Decoupling Modeling from Complexity Based on a Freedom-to-Act Architecture

Udo Inden
Cologne University of Applied Sciences
Research Centre for Applications of Intelligent Systems (CAIS),
Cologne, Germany
E-Mail: udo.inden@fh-koeln.de

Sergej Naimark
Controlling Chaos Technologies GmbH
Hannover, Germany
E-Mail: s.naimark@controlchaostech.de

Claus-Peter Rückemann
Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster (WWU) / North-German Supercomputing Alliance (HLRN),
Münster, Hannover, Germany
E-mail: ruckema@uni-muenster.de

Abstract—**Focusing on business systems we discuss that self-organizing, increasing operations' complexity is escaping from capabilities of modeling. This urges to decouple modeling from emergent domain complexity as well as management to eventually overcome control illusions inherent to conventional approaches. In a first step; we show that, on some conditions, agent-based modeling and multi-agent architectures are able of solving the problem. Conditions include aspects of service-based modeling, of semantic modeling and of freedom-based agent's behavior. The concepts are relevant for design, management and computing of highly complex operations.**

Keywords-complexity of modeling; agent-based modeling; freedom-to-act architecture; service-oriented architecture; semantic modeling.

I. Introduction to the Problem

Models reduce complexity of a real-world domain with regard to a particular purpose of control. Concepts in this paper are based on experience in highly complex environments, where struggling modeling complexity is part of daily business in strategic, tactical or real-time management. The development of operations complexity shows that improvements of modeling techniques did not ease this job because both, the complexity of operations and of modeling are coupled self-organizing developments.

From experiment, we learned that agent-based modeling (ABM) [37] and multi-agent technology (MAT) [29] [30] offer options of decoupling modeling from domain complexity. We argue that it is necessary if control about complex systems is to be maintained. ABM or MAT are well known. In spite of their advantages in handling complexity, they failed joining the mainstream of Information and Communication Technologies.

The reason is that managements and software developers consider emergent behavior to be an intimate enemy of the control of systems. In contrary, we argue that emergent behavior is the last resort of control of complex systems. Namely, we suggest changing the view at the control of complex systems which by principle cannot be reduced to models. The term control illusion [33] illustrates the difference between the views.

The following examples from aviation industry provide insight into the complexity operations' systems:

Airlines plan the service of aircrafts as sequences of flights executed in a period of time. The cost-efficiency of these '*rotations*' and the service quality they deliver to customers depend on a manifold of interacting factors spanning air- and ground operations and involving thousands of aircrafts and flights of hundreds of airlines or numerous supporting services. Well established international proceedings, synchronize flight plans global networks. Before a flight this abstract plan is to be particularized and confirmed.

But in execution, plans are troubled by a constant floor of interference which easily can get out of control [1]. The challenge lies in the continuous process of adapting to reality by correcting, mitigating or recovering active plans. In large cases ten thousands of autonomous actors and legacy systems are to be re-synchronized in almost real-time – repeatedly because the solution of the next problem may affect the solution of previous ones.

The 'time-to-volume' of industrial series production begins with the start of development of products and production processes and ends with reaching stable output as planned for amortization of invest. But, there are trade-offs, e.g., more engineering time drives costs and postpones the product launch, while less effort drives quality risk emerging from *butterfly effects* (non-linear behavior) or from *black swans* (long-tail risks).

In the cases of B787 or A380 such effects delayed ramp-up and planned volumes for years [2] [3]. For ramp-up, Boeing implemented a *virtual ramp-up system* (VRS, a simulator and planner) covering major stakeholders and components in the supply-chain – except the *fasteners* for carbon-fiber parts. Apparently, these tiny parts were ignored to limit model complexity. The supplier failed and fasteners became a problem. As example of *black swans* [5] the uncontained engine failure of a Trent 900 engine in Quantas A380, flight QF32, November 4th 2010 [4], may serve.

In airliner as in airframer business problems increase: Air traffic, thus the number of aircrafts and flights, is expected to more than double by 2020 while in America or Europe air- and ground infrastructures are lacking behind [34]. Airlines' competition gets harder and new competitors of Boeing and Airbus appear in emerging markets. Solutions are asked to increase service capabilities, reduce costs or CO_2 footprints, and get to volume faster. These examples suggest that architectures need to support, respectively, adapt to

 ISBN: 978-1-61208-226-4

- the pace of change, driven by competition that defines opportunity windows to adapt to change or affects trade-offs between costs and quality of modeling.
- the need of catching up with ignored or not captured aspects in terms of butterflies or black swans.
- the fact that more and more details matter (resolution of object) and become source or target of events (resolution of time: more events per unit of time) [6].

