Business Information Systems

Assessment of Value Analysis on an Example of Core Banking System Replacement Projects

Einar Polis
Tallinn University of Technology, Estonia
einarpo@gmail.com

Abstract. The question of value is being increasingly discussed in various disciplines in recent years. Value-Based Management and Value-Based Software Engineering (VBSE) have emerged as promising attempts for employing the value concept in generating information systems which satisfy the needs of all stakeholders. We analyzed several package software implementation projects for assessing applicability of value based approach and found that seldom sufficient value for stakeholders has been produced. VBSE might have helped to achieve better results, although the theory itself is not fully matured yet. We found that introduction of intrinsic value or viability of the system could significantly enhance the explanatory power of the theory. This facilitates understanding of many advanced qualities of systems and development of theoretical framework for transforming the enterprise together with its information system. Our analysis will be concluded with suggestions for conducting projects similar to those that analyzed in this article.

Keywords: Information systems, value-based software engineering, enterprise modeling, living enterprise, self-organization

1 Introduction

Some management theorists and practitioners [15; 12] have studied long-term success of enterprises and found useful to compare enterprises with living organisms. Life metaphor is frequently used by other kind of system builders too ("system has gone live", "deadline"). Nevertheless, for a more rigorous approach, elaborated theories are needed. To explain the nature of living systems, for years have been around theories like Living Systems Theory [29] or Viable Systems Model [5]. Architect Christopher Alexander [2; 3] has written extensively about living structures and has already influenced information systems domain heavily by introduction of the design pattern concept [1].

Development of information systems has frequently to deal with great complexity in a limited timeframe. It is commonly considered to be a social and creative activity, which is difficult to formalize. Appropriate mindset and constructive worldview are needed for achieving higher success rate of projects and for bringing more value to stakeholders. We claim that treating enterprises as living entities is beneficial at least for certain information systems development projects. However, despite the potential,

only few attempts (e.g. [27]) have been made for moving from generic concepts closer to the level of details necessary in practical development work.

We are trying to approach this issue from a different angle. Value-Based Management (VBM) and Value-Based Software Engineering (VBSE) have emerged in recent years as alternative ways of organizing enterprise change management and information systems development respectively. We used their tools for analyzing several core banking systems replacement projects. Although these tools appeared useful for explaining project outcomes, at their current state the methods are lacking ways for tackling some aspects of systems viability. Therefore we would like to propose a complementary approach to value analysis, mostly inspired by late works of Alexander [3].

2 Problem Statement

The issues tackled in this article are originated from relatively common package software implementation projects. In last decade, the authors have been dealing with implementing Temenos T24 financial software in several banks, with various rate of success. Main characteristics and results of the projects are summarized in Table 1.

Table 1. Summary of sample projects

#	*Main characteristics*	*Result*
A	Development mostly in HQ by local people and few external consultants. Low personnel turnover. Need to go live, because of expiry of legacy system licenses	Three projects completed in time and below budget
B	Development on site, by constantly changing external consultants. Virtually no involvement of local IT, maybe even slight resistance to the system. Political games	Project cancelled after two years
C	Development in HQ, by external consultants, under management of local IT. Good architecture and sophisticated end result	Core system replaced, but a year later and over budget
D	Development on site, by frequently changing external consultants. Political games, no significant involvement of local IT	Project cancelled after two years
E	Developed in HQ by external consultants. Later outsourced to cost-efficient country, reductionism to accounting. Bank acquired by another bank no reuse	Delayed one year, stopped right before going live
F	Development on site by external consultants. Strong local IT support, undemocratic but relatively reasonable decision making	Fast implementation, later problems in live
G	Too difficult and incompatible requirements were presented, unstable project team. Financial crisis caused the financial institution to close down	Five years of development led nowhere
H	Tiny bank. Implementation on site by external consultants. Reliance on certain individuals	Successful implementation, problems in upgrading and further developments

Modern banks are all relatively similar and heavily dependent on their core banking systems. Sample institutions were small to medium size, highly international and relatively similar. Three projects were conducted in Eastern and three in Western Europe, two in Middle East or from the other perspective, four in Northern and other four in Southern EMEA countries. These projects can be considered to be a good investigation objects, even without taking into account additional factors like regional differences or global economic situation.

Serious problems were encountered in all projects. Even in case of projects A, where the usual success criteria were met, high stress level caused a lot of suffering for stakeholders. Most projects (except F, G, H) were preceded or were expected to succeed by similar projects in other branches of the bank. Only in A the reuse of custom developments (and only by around 50%) and noticeable learning from previous experiences took place.

Poor result of projects had serious impact to organizations, which definitely were unable to function without efficient information systems. T24-based development was expected to possess a central role, frequently referred as the core banking system. However, we believe that encountered problems are not specific only to these particular institutions, but are variations of classic complex questions from enterprise information systems development area: Why so many information system projects fail? Why experienced developers and enterprises are not learning from failures? How to measure the success of enterprise/information systems designers?

3 Value Analysis

There are many possible reasons for failure of sample projects. Unrealistic goals, budget size, poorly managed risks (bad luck), availability of knowledgeable and motivated hardworking people are all important success factors. In some cases these issues can be tackled. In better performing sample projects (A) from the time/functionality/costs point of view, wide range and complex functionality was introduced while leaving the practical implementation work to modestly paid developers, who learned specifics of a particular software system by doing and also whose banking knowledge level was usually below the financially significantly better rewarded external consultants of other projects.

Similar problems with information systems development projects in general are frequently reported in literature. This leads to assumption, that inappropriate understanding of value is quite common in projects and can possibly trigger abovementioned undesirable consequences. From the other hand, if personal and organizational values are aligned and appropriate, high-quality information systems can be generated more likely.

Value-Based Software Engineering (VBSE) is an emerging theory originated mostly from the work of Barry Boehm [7]. By definition, its aim is to connect traditionally value-neutral computer science with value-based theories such as utility theory, decision theory, dependency theory and control theory. Central claim of VBSE is a theory W, also known as the Enterprise Success Theorem, which states

that "Your enterprise will succeed if and only if it makes winners of your success-critical stakeholders".

Financial value has central position in VBSE and it is no surprise that concentration on financial value is common for financial systems and institutions too. In some cases (like with project costs) it is quite well measurable and in some cases where it is not (like ROI), it is still comprehensible. Various tools have been proposed for performing actual value analysis in VBSE, though no method has become prevalent yet. Therefore we are using relatively informal approach for identifying stakeholder value satisfaction in sample projects. The results of this activity are presented in Table 5, with separation of traditional, mostly financial values and alternative values discussed below.

We simplified the roles of stakeholders by removing details not relevant to this kind of analysis, in order to make projects comparable and complexity manageable. Results of this effort are presented in Table 2. As all the projects were conducted with the help of third party companies, a distinction is made on how the roles are carried out by different organizations. In larger projects the roles were split into more specialized roles or assigned to teams. Alternatively in smaller projects one person could perform many if not most of the roles. Though some roles may seem more important than others, it is difficult to point out any roles which are not success-critical. Some people are also usually considered more valuable or important for the project, which is reflected in their appropriate (principal, senior etc) titles, in managerial hierarchies and ultimately in rates. Desire to keep the latter down has huge impact on outcome of the projects in globalized economy, because of opportunities and temptations for staffing or outsourcing parts of it into low-cost countries.

The analysis led to several obvious conclusions, like too heavy or too light emphasis on financial value can hurt the projects significantly. There are also less obvious findings, like domination of business analysts (B, D, G) or accountants (special kind of representatives of owners' interests) over developers tends to jeopardize the project success (D, E). Attempts to go live fast paid off, though when done in administrative way (F), caused some problems later. Other success factors were strong local involvement, keeping systems integrity, while business as usual and assuming changeless world led usually to undesirable consequences.

Most important success factor seems to be a personal attachment to the outcome. This surprisingly does not have to be reflected in huge amount of overtime hours, but more on whether the people really care about the project and the organization. If we feel connection with something, we most likely wish it to last, so the enterprise must possess some characteristics of the living entity (desire that the project lasts indefinitely is hardly justified). Indeed, even if financial aspects remain most important at least till a certain extent [19], it is difficult to care a lot about money-making machines or body-shopping companies. Softer aspects are hard to tackle in a reductionistic way and there is a frequently discussed gap between objective and subjective worldviews today [24]. Nevertheless, we suggest that not taking into account viability of the enterprise is causing serious problems for many information system development efforts. This leads to outcomes which are not able to survive in changing environment, if they become into existence at all.

Table 2. Stakeholder roles

Role	*Bank*	*Supplier*	*Description*
Owner	Usually large public or state-owned institution	Usually a small company owned by few people	Represented through management hierarchy, which can influence projects by financing, appointing key persons, steering the progress and conformance with strategic objectives. The ultimate source of organizational culture
Project mgmt	Project manager or managers, sometimes accompanied with administrative workers (PMO)	Usually more like a resource manager, sometimes team leader	Coordinates different activities or work-streams
Business Analyst	Business representatives and/or system analysts	Functional consultants, mostly with banking, sometimes technical background	Identifies business requirements, usually coordinates development activities in particular area. Usually no drastic changes to business are caused by such projects
Architect	Sometimes sets standards, requirements to interfaces etc	Usually just a title for praising outstanding developers or for lead developer	Should define high level structures, interoperability with other systems, development standards
Developer	Usually new to the software system, sometimes not present	Product experts from technical side, knowledgeable on software specifics	Complexity level of developments, sophistication of tools etc is not particularly high, which frequently causes unjustified underestimation of the developers role
Tester	Business Analysts in smaller, dedicated testers in bigger projects	Sometimes fully outsourced to separate company	Should ensure reasonable quality of the system
Infra-structure	System administrators	Configuration manager	Provides environments for growing the system. Usually quite transparent, but in case of failure affects everybody
User	Mostly noticeable in requirements elicitation, UAT (User Acceptance Testing) phases of the project and from feedback from the usage of the deployed system	Theoretically can be the bank's customers, but this relationship rarely matters	The people who are going to utilize the system in performing their business activities
Customer	Could be direct users of the system via electronic channels	Meta-customer for the supplier, but nevertheless important	Entities to whom the enterprise has to provide value in order to survive

4 Application of the Living Enterprise Theory

Systems development as an intellectual and creative activity depends heavily on the way of perceiving the world. According to the interpretive paradigm [17], enterprises have tendency to become more similar to what we are thinking of them. Respectively, if we consider them as living systems they are more likely to possess qualities usually attributable to living organisms. We believe that not only the top level managers have the power to influence the enterprise with their thinking, but all stakeholders.

