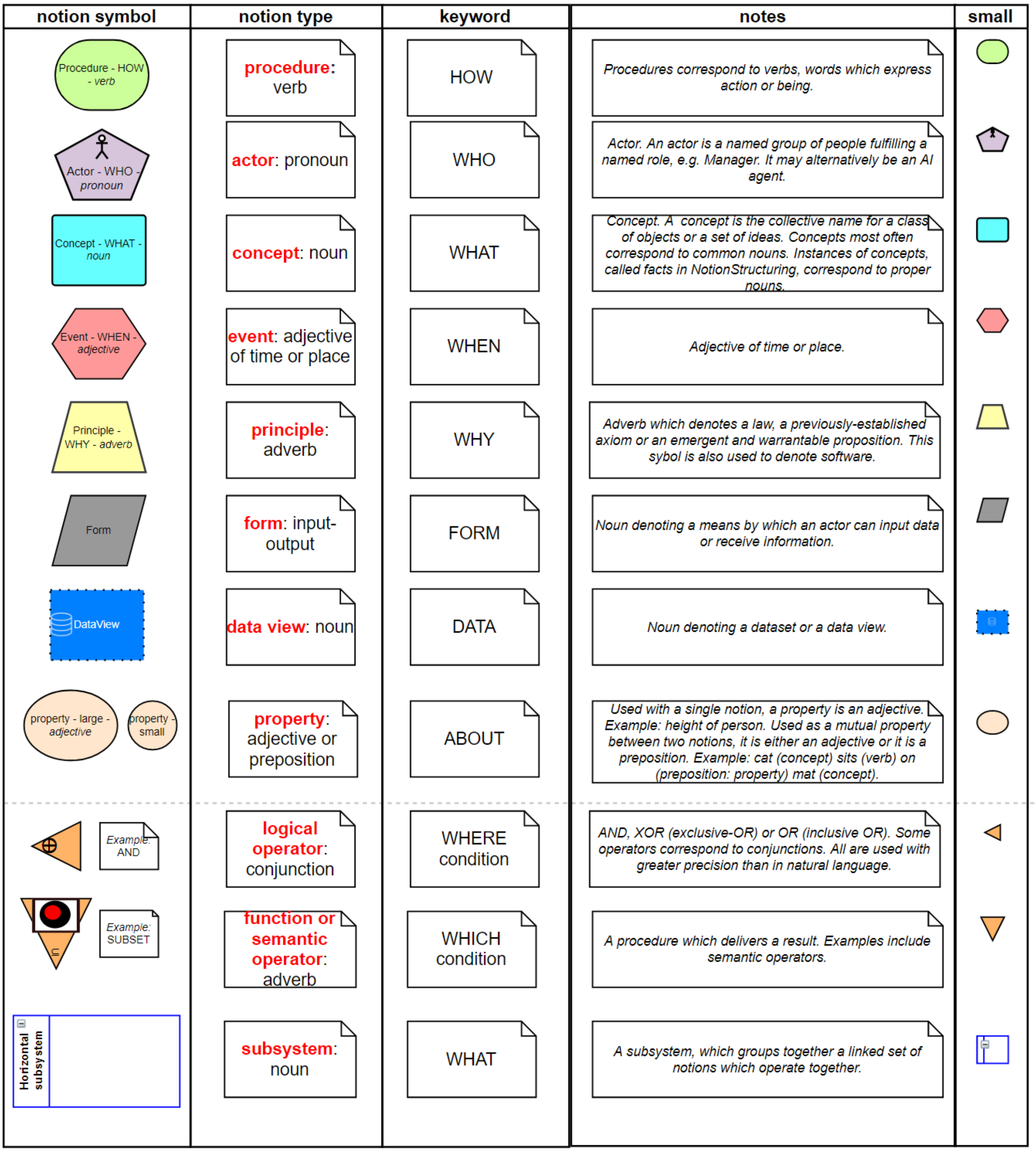


Figure 19 Notion types and analogies to parts of speech

6.5.What to do with complicated models: chunking in ABC

Modellers are advised to avoid excessive detail on any given diagram. As (Miller 1956) showed, our human short-term memory capacity is generally very limited.

A way in which to deal with this is by what is sometimes called chunking, in which we exploit hierarchy as we split up a model when it gets to be too big and complicated – in practice, if it will not fit on a printed page.

Chunking is described at https://en.wikipedia.org/w/index.php?title=Chunking_(psychology)&oldid=1006451017 accessed 22/02/2020.

Consequently, corresponding to nearly all the main notion types in ABC ***there are symbols which tell the reader to look somewhere else for the detail,*** see Figure 20.

Figure 20 Notion types that indicate that further detail appears elsewhere in a model

Note that these symbols have thicker borders with a darker version of the same colour used for ordinary notions. They can be used in complex models which require hierarchical decomposition.

6.6. Easing the task of learning ABC

A practical and cogent argument against all forms of conceptual knowledge modelling is that there are some students for whom learning in this area remains difficult or even sometimes impossible (Bennedsen and Caspersen 2008). In their article entitled “Systems Analysis for Everyone Else: Empowering Business Professionals through a Systems Analysis Method that fits their needs”, (Truex, Alter, and Long 2010) report on “a design science research project demonstrating a possible path toward addressing these longstanding problems by empowering business professionals to analyze systems in business terms, and not in UML, BPMN, or other formalisms that were developed for IT specialists”. The project is the use of the work systems approach, without conceptual modelling.