These aspects refer to the *law of requisite variety* saying that control fails if controllers' complexity does not match the complexity of the system to be controlled [15]. In result of developments sketched above models tend to fail delivering their core service that is reducing complexity with regard to a particular purpose. In consequence complexity of modeling is to be decoupled from complexity of domains.

This paper is organized as follows: Section two shows how mainstream architectures relate to complexity and its increase. In Section three, basic principles of the alternative freedom-to-act architecture are analyzed. After a summary, Section fife sketches selected aspects of further work.

II. Complexity in Function-, Process-, and Service-oriented Modeling Architectures

In the mainstream, different regimes of operations management and modeling developed: function-, process-, and service-orientation (with internet-based automation). Coming from its history in knowledge management also semantic modeling is about to enter the mainstream.

A. Encapsulating Complexity in Organizational Silos: The Function-oriented Regime

Modeling as effort and tool of operations management is initially bound to the work of Taylor [7] and operations strategies of Ford [8]. This thinking uses functional specialization to grow expertise of engineers or administrators (knowledge workers) and to compensate the lack of educated workers for the assembly lines by extreme simplification.

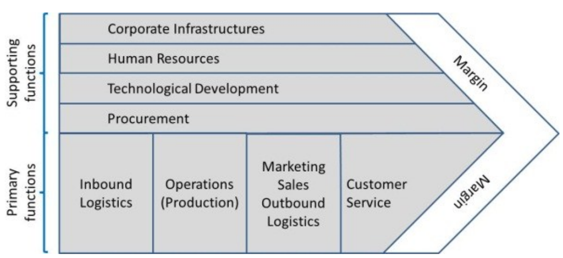

Figure 1. The Value Chain.

Figure 1 depicts a model of the '*value chain*', a later concept introduced by Porter [9]. It shows that value is produced by specialized departments differentiated in those directly contributing to the creation of value (lower group) and supporting ones related to a firm's infrastructures (facilities…, later also IT) or administration (upper group).

The picture also has a financial interpretation: its expanse represents the financial flow passing the organization. Reading the grey block as costs spent for direct and indirect operations, hopefully a positive difference to revenue is left, a profit: the spike of the arrow. In this light function-orientation obviously creates self-referential silo-behavior: if resources are to be distributed (more staff) or costs to be cut (the contrary) routinely competition appears: Who is more important, who to be blamed for failure? Also, careers are built on affiliation to silos. Accordingly, the value creating process is marked by silo-driven discontinuity turning into high coordination effort and long processing-times.

B. Tackling the Complexity of Interactions The Process-oriented Regime

In functional models, business processes are implicit. In order to overcome disadvantages they had to be made explicit. The motivation was induced by competition, again starting from car industry: When Taiichi Ohno, CTO of Toyota, visited Ford, he learned that silos or radical simplification of assembly jobs produce problems rather than solutions: High simplification turns into a waste of talent and silos into self-inflicted complexity [10]. Particularly, they obstruct the view at real challenges: effectiveness (value delivered) and efficiency (costs) of operations.

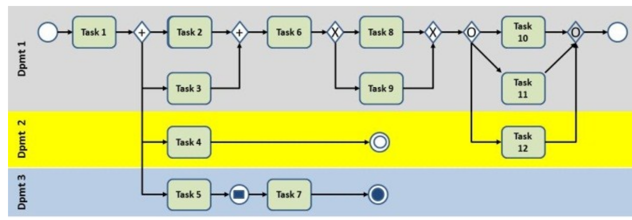

Figure 2. Business Process Modeling Notation.

Instead, Ohno invented strategies avoiding redundancy (*muda*) like *Just in Time* delivery replacing inventories or continuous improvement (*kaizen*) exploiting knowledge at *all* workplaces. Later these ideas were united as *Lean Management*. They aim at unobstructed flows of orders, material, or information as well as purposeful collaboration across fields of tasks and networks of suppliers. Workers, formerly just repeating simple jobs, became autonomous and creative partakers in implementing and improving operations. Knowledge and intelligence became strategic resources.

An MIT study [10], uncovering advantages of Lean Management, became a wake-up call to established car makers. Finally, the new strategy changed the rules of competition and formed a new *fitness landscape* [27]. Initially, managers in the USA or Europe misunderstood the call to become lean as a call for cutting cost. But, in fact, it was a call to change minds towards integrative thinking and modeling. Rather than optimizing functionality in the silos, the new heading geared towards the efficiency of functions linked across business processes and improving value propositions – a far more complex job than managing silos.