Ability to distinguish living from non-living seems natural to human mind, thus the life metaphor has a good potential to be well understood by different stakeholders of enterprise and it is also a good candidate for facilitating understanding and concentration on important things in information systems development. Relative popularity of life sciences at the moment can be beneficial too. However, finding clear definitions for concepts in this domain is relatively challenging and is still ongoing activity.

Some eminent management theorists have studied longevity of enterprises [15; 12] or buildings [8; 1] and found many similarities between them and living systems. Such theories are mostly intended to top-level management of companies and are lacking details of practical implementation. There are also several closely related and difficult to comprehend concepts like learning, intelligent, maybe even sustainable enterprise. We suppose that enterprise having listed characteristic is more alive, as these are not usually attributed to more deterministic machines.

Life is an emergent property of complex structures, which are usually having characteristics like metabolism, growth, reproduction, adaptation, homeostasis, autopoiesis etc. Several generic theories have been created for describing such systems, including Viable Systems Model (VSM) [5] or Living Systems Theory (LST) [29]. The former finds 5 subsystems necessary to meet the demands of surviving in the changing environment for any autonomous system. The latter presents a hierarchy of living systems and identifies 20 subsystems for processing matter/energy or information present in them all. From enterprise information systems development point of view, specialized theories could be even more applicable, like Alexander's approach [3] or Luhmann's theory of autopoietic social systems [26].

Certain qualities characterize a good design of systems. Alexander [1] has comprehensive discussion on difficulties in finding appropriate words for describing experiences with good buildings. Freedom, wholeness and quality without a name, are all closely related to liveliness. Opposite characteristics are attributable to lifeless, machine-like structures. This distinction is illustrated in Table 3. Realistically our creations contain elements both increasing and decreasing liveliness, thus making the whole more or less alive. Moreover, sometimes we may want to have a machine-like enterprise, only to produce concrete results in a particular timeframe. Universality, self-organization etc characteristics are in that case only hurting efficiency.

Table 3. Comparison of characteristics of the enterprise

Characteristic	*Less Life*	*More Life*
Organization	Allopoietic, structure is produced outside	Autopoietic, self-organization
Control	Centralized, autocratic	Decentralized, distributed, democratic
Optimization	Concentration on core processes and values	Diversification
Identity	Anonymous, interchangeable	Distinguishable, personality
Utility	Serving somebody's interests	Exists primarily for its own survival
Ownership	Changing, disrespectful	Tolerable, nonintrusive
Attitude	Indifference	Connectedness

Alexander has through years of careful observations identified around fifteen structural properties which are constantly present in living systems [3]. These are very briefly summarized in Table 4, with hints for implementation in transforming the enterprise. His basic element for explaining living structures is centre. We should treat environment as a field of wholes, each supporting and amplifying one another. One can be very precise and descriptive about these wholes, but cannot avoid looking at the totality at each step of the way. Centers are not isolated entities, but work together for supporting life and consist of other centers.

Having identified the importance of viability for enterprises, we would like to propose a modified Enterprise Success Theorem as "In a long term, enterprise succeeds if and only if it makes winners of success-critical stakeholders and maintains its viability".

5 Information System of the Living Enterprise

To explain the role of the information system in a living enterprise, we are presenting a model of the information system as a result of externalization of the information processing by networked hierarchy of entities (see Fig. 1).

5.1 Subject System

Hierarchy of living systems has indefinite number of levels. It is based on individuals, whose communications are building blocks for different kind of enterprises. Hierarchy tops with an entity containing all other entities and which is here called society. The latter is governed by nobody, but nevertheless it provides us with rules to follow and even with some data.

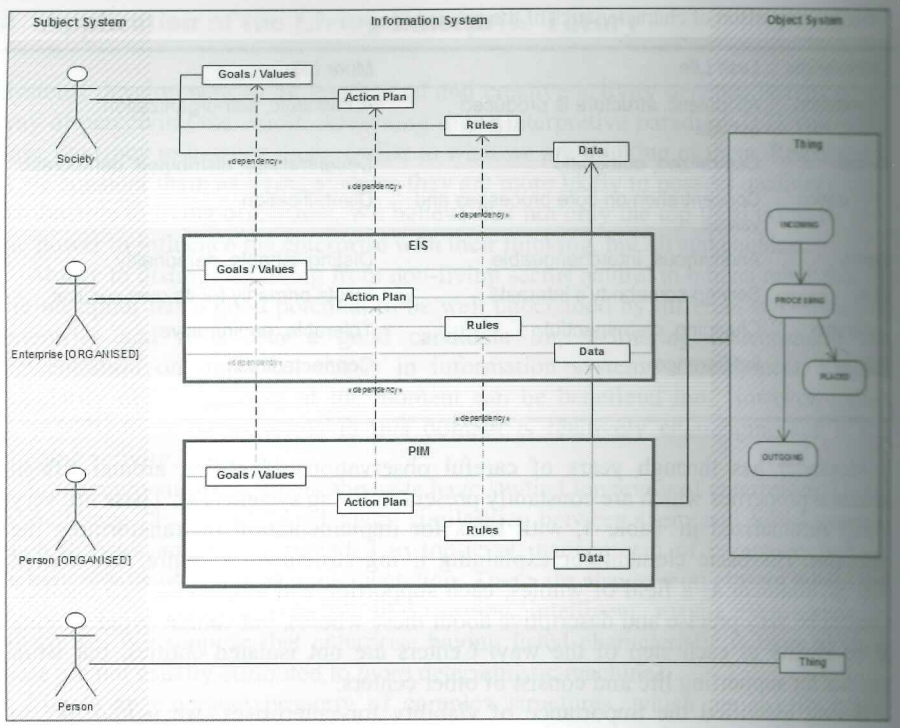

Fig. 1. Information systems of living entities

Technology optimists predict that in next decades if not machine intelligence then at least computing power will surpass the human brain [25]. Another prediction is that enterprises are going to store all related e-memories of their employees [6]. Systems with such abilities will be certainly very different from the systems we are able to build today.

Till those days, presence of human agent in ultimate customer role is expected in information systems. Somebody has to benefit from the existence of the system and conscious decisions are probably needed. Personal productivity expert David Allen has found [4], that for the latter we should free our minds by externalizing goals, action plans etc into personal information management systems (PIMS).

People gather into enterprises, causing new structures, culture and technology to arise. Each enterprise has its unique identity, which is restricted by its mission, laws, regulations etc. We may also consider characteristics of systems presented in Table 3. In case of more alive enterprises, they may be expressed in principles like sensibility to the economical environment, tolerance to differentiating viewpoints and conservative financing [15]. From the other hand, sometimes we may want to employ different principles and to treat enterprises as purposefully engineered money-making machines.

As with people, the enterprises may have different level of organization. In well organized enterprise its rules and policies, goals and values, actions for achieving them and data on important things are externalized into enterprise information system

(EIS). Among other reasons, this makes its pattern less vulnerable to loss of people comprising the enterprise.

5.2 Object System

Things seldom can be handled in most efficient manner only by relying on direct correlation between them and brain activities. Instead we may need to clarify our position regarding the thing and open corresponding records in our personal or enterprise information systems. These records should track lifecycle of the things in relation to us: when they came in, have been processed or put into right place and finally discarded.

Machines are expected to relieve us from mechanistic repetitive activities in dealing with things, including in related information processing. People should be left with freedom of choice, creative problems and possibility to grow with the system, which with respective rewarding feeling should make them happier [14]. Things themselves influence our feelings too. Alexander has asked people on liveliness of different objects and found much communality in their opinions [3]. Such experiments are dependent on artistic renditions of reality and Alexander's own books use images from renowned photographers, certainly capable of projecting their desired qualities.

From the other hand, we may intentionally want to make things (including information systems) look better for increasing their chances for a longer life and making their users feel better. Although beauty of buildings is extensively discussed, beauty in information systems design is generally considered unimportant. There is an interesting discussion related to making of ten-thousand year clock appealing to people, so that it will survive through all the uncertainties in the future [9].

5.3 Information System

According to LST, each living system has subsystems for processing information. As living systems, these subsystems form hierarchies, irrespective of whether they are externalized or not. Enterprise goals, action plans etc are dependent on similar items of other enterprises or society as a whole. In the same way they affect individual persons. Aggregation and summarization of data works mostly in opposite direction.

Brand describes how buildings change over time and adapt to their surroundings [5]. Lasting materials and flexible structures must be used for achieving longevity. Information systems have to adapt to changing environments too, possibly at a greater pace, to keep up with rapid advancement of information technology. Computer hardware and software becomes obsolete fast. Open source and good homebrew software with personal attachment may have better chances for survival, sustainable development, reuse and refactoring. Data must be kept alive by choosing open widespread formats if possible, upgrading them on necessity and ensuring it has value for the users.

6 Coevolution of Enterprise and Its Information System

Development of high-quality complex systems like business organizations or information systems is a non-trivial multidisciplinary activity having many stakeholders and a lot of uncertainty. It is not realistic to derive such systems entirely from a set of formal rules, at least not yet and for all systems possibly never. We could rather try to assess liveliness of an enterprise and plan steps for transforming it to be more alive or less vulnerable to risks. How liveliness can be interpreted in more or less scale or as a position in a lifeline of the enterprise is illustrated in Fig. 2.

