Engineering, Responsibilities, Sign Systems

Coen Suurmond
RBK Group, Keulenstraat 18, Deventer, the Netherlands
Csuurmond@Rbk.Nl

Keywords: Organisational Semiotics, Enterprise Engineering, Business Modelling, Sign Systems.

Abstract: This paper will address the question of how an engineering approach to enterprise modelling and system development might be combined with an approach to enterprises that does justice to its inherent social character, and where the members of the organisation are responsible for their decisions and not just operators of the systems. In the analysis of this question the concept of engineering will be discussed, along with the characteristics of different kinds of sign systems. System based sign systems (as used in ICT and engineering, and suited to the use of mathematical and logical formulas) will be contrasted with human based sign systems (natural language, appropriate for the adequate representation of values and of individual cases).

1 INTRODUCTION

The question is how we can combine the concept of the enterprise as a social system with the concept of software development as an engineering discipline. An enterprise has many different aspects: as legal entity, as economic actor, as an organisation of people with a common goal, as a community of individuals, as an actor in society. Each of those aspects has its own view on reality, with a perception of itself and of its environment, with its own language, and with norms that determine its behaviour. An enterprise has a vision, a mission and a strategy. An enterprise has processes, structures and employees. An enterprise has stakeholders, and fulfils a number of needs of those stakeholders. All of these are social concepts, invented and executed by and for people. To do so the enterprise uses a number of technical artefacts: buildings, transport of goods and people, machines for processing materials, and ICT for communication and information.

Over the past few decennia there has been a shift in the use of ICT, from EDP (electronic data processing) to information supply. While the focus used to be on the automation aspect, nowadays it is on the information aspect (while the automation of processes has meanwhile continued unabated). The computer based information systems are at the same time by their construction of a technical nature and by their use and purpose of a social nature. This intersection of technical and social system has proven to be problematic.

Firstly there is the problem of analysis and requirements: the developer tries to uncover the logic of the social system, but he is usually handicapped by being an outsider, because he does not speak the language and because he tries to uncover the logic of patterns that have evolved over time. From a systems thinking perspective and from the rationalistic/mechanistic world view of the IS-developer (and he shares this world view with many modern all-round managers) Business Process Reengineering has been presented as a remedy against the perceived irrationality of the business processes. Rationalisation first, systems development next.

Secondly there is the problem of the use of the information systems: does the newly developed system fit the social reality of the users? If that is not the case (in the subjective perception of the users) it can have a number of results. The quality of the business processes deteriorates as a consequence of the impoverishment of the available information. The users maintain the quality of the business processes by setting up and using additional information channels (free text within and outside of the systems). Or the users create their own parallel solutions for the storing and processing of information in spreadsheets and word processors.

In other words, we are dealing with two different discrepancies in the development of information

Suurmond C.
Engineering, Responsibilities, Sign Systems.
DOI: 10.5220/0004461100530059
In *Proceedings of the Second International Symposium on Business Modeling and Software Design (BMSD 2012)*, pages 53-59
ISBN: 978-989-8565-26-6

systems for enterprises: (1) the use of technical artefacts in social contexts; and (2) the mechanistic/rationalistic world view of analysts and developers in the face of the organic reality of practice. These differences can be traced back to a difference in sign systems; to the lingual reality of employees within the processes and of the external experts. In the analysis this manifests itself in the problem of capturing the extra-lingual practice of the business processes in the lingual reality of the analyst, and vice versa the challenge for the practitioners of evaluating the documents produced by the analysts. And in the implementation this manifests itself in the confrontation between the newly designed sign systems of the information systems and the established practice of the business processes.

The question is now how to best tackle these problems. It is about finding the balance. It should be clear that an enterprise is a historically and organically grown entity, embedded in social practices within the enterprise and between the enterprise and its external stakeholders. At the same time the grown practice should not be held as absolute; it should be possible to critically evaluate it. However, the backgrounds against which such an analysis is carried out are important. Is it from a mechanistic/rationalistic world view in which the enterprise is viewed as a technical artefact? Or is it from a view of the enterprise as a more organic entity, subject to a multitude of social forces and an emergent phenomenon?