We do not in any way disagree with their premises or methods. However, we do think it is possible to go a little bit further in the direction of modelling ***how a system does or ought to work, and in understanding and explicating its mechanism***, whence ABC modelling of work systems. We are not certain that we have the “right” answer, and certainly have more work to do to establish its utility and further to address the difficulties associated with learning it.

ABC has been structured as a set of ***profiles***. The simple (starting) profile does require modellers to decide the type of each notion but includes very few relationship types. Beginners can produce maps which are useful for discussion and as the basis for subsequent refinement. Indeed, almost any explicit model is better than a poor or inexplicit one.

Almost all students can read an ABC model, especially if it is explained to them by the modeller. Co-modelling in conjunction with techniques such as structured walkthroughs (Bailey, Pearson, and Gkatzidou 2014) permits the creation of useful maps even when these are incomplete or contain errors.

Moving to the more general profile, which introduces further notion and relationship types, requires further teaching and benefits greatly from active mentoring.

7. Applications of work system thinking with ABC

7.1. A model of an Accounting Information System, illustrating advance organiser, feedback and feedforward

Figure 21 shows an ABC model of an Accounting Information System. This model has been used as an “advance organiser” (Ausubel 1963; 2000) in the teaching of an introduction to information systems module for students of a programme in accountancy and finance operated in the UK and in Hong Kong. An accounting information system necessarily forms a large part of accountancy work systems.

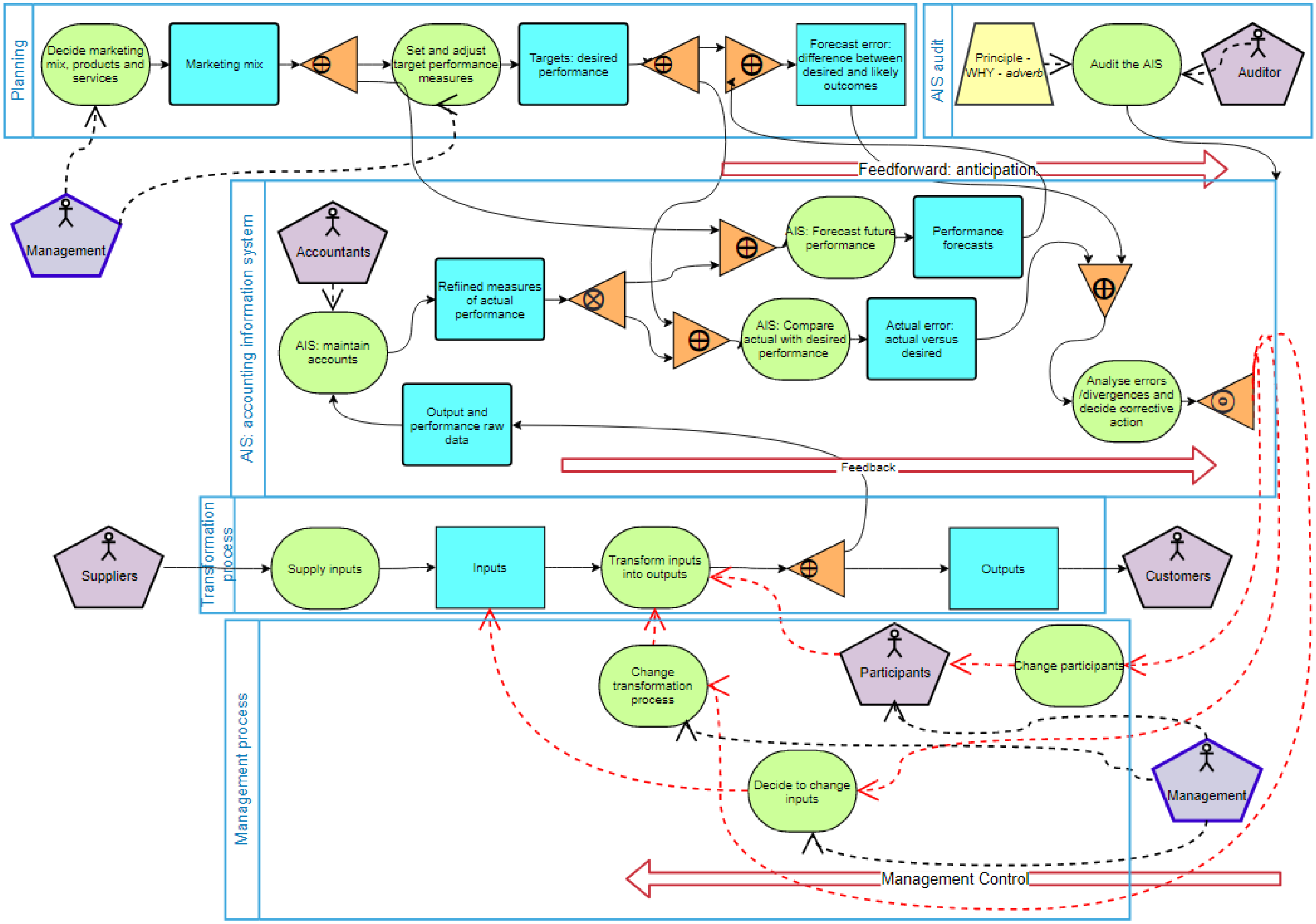


Figure 21 An Accounting Information System, illustrating feedback and feedforward