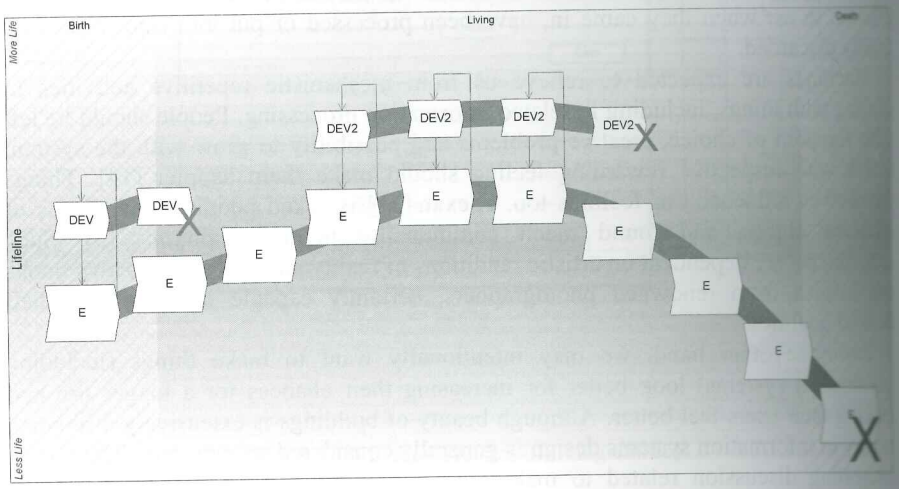

Fig. 2. The life of an enterprise

We could divide a lifeline into more stages (e.g. youth, sickness, prospering, decomposition) or hope that some entities will never face death. Opinions differ on necessity of dying, its importance in evolutionary adaptations and in adding value to life, both for humans [16] and enterprises [15]. Disappearance may cause significant economical damage, loss of knowledge and human suffering. System builders should consider these questions, though likelihood of finding consensus is small.

Staying alive is universally important for each individual being, whose ultimate goals can be defined as survival and replication, or long life and prospering. The ultimate goals can be broken down into sub-goals and selected subset of them expressed as requirements for transformation. From the other hand, entities have moral etc criteria which restrict the freedom in achieving these goals. They are based on speculations (free will, common good, consciousness, viability etc) on the nature of the world, which are still unavoidable in composing the action plan.

There are several ways for measuring the success of transformations. Some experienced consultants have given quite accurate predictions on chances of finishing the sample projects successfully. These predictions are based on personal experiences and inner feelings. We can try to build more formal composite indicator based on characteristics presented in Table 3, Alexandrian structural properties discussed below based on Table 4, financial performance, personnel turnover, domain-specific

fitness indicators, availability of risk management procedures, and whether different future scenarios have been discussed.

Having proper measurements is certainly very important for a development organization, which will be leading the transformational activities. From enterprise longevity perspective, this entity must be able to balance novelty seeking with benefits of long term knowledge acquisition. It should follow feedback for changes via measurements and look farther from project management, by having a development programme for consolidating subsequent projects in accordance to strategy of the enterprise. The latter should have longer view than upcoming few years. Many practical matters need to be managed properly for promoting liveliness by the development organization, including but not limited to:

- *Contracts*. Should leave freedom for adapting to constantly improving understanding of the system
- *Insourcing*. Balance has to be found between reliance on inner resources and "fresh blood"
- *Outsourcing*. Related organizations should have more than financial interest to the result
- *Creativity*. Enterprise should be sufficiently ready to support creative efforts of an individual
- *Ethics*. Illegal and immoral activities should be avoided for the sake of life of an enterprise

To deal successfully with so demanding tasks, development organization probably has to involve experts from various disciplines. Even more important is inclusion of people having long-term experiences within particular enterprise, willingness to understand and grow together with it.

Transformers of the enterprise need appropriate methods and tools, as enlivenment work may include significant amounts of information processing. In sample projects only generic office software (Microsoft Office) was used, in some occasions together with diagramming (Microsoft Visio), project management (Microsoft Project) and bug/defect tracking software. Even if using of predefined document templates and document management procedures was forced, rarely self-sufficient and reusable (except copy-pasting to produce all those documents required by CMMI) documentation was produced.

Dedicated tools may have several benefits over the generic tools. Possible structure of such tool is envisioned in Fig. 3. Proper goal, action and measurement handling have all their own specifics challenges. However, most effort is required to build an enlivenment-specific subsystem, called pattern repository in the model. This subsystem can be helpful for storing efficiently a large number of patterns (pattern libraries or pools for deriving pattern languages for particular cases), visualizing pattern interconnections (pattern net), linking patterns to goals and action planning system, measurement tracking and as a learning aid.

Though innovation may be routinely needed, in sustainable development it is expressed mostly by applying already proven solutions in novel context. Though genetic etc algorithms learnt directly from nature are quite extensively used in software-intense systems, more generic approaches for information systems development are based on design patterns movement.

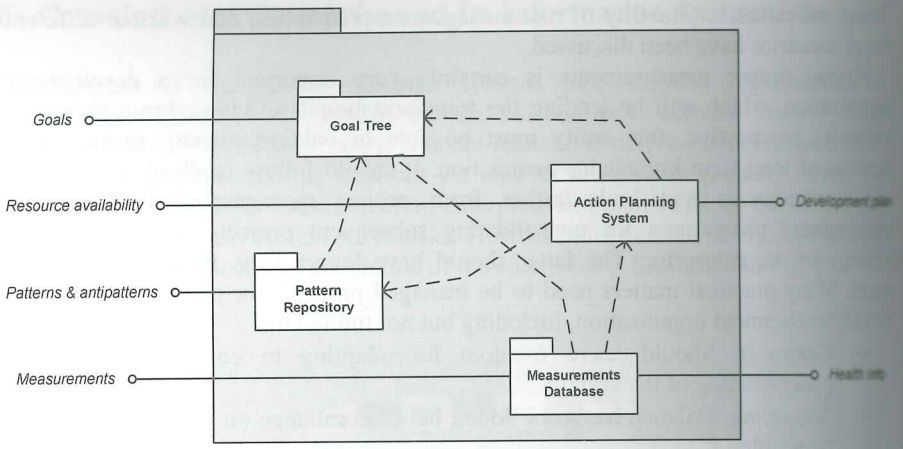

Fig. 3. High-level architecture of the enlivenment information system

Alexander saw design patterns emerging from repeating transformations as stereotypical centre configurations or transformations [1]. He considered finding of a real pattern to be as difficult as anything in theoretical physics [2], but once determined it could be reused efficiently by anybody from novices to masters. Patterns express collective knowledge and it is hard to give formal proof on why they are working. Respectively their application is more like an activity based on common sense and a shared vocabulary.

Design patterns are used extensively in many domains. They are particularly widespread in object-oriented programming [20]; but also in systems analysis [18], data modeling [23; 22], software configuration management [10], project management [11], business [30] or organizational [13] modeling, teaching etc. Several big software houses are publishing papers with patterns for using their products. A generic pattern theory may benefit from emergence of relevant mathematical theories, which so far have not added much practical value yet [21].

Unfortunately most pattern implementations are stressing utilitarian value of the individual pattern as "a solution to a problem in a context", by omitting most of Alexander's philosophy, generative power of pattern languages and thus possibility for explaining how true masters in the field are working. This might be one of the reasons, why design patterns are heavily criticized in recent times by acclaimed programmers [31].

Surrounding and organizational culture are important building blocks of the living enterprise [28]. This provides helpful insights for explaining the raise of design patterns and why they may be going out of fashion. It is still unclear whether we are culturally ready to grasp latest ideas of Alexander, which are even more unorthodox than the thinking which led to widespread recognition of design patterns in software development.

Nevertheless, recent ideas of Alexander [3] provide a comprehensive way for transforming the enterprise with its information system. Structure-preserving transformations (STP) are interpretations of structural properties from Table 4 applied to the system for increasing its overall liveliness. They facilitate the growth of basic

structures into complex living entities. From constructive point of view that means efforts for going live with particular subsystem development as fast possible, after the architecture capable to support the future growth is in place and the risks seem tolerable. Based on already living subsystem, further requirements can be stated. This correlates well with approaches present in many modern agile software development methodologies.

Table 4. Alexandrian properties of living structures with references to systems implementation

Property	*Narrative*	*Implementation*
Levels of scale	Within or around any given centre, exist smaller centers which are level of scale lesser in size	System must be divided into reasonable number of modules/subsystems (e.g. level of scale or 7 ± 2)
Strong centers	Centers have been reinforced by other centers and intensified by STP	System must have clearly identifiable non-competing concentration points
Boundaries	Strong boundaries enclose the centre and adjacent centers around it	Autopoietic systems are operationally closed, but interactionally open and structurally coupled with environment
Alternating repetition	Recurring structures throughout the design in non-trivial way	Basic elements should be reused in several places with minor modifications only
Positive space	The region of the design filled with latent centers	Not only what is designed has importance, but also what is intentionally not designed
Good shape	As centers become reinforced and intensified, latent centers become definite centers	Obscure structures should be avoided, if there is no absolute need for using them
...	...	...
Non-separateness	As the structure develops through its uncompleted forms, exaggerated differences are eliminated	The system has to be become maximally adapted to its surroundings

7 Alternative Analysis Example

We have tried to complement VBSE with the insight that personal attachment and appropriate mindset are vital for project, for long-term success of the enterprise and for each related individual. Table 5 in a next page attempts to present the results of this effort. Of course, there are many different aspects and different ways for representing the information. Moreover, we could fit here only a comparison of two quite opposite projects. Therefore we have to summarize main findings separately: in our project sample, better results were achieved in case of long-term partnership between institutions (evolution needs time) and in an atmosphere of mutual respect and trust (essential role of which is also emphasized by "community of trust" pattern in [13]).