2 ENGINEERING

Of old, engineering belongs to the world of the designing and building of technical artefacts. Etymologically, the word is related to the word engine. According to the OED, an engineer is nowadays (1) "one whose profession is the designing and constructing of works of public utility, such as bridges, roads, canals, railways, harbours, drainage works, gas and water works, etc"; (2) "a contriver or maker of 'engines'", (3) "one who manages an 'engine' or engines"; (4) "(with defining word, as human engineer, spiritual engineer), one who is claimed to possess specialized knowledge, esp. as regards the treating of human problems by scientific or technical means". The meaning of engineering is simply "to act as an engineer", again according to the OED (OUP, 1989). Henry Petroski cites a definition of structural engineering in his book "To Engineer is Human": "Structural engineering is the science and art of designing and making, with economy and elegance, buildings, bridges, frameworks, and other similar structures so that they can safely resist the forces to which they may be subjected" (Petroski, 1992, p40). The organisation behind this definition, the Institution of Structural Engineers, defines ""structures" as "those constructions which are subject principally to the laws of statics as opposed to those which are subject to the laws of dynamics and kinetics, such as engines and machines" (Thomas, 2012).

To put it briefly, the core task of an engineer is to design and make a technical artefact to serve some predefined function, and the artefact should be able to resist the forces applied to it without losing its usability.

The role of modelling in the work of an engineer is (1) to preview the design and (2) to study the forces that will be applied to the artefact when it is constructed and when it is in use. A beautiful example of such a model can be found in Barcelona, where Gaudi used some very simple materials (iron hoops, strings, and tiny sand bags) to study the forces for the design of his unorthodox buildings such as the Sagrada Familia and the Casa Milà.

The engineer uses a variety of scale models and prototypes, each representing one or more aspects of the intended artefact, as stepping stones to his final design. The artefact is the physical realisation of the design and the preparatory models.

In the field of information systems, the term engineering is used in different contexts. I will shortly discuss three of them. The first is Software Engineering, and in the book "The Road Map to Software Engineering" the following IEEE definition (Std 610.12) is used: "(1) The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software, that is, the application of engineering to software. (2) The study of approaches as in (1)". The author considers software engineering as engineering, because the process consists of related activities performed in response to a statement of *needs* and consuming *resources* to produce a *product*", in combination with systematic controlling and measurement processes (Moore, 2006, p3). Although the physical aspect of the technical artefact is lacking, all other elements of traditional engineering are present: predefined needs, a clear finished product, and a process of design and development which has a technical character. Dines Bjørner in his work about Software Engineering defines engineering as "the mathematics, the profession, the discipline, the craft and the art of turning scientific insight and human needs into technological products" (Bjørner, 2006). The technological product is central, and its role to serve human needs. Although the context is software, the definition applies equally well to technological artefacts as bridges and engines.

The second context is Systems Engineering, and here the issues are less clear cut. The website of the International Council on Systems Engineering (INCOSE) states: "Systems Engineering is an engineering discipline whose responsibility is creating and executing an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule compliant manner throughout a system's entire life cycle.". On another page of the website it is stated that "Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs". This is a far cry from a well defined technical artefact. Apart from the sloppiness of the statements, there is a fundamental difference between an obligation to produce a technical artefact, and the obligation to fulfil business needs and user needs.

