Enterprise Operating System

Alexey Sergeev[1,2]

[1] National Research University – Higher School of Economics
Dept. of Information Systems and Technologies,
Bol. Pecherskaya 25,
603155 Nizhny Novgorod, Russia
[2] Department of Engineering and Management, Instituto Superior Técnico
University of Lisbon, Portugal,
aisergeev@yahoo.com,

Abstract. In this paper, we introduce the term "enterprise operating system" as the essential component of enterprise which supports its viability. We further explain its notion, and investigate its relevance for the enterprise. We consider enterprise to be a viable system, therefore, enterprise operating system should correspond with the viable system model. We also explore concept of distributed systems to compare them to the enterprise.

Keywords: Enterprise operating system, viable systems, viable system model, distributed systems, enterprise architecture, enterprise engineering

1 Problem statement

Most enterprises operate in dynamic environment and constantly changing conditions. In order to keep existing, enterprises must adapt to the changes. However, in order to adapt, enterprises must comply with the following conditions [1]:

1. Have mechanisms to monitor external environment for changes. In case any change which may potentially affect enterprise occurs, some internal adaptation mechanisms should be triggered.
2. Have planning mechanisms. These mechanisms are responsible for resources and operations planning in order to comply with expected and natural environmental changes.
3. Have mechanisms for critical situations prevention. This means that enterprise should have mechanisms responsible for forecasting negative environmental changes which may potentially lead to critical situations. Such mechanisms will ensure that necessary resources for overcoming critical situations are in place, and they will support internal adaptation mechanisms in their reaction for environmental changes.
4. Have adaptation mechanisms. These mechanisms are triggered once environmental change affecting enterprise occurred, and they work in accordance with issue prevention mechanisms, if applicable.
5. Coordination and control mechanisms. These mechanisms are responsible for coordination between other mechanisms and subsystems, and controlling their behavior.
6. Have enterprise-wide policies regulating behavior of all other mechanisms and subsystems. Such mechanisms are ensuring that enterprise as a whole is following general rules of behavior and operation.

However, often enterprises do not have some of these mechanisms, or are not aware of existence of such internal mechanisms, which prevents their effective usage.

The problem statement for the thesis research can be formulated as follows:

Enterprises do not have or not aware of the internal mechanisms responsible for enterprise operation and adaptation, which leads to ineffective or lack of adaptation at all, which in turn may lead to enterprise end of existence.

The goal of the research is *to create a methodology which will make enterprise self-aware and ensure that all mechanisms necessary for enterprise operation and adaptation are in place.*

2 Proposed solution

The solution requires to introduce the notion of Enterprise Operating System (shortly EOS) first:

Enterprise operating system is the essential component of enterprise system that supports system's basic functions, controls the way a system works, manages system resources and their allocation among actors to make it possible for them to function and work together, adapt to changes and recover after critical situations.

"Essential component" means that the very existence and viability of an enterprise are dependent on its EOS. This question is elaborated in more detail in the next section. "System's basic functions" in terms of the enterprise are those which related to the creation and delivering products to customers. This EOS function is implemented through mechanisms of workload distribution within the enterprise, authority, responsibility and competence distribution. "Controls the way a system works" means that EOS has controlling and monitoring functions like monitoring flawless execution of transactions, checking for consistency of states and transitions, monitoring compliance to action rules, monitoring indicators of enterprise operation and performance, ensuring governance, risk management and compliance mechanisms are functioning. "Manages system resources and their allocation" means that EOS is responsible for ensuring resources (materials, tools, information, knowledge) are in place, and correctly and efficiently distributed within the enterprise. "Make it possible for them to function and work together" means that EOS is responsible for providing communication channels for actors, supporting mechanisms for authority delegation, resolving potential conflict situations, supporting teamwork.

EOS per se is the full set of mechanisms ensuring enterprise operation, viability and adaptability. We consider every enterprise to have its EOS in some form. We also consider that there is the Universal Enterprise Operating System (UEOS), which includes all the mechanisms required for ideal and most efficient enterprise operation, viability and adaptability.