Table 5. Excerpts from projects/stakeholders/value analysis

#	*Stakeholder*	*Value for Bank*	*Value for Supplier*	*Viability assessment*
A	Owner	Possibility to expand local product offerings	Regional partnership at reasonable profit rate	Financially the projects performed well and also other objectives were met. However, after completion aggressive ROI period has started, which led to break up of the development organization and partnership with the vendor. Support and future developments centralised to cost-efficient country, which made future developments and necessary software upgrade increasingly complicated
	Project mgmt	One more project within the same company	Volume-based compensation	
	Business Analyst	Opportunity for self-development	-	
	Architect	-	-	
	Developer	Usually cheaper than using the supplier	Relatively good income for no more than 40h of work per week	
	Tester	BA-s understood the cost of errors	Unit testing and fixing usually chargeable work	
	Infra-structure	Less laborious maintenance and production support	-	
	Users	Dissatisfied with the product, but with no real reason	-	
	Customer	Attractive products and good rates were offered to them	-	
.....				
D	Owner	Possibility to expand local product offerings	Easy money before financial crisis	The project was stopped after two years, restarted in other location by repeating the same mistakes and without success. Despite big financial losses, permanent staff was able to keep their jobs. External consultants were mostly satisfied with the project, many have found this time valuable part of their lives and formed social connections have survived over the years
	Project mgmt	Negative feelings compensated with money	Impressive title with little responsibility	
	Business Analyst	Learn something valuable in a job market	Good guaranteed income	
	Architect	Proposal of unrelated but professionally interesting services bus	1 week job to draw a later discarded diagram	
	Developer	Left because of low pay compared with supplier	Little professional growth, but pleasant atmosphere	
	Tester	-	Opportunity for third-party company	
	Infra-structure	Mostly one dissatisfied person who finally left	Extra responsibilities, but compensated	
	User	Effort for requirements presentation wasted	-	
	Customer	No better service as no new system	-	

8 Conclusions

The aim of this article is to advance theoretical basis for treating enterprises as living beings, based on wide body of literature and on personal field experience. Few similar attempts (e.g. [27]) have been made so far. For connecting these ideas with existing information system development theories, proposal for extending VBSE theory was made.

Good metaphors are needed for explaining information systems and for consolidating views of different stakeholders. Life metaphor with its explanatory power seems suitable for this purpose and also correlates well with modern approaches to systems development. From the other hand are living things, including living enterprises, generally more difficult to study than non-living things and experimentation with them is even more complicated. Each living organism is unique and it is hard to return to the same initial conditions for restarting the experiment. This work was intended to set the stage for more detailed and practical research. As large-scale experiments are very difficult to conduct, next steps could contain activities like analysis of longevity of different kinds of enterprise information systems, analysis of longevity of the program code or assessment of presented models with related predictions.

Viability of the system can be indeed useful for identifying requirements, consolidating viewpoints and ensuring longevity. Additional work is needed for elaborating presented concepts and testing them in practice. As concept of life covers wide range aspects, some potentially useful building blocks for theory were left untouched here. Maybe not so much for practical use, but for building more sound groundings, presentation might have benefitted from more rigorous semiformal language.

References

1. Alexander, C.: A Pattern Language. Oxford University Press, New York (1977)
2. Alexander, C.: The Timeless Way of Building. Oxford University Press, New York (1979)
3. Alexander, C.: The Nature of Order. Center for Environmental Structure, Berkley (2003)
4. Allen, D.: Getting Things Done: The Art of Stress-Free Productivity. Viking Penguin, New York (2001)
5. Beer, S.: Brain of the firm: the managerial cybernetics of organization. Wiley, Chichester (1981)
6. Bell, G., Gemmell, J.: Total Recall: How the E-Memory Revolution Will Change Everything. Dutton Adult (2009)
7. Biffl, A., Aurum, B., Boehm, B., Erdogmus, P., Grunbacher (eds.): Value-Based Software Engineering. Springer (2006)
8. Brand, S.: How Buildings Learn: What Happens After They're Built. Penguin Books, New York (1994)
9. Brand, S.: Clock Of The Long Now: Time And Responsibility: The Ideas Behind The World's Slowest Computer. Basic Books (2000)
10. Brown, W. et al.: Anti-Patterns and Patterns in Software Configuration Management. John Wiley & Sons, New York (1999)

11. Brown, W et al.: Anti-Patterns in Project Management. John Wiley & Sons, New York (2000)
12. Collins, J., Porras, J. I.: Built to Last. HarperCollins, New York (2004)
13. Coplien, J.: Organizational Patterns of Agile Software Development. Prentice-Hall, Upper Saddle River, NJ (2004)
14. Csikszentmihalyi, M.: Flow: The Psychology of Optimal Experience. HarperCollins, New York (1991)
15. de Geus, A.: The Living Company. Harvard Business School Press, Boston (1997)
16. de Grey, A.: Ending Aging. St. Martin's Press, New York (2007)
17. Dietz, J.: Enterprise Ontology: Theory and Methodology. Springer, Berlin (2006)
18. Fowler, M.: Analysis Patterns: Reusable Object Models. Addison-Wesley, Menlo Park, CA (1996)
19. Frey, B.: Happiness: A Revolution in Economics (Munich Lectures Series). The MIT Press (2008)
20. Gamma, E. et al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (1995)
21. Goertzel, B.: The Hidden Pattern. BrownWalker Press (2006)
22. Hay, D.: Data Model Patterns: A Metadata Map. Elsevier Morgan Kaufmann, Amsterdam (2006)
23. Hay, D.: Data Model Patterns: Conventions of Thought. Dorset House Publishing, New York (1995)
24. Kaufmann, S.: Reinventing the Sacred. Basic Books, New York (2008)
25. Kurzweil, R.: Singularity Is Near: When Humans Transcend Biology. Viking Penguin, New York (2005)
26. Luhmann, N.: Social Systems. Stanford University Press, Stanford, CA (1995)
27. McDavid, D.: Systems Engineering for the Living Enterprise. 18th International Conference on Systems Engineering, 16-18 Aug. 2005, pp. 244-249 (2005)
28. McCracken, G.: Chief Culture Officer. Basic Books (2010)
29. Miller, J. G.: Living Systems. Mcgraw-Hill, New York (1978)
30. Penker, M., Eriksson, H.: Business Modeling With UML: Business Patterns at Work. John Wiley & Sons, New York (2000)
31. Seibel, P.: Coders at Work. Apress (2009)

Long-term Strategies for Electronic Recordkeeping in a Large Infrastructure Project

Erik A. M. Borglund[1], Karen Anderson[1]

[1] Mid Sweden University, 871 88 Härnösand Sweden
[erik.borglund, karen.anderson]@miun.se

Abstract. Recordkeeping is about "making and maintaining complete, accurate and reliable evidence of business transactions in the form of recorded information." Yet the research on electronic record management systems (ERMS) has not focused on the problem of long-term preservation of electronic records. The aim of this paper has been to further elaborate the challenges long-term preservation impose on recordkeeping strategies and when the record quality criteria of evidentiality, integrity, authenticity of records is to be fulfilled. The research has been carried out as a qualitative case study in a large railway infrastructure construction project, where many records are operational for more than 100 years. In this research we can conclude that the long-term preservation issue is difficult to take into consideration if proactive strategies are not adopted. It is also important not to separate the current recordkeeping strategy from long-term recordkeeping strategy.

Keywords: Electronic record management systems, Proactivity, Record Continuum Model, Recordkeeping,

1 Introduction

This research is about recordkeeping, an area that has gained limited interest from the IS field. Recordkeeping is defined as "making and maintaining complete, accurate and reliable evidence of business transactions in the form of recorded information." [1] The definition of the term recordkeeping includes "the design, establishment and operation of recordkeeping systems" [1]

Records are sometimes described as sources for knowledge within organizations [2-4] and can be defined as "institutional memory" [4]. Capturing, storing and sharing of knowledge can give an organization advantages in productivity and innovation [5].

Records are evidence of business actions and transactions and are kept to support current business and accountability. Records need to be managed in such a way that their authenticity and reliability is protected if their evidential value is to be kept [6, 7].

Since the early 1990s when it became obvious that electronic records needed new managerial techniques [8] archival science has struggled to establish a basis for sound recordkeeping. To meet the new need for managing born-digital records [e.g. 9, 10-13], many organizations have implemented electronic record management systems (ERMS), and the effect of those implementations has gained interest from researchers. A study of the major scientific journals in the area of electronic recordkeeping, the Records Management Journal[2], and Archival Science[3], found many success stories about the positive impact organizations can gain by implementing electronic records management systems [10, 13-15]. Implementation of an ERMS is about change in organizations [12], and it is a long process [16].

Much of the current research on ERMS has focused on implementation, and organizational aspects of implementation of ERMS but not much on how to solve the problems with long-term preservation of records. Both the IS community and the archival community have seemed to separate the long-term preservation problem area from the ERMS context.

Electronic recordkeeping needs to adopt a proactive approach to be able to fulfill recordkeeping requirements, and to enable long term preservation [17]. Proactivity in recordkeeping implies that information systems involved in recordkeeping should be designed to fulfill recordkeeping requirements. That is, the successful capture, management and long-term preservation of digital records depend on appropriate design of the systems in which they are created and kept. Well-designed and carefully executed strategies for migration of the systems and the records they contain are also crucial for success. Attempts to preserve records as items, or individual digital objects, cannot preserve the context, which is essential to preserving the evidentiality, the integrity and the authenticity of records.

Long-term preservation of electronic records is a challenge, which has been argued for in both archival literature [see e.g. 8, 18-23], and in IS literature [24, 25].

In this paper we aim to further elaborate the challenges long-term preservation poses on recordkeeping strategies when the record quality criteria of evidentiality, integrity, authenticity [7] of records is to be fulfilled. We seek a deeper understanding of how long-term preservation problem is related to operational use of ERMS. We have carried out this research in a large infrastructure project of a railway construction, where many records are operational for more than 100 years, so long-term preservation is essential within the current recordkeeping system.

The remainder of the paper is as follows; first we present the applied research method and present the context in which the research has been carried out. Then we present the findings of the case study and an analysis of standards and methods for proactive design of electronic record management systems. The results are in the next section, which is followed by a discussion of the results. The paper ends with a conclusion and outlines for future research needs in this research domain are drawn.

[2] http://www.emeraldinsight.com/info/journals/rmj/rmj.jsp
[3] http://www.springerlink.com/content/105703/

2 Research method and research context

This paper is based upon a one-year research project, which aimed to identify problems, and potential research areas in a large railway infrastructure project. The research project was explorative and is best described as an interpretative case study. Interpretive case studies are a commonly applied research method in information system research [26]. We have used interviews, and official documents as the main data collection techniques.

This research conforms to the established Scandinavian IS research tradition, with an intertwined mix of technologies, application areas, and stakeholders [27].