More recently, the term engineering is also used in the context of enterprise engineering. One of the pioneers in the field is Jan Dietz, who is editor of the book series on Enterprise Engineering at Springer Verlag and a driving force behind the CIAO network. The website of the book series mentioned above tells us: "Enterprise Engineering is an emerging discipline for coping with the challenges (agility, adaptability, etc.) and the opportunities (new markets, new technologies, etc.) faced by contemporary enterprises, including commercial, nonprofit and governmental institutions. It is based on the paradigm that such enterprises are purposefully designed systems, and thus they can be redesigned in a systematic and controlled way. Such enterprise engineering projects typically involve architecture, design, and implementation aspects" (Dietz, 2011). Key in this statement is the presumption that enterprises are purposefully designed systems. All the same, this approach recognises the enterprise as essentially social systems (postulate 2 of the Enterprise Engineering Manifesto of the CIAO network), and it recognises the ethical necessity of taking the responsibilities of the people in an enterprise seriously (postulates 2 and 5). Here, the concept of engineering is taken from the technical environment to the organisational environment. The characteristics of engineering as a discipline that designs and builds artefacts are preserved by considering the enterprise as "purposefully designed systems", and by modelling the essential structures of an enterprise in its ontological model, which captures objectively the structure of the enterprise (postulate 4). In his book about Enterprise Engineering Dietz writes "The *engineering* of a system is the process in which a number of white-box models are produced , such that every model is fully derivable from the previous one and the available specifications. ... Engineering starts from the ontological model and ends with the implementation model" (Dietz 2006, p74).- The white-box model is a model that represent the structure and workings of a system, abstracted from implementation details. As such, it is an abstract model. The interesting question is, whether this engineering approach for the enterprise considered as a designable socio-technical artefact will hold up.

3 THEORY OF THE FIRM

An enterprise has a sustainable existence if and only if it produces products for its markets in a profitable way. The business processes represent the enterprise in action; there the products are created, sold and distributed and all activities that are needed to support and manage these primary processes happen there. The formal organisation of the enterprise is designed to structure the business processes in effective and efficient ways, and in accordance with the values and norms of the enterprise towards the internal and external stakeholders. The informal organisation, the collection of actual patterns that structure the business processes, is in a continuous evolutionary process as a result of the daily interaction of people and systems in the enterprise and its environment. An important driving force of the evolutionary processes of the informal organisation is the leadership and management of the enterprise, it shapes the values in the organisation. The corporate culture influences the way people behave in the enterprise; it supplies some background norms for their decisions. For James Taylor these aspects are so important, that he hypothesises that the organisation is "constituted in the ongoing processes of its members" (the title of a presentation earlier this year in Nijmegen was "Authoring the Organization"). Others are not going so far, but John Kay emphasises that the foundation of corporate success often lie in intangible factors such as reputation and internal architecture. He calls these factors the distinctive capabilities, and they determine the success and durability of the enterprise (Kay, 1993). The former strategic planner of the Shell Oil Company, Arie de Geus, makes a case of defining an enterprise as an organism, instead of a mechanism (De Geus 1997). When on the other hand an organisation would be considered in a mechanical way, and the behaviour of its members would be fully determined by explicit rules, we would have a perfect bureaucracy. And we know that the ultimate action weapon for a civil servant is to work to rule, making everything grind to a halt. In other words, the actual organisation, the factual business processes and the individual decisions of the members of the organisation are partly determined by explicitly defined structures and rules, partly by corporate culture, and of course partly by incidents and individual capabilities.

At the same time, the structure of an enterprise will be partly determined by objective factors, derived from the characteristics of its products and its markets. They have a kind of inherent process logic, determined by the conventions of the markets, by the products and the production and distribution processes, and by the social and legal conventions involved. That implies, that by knowing the markets and products of an enterprise, you can predict quite a lot about its internal process steps. One could consider these structures as the essential deep structures of the enterprise. What is much less predictable, however, is the way these structures are mapped unto business processes and the organisation. These mappings are the result of both conscious organisational decisions, and of evolutionary processes. Gradual changes of markets and products can drive evolutionary adaptations in the business processes, and the history of an enterprise might long be reflected in its operational ways.

For the analysis of an enterprise then, we have to deal with both process logic derived from its markets and products, and idiosyncratic characteristics derived from its specific history and evolution from the past. From the viewpoint of the enterprise, to comply with the process logic is necessary to be in business, to be unique and have distinctive capabilities is necessary to stay in business.

4 SEMIOTIC THEORY

4.1 Signs

Semiotics studies signs in use. One well known definition of a sign is: “A sign, or representamen, is something which stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign or perhaps a more developed sign. That sign which it creates I call the interpretant of the first sign. The sign stands for something, its object. It stands for that object, not in all respects, but in reference to a sort of idea, which I have sometimes called the ground of the representamen.” (Peirce, 1998, par2.228) Fundamental in the concept of the sign is the difference between the sign and that what it stands for, and the difference between the sign and its interpretation by the sign user. The process of interpretation of a sign (semiosis) is the subject of pragmatics, semantics studies the relation between the sign and its meaning (that what it stands for), and syntax in concerned with the formal relations between signs.