The problem statement can then be reformulated in the following way:

EOS does not include all required mechanisms of UEOS, which leads to ineffective or lack of adaptation at all, which in turn may lead to enterprise end of existence. The goal of the research is *to create a methodology which will help to ensure EOS includes all required mechanisms from UEOS.*

In order to create and use the methodology, it is needed to:

a) Create specification of UEOS. Such specification will be the reference for each enterprise to align their own EOS.
b) Find the modelling method for EOS representation. In order to be able to work with EOS, it needs to be modelled in some way.
c) Create the model of UEOS, which will be the reference model for each enterprise, using the modelling method defined.

After aforementioned steps are completed, the methodology will be:

1. Create EOS specification.

   In order to do that, authors propose to use the following procedure:

   1) Analysis of the ontological model of the enterprise.

   This step will help to quickly get high-level understanding of the enterprise processes and structure, and draw first ideas about EOS implementation of this particular enterprise.

   2) Interviewing enterprise representatives.

   These representatives can be enterprise owners, enterprise executives, enterprise managers, or any employees knowledgeable about enterprise processes and structure and who can contribute to the discussion. The proposal is to handle such interviews in the following way. First, explain the definition and notion of the EOS to enterprise representatives and check if their understanding is correct. Next, ask for their opinion about EOS implementation in particular case of their enterprise. After that, ask them to recall how changes and critical situations (both internal and external) were handled by this enterprise previously.

   3) Running series of case-study discussions with enterprise representatives.

   Continuing metaphor of operating system being applied to the enterprise, we can outline possible ways to explore EOS mechanisms.

   When user interacts with operating system of a personal computer, he/she does not see the operating system mechanisms and functions unless he/she tries to make changes to operating system environment or unless anything critical occurs with the system. Examples of changes to OS environment are installing or uninstalling applications and redistributing computing resources in case heavy computing workload is applied to the system - these are considered as internal changes to operating system state as they happen within OS, as well as changing hardware configuration of the personal computer (e.g. adding more system memory, and therefore, more resources are available for OS to distribute among tasks) – this type of changes is considered external to OS as it happens out of OS zone of control. Examples of internal critical events are memory access errors or failure of file system, examples of external critical events are hardware failures like hard drive failure or memory module failure, or power outages in electrical network which prevent personal computer from operation. The difference between changes to OS and critical events is their level of predictability – we consider predictable (expected) events as changes and unpredictable (unexpected) events as critical situations. The difference between internal and external events are level of control OS has – internal events are within OS zone of control, while external events are out of OS zone of control. Summing up, we can outline four types of events when OS mechanisms can be explicitly identified: internal changes, external changes, internal critical situations, and external critical situations. Other ways to understand OS mechanisms are either read OS specification or talking to OS developers directly, but usually such ways are not available for ordinary users.

Authors of this paper state that the same can be applied to enterprise operating system. Considering EOS, examples of internal changes are: cutting resources, opening or closing a department, adding new responsibilities and more workload, authority delegation. As an example of external change we can take the seasonal cycles of business operation: e.g. flower industry in Russia has high seasons in spring and autumn with huge spikes for the beginning of the school year (the 1st of September), International Women's Day (the 8th of March) and Victory Day (the 9th of May) and lower spike for the St. Valentine's Day (the 14th of February), therefore, flower companies should make changes in their operation, having more resources during spring and autumn and significantly more resources for spikes described above. EOS should control all such changes when adjusting to business operation cycles. Examples of internal critical situations are the large number of workers leaving company unexpectedly, one or more executive officers leaving to competitor company, mass strike of workers. Examples of external critical situations are economic crisis, fire at one of the company's plants, unexpected bankruptcy of main supplier or distributor. In case of EOS, the developers are enterprise owners and managers, and they are the users at the same time, so theoretically it might be possible to extract some EOS mechanisms from conversations with enterprise representatives and from ontological model of the enterprise

In case (and most likely) there are very few real-life examples of enterprise dealing with changes and critical situations recalled in enough detail, it is needed to get required info by discussing synthetic case-studies. The idea is to synthetically model certain situations and explore enterprise behavior in these situations as enterprise representatives expect it to be. Authors propose the following examples of case-studies (the full list is being developed now): cutting 20% of workforce, 3 executive officers unexpectedly leaving to work for competitor company, internal reorganization, opening new department or business unit, closing one department or business unit, bankruptcy of main supplier (or other major failure in supply chain), bankruptcy of main distributor (or other major failure in distribution chain), economic crisis, one of the company plants or offices stop working for 2 days due to force-majeure, major breakdown in email server, and other possible situation depending on the enterprise specifics.