The research was carried out in the project *Ådalsbanan* that is managed by the Swedish Rail Administration (SRA). The Swedish Rail Administration has overall responsibility for the railway transport system in Sweden. This includes the role of leading large infrastructure projects, which have an estimated lifespan of 150 years or more after construction is completed. The *Ådalsbanan* is one of the largest railway projects in modern times in Sweden, and is also the first project of that size in which all records are born digital. The project, *Ådalsbanan*, also involves several other official agencies: e.g the National Land Survey, the County Administrative Board and the Swedish Road Administration. Altogether there are more than 16 public organizations that are involved in the project. As well as the public organizations, hundreds of sub-contractors are involved in the project. The budget for the project is 6,6 billion Swedish kronor (approx 650 million €). It takes a very long time to repay investment in new railways.

This paper has an internal logical structure, wherein we first present a set of current and existing standards for electronic recordkeeping, together with relevant archival theories, which serve as basis for the analysis of the empirical data. We then present how recordkeeping is carried out by the SRA in general and specifically in the project "*Ådalsbanan*". Based upon the raw empirical data, which in the paper is presented in the results section, an analysis has been done in order to identify:

1. Problems with current recordkeeping strategy at SRA;
2. Design implications based upon 1.

We are fully aware that the findings from the SRA cannot be generalized. However the results can be transformed to other similar organizations, which also manage records that are operational for long time.

3 Theoretical framework

In this section we present the theoretical foundations for this paper.

3.1 Records management

Records and archives are the two concepts that form archival science [28]. And two of the most widespread definitions of records are:

> *"Information created, received, and maintained as evidence and information by an organization or person, in pursuance of legal obligations or in transaction of business."[7]*

> *"Recorded information in any form or medium, created or received and maintained, by an organization or person in the transaction of business or the conduct of affairs."[29]*

The record definition above does not make any separation between electronic records or paper-based records. All records have content, structure/form and are created in a context [30].

Records management, according to the ISO 15489 standard, is defined as:

> *"field of management responsible for the efficient and systematic control of the creation, receipt, maintenance, use and disposition of records, including processes for capturing and maintaining evidence of and information about business activities and transactions in the form of records "[7]*

In this paper we have used both the term recordkeeping and records management, which are treated as similar concepts. Recordkeeping is a term used foremost in Australia, and is used by those who conform to the Australian archival tradition as a broader term that covers both records management and archives, where archives are those records that are of enduring value which must be preserved for the long term.

The evidential value of a record is central, and records are preserved for the evidence they represent [e.g.4, 28, 31]. According to Cox [32] the evidential value of a record can only exist if the content, structure, and context are preserved. The context is the link between different records that belong together and also to the process where the record was created. The record's relationship to transactions is both what makes records different from information and enables the evidential functionality of records [6]. In order to have evidential value, records must have two other criteria: Authenticity, and reliability. A record can never serve as evidence if it is not reliable and authentic [6, 7, 20].

3.2 Records Continuum Model

The continuum model is developed by Frank Upward, and the aim of the model is to support archivists in their concern with the relationship between recordkeeping and accountability [33]. According to Upward [33] a characteristic of the model is the view of records as unstable. A recordkeeping model should consider both an object oriented approach and a system-based approach. There are no end products in an archival institution so there is a need for continuing addition of process metadata while they change across space-time. The model is four-dimensional (see figure 1). According to Upward [33] traditional archival methods are creating one-dimensional documents and two-dimensional records, or three-dimensional archive but technologies enable a four-dimensional approach. Upward [34] writes that records can have multiple lives in space-time, and a record is never finished in its creation, it is continuously in change.

The transaction axis is about the core business, its processes and functions. The identity axis focuses on the information's relation to the organization and its actors. The evidence axis describes the status of the information, and what role it plays in the organization. In the recordkeeping container axis the focus is about what kind of aggregated information units the original record can gather later.

Very simply one can describe the four axes as: Who (Identity) did what (Transactionality)?; What evidence exists upon this (Evidentiality)?; and what can be retrieved from the documents, records and archives (Recordkeeping containers)? The model implies that already when a record is born, it should fulfill what the other dimensions require. The model is the basis for the proactive approach where at creation a record should fulfill future requirements.

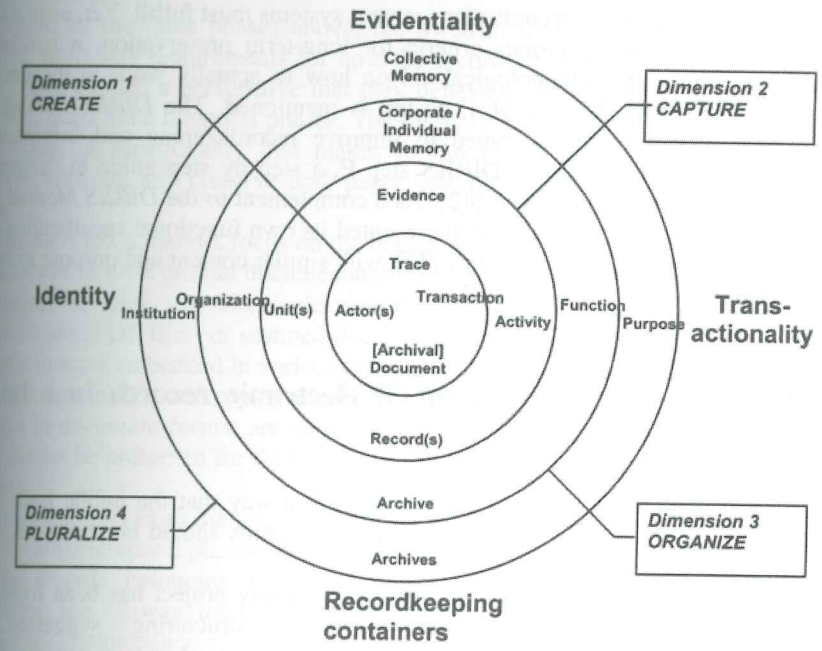

Fig. 1. The Records Continuum Model

The Records Continuum Model also visualizes the relationship between records creation and the evidential value of records over time, i.e. corporate knowledge and collective knowledge.

3.3 Existing standards for ERMS

Several standards have been developed in the domain of electronic records management. In *MoReq2*, the *Model Requirements for the Management of Electronic*

Records [35], the functional requirements of electronic record management systems are described. MoReq2 is an effort by the European Commission that was necessary to underpin transparent transnational e-services in Europe. In the United States of America the Department of Defense has a design criteria standard for electronic records management software [36]. Software vendors in the United States of America that have electronic records management software must be DoD-certified to be able to sell their software to public agency customers. In Norway they are working on an improved national standard for the Norwegian archive system, *Noark 4*. The new standard *Noark 5* is inspired by the work the European Commission did to improve *MoReq* [37]. The International Council on Archives has also developed a set of *Principles and Functional Requirements for Records in Electronic Office Environments*. [38-40] The four contributions presented above deal with the requirements that electronic records management systems must fulfill. Yet, even if the standards imply and state various criteria for long-term preservation, it has been impossible to find any methodological aid on how to actually work in the design process. In the *DIRKS Manual* [41], design is mentioned. The *DIRKS Manual* is based upon an 8-step process aimed to improve recordkeeping and information management in organizations. In DIRKS step F, a step by step guide to designing recordkeeping systems is presented [42]. As a complement to the *DIRKS Manual* the National Archives of Australia has also presented its own functional specification for electronic records management systems [43], with similar content and purpose as both *MoReq2*, and *Noark*.

4 Results – The management of electronic records in a large infrastructure project

In Sweden all public records must be kept in such a way that the public have free access to the records. It also means that all public records should be preserved, if a decision for destruction is not taken.

The records management in this infrastructure railway project has been found to exist in three temporal structures. The notion of temporal structuring is suggested as a way to study and understand the role of time in organizations. Actors in organizations produce and reproduce a variety of temporal structures, which in turn make the temporal rhythm and form the ongoing practice [44-46]. Organizations have different temporal structures and rhythms that are at the core of their practice.

The three temporal structures are:

- The project phase, which includes all sub phases that are within the project;
- The maintenance phase, which is the phase where the railway is operational;
- The curation phase is the phase when the railway is no longer operational.

They are also seen in figure 2.

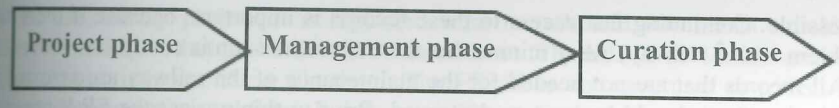

Fig. 2. The three temporal structures of records management in Ådalsbanan.

Beside the records born within the project it is worth noting that many records are born within 16 other public organizations that have business that is affected by the railway project in some way. The full extent of how many records this is, and in what way it really affects the railway project in its three phases is unknown. No comprehensive cross-organisation records survey for the Ådalsbanan Project has ever been undertaken.

Within all the three phases above, the Swedish Rail Administration (henceforth named SRA) sets requirements for how the records in the ERMS will be managed. They have applied a perspective that they believe covers the entire life cycle of the records, ie all three temporal phases. The SRA's aim is that their internal regulations will result in functionality that makes it possible to create, metadata tag, store, archive, sort out, and preserve both paper based and electronic records for the long term.

The SRA has a distinct focus on the management of records that are in the form of a document in their official documentation. I.e. either the record is physically a paper document or they are electronic documents, such as Word documents, Excel spreadsheets, PDF files, or scanned documents such as TIFF or JPG pictures. All the records that are embedded in various business information systems are not covered by the SRA internal recordkeeping regulations. The consequence is that the records that are not in document format are uncontrolled. Therefore there is a very high risk that they cannot be preserved for the very long term required by this railway infrastructure project.

Empirical data from the project phase is presented below, but we also present how the two other temporal phases are organized and how they work.

The records belonging to the Swedish railway are geographically ordered, following the railway tracks. The majority of the records that are kept for the continuous maintenance of the railway are blueprints of buildings, the power supply, the signal devices, freight yards, and roads for which the SRA is responsible.