For the pragmatist the concept of meaning is directly connected to its use, as Wittgenstein writes: “The meaning of a word is its use in the language” (Wittgenstein, 2009, par. 43). Knowing the meaning of a word is knowing when and how to use the word. Related to this concept of meaning is the Wittgensteinian concept of language games. A language game involves a community of users in a certain context, along with the (unwritten) rules how to use language in this context. This pragmatist (not to confuse with pragmatic!) approach of meaning emphasises the primacy of the actual use of signs, the lexical meaning is an abstraction and record of the stable kernel of the meaning in use. Even when definitions of words are explicitly written down, the actual use in the community of users will ultimately determine the meaning. Examples of this mechanism can be found in law, see for example the differences between written law and its interpretation in jurisprudence in European law, and in the workings of case law in Anglo-Saxon practices.

4.2 Sign Systems

The concept of the sign system is widely used in semiotics, but rarely defined. Intuitively it is easy to give a number of examples of clearly different sign systems. Consider the difference between natural languages (English, Japanese), formal languages (predicate logic, C#), visual languages (pictograms on airports and stations). Hereafter it quickly becomes more difficult: can different language families be considered different sign systems? Different languages within one language family? Different dialects? The language games of

Wittgenstein? Are the social media examples of new sign systems, with their own vocabulary, usage rules and expressive power? Hereafter, I will talk about "system based sign systems" for the formatted information that is recorded, processed and presented in available business information systems, and "human based information systems" for information that is processed by human beings (both linguistic and non-linguistic).

Above, more or less formal and/or structured information flows were mentioned in the context of the organisation. The ledger system in a financial administration has a prescribed formal main structure that allows external stakeholders (taxes, chartered accountants) to know in advance what information can be found where. Within this prescribed main structure an organisation can make its own choices, but the main grouping is recognisable for all stakeholders. This is an example of an engineered sign system. A very different example is the way in which all kinds of things are designated in an organisation ("office of Jean-Claude" although Jean-Claude has left 30 years ago, or volumes designated by some odd measure such as "two mills of French fresh", indicating a certain amount of a certain quality of shrimps). These are historically grown practices that refer back to field names, to an amount and a quality, and to a measure that is long abolished in trade, but still used in production planning. Both sign systems are not readily accessible by an outsider. The extent to which a sign systems is defined can differ greatly, but it is true for every sign system that it has to be learned in practice.

4.3 Characteristics of Sign System

For an analysis of the characteristics of sign systems the dimensions of the Information Space as defined by Max Boisot can be useful. He distinguishes in his analysis of information the following three dimensions: (1) degree of codification, (2) degree of abstraction, and (3) degree of diffusion (Boisot, 1998). His analysis is directed towards knowledge assets; applied to sign systems we could distinguish between the way information is codified, abstracted and diffused in a sign system. On the first sight, there is a huge difference between (1) system based sign systems that are a combination of highly codified information (structured data that might be interpreted by the computer programs) and less codified information (free text that is recorded and made available to the user) and (2) The human based sign systems that are much less codified and based on conventions (the Wittgensteinian language games).

However, the real difference might lie elsewhere, not so much in the degree of codification as well in the way of codification. Experienced employees can communicate in a very precise manner, but the communication might require a rather long initiation process. Without a professional background, the words just do not make sense to the outsider. The system based sign systems on the other hand seem easy to use but are often imprecise. The available codification in the ICT system might just not match the needs and the categories of the user. In other words, the first difference between system based sign systems and human based sign systems lies in the nature of the codification processes. In talking and writing an experienced user can code a lot of information into his utterances. The degree of imprecision can be stated in a very subtle and precise way ("the same quality as last time, but remember last week!"). The modalities and illocutionary force of speech acts are hard to codify in ICT systems ("please keep in mind that this customer might order next week, and that we will have to deliver within reason, but due to the unpredictable weather he probably will not order until Friday").

