

Using Fractal Enterprise Model to Assist Complexity Management

Ilia Bider & Erik Perjons

DSV, Stockholm University, Stockholm
{ilia|perjons}@dsv.su.se

Abstract. This paper deals with the problems of a complex organizational system in which not each of its parts is directly connected to all other parts. For such a system, it is important to identify which parts/sub-systems need to be directly connected to each other, and which could be left without such connections. The paper puts forward a hypothesis that a suitable enterprise model could be used for this end, and investigates the suitability for this end of one particular enterprise modeling technique called Fractal Enterprise Model.

Keywords: complexity management, Ashby law, requisite variety, enterprise modelling.

1 Motivation

In this position paper, we use a classical definition for complexity management in the form of Ashby law of requisite variety [1]. The law states that for a system to survive and prosper in a dynamic environment, its variety should match or be greater than the variety of the environment. The variety itself can be roughly defined as a number of possible states of the environment or system respectively. In essence, the law requires that for any relevant for the system perturbation in the environment, the system needs to have an appropriate response.

There are two ways for managing variety, either to decrease variety of the environment, using so-called attenuators, or increase variety of the system, using so-called amplifiers. Normally, a combination of both is used. One of the accepted ways of increasing the variety of a system is to structure the system in a number of subsystems, each of which is responsible for its own segment of the environment, and does not look at other segments at all. As the result, the variety of the system will be much greater than when each system's element is related to each segment of the environment.

One way of structuring the system to achieve greater variety is suggested in Viable System Model (VSM) of Stafford Beer [2], where each unit of System 1 is responsible for its own part of the environment. In addition, such a unit can be recursively decomposed according to VSM, thus splitting the unit to smaller parts – System 1 units of the next level. Another way of structuring a system to increase variety is a layered architecture. A typical example of how such architecture works is presented by Michael Po-

lanyi in the essay "Life's irreducible structure" [3], where different layers concerns different aspects of reality, e.g. physical vs. mental, and each layer provides a mechanism for the next layer to implement its functionality. A similar concept is used in the IT world, e.g. for creating communication protocols.

The maximum variety that could be achieved by system structuring is when the environment can be represented as a set of completely independent segments, which never happens in reality. Therefore, it is impossible to structure a system so that all its subsystems are completely independent from each other. Some mechanism of interaction between the subsystems is needed. In VSM, coordination between the System 1 units that work on the interdependent or intersecting segments of environment is entrusted to so-called System 2, while System 3 has an overall responsibility for cohesion of work of System 1 units. In the layered architecture of Polanyi, the coordination between the layers is achieved by the next layer setting constraints on the functioning of the previous layer.

When the subsystems need to interact with other subsystems, the increase of system variety will be maximized if each subsystem interacts only with few other subsystems, not all of them. This will create a really complex system in terms of Niklas Luhmann [4], who differentiates simple complexity, where each element of the system is connected to each other element of it, and complex complexity, where total interconnectivity is not achievable, and each element is connected only to few other elements.

An interesting example of complex complexity is given by Michael Polanyi in his essay "The republic of science: its political and economic theory" [3]. The question raised in the essay is how Science can function as a whole given extreme specialization of scientists. He shows that cohesion in Science as a system is achieved through the interdisciplinary fields where two or more "pure" scientific fields become interconnected. This interconnections help to propagate new achievements in one field to some distance to a completely different field, ensuring wholeness of the system.

The question that we want to raise in this paper is how to find which subsystems of an enterprise, or any other type of organizational system, need to be directly connected, and which do not need such a connection. We aim to use a special type of enterprise models for this end that is called a Fractal Enterprise Model (FEM). It was first introduced at PoEM 2012 [5], and then extended and improved in [6]. FEM has a form of a directed graph with two types of nodes *Processes* and *Assets*, where the arrows (edges) from assets to processes show which assets are utilized by which processes and arrows from processes to assets show which processes help to have specific assets in healthy and working order. The arrows are labeled with meta-tags that show in what way a given asset is utilized, e.g. as *workforce*, *reputation*, *infrastructure*, etc., or in what way a given process helps to have the given assets "in order", i.e. *acquire*, *maintain* or *retire*.

A FEM is built recursively by using a so-called unfolding procedure and two types of archetypes: *process-assets archetypes* that show which kind of assets might be needed for running a process, and an *asset-processes archetype* that shows which processes are needed to maintain an asset in order. Unfolding starts with a primary process - a process that delivers value to a customer/beneficiary - by applying process-assets archetypes and alternating them with the asset-processes archetype.

The rest of the paper is structured in the following manner. As FEM is not a commonly used model, in Section 2, we give a short overview of it. The reader interested in the details, and the reasons why the model is called fractal and what was the motivation for its development could get this information from [6]. In Section 3, we discuss how FEM can assist in finding which subsystems of an organizational system need to be interconnected. Section 4 is devoted to concluding remarks and plans for the future.