The records are kept physically in the regional SRA archive that is closest to the railway, and its tracks. Currently the records are paper-based in the SRA regional archives, and there is not yet any proposal for how to manage the digital records that result from *"Ådalsbanan"*. The SRA has chosen a solution for managing digital records where they preserve a clone (i.e. an exact copy) of the ERMS, believing that in this way all records can be accessed even after the project is completed. In the project *"Ådalsbanan"* there are many digital records that are 3D CAD blueprints that cannot be transformed to paper. There are detailed regulations about what records should be preserved during this temporal structure. The delivery of records follows a standardized process, where the project organization delivers the records to the management organization: that is, to the SRA operational division. Only the records that are necessary for the operational management of a railway are kept and

accessible. Continuing fast access to these records is important, because if there is a problem with a railway, every minute that the train cannot run is costly.

All records that are not needed for the maintenance of the railway are assessed as to whether they should be kept or destroyed. Prior to this project the SRA has only needed to handle paper-based records. These records have been kept in the SRA's regional archives for a couple of years until the SRA delivered them to the Swedish National Archives. The SRA has no defined and stated strategy on how to preserve the *"Ådalsbanan"* project's electronic records for the long term. The solution SRA has chosen for the long-term preservation of records that are not needed for the maintenance of the railway, is to make a clone, i.e. an exact copy of the ERMS. At this point the SRA does not separate how they manage the records for maintenance and those for long-term preservation purposes.

4.1 Project phase in details

The railway project *"Ådalsbanan"* has a planned life of 10 years, and acts as an independent organization during this project phase. The project phases consist of three major sections: Investigation, Planning, and Construction (See figure 3.)

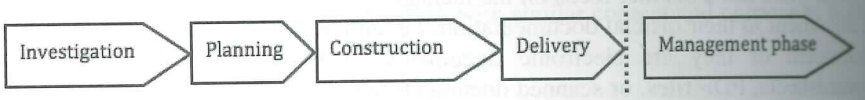

Fig. 3. The project phases and its sub phases

Within the *"Ådalsbanan"* project they have well-described goals for electronic records management during the project phase. The following goals for record management are set out in the project description:

- The right information should be available at the right place, in time, with the correct status, and fulfilling the user requirements
- The information and records management should fulfill the legal requirements for both public and secret records.
- All records should be traceable, and should be kept in the stated formats, follow stated standards, be managed in the stated way according to the SRA and the project's internal regulation
- All employees within the project and all internal and external partners should follow the stated regulations on records management in the project
- The records that should be delivered for continuous management of the railway should follow the stated internal regulations for those records.

The project *"Ådalsbanan"* has an electronic records management system (ERMS), which contains more than 250 000 records, each of which can be many documents. Everyone who is involved with records and documents creation or use has access to the ERMS. The main purpose of the ERMS is to guarantee that all records that should be available to the maintenance phase of the railway are managed correctly. For each

part of the railway project, it is regulated what records and documents should be delivered, and how that phase should be carried out. The ERMS at *"Ådalsbanan"* is rather simple: it looks like a file three. The ERMS is not structured to reflect and support the project's workflow. It acts more like a document repository, where the records are kept and preserved in a controlled environment. The ERMS is accessible through the SRA Intranet.

In the regulations for records management in the project *"Ådalsbanan"* one can find nothing about the importance of preserving context. This means that "Why-questions" about actions are difficult to answer. Even if the project *"Ådalsbanan"* has not yet faced any situations where they felt that they needed to search for how other projects solved specific problems, this is something the SRA was aware of. There is no tradition within the SRA to make sure that the wider knowledge context attached to the records is preserved. SRA does preserve the transactional context, although they don't seem to have a strategy for preserving future metadata that should be captured if the records are used after the ERMS is cloned.

5 Discussion

Electronic records management in this infrastructure project has three main purposes, which are:

1. To support the work carried out within the project, and during the construction of the railway;
2. To support the Swedish Rail Administration in its continuous management of all parts of the railroad during the life-period of the railway;
3. To support the public's right of access to public records and future research requirements.

In this research we have seen and presented how electronic records in the *"Ådalsbanan"* project are managed within the ERMS, which is something that many organizations do. What makes the *"Ådalsbanan"* project unique is that they produce records that are going to be operationally used for at least 100 years and most likely up to 150 years. In the available standards for electronic record management systems it is very difficult to find any requirements that deal with this kind of need. This is an acute problem for SRA, and the fact they plan to "clone" the whole ERMS when the railway is moved from construction/project phase to maintenance phase, seems to be a rather panicky idea. Such a solution will act like a curation of a dead system and is something museums do. This simple strategy does not ensure that the software- and hardware-dependent system will remain accessible over the long-term, nor will it be possible for the records within the system to continue to aggregate metadata if they are accessible and used after cloning. There are so many problems to solve if you need to preserve a system. Even if SRA has a records management strategy, they only mention the need for long-term preservation. They have not yet identified an effective strategy to guarantee long-term preservation.

The SRA has designed and developed an ERMS that is aimed only at managing the records during the project phase, and not for the project's maintenance phase. The

ERMS does not capture and manage digital records that are not documents e.g. the 3D CAD records, but also others, e.g. databases that contain records.

This creates a situation where the current recordkeeping strategy is not in harmony with the long-term recordkeeping strategy. This is visualized in figure 4.

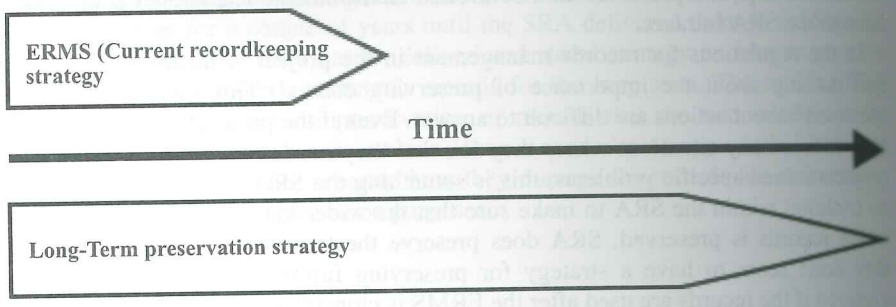

Fig. 4. Difference between ERMS strategy and long-term preservation strategy

In the best of worlds, the current recordkeeping strategy should be integrated and embedded in the long-term preservation strategy. The Records Continuum Model [33, 34, 47] implies that these two strategies should have been united.

In the SRA recordkeeping strategy they do not focus on describing business rules or policies and mandates. The business activities and processes are not documented at a detailed level. Neither do they regulate in detail how the actors in the projects should be described. According to the ISO standard 15489, it is required to manage metadata about;

a) the record itself;
b) the business rules or policies and mandates;
c) agents who are actors in the transactions;
d) business activities or processes in which the transactions take place;
e) records management processes.

As presented above records and documents are sometimes described as sources for knowledge within organizations [2-4, 48] and can be defined as the memory of an organization [4]. By studying the Records Continuum Model, you will find that dimension 3 & 4 is about the pluralization of records and their value so that they become individual and organizational memories. Capturing the context surrounding the records supports such activities and is implicitly involved in an organization's knowledge management. From a sound recordkeeping tradition it is important to capture the context in which every record is created. The ERMS in the *"Ådalsbanan"* project has a limited capacity to capture context. In archival science the central concepts of provenance and original order are about capturing and documenting the context and the relationship between records in a business [3]. Provenance is the link between records and transactions, processes and organizations and how they relate to each other. Without the provenance and the contextual links between records, records cannot be demonstrated to be authentic and reliable, evidentiality is lost and using the records for knowledge and understanding about what has happened will be difficult.

The SRA does not capture and preserve the knowledge context and this makes it almost impossible for the records to act as sources for knowledge. For example, the SRA cannot use records as sources for reducing the risk of making the same mistakes in future projects.

When the ERMS used by SRA was designed and developed none of the available standards were followed. This has resulted in a situation where the ERMS not can be verified to fulfill the minimum quality requirements, i.e. authenticity, reliability, integrity, and usability [7]. Furthermore, the ERMS is only one of many possible IS/IT systems which are part of an organization's knowledge management strategy and play an important role in managing explicit knowledge [49]. However, if correctly developed, the ERMS could be an important asset for contributing to knowledge management capacity in an organization such as SRA. The Records Continuum Model (figure 1) implies that in the dimension 'organize', the records take a form that also makes them useful as corporate memory in an organization. When the project *"Ådalsbanan"* delivers its material to the SRA after the project formally has ended, the records will not be tagged with metadata that will make them a useable part of the corporate memory.

The ERMS in *"Ådalsbanan"* project is not structured to reflect and support the project's workflow. Records arise organically from business transactions. They should be maintained in their original order to preserve provenance, reflect business processes, to preserve evidence for accountability and preserve knowledge in its original context.

Records management and records management systems should support the business, which is at the heart of the ISO 15489 standard [7, 50]. This realizes the benefits of sound recordkeeping, including the capture of corporate memory. In the *"Ådalsbanan"* Project the ERMS acted as a repository, and work process analysis to identify where records are born was not found. By using a recordkeeping strategy that follows the work process the context is more naturally preserved, and the context is also implicitly embedded in the understanding of the process. The EDRMS that the *"Ådalsbanan"* and SRA used was more like a modern file server with integrity rules than a recordkeeping system.

As seen above the SRA has designed and developed a strategy for recordkeeping that attempts to take into account the long-term preservation, i.e. a temporal aspect on recordkeeping. However the strategy seems to produce only a paper product, when the ERMS has vanished at some time in the future. The ERMS only manages current document-type records, and the system for preservation of operational records in the maintenance phase is not designed to fulfill long-term preservation requirements.

Derived from the Records Continuum Model [33, 34, 47] you must act proactively with electronic recordkeeping [51]. The proactivity implies that you answer at the systems design stage, before the records are created, questions like:

- what records will be preserved?
- what metadata is needed at creation, and throughout the records' existence to preserve context and ensure that the records remain evidential?
- how long should the records be preserved, and what file formats are suitable preservation formats?
- what are the business needs for records' usability throughout its exstence? i.e. long-term user requirements on the records.

- who will be able to access and use the records in the future, and for what purpose?