Organizations had to learn how to create new value from this. Self-inflicted complexity was to be exchanged by value-driven (money-making) complexity. It asked taking care for interdependencies beyond boxes and to enrich responsibility of knowledge- and of assembly-line workers: Managerial excellence is marked by capabilities of model-literacy and self-management on *all* hierarchical levels [11].

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-226-4

So, models of functions and processes became representations of complex interdependent activity. In parallel, ICT became a driver of model complexity since automation asked elaborating models with high precision and detail. Figure 2 depicts a diagram accordingly to the BPMN (Business Process Modeling Notation) [35], an advanced standard, capturing organizational structures in horizontal lanes (remains of the silos) as well as structures and rules of proceedings and interaction (connectors or auxiliary information).

It answers the question: Who does what (why), where, when, how and with what and whom? Functions now are embedded as physical or intellectual resources [12]. With growing vertical (along managerial hierarchy) and horizontal (same level of activity) integration of automation current ICT covers the value chain depicted in Figure 1, while process-cost accounting [13] enabled new strategies of controlling economics of managing operations complexity.

But, scale and complexity of models grew (consider the complexity of operations landscapes drafted in the introduction) and in accordance to Ashby's Law of requisite variety [14], richness and heterogeneity of detail or the manifold of interfaces did not turn into economies of scale but became drivers of costs and risks to run out of budgets. Complexity started escaping from modeling capabilities.

C. Modularizing and Encapsulating Functionality The Regime of Services

There are solutions to this problem: modularization and virtualization. Modules with standardized interfaces reduce variety and hide details of the functionality they provide. So, challenged by faster change and increasing demandingness in markets as well as by increasing costs, automotive industry introduced *platforms* that decouple the variety of car-models from the variety of parts they are built from.

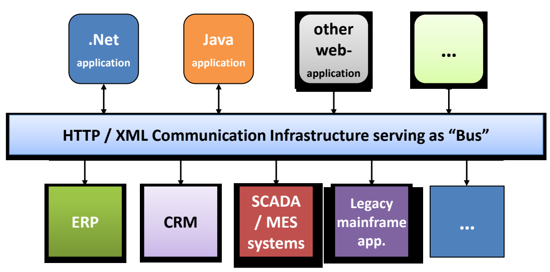

Figure 3. SOA-based Application Architecture.

This concept of *functionality* provided by a *module* also was adopted by ICT in service-oriented architectures (SOA) for services collaborating via the Internet (Figure 3). Promises reach from risk reduction or portability and re-usability in software development to integrated, intelligent operations like in the vision of the *Internet of Things and Services* [15] [16]. And, as we shall see later, in a general view, everything can be seen as provider or demander of service.

Based on service description standards [17], web services are capable of autonomously composing complex dynamic networks involving things (in the simplest case via radio tags, in future also enabled by embedded multi-core computing capacity), legacy systems as well as human users in the roles of operations' supervisors or IT operators.

However, due to latency times or high customization, legacy IT is hard to be integrated into service landscapes or to be decomposed into sub-services [18]. The competition of big players (IBM, Oracle, SAP, etc.) in direction will fragment web-standards. Or, the vulnerability of Internet-based services and dependencies on intermediaries raise security concerns. Yet, in spite of achievements, also SOA does not sustainably reduce the complexity of models [19] [20].

To not get lost in variety Volkswagen implemented a new platform, the *Modular Transverse Matrix*. It reduces the number of modules by up to 90 % across 10 brands (Seat, VW … Audi, Porsche, Bentley) [21]. However, it is hard to believe that this is applicable to examples in aviation industry. And, by and by, markets also will coerce VW to accept and accommodate new variety. Accordingly, complexity will reconquer modularized operations and related models or applications' architectures [22] [23].

D. Introducing Meaning The Regime of Semantic Modeling and Ontologies

Adopting Internet-based communication and cooperation semantic models became relevant: Languages are sets of tools to create meaning of signs or symbols in communities. So far, however, technology only can interpret sequences of signs obeying syntactic rules, e.g., as command to delete a text. There is no understanding of meaning.

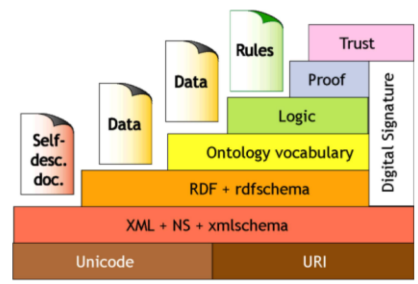

Figure 4. The Semantic Layer in the W3 Architecture [24].