2 Fractal Enterprise Model

A Fractal Enterprise Model (FEM) includes three types of elements: business processes (more exactly, business process types), assets, and relationships between them, see Fig. 1 in which a fragment of a model is presented. The fragment is related to a hypothetical company that sells books over the Internet. Graphically, a process is represented by an oval, an asset is represented by a rectangle (box), while a relationship between a process and an asset is represented by an arrow. We differentiate two types of relationships in the fractal model. One type represents a relationship of a process “using” an asset; in this case, the arrow points from the asset to the process and has a solid line. The other type represents a relationship of a process changing the asset; in this case, the arrow points from the process to the asset and has a dashed line. These two types of relationships allow tying up processes and assets in a directed graph.

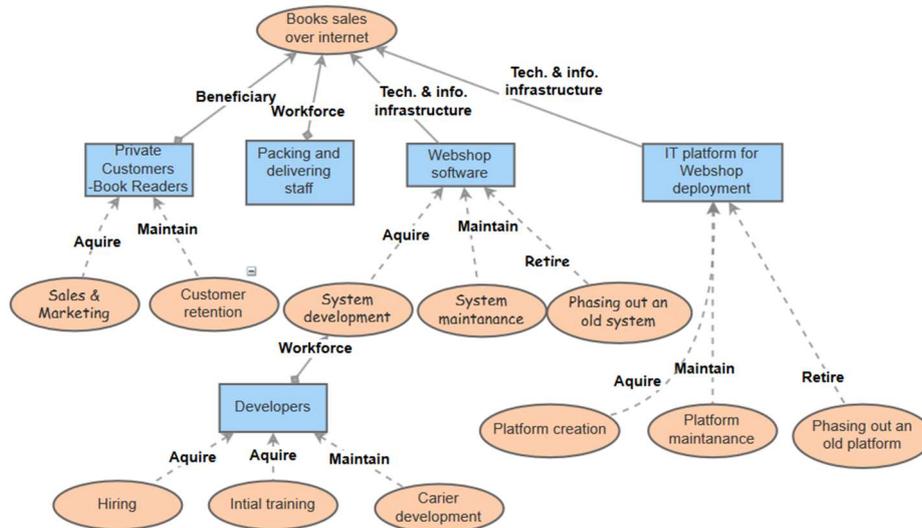


Fig. 1. A fragment of a FEM, adapted from [7], drawn with the help of Insight maker.

In FEM, a label inside an oval names the given process, and a label inside a rectangle names the given asset. Arrows are also labeled to show the type of relationships between the processes and assets. A label on an arrow pointing from an asset to a process identifies the role the given asset plays in the process, for example, *workforce*, *infrastructure*, etc. A label on an arrow pointing from a process to an asset identifies the way

in which the process affects (i.e. changes) the asset. In FEM, an asset is considered as a pool of entities capable of playing a given roles in a given processes. Labels leading into assets from supporting processes reflect the way the pool is affected, for example, a label *acquire* identifies that the process can/should increase the pool size.

Note that the same asset can be used in two different processes playing the same or different roles in them, which is reflected by labels on the corresponding arrows. It is also possible that the same asset can be used for more than one role in the same process; in this case, there can be more than one arrow between the asset and the process, but with different labels. Similarly, the same process could affect different assets, each in the same or in different ways, which is represented by the corresponding labels on the arrows. Moreover, it is possible that the same process affects the same asset in different ways, which is represented by having two or more arrows from the process to the asset, each with its own label.

In FEM, different styles can be used for shapes to group together different kinds of processes, assets, and/or relationships between them. Such styles can include using dashed or double lines, or lines of different thickness, or colored lines and/or shapes. For example, a diamond start of an arrow from an asset to a process means that the asset is a stakeholder of the process (see the arrows “Workforce” in Fig. 1).

Labels inside ovals, which represent processes, and rectangles, which represent assets, are not standardized. They can be set according to the terminology accepted in the given domain, or be specific for a given organization. Labels on arrows, which represent the relationships between processes and assets, however, can be standardized. This is done by using a relatively abstract set of relationships, like, *workforce*, *acquire*, etc., which are clarified by the domain- and context-specific labels inside ovals and rectangles. Standardization improves the understandability of the models.

3 Using Fractal Enterprise Model for Complexity Management

To use FEM for identifying systems interactions, we need to introduce some kind of systems related to FEM's nodes. For this purpose, we consider a Business Processes (BP) from a systems perspective suggested in [6]. According to this suggestion, we can identify two types of systems related to BPs, one belongs to the level of process instances the other to the level of process types:

1. On the instance level, we consider a Business Process Instance (BPI) as a temporal system created for handling a specific situation, for example a request for quotation. After the situation has been dealt with, the temporal system is disbanded. This interpretation corresponds to the idea of respondent system from [8].
2. On the type level, we consider that for each type there exists a permanent socio-technical system that is responsible for starting and monitoring process instances of the given type, and supplying them with resources/assets needed for attaining the instances operational goals, such as people, tools (e.g. IT systems), procedures (e.g. manuals, process maps), etc. We call this system a Business Process Work System or BPWS for short. Note that term *work system* was introduced by S. Alter, see, for example, [9], and BPWS can be considered as a particular class of work systems.