2. Model EOS using defined modelling method. At this stage, enterprise will be made aware of its current EOS.

3. Compare EOS model to reference UEOS model. At this step, it will be possible to identify the gaps in EOS.
4. Propose changes to EOS in order for it to comply with UEOS.
5. The last step would be to help enterprise to implement changes required for EOS to comply with UEOS.

All these should lead to better organization of the enterprise in terms of its structure and processes.

3 Existing work

The notion of EOS is not new in the literature, however, notion of EOS is typically tied to the software implementation. For example, Guerreiro, van Kervel and Babkin define EOS as an Enterprise Information System, which controls the business transactions operation in an organization. [2; 3]. In contrary to that, our understanding of EOS is wider. In this work we follow view of Tribolet [4] on the enterprise as a semantic web of active servers (agents), either silicon based or carbon based, running “internally” their own apps and interacting “externally” through the web, in real-time. As long as enterprise is a web of servers, and servers are interconnected and communicating with each other, we can continue this metaphor and consider that enterprise has its own operating system, which we call EOS.

Our work is also related to the notion of viable system which by definition is a model of the organizational structure of any viable or autonomous system.

VSM considers an enterprise interacting with its environment [5, 13, 14] in two ways:

• Operation – the primary functions where all basic works are being done (like production, distribution, etc.);

• Metasystem - the secondary functions which support all units working together in an integrated way (accounting, scheduling, strategic planning, etc.)

Figure 1 illustrates the basic VSM [13, 14]:

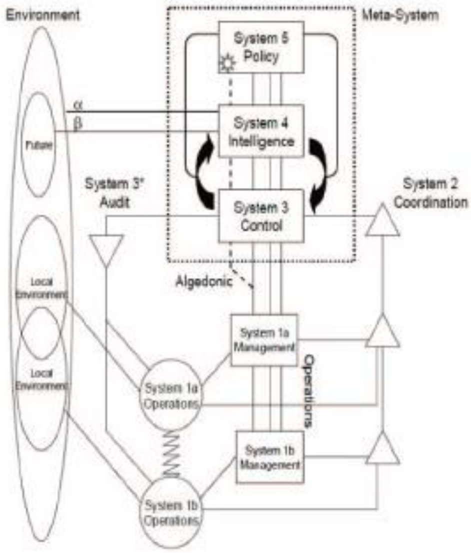

Figure 1. VSM basic model.

VSM has five key systems, one at the Operation and four at the Metasystem (adopted from [1, 5, 13, 14]).

We can find at the Operation:

System 1 – Implementation / Operational Units

Operational units where primary activities are done. The Operational units are responsible for producing the products or services.

We can find at the Metasystem:

System 2 - Co-ordination / Conflict solver

The system responsible for stability/resolving conflict between Operational units and to co-ordinate the interfaces of its value-adding functions and the operations of its primary sub-units.

Examples of the services in a complex organization that may come under System 2 [15]:

- Computer/ICT services
- Documentation
- Purchasing
- Scheduling of common facilities
- Safety and Security
- Tax compliance
- Training in existing practices

System 3 – Control / Optimizer

The system responsible for optimization/generating synergy between Operational units via a two-way communication between the Operation and the Metasystem.

Management accounting, budgeting and production control are typical of functions provided by System 3 [15].