https://app.diagrams.net/#G1zho7OByVX8Cvviu5PXPiXuZ-2rfi1byM

7.2. Data and knowledge yield information

We suggest a Data, Knowledge, Information DKI work system model following Kettinger and Li (Kettinger and Li 2010). Their work has extended the infological equation of (Langefors 1980), suggesting that information is the joint function of data and knowledge. They name their approach the KBI theory, the knowledge-based information theory. They put forward the following initial definitions:

- Data are the measure or description of states of objects or events, usually referred to as a set of interrelated data items that measure the attributes of the objects or events.
- Knowledge is justified true belief of the relationship between concepts underlying these states.
- Information is the meaning produced from data based on a knowledge framework that is associated with the selection of the state of conditional readiness for goal-directed activities.

Information, representing a status of conditional readiness for an action, is generated from the interaction between the states measured in data and their relationship with future states predicted in knowledge. They view data, information, and knowledge as being core to the Information System (IS) field. In response to limitations in existing models, they propose a knowledge-based theory of information. This is extended from (Langefors 1980)' infological equation, suggesting that information is the joint function of data and knowledge. Different forms of IS are conceptualized as being capable of transforming specific categories of data into information for business operations and decision-making.

They conclude that the production of information from data needs knowledge, and when knowledge varies, so does information.

7.3. Operationalising "data and knowledge yield information" as a work system

To make this theoretical insight more practical, I would suggest it is useful to operationalise the approach as a work system whose working can be represented as an ABC model. See Figure 22 for a suggested model of such a work system.

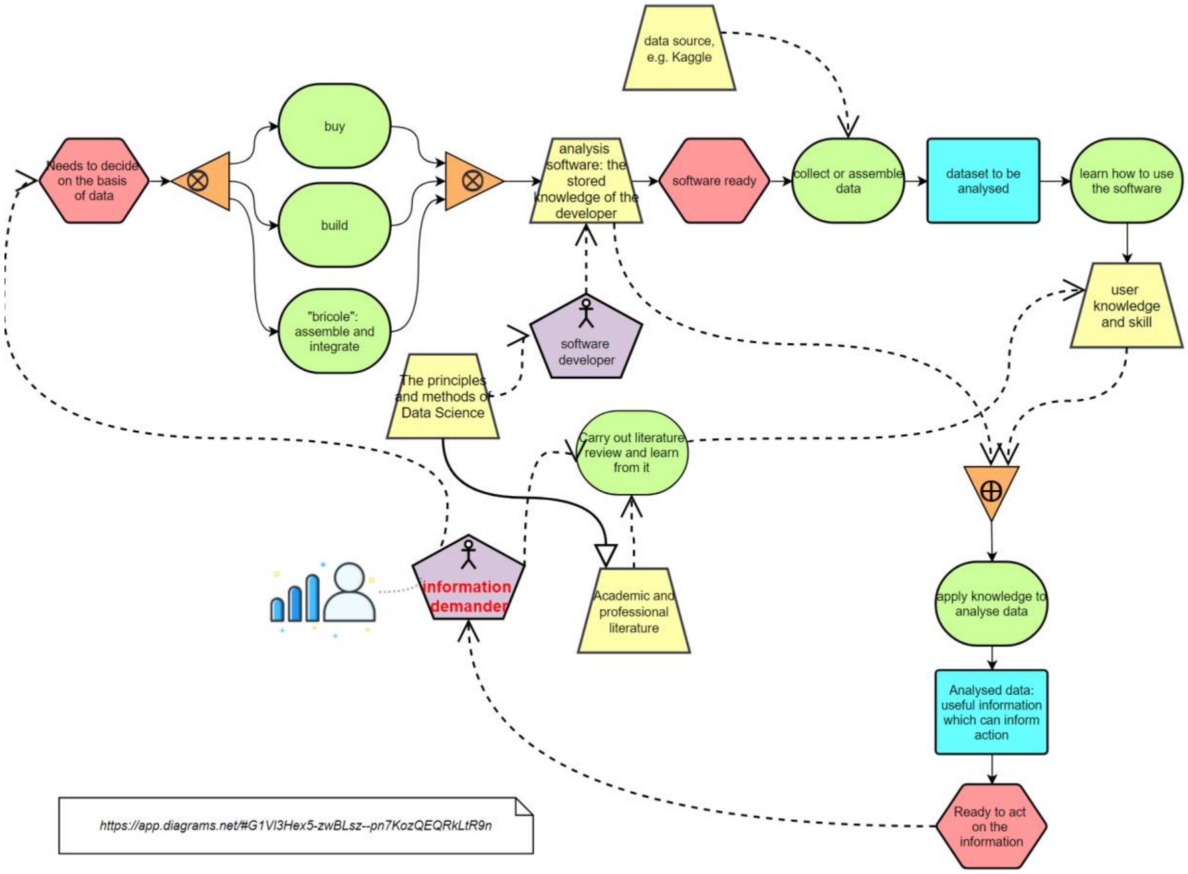
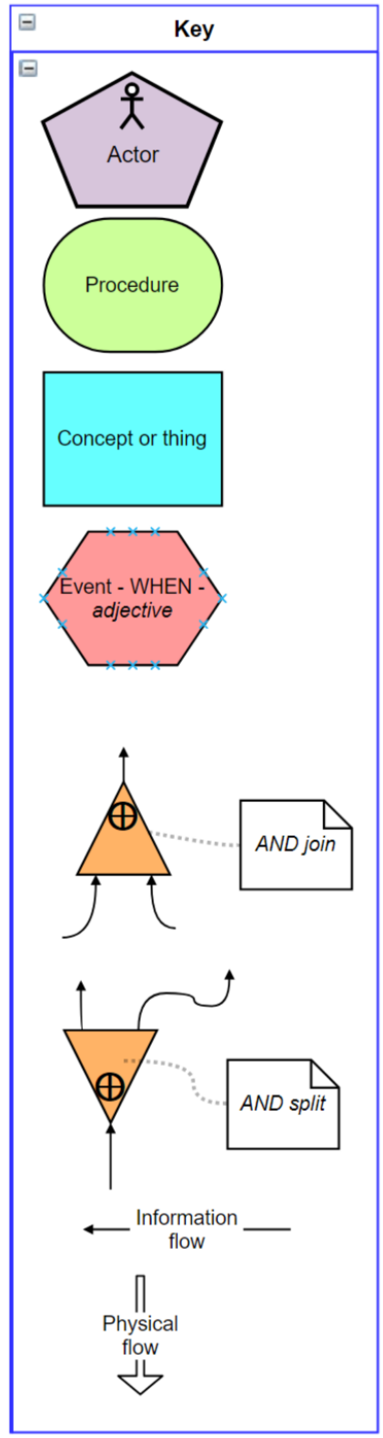