The idea of the semantic web [25] is to encode semantic information into web-pages. In the W3-architecture of Berners-Lee (Figure 4) [24] ontologies provide the *vocabulary*, more generally, the knowledge enabling to recognise and associate services. The common vocabulary enables modelling relations to other objects and properties relevant to relations.

Reading meaning, a sequence of signs would not only be just a code for deterministic execution of commands, but for understanding *know-why* and context, i.e., *the content of the service it provides*: Which idea is leading the search? What is a function about? Does it fit into a particular scene? What is the meaning of or responses to events in contradictive context (costs, quality, security, operations' footprints)?

Ontologies, among others, enable capturing local knowledge about objects related to operations like orders or resources. Given that these local models are globally consistent (non contradictive), complete (no relevant aspects missing) and clear they conceptually and practically support integrating large and massively distributed operations' systems [26] and a designer of a web service can describe a service without knowing other services.

 ISBN: 978-1-61208-226-4

On the other side, semantic interoperability needs a non-trivial degree of accuracy of the model. It is obvious that this problem increases with the complexity to be captured. Although semantic modeling is a great step for itself, it does not decouple from complexity.

III. Implicit Aspects of Decoupling: Towards Freedom-to-Act-Architectures

A. Decoupling Modeling from Domain Complexity

The example of Boeing's or Airbus ramp-ups shows that the approach of modeling domain complexity forces a trade-off between simplification and risk that butterflies, long-tail risks can materialize (*black swans*) or open events may affect operations (*openness to environments*) each speedily grow to a 7-, 8-, or even 9-digit € problem. In a complex domain simplification is evidence that this complexity is out of control of modeling and management. It is a bad bet. Decoupling by agent-based modeling (ABM) [37] is the alternative.

A first step is realizing that complexity is a property of dynamic systems [29] [30]. It emerges from degrees of freedom of interacting elements (or agents) the system consists of. These agents may act autonomously (executing individual decisions for achieving individual objectives with individual resources like humans or *things that think*) or not (simple things just connected by sensing technology).

Decoupling relies on a division of labor amid ABM and Multi-agent Systems (MAS). Instead of modeling operations' complexity, ABM focuses on the behavior of agents and the framework of interaction complexity emerges from. The knowledge required is in the relationships and properties of objects or in target functions, rules and protocols of agents' communication. It can be captured in ontologies as local knowledge from or also directly modeled by the people working in the domain. Software-agents can read ontologies and MAS organize interaction. Thus, complexity is not in the model, but emerges in the MAS. Since ABM is scalable to resolution of objects [5] or open to any new object, it also enables capturing butterfly effects or black swans.

Occasionally, as shown by the sequence of paradigms discussed in Section II, there also is a revolutionary and irreversible change of sets and settings of agents and of patterns of interaction. So, the success of the Lean Regime does not allow returning to Ford's strategies. This *evolutionary aspect* of operations' systems is explained by the theory of *fitness landscapes* [27] as discriminating change of control requirements and capacities of self-reproducing (autopoietic) systems [28]. But, although it may require a thoroughly review of models, it still can be handled by principles of ABM.

B. The Definition of Freedom-to-Act (FTA) Architectures

Dynamics of systems formed by agents can be described as the change of states of agents. These states are results of previous activity and, at the same time, resources of subsequent action; they imply degrees of freedom future states emerge from. This view we consider to be relevant for effectively modeling and managing complex dynamic systems.

Since the behavior of complex systems hardly is reducible to models and to be predicted, they ask to adapt to unexpected, thus unplanned change. Therefore, FTA are the most decisive resource or control parameter of managing complex systems and the viability of any solution depends on the physical (feasibility of the solution) and the legal (legitimacy) availability or exploitability of FTA.

For example, agents representing 'busy' service trucks in airport operations will not respond to another order, except there are FTA allowing to shift current jobs in time or to transfer them to other agents. Agents representing parts to be supplied to an assembly line may have the state 'delayed' and ask for mitigating action, i.e., other agents to take action in reach of *their* FTA. Or, due to a satisfying crash-test of new material, agents representing the respective objects in engineering may release contingency budget by reducing the value of the event risk "readiness for manufacturing delayed" and by this release the FTA of other agents.

C. Lean Modeling, Service-orientation and Semantic modesty

In agent-based modeling, ontologies do not only serve as dictionaries but primarily as frameworks of agents' behavior. They need capturing dimensions of acting like space (aircraft negotiate new routes) or organizational affiliation of staff (experts may be sent for supporting suppliers).