Note that not only a BPWS is a socio-technical system, but also a BPI. The latter can be quite complex and it can have a long life length. However, our focus in this work is more on the BPWS than its BPIs. Note also that a BPWS does not coincide with a particular department that is responsible for the process, but it includes all people, methods, tools engaged in the process.

Returning to FEM, we can consider that a process node with all connected assets nodes that are used in the process represents a BPWS related to this process. Having this view, two BPWSs can be considered as connected to each other if there is at least one common asset to which both have links. Such connections can be classified in three categories:

1. Both BPWSs manage the same asset. For example, both BPWS *System development* and *System maintenance* in Fig. 1 manages *Webshop software*. Naturally, they need to interact with each other. *System development* needs to provide system documentation and answer questions when needed. *System maintenance* needs to report the problems to be addressed in the next version of the system. We refer to this type of interactions as to *horizontal interactions*.
2. One BPWS manages an asset used in another BPWS. For example, *Hiring* manages asset *Developers* that is used in BPWS *System development* as *Workforce* in Fig. 1. Naturally, these two BPWSs, *Hiring* and *System development* need to interact. *System Development* provides requirements on what qualifications are required to *Hiring*. *Hiring* provides information on the state of the job market. The latter may be used for redesigning the system development process. We refer to this type of interactions as to *vertical interactions*.
3. Both BPWS share the same asset. For example, two BPWS may use the same IT system (there is no example in Fig. 1 for this case). Naturally, these two BPWSs need to interact to come to agreement, for example, on the requirements for shared asset, or/and on the way it is shared. We refer to this type of interactions as to *coordinating interactions* (it has the same meaning, more or less, as System 2 in VSM).

Note that our classification of interactions between BPWS is somewhat related to the classification of interactions between work systems suggested in [10]. The difference lies in that we consider a specific type of work systems - BPWS, and specific interactions between BPWSs being connected to the same asset.

4 Concluding remarks

This is a position paper aimed at raising interesting questions for discussion, rather than giving the answers. The contribution of this paper is threefold. Firstly, we formulated a question for discussion on "how to identify which sub-systems of a really complex organizational system need to be connected", where by really complex we mean a system in which not each of its parts is directly connected to all other parts. Secondly, we put forward a hypothesis that an enterprise model could be used as a tool for answering this question. Thirdly, we investigated a particular enterprise modeling technique, FEM, to

see how this hypothesis could be implemented in practice. The investigation is preliminary, but it gives some direction for further research.

As far as practical application of using FEM for complexity management is concerned, we see FEM, in the first place, as a diagnostic tool that helps to identify the places where interactions should exist and assess the quality of the channels for these interactions. Besides ensuring that interactions are happening in a timely and consistent manner, these channels should be decentralized. The latter ensures that a channel itself does not create extra connections to the sub-systems that do not need to be connected.

As far as directions for further research are concerned, the most important one is answering the question of how interactions between connected BPWSs could be practically arranged. This can be answered by analyzing current practice by conducting case studies. Another research direction is comparing the usage of FEM for complexity management with other modeling techniques, e.g. VSM, to find out benefits, drawbacks and limitations. Note, however, that different modeling techniques decompose a system into subsystems differently. Thus, most probably, the results achieved while applying different techniques will be rather complementary, than contradictory. For example, VSM places focus on coordinating units of System 1, which constitutes System 2. The issues of coordination between System 1 units and supporting functions, which are spread through System 3, 4 and 5, are left in the shadow. The latter is what using FEM can help with.

References

1. Ashby, W. R.: *An introduction to cybernetics*. Chapman & Hall, London (1956)
2. Beer, S.: *The Heart of Enterprise*. Wiley (1979)
3. Polanyi, M. S.: *Knowing and Being*. University of Chicago, Chicago (1969)
4. Luhmann, N.: *Introduction to Systems Theory*. Polity Press (2013)
5. Bider, I., Perjons, E., Elias, M.: *Untangling the Dynamic Structure of an Enterprise by Applying a Fractal Approach to Business Processes*. In : *Proceedings of PoEM 2012*, Springer, LNBIP 134, pp.61-76 (2012)
6. Bider, I., Perjons, E., Elias, M., Johannesson, P.: *A fractal enterprise model and its application for business development*. *Software & Systems Modeling* (2016)
7. Bider, I., Perjons, E.: *Using a Fractal Enterprise Model for Business Model Innovation*. In : *BPMDS 2017 RADAR*, CEUR Vol 1859, pp.20-29 (2017)
8. Lawson, H.: *A journey through the systems landscape*. College Publications (2010)
9. Alter, S.: *The Work system method: Connecting people, processes, and IT for business results*. Work System Press (2006)
10. Alter, S.: *System Interaction Theory: Describing Interactions between Work Systems*. *Communications of the Association for Information Systems: Vol. 42 , 42(Article 9)* (2018)