Figure 22 Data and knowledge together yield information

This knowledge or concept map suggests that applying their own knowledge, augmented by the knowledge stored in software by developers who are knowledgeable about data science, can yield information of use to information demanders. This model seeks to make the KBI insight more approachable and more practically applicable. This model has been applied in the teaching and learning of a data analytics module both at MBA level and at post-experience degree apprentice level.

7.4. How to learn how to use ABC in work systems modelling



To create an ABC model of a work system:

- ❑ Choose a simple ***work system*** which you already know well.
- ❑ ***Download*** (but do not open) the Simple custom shape library https://drive.google.com/file/d/1anyZFoV9TgpGicSIRC7GxAR-SdwtHGbQ/view?usp=sharing to your computer and save this, e.g. in Downloads.
- ❑ Download and open with diagrams.net a sample model from Google Drive: https://drive.google.com/file/d/1kzhrt_zqFlSoEyYdIyp9CdgYa-ft2udt/view?usp=sharing.
- ❑ Ensure that the Simple custom shape library that you have just downloaded is open in the Sample model: File / Open library from <downloaded custom shape library>. Then create a new blank model. This can be used as the basis for a first modelling attempt by the user.
- ❑ Create a new window in the sample model: click the + sign at the bottom of the model.
- ❑ Decide the main notions that you need to include on your diagram – see the adjacent Key for the main notion type.
- ❑ Follow a basic diagrams.net tutorial, e.g. https://www.youtube.com/watch?v=-0qxOIP05tw.
- ❑ Start to build your model, dragging shapes onto the canvas.
- ❑ Follow the suggestions made in section 4.6.

8. Evaluating work system modelling using ABC

We need both to evaluate the models that are being produced and the process of modelling of work systems. We would hope to see evidence both of increased understanding by the modeller and by participants in a work system.

8.1. Is ABC pragmatically useful in the modelling and understanding of work systems?

The emphasis in this paper has been on the usefulness of ABC in modelling how work systems ***function*** as what Bunge identifies as ***mechanisms*** (Bunge 2004).

Examples of models of ABC models of work systems appear in this paper as Figure 4 and Figure 21. Figure 22 is a work system which exploits knowledge and data to yield information.

A cohort of degree apprentice students have modelled a work system in their various employing organisations, with varying degrees of success – very high when mentored. A fuller evaluation will be available in the spring of 2021.

Two small cohorts of EMBA students in Romania have previously produced good work, in one instance in a “live” business process investigation.

Our provisional conclusion is that ABC is useful in the modelling, understanding and use of work systems.

8.2. The relationships between systems, information systems and work systems

These are modelled in Figure 23, which features additional typed relationships from the general profile. These are specialisation / generalisation and composition.