6 Concluding remarks

The aim of this paper has been to further elaborate the challenges long-term preservation imposes on recordkeeping systems strategies and design, when the quality criteria of evidentiality, integrity, authenticity [7] of records must be fulfilled. We have presented empirical results from a one year long case study which provided a deeper understanding about how the long-term preservation problem is related to operational use of ERMS.

In this research we can conclude that the long-term preservation issue is difficult to achieve if the business does not proactively take this aim into consideration at the systems design stage. Based on the fundamentals of the Records Continuum Model, sound recordkeeping that takes long-term preservation issues into consideration requires proactivity. The result from this case study shows a situation where, due to low quality recordkeeping with poor documentation and capture of context, the organization will not be able to use their records as the knowledge reservoir they could be. It is not a learning organization and they risk making the same mistakes over and over again. The case study also shows that the strategy for current recordkeeping i.e. recordkeeping that will mostly be handled by an ERMS, must not be separated from the long-term preservation strategy. In this case study the responsibility for current recordkeeping, and long-term preservation was found in two different parts of the organization, which also affected the end product.

A natural continuation of this research would be to establish and test a method for integrating current recordkeeping with long-term recordkeeping strategies based upon the Records Continuum Model.

References

1. National Archives of Australia: Digital Recordkeeping: Guidelines for Creating, Managing and Preserving Digital Records, (2004) http://www.naa.gov.au/Images/Digital-recordkeeping-guidelines_tcm2-920.pdf.
2. Gladney, H.M.: Trustworthy 100-Year Digital Objects: Evidence After Every Witness Is Dead. ACM Transactions on Information Systems vol. 22, pp. 406--436, (2004)
3. McKemmish, S., Piggott, M., Reed, B., Upward, F. (eds.): Archives: Recordkeeping in Society Charles Sturt University, Centre for Information Studies Wagga Wagga (2005)
4. Sprehe, T.J.: Integrating Records Management into Information Resources Management in U.S. Government Agencies. Government Information Quarterly vol. 17, pp. 13--26, (2000)
5. Nunes, M.B., Annansingh, F., Eaglestone, B., Wakefield, R.: Knowledge management issues in knowledge-intensive SMEs. Journal of Documentation, vol. 62, pp. 101--119, (2006)
6. Reed, B.: Records. In: McKemmish, S., Piggott, M., Reed, B., Upward, F. (eds.): Archives: Recordkeeping in Society Charles Sturt University, pp. 101--130. Centre for Information Studies, Wagga Wagga (2005)

7. International Standards Organization: ISO 15489-1. Information and Documentation and Records Management Part 1: General. International Standards Organization, Geneva (2001)
8. Bearman, D.: Electronic evidence : strategies for managing records in contemporary organizations. Archives and Museum Informatics, Pittsburgh (1994)
9. Henriksen, H.Z., Andersen, K.V.: Electronic records management systems implementation in the Pakistani local government. Record Management Journal vol. 18, pp. 40--52 (2008)
10. Williams, D.J.: EDRM implementation at the National Weights and Measures Laboratory Records Management Journal, vol. 15, pp. 158--166, (2005)
11. Biagio, M.L.D., Ibiricu, B.: A balancing act: learning lessons and adapting approaches whilst rolling out an EDRMS. Record Management Journal, vol. 18, pp. 170--179, (2008)
12. Gunnlaugsdottir, J.: As you sow, so you will reap: implementing ERMS Record Management Journal, vol. 18, pp. 21--39, (2008)
13. Maguire, R.: Lessons learned from implementing an electronic records management system. Records Management Journal, vol. 15, pp. 150--157, (2005)
14. Gregory, K.: Implementing an electronic records management system: A public sector case study. Records Management Journal, vol. 15, pp. 80--85, (2005)
15. Smyth, Z.A.: Implementing EDRM: has it provided the benefits expected? Records Management Journal, vol. 15, pp. 141--149, (2005)
16. Wilkins, L., Swatman, P.M.C., Holt, D.: Achieved and tangible benefits: lessons learned from a landmark EDRMS implementation. Record Management Journal, vol. 19, pp. 37--53, (2009)
17. Borglund, E.A.M.: Electronic Records use Changes Through Temporal Rhythms. Archives & Social Studies: A Journal of Interdisciplinary Research, vol. 2, pp. 103-134, (2008)
18. Dollar, C.M.: Authentic electronic records : strategies for long-term access. Cohasset Associates Inc., Chicago, Ill. (2000)
19. Duranti, L.: The Impact of Technological Change on Archival Theory. International Council on Archives Conference, September 16, 2000, Seville, Spain. (2000)
20. Duranti, L.: Concepts, Principles, and Methods for the Management of Electronic Records. The Information Society, vol. 17, pp. 271--279, (2001)
21. Gilliand-Swetland, A.J., Eppard, P.B.: Preserving the Authenticity of Contingent Digital Objects. D-Lib Magazine, Vol. 6 (2000)
22. Gilliland-Swetland, A.J.: Enduring paradigm, new opportunities : the value of the archival perspective in the digital environment. Council on Library and Information Resources, Washington, D.C. (2000)
23. McInnes, S.: Electronic Records: the new archival frontier. Journal of the Society of Archivists, vol. 19, pp. 211--220, (1998)
24. Chen, S.-S.: The Paradox of Digital Preservation. IEEE Computer, vol. 34, pp.24--29, (2001)
25. Sprague, R.H.: Electronic Document Management: Challenges and Opportunities for Information Systems Managers. MIS Quarterly, vol. 19, pp. 29--49, (1995)
26. Walsham, G.: Interpretative Case Studies in IS Research: Nature and Method. In: Myers, M.D., Avison, D.E. (eds.): Qualitative research in information systems : a reader, pp. 101--113. SAGE, London (2002)
27. Dahlbom, B.: The new Informatics. Scandinavian Journal of Information Systems vol. 8, pp. 29--48 (1996)
28. Thomassen, T.: A First Introduction to Archival Science. Archival Science vol. 1, pp. 373-385 (2001)
29. International Council on Archives: ISAD(G) : general international standard archival description : adopted by the Committe on Descriptive Standards, Stockholm, Sweden, 19-22 September 1999. Subdirección General de Archivos Estatales, Madrid (2000)
30. Hofman, H.: Lost in cyberspace : where is the record? In: Abukhanfusa, K. (ed.): The concept of record : report from the Second Stockholm Conference on Archival Science and

the Concept of Record, 30-31 May 1996, pp. 115--129. Swedish National Archives, Stockholm (1998)
31. Schellenberg, T.R.: Modern archives : principles and techniques. SAA. (Original work published 1956), Chicago (1956/1998)
32. Cox, R.J.: Managing records as evidence and information. Quorum Books, Westport, Conn.; London (2001)
33. Upward, F.: The records continuum. In: McKemmish, S., Piggott, M., Barbara, R., Upward, F. (eds.): Archives: Recordkeeping in Society, pp. 197--222. Charles Sturt University, Centre for Information Studies Wagga Wagga (2005)
34. Upward, F.: Modeling the continuum as paradigm shift in recordkeeping and archiving processes, and beyond - a personal reflection. Records Management Journal, vol. 10, pp. 115-139, (2000)
35. European Commission: MoReq2: Model requirements for the management of electronic records. Office for Official Publications of the European Communities, Luxembourg (2008)
36. Department of Defense: Electronic Records Management Software Applications Design Criteria Standard: DoD 5015.02-STD. Department of Defence (2007)
37. Riksarkivet: Noark-5 Norsk arkivsystem. Riksarkivet Norge (2007)
38. International Council on Archives: Principles and Functional Requirements for Records in Electronic Office Environments - Module 1: Overview and Statement of Principles. International Council on Archives (2008)
39. International Council on Archives: Principles and Functional Requirements for Records in Electronic Office Environments - Module 2: Guidelines and Functional Requirements for Electronic Records Management Systems. International Council on Archives (2008)
40. International Council on Archives: Principles and Functional Requirements for Records in Electronic Office Environments - Module 3: Guidelines and Functional Requirements for Records in Business Systems. International Council on Archives (2008)
41. National Archives of Australia: Part 1-The Dirks Methodology: A Users Guide (2001) http://www.naa.gov.au/Images/dirks_part1_tcm2-935.pdf.
42. National Archives of Australia: Step F – Design Of A Recordkeeping System (2001) http://www.naa.gov.au/Images/dirks_stepF_tcm2-963.pdf.
43. National Archives of Australia: Functional Specifications for Electronic Records Management Systems Software (2006) http://www.naa.gov.au/Images/ERMSspecifications_tcm2-1007.pdf.
44. Ancona, D.G., Okhuysen, G.A., Perlow, L.A.: Taking Time To Integrate Temporal Research. Academy of Management Review, vol. 26, pp. 512--529, (2001)
45. Orlikowski, W.J., Yates, J.: It's About Time: Temporal Structuring in Organizations. Organization Science, vol. 13, pp. 684--700, (2002)
46. Reddy, M., Dourish, P.: A finger on the pulse: temporal rhythms and information seeking in medical work Proceedings of the 2002 ACM conference on Computer supported cooperative work pp.344-353, New Orleans, Louisiana, USA (2002)
47. Upward, F.: The Records Continuum and the Concept of an End Product. Archives and Manuscripts, vol. 32, pp. 40--62, (2004)
48. Menne-Haritz, A.: Access-the reformulation of an archival paradigm. Archival Science, vol. 1, pp. 57--82 (2001)
49. Carvalho, R.B.d., Ferreira, M.A.T.: Using information technology to support knowledge conversion processes. Information Research, vol. 7, (2001) [Available at http://InformationR.net/ir/7-1/paper118.html]
50. International Standards Organization: ISO 15489-2. Information and documentation – Records management — Part 2: Guidelines. International Standards Organization, Geneva (2001)
51. Borglund, E.A.M.: Design for recordkeeping: areas of improvement. Department of Information Technology and Media, Mid Sweden University, Sundsvall (2008)

The Zachman Framework with Archetypes and Archetype Patterns

Gunnar Piho[1,2], Jaak Tepandi[1], Mart Roost[1]

[1] Department of Informatics, Tallinn University of Technology, Raja St. 15, Tallinn 12617, Estonia
[2] Clinical and Biomedical Proteomics Group, CRUK, LIMM, St James's Univ. Hospital, Beckett St, Leeds LS9 7TF, UK
g.i.piho@leeds.ac.uk, tepandi@cc.ttu.ee, mroost@staff.ttu.ee

Abstract. Archetypes based development (ABD) utilizes archetypes and archetype patterns to increase dependability of software, reduce semantic heterogeneity of models and data types, improve maturity of the software development process, lead development of one-off software towards software factories, as well as satisfy the needs of small and medium sized software houses. Zachman framework (ZF) for enterprise architecture has been widely accepted as a standard for identifying and organizing descriptive representations that have critical roles in enterprise management and system development. To enable systematic usage of diverse enterprise architecture views and aspects in ABD, we analyze the conformity of the central idea of ABD (business archetype patterns - *party and party relationship, product, inventory, order, quantity and rule*), as well as the ABD itself with ZF. We describe and explain the meaning of each column, row and cell of ZF from the ABD perspective. As a result the ZF with business archetypes and business archetype patterns is presented. This ZF with archetypes and archetype patterns helps developers to better understand business requirements, to design cost effective enterprise applications through systematic reuse of archetypal components by enabling supply chains of product families, as well as to justify and explain solutions to business side stakeholders.