Models also need support sensitivity to events (aircraft passed control-point; CF-component is ready for manufacturing) as well to context in terms of domains of operations (engineering or administration) or to possibly contradicting objectives (reducing costs versus reducing environmental footprints). Also, connectivity to sensors or other sources is required.

Undisputedly, semantic models can be very complex and under conditions of fast change *the challenge is keeping ontologies complete and consistent in order to enable MAS of continuously providing viable solutions in spite of frequent updates*. One problem may be the *manifold of relationships between objects*, another one the *sophistication of semantics*. To prevent complexity from returning through backdoors, we therefore, suggest two principles of modeling: service-orientation and semantic modesty.

As discussed, *service-oriented architectures* support modularization and encapsulation of functionality. Mainly they are used for modeling web-services. *The same strategy can be applied in ontologies by organizing models of agents' behavior in a service-oriented way*. On that base, agents form dynamic networks by offering and consuming services on virtual markets which allow controlling activity by cost, price, or margins [30].

Service-oriented modeling can significantly contribute to the clarity of ontologies since *any* relation may be modeled as a service-relationship, since FTA can be understood as capabilities or needs to provide or consume a service and since any property of agents can be arranged accordingly. For example, a truck may offer transport services and demand services of gas stations – both substantiated by FTA in terms of maximum payload or level of fuel. Drilling deeper, payload is a service offered by structural components of the truck …

This concept also works in less usual cases. So, the relation *table "has" legs* (*leg "is part of" table*) translates into:

 ISBN: 978-1-61208-226-4

"*Legs provide the service of keeping distance from ground by 72 cm*" and into sequences like: "*user agent asks for table of heights of 72 cm*" → "*table agent asks …*". This example from ergonomics and logistics of modularized office furniture system in a large organization may look strange, but it works and provides clarity.

Semantic sophistication is about risks of modeling complexity *into* ontologies. Certainly, there are applications that cannot avoid this problem like a semantic search engine that may have to discern the meaning of "time" in philosophical, physical, economical, or sociological contexts. The problem increases with distance from models in natural sciences, engineering or direct business operations. In the latter 'time' is the clock-time planned or elapsed between events. Already in more abstract economic contexts like *time to volume* (achieving the crestline of production) and *time to amortization* (progress of effective sales) the meaning complicates. New managerial regimes introduce new concepts of time, like 'synchronization' which is highly relevant to lean management but far less for Henry Ford's functional silos.

Consequently, by *limiting the variety of relations* and being *modest* in terms of semantic complexity the global consistency of local modeling is significantly easier to validate. Like in Lean Management, it abandons self-inflicted complexity and compares to the VW Matrix which at least for the time being abandoned 90% of variance.

Semantic, service-oriented agent-based modeling also is *scalable to increasing resolution* of object and time [5]. If another 'butterfly' or 'black swan' is identified to affect the behavior of the system it can be added as another agent providing and consuming services as well as owning the properties related to these services. And subsequently the MAS will process the model.

D. Processing Freedom-to-Act Architectures and the Example of Ontology-based Multi-Agent Systems

Why don't wheeled animals exist? Because wheels do not provide animals with FTA required sustaining in their environments. Life is the machine producing survival or extinction by processing real animals' FTA. Alike, the leanness of firms is processed by markets returning profit or loss.

In real-time business operations, FTA are explicitly processed in a mode inofficially called *improvisation*. Experienced, focused and observing dispatchers or operators at any time know the FTA of "their" resources. If one fails in a tight situation they know alternatives that can serve for mitigation or remedy (compare idle slots in Figure 5). The same works in MAS: FTA-awareness of dispatchers is replaced by collaborating agents, each by definition knowing its FTA and the proceeding of exploiting them at the best.

Multi-agent systems are *the* model of software for processing FTA, whether they are formed by software agents acting in the local memory of a server or across a grid of servers, by agents acting in the Internet (web-services, things like aircraft2aircraft [34] or car2car communication in future traffic management scenarios [36]) or in collaboration with human analysts, planners or deciders.

In MAS, agents process semantic models of relations, properties or other aspects of the framework like objectives, metrics, negotiation or reporting protocols that shape agents' behavior. With some simplification, relations between agents compare to relations between actors playing a role and ontologies compare to the role scripts. Performance's quality relies on both, the quality of scripts and the talent of actors comparing to the quality of code and properties of agents.