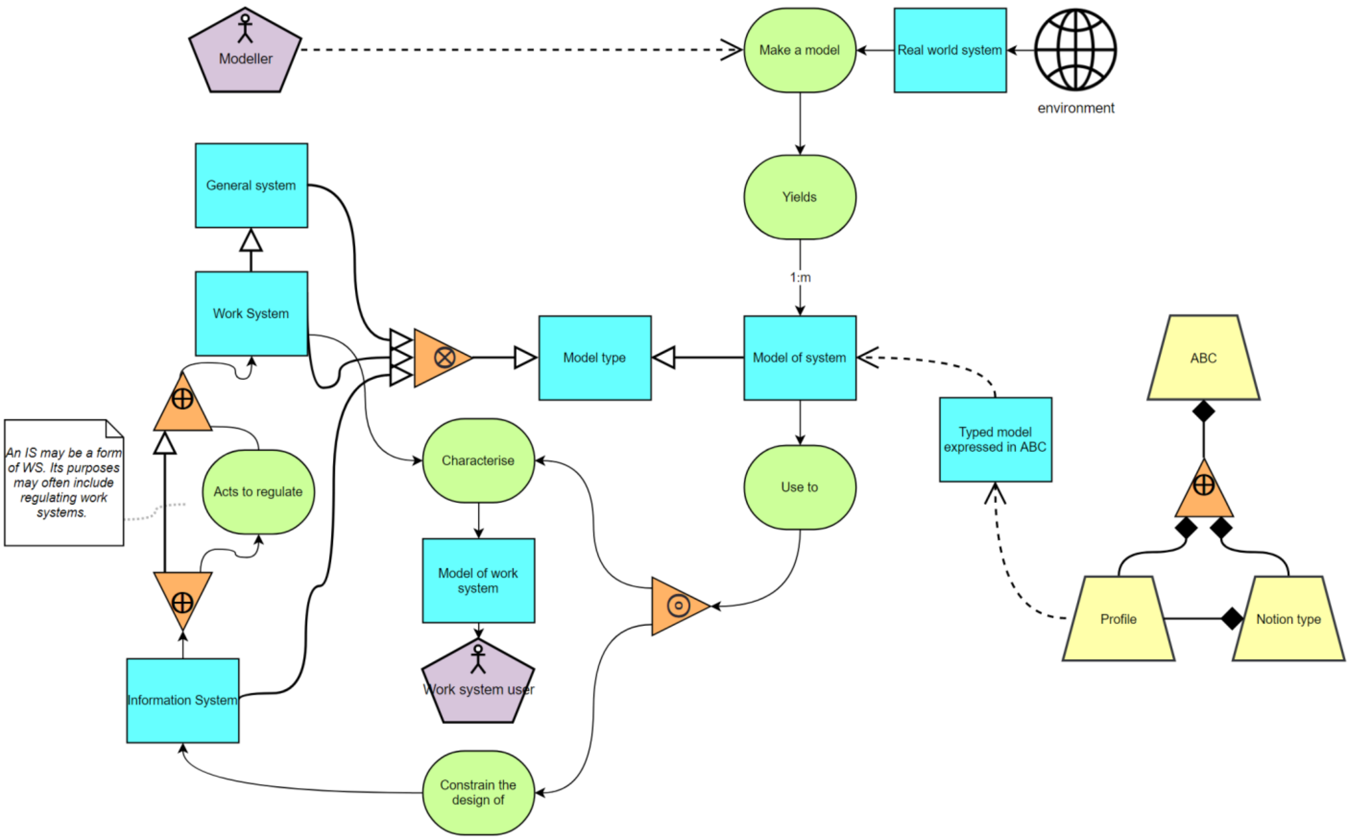


Figure 23 Systems and models of systems

https://app.diagrams.net/#G1GgF3Hq5zipnqoI1Kuku_3aLnYSINxNsO

8.3.Conclusions and further work

This paper has sought to extend the Work System Method WSM of Steven Alter by means of a modelling notation and mechanism called ABC. We have reflected on the ongoing importance of modelling systems, with an emphasis particularly on work systems (thinking about which is opposed to largely arid debates about the nature of the "IT artefact").

We have used ABC to show or to conjecture how things, processes and events interrelate. We have demonstrated by example how to discern conceptual models, so as to understand and, potentially, to improve by design, active models – specifically, the information systems which support work systems. Thus we aim to regulate work systems, whether actively by explicit control or implicitly by aiding learning, understanding and self-control by modellers and participants.

Models are necessary; the IS community with which I identify has a duty to help people understand that.

Control – management – needs and should mandate good modelling aiding requisite variety in the controller (or manager).

Models take different forms, but we should seek to "surface" them; we must help to make them more explicit and perhaps to improve them. Aspects of certain models are active or dynamic, e.g. tables of summary data used to support decisions presented by information systems. Others are representational and apparently passive, but they may influence or control the behaviour of active agents.

The work system framework is an excellent meta-framework which can also usefully be extended by conceptual modelling, for example using ABC.

We should endeavour to build good regulators, specifically good information systems, and to help others to do so.

Renewed emphasis should be given to modelling within the information systems curriculum. In addition to WHAT and WHY, we often need to know HOW. We have sought to reinforce the usefulness of work system thinking and suggested a way of understanding how such systems work by means of a modelling process which ***classifies notions and their relationships by type***. This encourages model-based reasoning and will often make errors or inconsistencies or gaps more clearly discernible to the modeller. This will in turn facilitate our aim, which is the felicity of the model to reality, resulting in increased understanding in particular by business students and professionals.

We can contend – on the basis of much individual and some collective experimental learning – that the modelling of work systems has great utility and can be taught and learned by any engaged and mentored student and would-be modeller. The paper itself has given an introduction and tutorial in the ABC modelling approach as a means for modelling such systems.