System 3* – Auditing

This systems fulfills the need for an audit channel that can delve into detail without taking over and micromanaging. The financial audit is the most obvious example, but there could be an energy audit, a security audit, an IT compatibility audit, a study of customer complaints and others. Sporadic employee satisfaction surveys and needs analyses are other examples. [15]

Taken together, the management functions of Systems 1, 2, 3 and 3* account for the as-is run time operations of the organization. Note that the only direct connection to the environment exists in the linkage between it and the System 1 operations. Note also that these are functions, not names on an organization chart. It is possible, even likely, that an individual could play a role in delivering a product or a service to a customer and in managing that operation. System 3 often includes representatives from management at System 1 and almost everyone enacts roles in System 2, at least by observing the protocols. [15]

System 4 – Intelligence, Planning, Strategy and Adaptation

The system responsible for the future plans and strategies and adaptation to a changing environment, it implements the two-way link between the Viable System and its external environment getting continuous feedback from the exterior and projecting the organization identity to the exterior.

Recruitment, staff development, benchmarking, participation in trade shows and conferences, market research and lobbying are concerned with learning about and affecting the outside and future. Research and development, strategic planning, borrowing policies and marketing use that knowledge to make internal modifications to be ready for coming changes. [15]

System 5 - Policy

The system responsible to guarantee that the organization works as a whole making the policies and providing clarity about the overall direction, values and purpose of the unit.

VSM scheme can be redrawn in the following way (Figure 2):

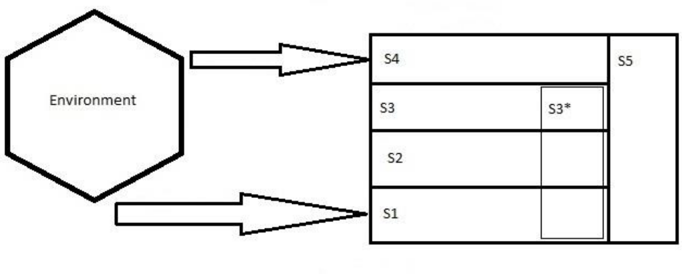

Figure 2. VSM basic model - adaptation.

Arrows mean information gathering flows, they are purposefully unidirectional since we are interested only on gathering information from the environment in order to understand how environment affects enterprise operation. Information gathering flow from System 1 to environment is important for System 1 only, while flow from System 4 is crucial for the whole enterprise.

Since System 5 defines policy for the whole enterprise, it is positioned vertically to touch every other System of VSM.

The theory of distributed systems (in particular distributed operating systems) is applicable to our research, as long as enterprise is a web of servers, and servers are interconnected and communicating with each other [16]. The model of distributed systems (DS) is widely used in many spheres of business, enterprise organization and especially popular in operating systems development. The following principles of the distributed operating system design are described in the literature:

- Transparency – this feature allows DS to transmit data in a transparent way, so user does not notice any difference comparing to the centralized (non-distributed) system;
- Flexibility - providing a large set of higher-level services;
- Reliability - the ability to prevent and recover from errors;
- Efficiency – multifunctional system with the effectively distributed resources;
- Scalability – the system can be scaled-out depending on the current needs.

According to the examples in the literature [16; 17; 18; 19; 20; 21], we can extract the main features of DS:

- System element roles and interconnect network channels are crucial for the system functionality, and therefore from the analysis and synthesis points of view;
- The mechanisms for functions and resources distribution between the elements is typical for a DS;
- Functions sharing and metasystems ensure the fault tolerance - failure of one node does not lead to a full crash of the whole system;
- Parallelism and task-sharing controller - parallel data processing on several nodes simultaneously. This leads to the speeding up the system's work. The program speed increases because of the parallelization, which is explained by Amdahl and Gustafson law. In addition, it helps the DS to support the concurrent work of several users;
- Data and functions duplication. These features are implemented by e.g. the RAID usage for data storage or, for example, having a backup person for each employee, who is capable to cover some functions of the latter and is able to replace the person in critical situation;
- Fault tolerance - DS reliability - multiple nodes' failure insignificantly reduces productivity.

All the characteristics above can be applied to the enterprise, considering it to be a distributed system. Here we can combine computing nodes and human resources, use different interaction models and assets. Within the enterprise, roles and resources can be distributed instead of the computing nodes.