Instead of fighting complexity, the peer2peer architecture of MAS take advantage of it by exploiting FTA, i.e., disposable capacity. If the behavior of the system at least is statistically predictable, an optimal, e.g. cost minimizing plan can be delivered that, except in terms of contingency buffers, avoids idleness of resources (*muda*). Disposable capacity is avoided and no exploitable FTA are left.

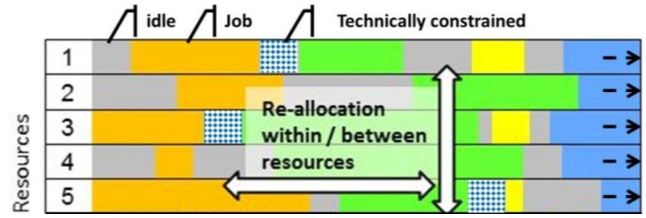

Figure 5. Gantt Diagram with Jobs and Idle Time.

The Gantt Diagram shown in Figure 5 displays the allocation of resources to jobs as well as idle slots which an optimal plan tries to avoid. But, in another view, these idle slots also represent a part of the relevant spectrum of FTA: disposable time, a major resource of adapting to unplanned events. Again, there are counter-intuitive aspects:

- On one hand, objectives of optimization are incompatible with the unpredictability of complex systems since the validity and viability of any optimum relies on the reliability of underlying assumptions.
- On the other hand, muda to be avoided accordingly to principles of optimization is the major resource of adaptiveness to unexpected events. Thus, principles of optimization may detract resources required to adapt to unexpected events.

These problems significantly increase with complexity: As in reality, FTA (agent's disposability) are floating in a non predictable way. And the more complex, thus uncertain the scene the more likely there is a need for redundancy that may serve as resource to a solution.

IV. SUMMARY

In the light of Ashby's Law [14] '*competing'* can be conceived as effort to achieve, maintain or increase control in a competitive environment, i.e., to dispose of freedoms to act competitors cannot control. Therefore, competition in direction drives complexity. That way Ford's mass-production system (standing for all car-makers of this type) with its almost strict decomposition of work exceeded complexity of car-by-car garage production.

Then, overcoming self-inflicted complexity of silos, Toyota invented lean manufacturing systems, again more complex than the regime they attacked. Consider demands of strategies like Kaizen or Just-in-Time (JIT) for process-spanning thinking, model literacy or continuous learning on all hierarchical levels: Lean management needs, employs and develops organizational and collaborative intelligence.

Since competition does not stop driving complexity, platform strategies answered in operations as in modeling (SOA) or ICT, among others for automation of operations (web services). The Internet of things and services or the semantic web are next steps.

Modeling architectures are running after this still accelerating development. To maintain chances of effective modeling it becomes paramount decoupling them from domain complexity by agent-based as well as lean concepts. Multi-agent systems are able of processing these models and exploiting FTA in response to unexpected events.

V. Selected Topics of Future Work

The view at FTA should serve as bridge between traditional and future handling of complexity. Among others, MAS, the reference used here, likely failed joining the mainstream because of a lack of such bridges in the thinking of users and developers [31]. In the following, selected aspects of further proceeding will be sketched.

A. The Acceptance Problem: Trust in a Black-Box?

There are serious technical problems and security concerns which finally may find acceptable solutions. But, the most fundamental problem is not about technology or automation but about organizational integration.

In business environments, conceiving and tackling complexity as a resource commonly interferes with conventional concepts of governance and management: Still the ideas of calculability and optimizability (full control) prevails and not principles of continuous and potentially experimental adaptation and approximation. Redundancy is not considered to be a resource.

With major EU industry, we currently are developing strategies on integrated risk management across large-scale operations landscapes. The approach is driven by competition and complexity that reduces time windows to be passed for amortization of capital-intensive projects and hardly leaving alternatives to new, ABM- and FTA-related strategies.

B. Improving Modeling and Auto-code Technology

There are ontology editors and debugging tools available and the strategy of reducing semantic complexity has the potential of substantially simplify the development of ontologies. But, there is still a long way from ontology to effective code. As an example, let us take the fasteners neglected in Boeing's Virtual Ramp-up System. Including such aspects at a later point of time, may need implementing new classes of objects providing respectively asking new services.

On their own resources, users may easily implement new classes of objects and services into ontologies. Implementing it into a multi-agent system requires software developers. Therefore, strategies, architectures and tools have to be elaborated enabling users (more) directly, at the best without developer support, implementing new classes objects into processing systems (e.g., MAS).

We call this objective "WYKIWYG" – what you know is what you get. It may have an impact in terms of model driven software engineering or possible automated code generation and with this increase acceptance of users.