In further work, we will seek greater certainty in ensuring that we have met the objection of (Truex, Alter, and Long 2010) which is discussed above in section 6.6.

A more ambitious project, and one not yet funded, will create an integrated development environment for ABC. This is intended:

1. To enforce the grammar rules that should govern the valid types of links that may join the knowledge types - cf. (Paquette 2010). However, the current diagrams.net implementation of ABC does not enforce these rules.
2. To provide fuller support for a ***notion dictionary***, to be integrated with the modelling component currently provided by diagrams.net. We have not discussed the dictionary in this paper. It is, however, essential. Purely pictorial representations ***cannot*** replace natural language or propositional statements. See (Vervaeke and Green 1997)'s refutation of the work of George Lakoff on idealised cognitive models (Lakoff [1987] 1990): "A diagram, however, is just a picture, schematic though it may be, that exists under a description, so to speak. The description alone serves to reduce the referential indeterminacy of the picture. The claim, then, that image-schemata are really something like diagrams amounts to the admission that much of the cognitive work is being done by the descriptive - that is, propositional -apparatus that assigns a unique referent to the diagram. That is, because what assigns a determinate meaning to a diagram is a description, it is very likely that the conceptual apparatus is, in the final analysis, propositional in nature." (Vervaeke and Green 1997, p.76)
3. To guide learners and users of ABC in the modelling and design of work systems and - optionally – in documenting information systems requirements and functionality.

8.4.Easing the learning process concerning work systems and ABC

In the interim, we have presented and exemplified ABC as an already usable, useful and used means for modelling how work systems function and how they can be perceived and redesigned. Learning how to use ABC as a means to model how work systems are (or should be) structured requires a learning effort on the part of modellers and their mentors. To this end, existing and developing web-based learning aids will shortly be made more generally available, in the hope that the ABC approach to modelling will see wider application.[2] A subsequent paper will discuss the principled design of the ABC language in more detail: its multiple parentage in conceptual modelling, knowledge cartography, and critical realist philosophy, and why ABC takes the form that it does.

8.5. Some advantages of modelling work systems in ABC

We argue that ABC can bring the ontological clarity associated with information systems modelling in the BWW tradition to the modelling of work systems and their mechanisms.

ABC is not immediately easy to learn and apply to its full extent; however, its structure as a series of profiles does ease initial and subsequent learning and the starter profile has been successfully used by people who are not modelling specialists.

ABC remains a work-in-progress, but one which already shows promise as a ***useful***, ***usable*** and ***used*** conceptual knowledge modelling approach:

ABC has been (1) ***useful*** for PhD research, for applications in commerce and industry and now for modelling work systems.

It is (2) ***usable*** now. See section 7.4 for guidance.

ABC has (3) now been ***used*** by students, unaided or mentored, and by business professionals.

REFERENCES

Abrahamsson, Pekka, Kieran Conboy, and Xiaofeng Wang. 2009. '"Lots Done, More to Do": The Current State of Agile Systems Development Research'. *European Journal of Information Systems* 18 (4): 281–84.

Ackermann, Fran, Colin Eden, and Steve Cropper. 1992. 'Getting Started with Cognitive Mapping'. In . Banxia Software. https://banxia.com/dexplore/resources/whats-in-a-name/.

Alter, Steven Lewis. 2002a. *Information Systems: Foundation of E-Business, 4/e*. 4th ed. Pearson Education.

[2] References to these aids are withheld in this paper because they would make blind reviewing of planned later papers impossible.

———. 2002b. ‘The Work System Method for Understanding Information Systems and Information System Research’. *Communications of the Association for Information Systems (Volume 9, 2002)* 90 (104): 104.

———. 2006. *The Work System Method: Connecting People, Processes, and IT for Business Results*. Work System Press.

———. 2008. ‘Defining Information Systems as Work Systems: Implications for the IS Field’. *European Journal of Information Systems* 17 (5): 448–69.

———. 2011. ‘The Work System Method: Systems Thinking for Business Professionals’. In *Proceedings of the 2012 Industrial and Systems Engineering Research Conference*. Orlando, Florida. https://repository.usfca.edu/cgi/viewcontent.cgi?article=1031&context=at.

———. 2013. ‘Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future’. *Journal of the Association for Information Systems* 14 (2): 72–121.

———. 2015. ‘The Concept of “IT Artifact” Has Outlived Its Usefulness and Should Be Retired Now’. *Information Systems Journal* 25 (1): 47–60. https://doi.org/10.1111/isj.12048.

Alvesson, Mats, and Kaj Sköldberg. 2009. *Reflexive Methodology: New Vistas for Qualitative Research*. Sage Publications Ltd.

Ashby, W. Ross. 1956. *An Introduction to Cybernetics*. Chapman & Hall London. http://pcp.vub.ac.be/books/IntroCyb.pdf.

Ausubel, David P. 1963. ‘The Psychology of Meaningful Verbal Learning.’ http://psycnet.apa.org/psycinfo/1964-10399-000.

———. 2000. *The Acquisition and Retention of Knowledge: A Cognitive View*. Kluwer Academic Pub.