The second difference lies in the diffusion processes. It is clear that the diffusion processes are greatly influenced by ICT systems. When we restrict ourselves to the use of business information systems (rather than communication techniques such as video links, skype, et cetera), these networked systems make all information in them instantaneously available to all users (when they are granted the right authorisation). This diffusion processes create problems of their own, such as uncontrolled interpretation by inexperienced people and inconsistent information, that I will not discuss here.

Abstraction is needed whenever general rules are applied to individual cases. In system based sign systems, the abstractions are predefined in its data structures and individuals are coded as objects. The possible abstractions are predetermined by a combination of the more or less fixed data structures, software code and configuration. In the human based sign systems, the abstraction coincides with the interpretation and is contextually determined. This prototypical case for this kind of abstraction processes is the verdict of the judge. The application of the law can be seen as the subsumption of the individual case under the general rule. The perception of the individual case involves abstraction processes, and this abstracting perceptual power is exactly where the added value of the human being is. That applies not only to professional experts, but it applies also to all kind of work where people are accountable (paid to use their own judgement, and not just 'hired hands').

4.4 Sign Systems and Business Processes

So, basically we are dealing with two differences between system-based sign systems and human based sign systems: (1) the way the signs are captured and coded, and (2) the interpretation capabilities. For practical purposes, two questions arise. The first question is about the costs and effects of information processing by ICT systems. What is the ratio between the efforts to capture the requirements and to subsequently engineer the solution, and the resulting savings and process improvements in the business processes? The second question is not about calculations of return of investment, but more fundamentally which decisions can be made by ICT systems and which decisions have to be made by the members of the organisation. The capability of the human mind to interpret information in context, to absorb new situations and to judge between norms cannot be met by ICT systems. This basic capability is fundamental to the concept of accountability.

The characteristics of the business processes in an enterprise are partly determined by its markets and products, and partly by its idiosyncratic elements like location, infrastructure, corporate culture and so on. When we bear in mind what Arie de Geus has written about the living company, and what John Kay has written about distinctive capabilities, we should be aware of the risks in considering an enterprise as a whole as an artefact that can be engineered.

At the same time, the inherent process logic, as determined by the markets and products, is much more suited to an engineering approach. One could consider this process logic as the essential structure of the enterprise, as it represents the invariant structural elements of the organisation. However, this objectified approach to the inner workings of the enterprise should not be confused with the elements that define the unique position of the enterprise. The way the process logic as an abstract model is implemented in the real enterprise systems is decisive for the operations of the enterprise, and in this area an analysis of the sign systems in use is relevant.

4.5 Representation of Elements from Business Processes in Software

Item coding of semi finished products in food processing industry is a notoriously difficult problem, because the variable characteristics of the products cannot be mapped uniquely on fixed item codes. This is quite different from item codes of engineered and standardised materials such as M3 bolt. In practice knowledge of the customer codetermines which lots are suited for a sales order. The item code is essential in the recording of the sales order, the consignment note, and the invoicing, but in itself insufficient for the preparation of the sales order. Information systems that pretend to calculate the internal processes from the sales orders will fail, the judgment of an experienced human planner is undispensable.

Another modelling challenge lies in the representation of objects. What is considered to be one object by one department, might be more objects for other departments. Some time ago, I saw a freight train consisting of pairs of freight carriages. Each pair was connected by a hooded transition in between. This creates a typical modelling problem: do we have one four-axle carriage, or two two-axle carriages? For operational logistic planning purposes it will probably be the first (each pair is a fixed capacity unit), and for the maintenance department it will probably the latter (individual carriages with each its own maintenance history).

Software often has problems with these kinds of constructs, and the requirement analyst wants to know "what it really is". This is an expression of a certain world view, where atomistic facts are presupposed. It is a world view that fits perfectly well to relational databases. Unfortunately, it is not a world view that fits to the actual world, where technical artefacts can be both one and two at the same time. However, this kind of problem is still essentially a solvable engineering problem. It may lead to complicated software, but the several interpretations can be modelled unequivocally and converted into software solutions.