C. Criticality Management, an Application Example

Criticality is a control parameter of complex system defined as the scale-free point of a phase transition, e.g., from liquid to solid, from stable to unstable (a pile of sand) or from able to unable of response to unexpected events, disposing or not of respective FTA. To effectively manage criticality it is to be estimated.

As a parameter of complexity, also criticality emerges in operations, and is to be modeled as global property (like the temperature of a solid object). Air traffic or manufacturing systems are dynamic networks formed by large numbers of agents. In simulation or real operations' control respective FTA can be logged. Stochastic models could be explored to estimate criticality and working it up for management. These aspects are very closely related to research on risk management mentioned.

D. Convergence with HPC-Problems

There is serious indication that modeling and computing of business operations' on one and domains and issues of High Performance or High End Computing (HPC, HEC) on the other side converge. Operations' models may be smaller as, e.g., climate models.

But, in terms of distribution and non-linearity these applications compare to typical HPC applications. In terms of the variety of agents (including autonomous ones) and the manifold of parameters they may even be more complex. In both domains, FTA are a relevant concept. Reversely, HPC addresses real-time collaboration and learning in a way comparable to a project on intelligent manufacturing ramp-up, we shall start end of 2012 [32] [5].

References

[1] Inden, U., Tieck, St., and Rückemann, C.-P. (2011). Rotation-oriented Collaborative Air Traffic Management. In: C.-P. Rückemann, W. Christmann, S. Saini, & M. Pankowska, (Eds.), Proceedings of The First International Conference on Advanced Communications and Computation, INFOCOMP, October 23-29 2011. Pp. 25-30, ISBN: 978-1-61208-161-8.

[2] Aero International (2009). Boeing 787 – Der schwierige Weg zum Erstflug. Aero International 2009, Vol. 4. Pp. 40-43.

[3] Kinsley-Jones, M. (2010). Giant steps – Has the latest Airbus A380 production ramp worked. Flightglobal, July 12th 2010, Retr. Feb. 13th 2011. http://www.flightglobal.com/news/articles/farnborough-giant-steps-has-the-latest-airbus-a380-production-revamp-343814/.

[4] Ostrower, J. (2010). A380 fleet grounded following Trent 900 failure. Flightglobal Blog, Nov. 4th 2010. Retr. May 1st 2012, http://www.flightglobal.com/blogs/flightblogger/2010/11/qantas-a380-fleet-grounded.html.

[5] Taleb, N. N. (2007). Black Swans and the Domains of Statistic. The American Statistician Association, Vol. 61, Issue 3, 2007. DOI: 10.1198/000313007X219996. Pp. 1-3.

[6] Müller-Stewens, und G., Fleisch, E. (2008). High-Resolution-Management: Konsequenzen des Internet der Dinge auf die Unternehmensführung. Führung & Organisation 77 (5), Pp. 272-281.

[7] Taylor, F. W. (1911). Principles of Scientific Management. New York and London, Harper & brothers.

[8] Hughes, T. P. (2004). American Genesis: A Century of Invention and Technological Enthusiasm, 1870-1970. The University of Chicago Press. IBN: 0-226-35927-1.

[9] Porter, M. E. (1985). Competitive Advantage: Creating and Sustaining superior Performance. The Free Press, New York (1998). ISBN 0-684-84146-0.

[10] Womack, J., Jones, D., and Roos, D.: The Machine that changed the World – The Story of Lean Production. Harper Collins, New York 1990, ISBN 978-0-06-097417-6.

[11] Ohno, Taiichi (1995), Toyota Production System: Beyond Large-scale Production, Productivity Press Inc., ISBN 0-915299-14-3.

[12] BPMN standard: http://www.bpmn.org/. Retr. Mai 12th 2012

[13] Mayer, R. (1998). Kapazitätskostenrechnung: Prozesskostenrechnung, Lösungsansatz für indirekte Leistungsbereiche, Vahlen, München. ISBN 3-8006-2366-8.

[14] Ashby W.R. (1956). An Introduction to Cybernetics. London, U.K.: Chapman & Hall Ltd. Pp. 206–212.

[15] Ten Hompel, M. et al. (2008). Künstliche Intelligenz im Internet der Dinge: Die Zukunft der Materialflusssteuerung mit autonomen Agenten. Jahrbuch der Logistik 2008. Pp. 24-29.

[16] Karnouskos, S., Savio, D., Spiess, P., Guinard D., and Trifa V., Baecker O. (2010). Real-world Service Interaction with Enterprise Systems in Dynamic Manufacturing Environments. In: Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management. ISBN 978-1-84996-118-9, Springer, Pp. 423–457. Retr. Sept. 22nd 11.