Bailey, Christopher, Elaine Pearson, and Voula Gkatzidou. 2014. ‘Measuring and Comparing the Reliability of the Structured Walkthrough Evaluation Method with Novices and Experts’. In *Proceedings of the 11th Web for All Conference*, 1–10. W4A ’14. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2596695.2596696.

Basque, Josianne, Gilbert Paquette, Béatrice Pudelko, and Michel Léonard. 2008. ‘Collaborative Knowledge Modelling with a Graphical Knowledge Representation Tool: A Strategy to Support the Transfer of Expertise in Organisations’. In *Knowledge Cartography: Software Tools and Mapping Techniques*, by Alexandra Okada, Simon J. Buckingham Shum, and Tony Sherborne. Springer-Verlag New York Inc.

Bennedsen, Jens, and Michael E. Caspersen. 2008. ‘Abstraction Ability as an Indicator of Success for Learning Computing Science?’ In *Proceeding of the Fourth International Workshop on Computing Education Research*, 15–26.

Boehm, Barry W. 1983. ‘Seven Basic Principles of Software Engineering’. *Journal of Systems and Software* 3 (1): 3–24.

Booch, Grady, James Rumbaugh, and Ivar Jacobson. 2005. *Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series)*. Addison-Wesley Professional.

Boulton, Jean G., Peter M. Allen, and Cliff Bowman. 2015. *Embracing Complexity: Strategic Perspectives for an Age of Turbulence*. OUP Oxford.

Bunge, Mario Augusto. 1977. *Treatise on Basic Philosophy: Volume 3: Ontology 1: The Furniture of the World*. Vol. 3. Dordrecht: Reidel.

———. 1979. *Treatise on Basic Philosophy. Volume 4: Ontology II: A World of Systems*. Vol. 4. Dordrecht: Reidel.

———. 2000. 'Systemism: The Alternative to Individualism and Holism'. *The Journal of Socio-Economics* 29 (2): 147–57.

———. 2003. *Emergence and Convergence: Qualitative Novelty and the Unity of Knowledge*. University of Toronto Press.

———. 2004. 'How Does It Work?: The Search for Explanatory Mechanisms'. *Philosophy of the Social Sciences* 34 (2): 182–210. https://doi.org/10.1177/0048393103262550.

Checkland, Peter B. 1981. *Systems Thinking, Systems Practice*. Chichester: Wiley.

———. 2012. 'Four Conditions for Serious Systems Thinking and Action'. *Systems Research and Behavioral Science* 29 (5): 465–69.

Checkland, Peter B., and John Poulter. 2006. *Learning for Action: A Short Definitive Account of Soft Systems Methodology, and Its Use Practitioners, Teachers and Students*. John Wiley & Sons.

Chen, Peter Pin-Shan. 1976. 'The Entity-Relationship Model—toward a Unified View of Data'. *ACM Transactions on Database Systems (TODS)* 1 (1): 36.

Church, Alonzo. 1940. 'A Formulation of the Simple Theory of Types'. *The Journal of Symbolic Logic* 5 (2): 56–68. https://doi.org/10.2307/2266170.

Cockburn, Alistair. 2001. *Writing Effective Use Cases*. Addison-Wesley.

Conant, Roger C., and W. Ross Ashby. 1970. 'Every Good Regulator of a System Must Be a Model of That System'. *International Journal of Systems Science* 1 (2): 89–97.

Dijkstra, Edsger W. 1982. 'On the Role of Scientific Thought'. In *Selected Writings on Computing: A Personal Perspective*, 60–66.

Eden, Colin. 1988. 'Cognitive Mapping'. *European Journal of Operational Research* 36 (1): 1–13.

Espejo, Raul, and Alfonso Reyes. 2011. *Organizational Systems: Managing Complexity with the Viable System Model*. Heidelberg: Springer. https://www.dawsonera.com/Shibboleth.sso/Login?entityID=https://shiblogin.hull.ac.uk/idp/shibboleth&target=https://www.dawsonera.com/shibboleth/ShibbolethLogin.html?dest=http://www.dawsonera.com/depp/reader/protected/external/AbstractView/S9783642191091.

Gabbay, Dov M., and John Woods. 2005. *A Practical Logic of Cognitive Systems: The Reach of Abduction: Insight and Trial*. Elsevier.

Gemino, Andrew, and Yair Wand. 2004. 'A Framework for Empirical Evaluation of Conceptual Modeling Techniques'. *Requirements Engineering* 9 (4): 248–60.

Greca, Ileana Maria, and Marco Antonio Moreira. 2000. 'Mental Models, Conceptual Models, and Modelling'. *International Journal of Science Education* 22 (1): 1–11.

Guizzardi, Giancarlo, and Terry Halpin. 2008. 'Ontological Foundations for Conceptual Modelling'. *Applied Ontology* 3 (1/2): 1–12.

Hassan, Nik R. 2011. 'Is Information Systems a Discipline? Foucauldian and Toulminian Insights'. *European Journal of Information Systems* 20 (4): 456–76.

Inhelder, Bärbel, and Jean Piaget. 1955. *De La Logique de l'enfant à La Logique de l'adolescent: Essai Sur La Construction Des Structures Opératoires Formelles*. Presses universitaires de France.