The study of Complex Adaptive Systems (CAS) is also relevant for this research. Enterprise can be considered as complex adaptive system [6, 7, 8, 9], because it complies with the specification of CAS: it consists of agents – employees – which has connectivity and dimensionality, it operates towards self-organization and emergence, it may have non-linear changes and non-random future, it operates in dynamic environment. Authors are now looking deeply into CAS theory to find more relevant ideas for current research. The theory or inquiring systems may also be relevant for this research [10], as well as the notion of Complex Adaptive Inquiring Organization, which, as shown by Kuhn [9], enterprise is.

Control theory is also applicable for this research [11], since the concept of feedback loop is important for adaptation to the environmental changes. The concepts of microgenesis and macrogenesis [6, 7] are relevant because they help to create mechanisms for adaptation and recovery after critical situations. We plan to integrate some parts of Enterprise Dynamic Systems Control developed by Guerreiro [7], and GOD-theory developed by Aveiro [6]. The theory of autopoiesis developed by Varela and Maturana [12] may be also applicable given the nature of the enterprise, but no further exploration has been done yet. The concept or real-time steering of the enterprise, discussed by Pascoa [8], may be relevant for exploration of some adaptation mechanisms like environment monitoring mechanism, control mechanism and planning mechanism. This idea is to be further developed in later research.

4 Current state of the research and future plans

Currently research is in its initial phase. Authors are examining different theories as was highlighted in section 3. Viable systems theory is being examined in more detail as it is considered to be the key part.

Based on the literature review [13, 22; 23; 24; 25; 26; 27; 28], the proposal to extend VSM is being made.

The proposal is to explicitly specify Immune System, Language, Competence, Tools and Sensors/triggers.

1. Immune System

The set of deep crucial mechanisms which can intervene into any System's operation and make changes to Systems in the situation of crisis. On one hand, such mechanisms are created by S3, S4 or S5, but on the other hand they can change parent System.

Examples within enterprise:

- Board of Directors of a company
- Shareholder meeting
- Legal attorney can veto any activity if it contradicts with the law

2. Language

Everything which helps enterprise members (including VSM Systems) to communicate and understand each other. Based on the literature review, it is needed to explicitly specify:

a) Communication channels – ways, format and media for communication. Examples of defined communication channels may be:

- Meeting minutes should be sent via email
- Interview with job applicant should be done either face to face or via phone

b) Shared vocabulary – enterprise (and departments within enterprise) may have certain specific terms for certain specific things. It is highly important that communicating subjects are sharing the same vocabulary.

Examples:
- Specific terms for yearly performance review sessions with managers
- Specific terms for the level of employee on the career ladder (e.g. junior engineer – engineer – senior engineer – principal engineer)

c) Context – understanding of the meaning of the message is dependent on the context. It is important to explicitly define context for certain communication channels and information flows.

Examples of the importance of context:
- Term “BMC” may mean “Baseboard Management Controller” for hardware engineer, while for manager it may mean “Business Model Canvas”

3. Tools

It is important to explicitly specify hardware and software tools which enhance communication by making it faster, more efficient, and/or secure.

Examples:
- Access to corporate email from mobile devices (faster communication)
- VPN connection to corporate network assets (more secure communication)

4. Competence

Literature review shows that individuals are not taken into account when using standard VSM, while people are the essential part of every enterprise. One of the main attribute of employees from employer’s perspective is their competences [29; 30; 31; 32; 33]. Review in [34] shows that competences are not explicitly considered in enterprise engineering field of research as well. The proposal is to add explicit specification of competences to the extended VSM. Using terms from the Four Stages of Competency model [35], the goal is to make enterprise consciously competent about its employees and about itself.

5. Sensors/triggers

We want to replace and extend the notion of homeostat by explicit specification of:
Sensors – mechanisms for information gathering and state monitoring.
Triggers – thresholds for sensor values and reactive actions for thresholds violation.
Importance of sensors is highlighted in [11], where authors state that sensors are crucial for the system to be observable. In order to be controllable, system must be observable and have mechanisms which react on certain sensor outputs – i.e. triggers are necessary. Having sensors and triggers is a mandatory condition to steer the organization.