[17] UDDI standard: https://www.oasis-open.org/committees/tc_home.php? wg_abbrev=uddi-spec. Retrieved May 12th 2012

[18] Weiss, O. (2001). ERP-Systeme müssen flexibler werden. Computerwelt Sept. 9th 2011. Retr. October 17th 2011. http//www.computerwelt.at/?id=251&tx_ttnews[tt_news]=45815.

[19] den Haan, J. (2007). Model-Driven SOA. Sept. 13th 2007. http://www.theenterprisearchitect.eu/archive/2007/09/13/model-driven-soa. Retr. Jan. 8th 2012.

[20] de Groot, R. (2008). Top 10 SOA Pitfalls: SOA does not solve complexity automatically. May 19th 2008. http://blog.xebia.com/2008/05/19/top-10-soa-pitfalls-6-soa-does-not-solve-complexity-automatically/. Retr. Jan. 8th 2012.

[21] VW (2012). Volkswagen introduces Modular Transverse Matrix. Retr. Feb. 12th 2012. http://www.volkswagenag.com/content/vwcorp/info_center/en/themes/2012/02/MQB.html.

[22] Muhammad, S.S., Myers, D., and Sanchez C.O. (2011). Complexity Analysis at Design Stages of Service Oriented Architectures as a Measure of Reliability Risks. In: Milanovic N., Engineering Reliable Service Oriented Architecture: IG Global. ISBN13: 9781609604936. Pp. 292-314.

[23] Tran, H., Zdun, U., and Dustdar, S. (2007) View-based and Model-driven Approach for Reducing the Development Complexity in Process-Driven SOA. International Conference on Business Process and Services Computing, volume 116 - Lecture Notes in Informatics. P. 105—124.

[24] Berners-Lee, T. (2000). Presentation. Semantic Web on XML. XML 2000. Slide 10. Retr. Jan. 24th 2007 http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

[25] Berners-Lee, T., Hendler J., and Lassila O. (2011). The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American Feature Article: The Semantic Web.

[26] Stuckenschmidt, H. (2009). Debugging OWL Ontologies – A Reality Check. In Petrie Ch.: Semantic Web Service Challenge: Proceedings of the 2008 Workshops. Stanford Logic Group, Computer Science Department. Stanford University. Retr. Dec. 12th 2010. http://logic.stanford.edu/reports/LG-2009-01.pdf.

[27] Kauffman, St. (1995). At Home in the Universe. Oxford University Press, New York. P. 200 ff.

[28] Maturana, H., and Varela, F. (1992). The tree of knowledge: biological roots of human understanding (1984). Boston, MA: Shambhala Publications, Inc.

[29] Skobelev, P. (2011). Bio-Inspired Multi-Agent Technology for Industrial Applications. In: Alkhatheb F., Al Maghayreh E., Doush A.: Multi-Agent Systems – Modeling, Control, Programming, Simulations and Applications. Pp. 495-522. InTech, ISBN 978-953-307-174-9.

[30] Rzevski, G. (2011). A practical Methodology for Managing Complexity. Emergence: Complexity & Organization.13 (1-2), Pp. 38-56.

[31] Vrba, P., Kadera, P., Jirkovsky, V. Obitko, M., and Marik, V. (2011). New Trends of Visualization in Smart Production Control. In: Marik V. et al.: Holonic and Multi-Agent Systems for Manufacturing, HoloMAS 2011. Springer, Berlin ISBN 978-3-642-23180-3. Pp. 72-83.

[32] ARUM – Adaptive Ramp-up Management. European Research Framework Program, Project 312056. Planned Start: Sept 1st 2012. Project Management: EADS.

[33] Sloman, St. A., and Fernbach, P.M. (2011). Human representation and reasoning about complex causal systems. Journal Information, Knowledge, Systems Management. IOS Service. Volume 10, Number 1-4 / 2011, Chapter 5. Pp. 85-99.

[34] SESAR, Single European Sky ATM Research: http://www.eurocontrol.int/sesar/public/standard_page/overview.html. Retr. Nov. 14th 2009

[35] On Business Process Model and Notation (BPMN). http://www.omg.org/spec/BPMN/1.2/. Retr. Jan. 19th 2012

[36] Car-to-Car Communication Consortium. http://www.car-to-car.org/. Retrieved May 12th 2012

[37] Allan, R.J. (2009). Survey of Agent Based Modeling and Simulation Tools. Retr. June 21st 2012. http://epubs.cclrc.ac.uk/work-details?w=50398.