Jones, Stacy Holman. 2005. 'Autoethnography: Making the Personal Political'. In *The SAGE Handbook of Qualitative Research*, edited by Norman K. Denzin and Yvonna S. Lincoln, 763–91. Sage Publications.

Kettinger, William J., and Yuan Li. 2010. 'The Infological Equation Extended: Towards Conceptual Clarity in the Relationship between Data, Information and Knowledge'. *European Journal of Information Systems* 19 (May): 409–21.

Lakoff, George. (1987) 1990. *Women, Fire, and Dangerous Things*. Univ. of Chicago Press.

Langefors, B. 1980. 'Infological Models and Information User Views'. *Information Systems* 5 (1): 17–32.

Magnani, Lorenzo. 2009. *Abductive Cognition: The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning*. Springer Berlin Heidelberg. http://link.springer.com/content/pdf/10.1007/978-3-642-03631-6.pdf.

Magnani, Lorenzo, Nancy Nersessian, and Paul Thagard, eds. (1999) 2012. *Model-Based Reasoning in Scientific Discovery*. Softcover reprint of the original 1st ed. 1999 edition. Springer.

Miller, George A. 1956. 'The Magical Number Seven, plus or Minus Two: Some Limits on Our Capacity for Processing Information.' *Psychological Review* 63 (2): 81.

Nassi, Isaac, and Ben Shneiderman. 1973. 'Flowchart Techniques for Structured Programming'. *ACM Sigplan Notices* 8 (8): 12–26.

Nersessian, Nancy J. 1999. 'Model-Based Reasoning in Conceptual Change'. In *Model-Based Reasoning in Scientific Discovery*, edited by Lorenzo Magnani, Nancy J. Nersessian, and Paul Thagard, 5–22. Springer US.

Novak, Joseph D., and Alberto J. Cañas. 2008. 'The Theory Underlying Concept Maps and How to Construct and Use Them'. *Florida Institute for Human and Machine Cognition Pensacola Fl, Www. Ihmc. Us.[Http://Cmap. Ihmc. Us/Publications/ResearchPapers/THeoryCmaps/TheoryUnderlyingConceptMaps. Htm]*.

Okada, Alexandra, Simon J. Buckingham Shum, and Tony Sherborne. (2008) 2014. *Knowledge Cartography: Software Tools and Mapping Techniques*. 2nd ed. 2014 edition. New York: Springer.

Paquette, Gilbert. 2010. *Visual Knowledge and Competency Modeling - From Informal Learning Models to Semantic Web Ontologies.* Hershey, PA: IGI Global.

Pooley, Rob, and Perdita Stevens. 1999. *Using UML: Software Engineering with Objects and Components*. 2nd ed. Harlow, UK: Addison-Wesley Longman.

Psillos, Stathis. 2009. 'An Explorer upon Untrodden Ground: Peirce on Abduction'. *Handbook of the History of Logic: Inductive Logic* 10: 117–51.

Rodriguez, Noelie, and Alan Ryave. 2002. *Systematic Self-Observation*. Sage Publications, Inc.

Romiszowski, Alexander Joseph. 1984. *Designing Instructional Systems: Decision Making in Course Planning and Curriculum Design*. Psychology Press.

Rosemann, Michael, and Peter F. Green. 2002. 'Developing a Meta Model for the Bunge–Wand–Weber Ontological Constructs'. *Information Systems* 27 (2): 75–91.

Scheer, August-Wilhelm. 2000. *ARIS--Business Process Modeling*. Springer.

Scheer, August-Wilhelm, O. Thomas, and O. Adam. 2005. 'Process Modeling Using Event-Driven Process Chains'. In *Process-Aware Information Systems: Bridging People and Software Through Process Technology*, edited by Marlon Dumas, Wil M. P. van der Aalst, and Arthur H. M. ter Hofstede. John Wiley & Sons.

Simon, Herbert A. (1970) 1996. *The Sciences of the Artificial*. The MIT Press.

Snowden, David J, and Mary E. Boone. 2007. 'A Leader's Framework for Decision Making'. *Harvard Business Review* 85 (11): 68–76.

Truex, Duane P., Steven Lewis Alter, and Cherie Long. 2010. 'Systems Analysis for Everyone Else: Empowering Business Professionals through a Systems Analysis Method That Fits Their Needs'. In *Proceedings of the 18th European Conference on Information Systems*. Pretoria, South Africa.

Vervaeke, John, and Christopher D. Green. 1997. 'Women, Fire, and Dangerous Theories: A Critique of Lakoff's Theory of Categorization'. *Metaphor and Symbol* 12 (1): 59–80.

Vincent, Paul, Kimihiko Iijima, Mark Driver, Jason Wong, and Yefim Natis. 2019. 'Magic Quadrant for Enterprise Low-Code Application Platforms'. *Gartner Report*.

Wand, Yair, and Ron Weber. 1990. 'Mario Bunge's Ontology as a Formal Foundation for Information Systems Concepts'. In *Studies on Mario Bunge's Treatise*, edited by Paul Weingartner and Georg JW Dorn, 18:123–49. Atlanta: Rodopi.

Yourdon, Edward Nash, and Larry LeRoy Constantine. 1976. *Structured Design*. Yourdon, Inc.