Example:
Sensor: regular monitoring of governmental legislation for changes which may affect enterprise operation.
Trigger: if there is a new legislation which may affect enterprise operation, legal department must review it and propose respective changes to comply with new legislation.

Comparison between viable systems and distributed systems has been made to understand similarities and extend each other. We try to apply the VS features to the DS to make it more stable and adaptation capable. Let’s project the five VS levels to the DS.

At the level 1 – Operations – we can locate DS objects, nodes, primary computing units. Each unit controls general interactions in accordance with its scope. At this level the system has an ability to interact with external agents in terms of general scope – sales, supply chain, services, etc. and also interconnects with the upper system levels. Level

2 – Coordination – the information flows between nodes with no dependence on the exact object location. It is the information exchange in terms of providing invariants during parallel computation, that can be a domain controller or any additional infrastructure helping to the main system – finance department, HR, legal, etc. This level assumes internal iterations. Level 3 – Optimization – controller nodes initiate services improvement and internal audit (level 3*). That can be arranged as a checksum gathering or periodical state message exchange provided by the managing node. Moving to level 4, we can see that in general meaning the DS doesn't have this level explicitly. Thereafter the level 5 is also not explicitly presented due to the lack of nodes (or agents) in the DS capable to conduct such ontological tasks as strategic planning and policy development. However, these two levels provide the stability and adaptation mechanisms that can extend the DS and make it more sustainable in the dynamically changing environment.

We can model the DS in accordance to the five VSM levels and see how it can help to improve the DS structure. To do that we need to apply level 4 and level 5 concepts to the DS model.

Level 4 is an anticipation of change caused by external environmental disturbances and can be built-in as a strategic planning stage. We need a special interface to communicate with external agents. It may be implemented using a set of sensors, monitoring the external data flow change and getting information from the outer share points and information hubs. For an enterprise, we suggest a practice of feedback collection from the external agents on each level – first of all – customers and any third party agents dealing with the system. This practice may help to understand the reason for any system fluctuation and act accordingly. At this level a controller node gathers data from the external sensors and use it for further planning and strategy development. Data are stored to make an analysis and perform calculations to see current trends and outline the policy changes required for the system adaptation. At this level the responsible agent conducts the strategic planning after the complete data analysis. Strategic planning is an ontological act, so building the levels 4 and 5 of the viable DS we need to include an agent capable for decision-making into the DS structure. Accordingly, DS requires a human being to be integrated to the system, because as we mentioned above, only human being can be capable to deal with ontological tasks due to the responsibility concerns and its exclusive cognitive abilities.

At the level 5 responsible agents implement appropriate changes into the system stricture to improve the system and adapt it to the external changes.

A combination of exercises included in the level 4 and 5 are implied with the certain intervals. Namely, statistics gathering and data monitoring are the regular actions – appropriate actors make an analysis based on the complete and relevant data to be able to react in a timely manner and be prepared to prevent crisis situations. Actions of the level 5 should be applied when the necessity for policy changes occurs, according to the strategic planning outcomes on the level 4.

Thus, applying proposed changes we result in the DS which has the full set of functions and resources distributed between multiple nodes with no dependence on the exact location, at the same time this system as able to adapt to the dynamically changing environment and react to the external signals.

This example shows that the viable DS is more sustainable due to the adaptation mechanism. Getting an interface for gathering information from external environment, DS becomes inseparably related to the external environment, gets an orientation mechanism and receives a viable data. Adding new type of agents responsible for the decision-making to the DS enables the mechanism to operate with the information from the ontological sources.

Currently authors work on the real case in order to specify EOS of one business group within large enterprise. It should give better understanding of which mechanisms are part of EOS and which are not, and it should lead to first draft of UEOS specification, as well as should help to choose good modelling method for EOS. The case is not ready to be published yet.