Whenever there is a misfit between the sign system of the ICT solutions and the sign systems of the user and his processes, the ICT solutions will cause problems. It leads to distortion of processes, of information, or both. A habitual cause is a rationalistic engineering approach of ICT designers to user processes that are not well understood. Close observation and sometimes participation in the user processes is needed to detect the rationale behind the patterns in the processes, or at least to understand which information matters for the user.

A second element of a successful ICT system is the design of a system of references that satisfies the needs of both the ICT system and the users. It might sound as a trivial requirement, but the solution is often difficult. It involves an understanding of the user processes, and the way the user integrates information from the ICT systems with information from other sources. At the same time, the integrity of the ICT systems must be warranted. The system engineer must understand that his system is not the centre of the world, and that it should serve its users. A sales order is an agreement between buyer and seller from the moment that they make that agreement. When and how the agreement is registered in the ICT system is relevant, but not decisive. If the order is split in the system into more sales orders for some reason, it will still be one sales order for the customer.

5 CONCLUSIONS

The engineering approach is the right approach whenever a well defined artefact is to be delivered, as it is able to deal with the well defined intended use of the artefact. An engineering approach is bound to use formalised sign systems, that allow for precise and thorough testing, based on logic and causal relations. Engineered artefacts (and their preparatory models) are subject to the rigid laws of physical forces (bridges, engines) and of logic (abstract software models and enterprise models).

However, when social factors and individual and collective decisions determine the actual use of the technological artefacts, it is not matter of engineering anymore. That applies to bridges, where traffic might be regulated by the marking of traffic in lanes, by traffic lights and by speed limits. That applies also to software systems, that must be configured for use in an organisation with a specific corporate culture, with a given composition of employees with knowledge and experience, and with a distribution of tasks and responsibilities. Decisions in this realm are partly based on goals and values, and cannot be expressed in formal sign systems.

For system development, it is a big challenge to combine an engineering approach for capturing the process logic of an enterprise, as determined by its markets and products, expressed in formal sign systems, with an implementation of this abstract process model that does justice to the idiosyncratic elements of the enterprise. The analysis and expression of these idiosyncratic elements calls for the use of human based sign systems (natural language with all its capabilities to express norms and values with all its subtleties) and might be compared to anthropological analysis.

The analysis presented in this paper should enhance the awareness of the nature of the problem. It should help to demarcate the domains where formal sign systems as used in engineering are applicable, and the domains where human based sign systems are called for. Not discussed in this paper, but most important, is the question how to reconcile the use of these different kinds of sign systems in the development of a concrete information system.

REFERENCES

Bjørner, D.. 2006. *Software Engineering 1*, Springer Verlag. Berlin.

Boisot, M.H. 1998. *Knowledge Assets*, Oxford University Press. Oxford.

De Geus, A, 1997. *The Living Company*, Nicholas Brealey Publishing. London.

Dietz, J.L.G., 2006. *Enterprise Ontology*, Springer Verlag. Berlin.

Dietz, J.L.G. (ed.), 2011. *Enterprise Engineering – The Manifesto*, CIAO Network. From: http://www.ciaonetwork.org/publications/EEManifesto.pdf.

Kay, J., 1993. *Foundations of Corporate Success*, Oxford University Press. Oxford.

Moore, J.W., 2006. *The Road Map to Software Engineering*, John Wiley & Sons. New York.

OUP., 1989. *The Oxford English Dictionary*, Oxford University Press, Oxford.

Petroski, H.., 1992. *To Engineer is Human*, Vintage Books. New York. Smith, J., 1998. *The book*, The publishing company. London, 2nd edition.

Peirce, C.S., 1998. *Collected Papers of Charles Sanders Peirce*, Thoemmess Press. Bristol.

Thomas, R., 2012. *History of the Institution of Structural Engineers*, Institution of Structural Engineers. From: http://www.istructe.org/webtest/files/e1/e1faa2cf-e604-47d0-a888-31a9d6758fa8.pdf

Wittgenstein, L., 2009. *Philosophical Investigations*, John Wiley & Sons. New York, 4th edition.