Future plans include:

1. Study CAS theory, theory of inquiring systems, notion of Complex Adaptive Inquiring Organization, control theory, concepts of microgenesis and macrogenesis, and autopoiesis theory in more detail in order to find the links between them and viable systems theory, and to understand what may be applicable for the notion of EOS.
2. Finishing the real case exploration. The results are planned to be published.
3. Based on the case results, stakeholders' feedback, and further theory exploration, the draft of UEOS specification will be created.
4. Based on the case results and stakeholders feedback, the modelling method for EOS should be chosen. It may be based on DEMO models with some enhancements.
5. Having modelling method and draft of UEOS, this draft will be validated on the second real case. After that some adjustments may be needed for both modelling method and UEOS specification, as well as for the overall methodology.

5 Conclusion

The research is still in its initial phase. Comprehensive review of viable system approach and model has been done, as well as review of complex adaptive systems theory and inquiring systems theory, in addition to initial exploration of other areas such as control theory, concepts of microgenesis, macrogenesis, and autopoiesis. While deeper investigation of mentioned concepts is already planned, it will be valuable to get additional feedback from the enterprise engineering community about which areas to focus on in order to have better picture of EOS concept, and its specification as well as implementation. Deeper review of VSM with proposal to extend it has been done. Comparison between viable systems and distributed systems has been done.

On the positive side, real case is under development now, and valuable insights are expected as a result of this case exploration.

References

[1] Beer S. Brain of the Firm / S. Beer - Allen Lane, The Penguin Press, London, Herder and Herder, USA. 1972

[2] Guerreiro S., van Kervel S., Babkin E. Towards Devising an Architectural Framework for Enterprise Operating Systems. In Proceedings of the 8th International Joint Conference on Software Technologies, pages 578-585.

[3] van Kervel S. Ontology driven Enterprise Information Systems Engineering. SIKS Dissertation series nr. 2012-50. ISBN: 978-90-9027133-0

[4] Tribolet J. An Engineering Approach to Natural Enterprise Dynamics - From Top-down Purposeful Systemic Steering to Bottom-up Adaptive Guidance Control / J. Tribolet - ICEIS 2014 keynote - http://www.iceis.org/KeynoteSpeakers.aspx?y=2014#4

[5] Walker, J. The VSM Guide / J. Walker - http://www.esrad.org.uk/resources/vsmg_3/screen.php?page=1qguide, [available online on the 25th of September 2014]. 2006

[6] Aveiro D. Towards a GOD-theory for organizational engineering: continuously modeling the (re)Generation, Operationalization and Discontinuation of the enterprise. PhD thesis, Instituto Superior Tecnico, 2009.

[7] Guerreiro S. Enterprise Dynamic Systems Control enforcement of run-time business transactions using DEMO: principles of design and implementation. PhD thesis, Instituto Superior Tecnico, 2012.

[8] Páscoa C. Organizational and Design Engineering of the Operational and Support Dimensions of an Organization: The Portuguese Air Force Case Study / C. Pascoa – PhD thesis, Instituto Superior Técnico, Lisbon, Portugal. 2012.

[9] Kuhn J. A Theory Of Complex Adaptive Inquiring Organizations: Application To Continuous Assurance Of Corporate Financial Information. PhD dissertation, University of Central Florida. Orlando, Florida, USA, 2009.

[10] Churchman C.W. The Design of Inquiring Systems: Basic Concepts of Systems and Organization. Basic Books, New York. 1971

[11] Abraham R., Tribolet J., Winter R. Transformation of multi-level systems – theoretical grounding and consequences for enterprise architecture management. Advances in Enterprise Engineering VII. Lecture Notes in Business Information Processing Volume 146, 2013, pp 73-87

[12] H.R. Maturana, F.J. Varela (1980). "The cognitive process". Autopoiesis and cognition: The realization of the living. Springer Science & Business Media. p. 13. ISBN 978-9-027-71016-1.

[13] Zadeh M.E., Lewis E., Millar G., Yang Y., Thorne C. The Use of Viable System Model to Develop Guidelines for Generating Enterprise Architecture Principles. 2014 IEEE International Conference on Systems, Man, and Cybernetics October 5-8, 2014, San Diego, CA, USA

[14] Millar G. The Viable Governance Model: A Theoretical Model of IT Governance within a Corporate Setting. DIT Unpublished doctoral dissertation, University of New South Wales, Canberra, 2009.

[15] Leonard A. The Viable System Model and Its Application to Complex Organizations. Syst Pract Action Res (2009) 22:223–233

[16] Kiang M. Y., Chi R. T., Tam K. Y., “A Distributed Knowledge Acquisition System in a DSS”, Journal of Management Information Systems, Vol. 9, No. 4 (Spring, 1993), pp. 59-82
[17] Lenzen C., Wattenhofer R., “Distributed algorithms for sensor networks” Philosophical Transactions: Mathematical, Physical and Engineering Sciences, Vol. 370, No 1958, Sensor network algorithms and applications (13 January 2012), pp. 11-26.
[18] Tan G., Feng S., Sun N. Parallel multiple sequences alignment in SMP cluster //High-Performance Computing in Asia-Pacific Region, 2005. Proceedings. Eighth International Conference on. – IEEE, 2005. – C. 6 pp.-431.
[19] Liu Sheng O. R. “Analysis of Optimal File Migration Policies in Distributed Computer Systems”, INFORMS, Management Science, Vol. 38, No. 4 (Apr., 1992), pp. 459-482
[20] Giere R. N. “The Role of Agency in Distributed Cognitive Systems”, Philosophy of Science, Vol. 73, No. 5, (December 2006), pp. 710-719
[21] Dietz J. L. G. Enterprise Ontology: Theory and Methodology. B., Heidelberg, N. Y.: Springer, 2006 – ISBN-10 3-540-29169-5.
[22] Espejo R., Reyes A. Organizational Systems. Managing Complexity with the Viable System Model. Springer-Verlag Berlin Heidelberg 2011. ISBN 978-3-642-19108-4
[23] Perez Rios, J. Design and Diagnosis for Sustainable Organizations. The Viable System Method. Springer-Verlag Berlin Heidelberg 2012. ISBN 978-3-642-22317-4
[24] Hildbrand S., Bodhanya S. The potential value of the Viable System Model as a managerial tool. Management Dynamics Volume 22 No. 2, 2013
[25] Nechansky H. Issues of organizational cybernetics and viability beyond Beer’s viable systems model. International Journal of General Systems, 2013 Vol. 42, No. 8, 838–859
[26] P. Checkland, “Are Organisations Machines?” Futures, vol. 12, pp. 421- 424, 1980,.
[27] W. Ulrich, “A critique of pure cybernetic reason: the Chilean experience with cybernetics”, Journal of Applied Systems Analysis, vol. 8, 1981, pp. 33-59.
[28] R. L. Flood, and E. R. Carson, Dealing with complexity: an introduction to the theory and application of systems science (2nd ed.). New York: Plenum Press. 1993.
[29] Munkvold B. E., Hustad E., "IT-Supported Competence Management: A Case Study at Ericsson," Information Systems Management, 2005, pp. 78-88.
[30] Nordhaug O., Human Capital in Organizations. Scandinavian University Press, 1993.
[31] Hachicha R.M., Dafaoui E.M., Mhamedi A.E.. Competence evaluation approach based on 2-tuple linguistic representation model. 978-1-4244-3672-9/09/$25.00 ©2009 IEEE
[32] Czelusniak, J., Abreu, A., Dergint D., Hatakeyama, K. 2010. Proposal of Agent’s Software for Support Competence Management Process. PICMET’10 Conference.
[33] Sanghi S. The Handbook of Competence Mapping: understanding, designing and implementing competency models in organizations. Sage Publications, 2004.

[34] Sergeev A., Babkin E. Towards Competence-Based Enterprise Restructuring Using Ontologies. / D. Aveiro et al. (Eds.): EEWC 2015, LNBIP 211, pp. 34–46, 2015.
[35] Stephens J., Haslett T. A Set of Conventions, a Model: An Application of Stafford Beer's Viable Systems Model to the Strategic Planning Process. Syst Pract Action Res (2011) 24:429–452