Keywords: archetypes and archetype patterns based development, domain engineering, Zachman Framework

1 Introduction

The growing pressure to improve software quality as well as reduce cost and time to market has in recent years catalyzed a transition to more automated software development methods. Software factories (SF) approach is one of these automated software development or mass customization methods, which promise greater gains in productivity and predictability by making application assembly more cost-effective through systematic reuse and by enabling the formation of supply chains. Software factories address a number of challenges that the software development faces today, such as increasing dependability [1] of developed software, reducing semantic heterogeneity [2; 3] of models and data types, improving the maturity [4; 5] of the

software development process, reducing cost and time to market, and leading to development of one-off software towards automated software development [6; 7].

The business archetypes and archetype patterns [8] introduced by Arlow and Neustadt are among of the key innovations for developing better software with less recourses. Our industrial background in developing software for clinical laboratories have stimulated ongoing research in archetypes based development (ABD) [9; 10; 11; 12]. We investigate the research topics [13] posed by Björner, by combining the software engineering triptych [14; 15; 16] methodology with archetypes and the archetype patterns (A&AP) initiative by Arlow and Neustadt [8]. According to the software engineering triptych, in order to develop software we must first informally or formally describe the domain ($\mathcal{D}$); then we must somehow derive the requirements ($\mathcal{R}$) from these domain descriptions; and finally from these requirements we must determine software design specifications and implement the software ($\mathcal{S}$), so that $\mathcal{D}, \mathcal{S} \models \mathcal{R}$ (meaning the software is correct) holds [13]. ABD [11] is software triptych process with business archetypes and business archetype patterns.

In the process of elaborating the principles for ABD it soon became evident that there are a large number of views, issues, and artefacts to be developed and used. It is vital to understand, arrange, and utilize them in a systematic way. The best approach for this is to apply an already established overwhelming framework for enterprise architecture. There exist various frameworks, and the Zachman framework (ZF) [17] for enterprise architecture has been widely accepted as a standard for identifying and organizing descriptive representations that have critical roles in enterprise management and system development. For this reason the ZF was selected as a reference model for presenting the various ABD facets.

To enable systematic usage of diverse enterprise architecture views and aspects in ABD, we analyze the conformity of the central idea of ABD (business archetype patterns - *party and party relationship, product, inventory, order, quantity and rule* [8]), as well as the ABD itself with ZF. We describe and explain the meaning of each column, row and cell of ZF from the ABD perspective. As a result the ZF with business archetypes and business archetype patterns is presented. This ZF with archetypes and archetype patterns helps developers to better understand business requirements, to design cost effective enterprise applications through systematic reuse of archetypal components by enabling supply chains of product families, as well as to justify and explain solutions to business side stakeholders.

The rest of paper is organized as follows. In Section 2, we introduce some basic concepts and principles. We present the proposed framework in section 3. In Section 4 we exemplify the suggested business archetypes and business archetype patterns based approach in terms of clinical laboratory and show how we use this approach in Clinical and Biomedical Proteomics Group, at the Leeds Institute of Molecular Medicine, University of Leeds when developing software for laboratory in question. Finally, we make conclusions and suggest some comments for future works.

2 Basic Concepts

Business archetype is a primordial thing that occurs inconsistently and universally in business domains and in business software systems [8]. Business archetype pattern is a specification of collaboration between business archetypes. The following is a short description of particular business archetype patterns as described by Arlow and Neustad [8].

The *party archetype pattern* abstracts an identifiable and addressable unit that may have a legal status and that has some autonomous control over its actions. The *party relationship archetype pattern*, capturing a fact about semantic relationship between two parties in which each party plays a specific role, abstracts the relationships between parties. For example, the *CRM* (customer relationship management) *archetype pattern* concretizes the party relationship archetype pattern and abstracts features needed for customer management; similarly the *notification archetype pattern* [18] also concretizes the party relationship archetype pattern but abstracts agent reports about terrorist's organization or terrorist (someone who acts on behalf of a terrorist's organization) activities. The *product archetype pattern* abstracts product features and product related activities. The *inventory archetype pattern* abstracts a collection of inventory entities held in the stock by a business. The *order archetype pattern* abstracts selling as well as buying orders; the *rule archetype pattern* abstracts constraints on the operations of the business; and the *quantity archetype pattern* abstracts measures (as an example, money is subtype of quantity).

The main target of ABD [11] is rapid application development (RAD) of product families. While general-purpose RAD uses logical information about the software captured by general-purpose development artefacts [7], the ABD uses conceptual information captured by domain specific models. ABD focuses on software factories, i.e. on family-based development artefacts (domain specific languages, patterns, frameworks, tools, micro processes, and others) that can be used to build the family members. Our ABD process model is based on the software engineering triptych (from domain via requirements to software) model [14; 15; 16] and roughly consists of the following:

- Analysis of the business domains using domain analysis methodology similar to one suggested by Björner [16]. Björner's domain analysis methodology is based on domain stakeholders and on pragmatically chosen domain facets (intrinsic, business processes, supporting technologies, management and organization, rules and regulations, scripts and human behaviour). We suggest ZF based approach in combination with archetypes and archetype patterns (Section III). In this approach, we use the question provided by ZF (what, how, where, who, when and why) and answers to these questions provided by Arlow and Neustadt. It means we use archetypes and archetype patterns [8] as the meta-model when developing the business domain models (for example the domain models for clinical laboratory).
- In ABD, as common for software factories, all models are software artefacts and not only documentation artefacts. For instance, all our clinical laboratory models – the industrial case study of our research - are realized as .NET class libraries by using test driven development (TDD) [19] methodology.

- We use these domain models as the "ubiquitous language" [20] for prescribing and formalizing requirements from customers. These customer requirements we can validate according to the domain models.

We investigate possibilities to generate the software by using domain models and requirements formalized according to this domain model. Our aim is to verify the generated software according to the requirements and to the domain model.

Look informal explanation of ABD from [11].

3 ABD from ZF Perspective and Vice Versa

The Zachman Framework (ZF) [17; 21; 22] is a framework for enterprise architecture, which provides a formal and structured way for describing an enterprise. The ZF is a two dimensional matrix consisting of 6 rows and 6 columns.

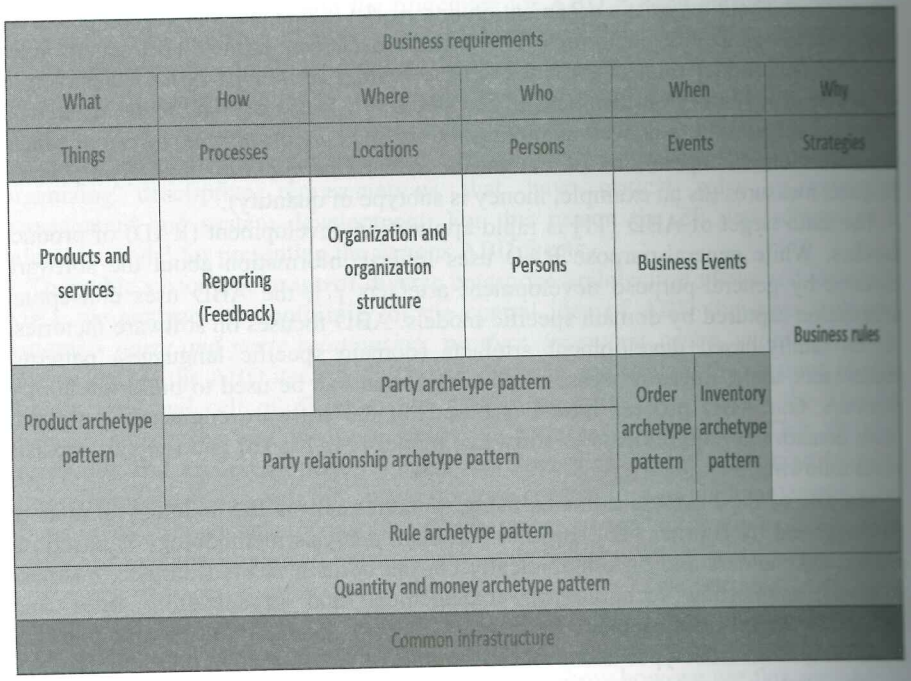

Fig. 1. ZF columns with archetypes and archetype patterns

Each row represents a view of the enterprise from the particular perspective: row 1 (scope) represents the context that establishes the universe of discourse; row 2 represents the conceptual usage characteristics of the end product (what the owner(s) are going to do with the end product); row 3 represents the logical designers view by combining the end user desirables with physical and technical possibilities of builder; row 4 represents builders technological perspectives to the system; row 5 is a detailed

description that disassociates the parts or pieces of the complex object for manufacturing purposes; and row 6 is the actual functioning system.

Each column describes represented by rows perspectives from particular abstraction by asking the common questions what, how, where, who, when and why. By answering these questions we get information about things (what), processes (how), locations (where), persons (who), time (when) and motivation (why).

More precise ZF descriptions are given in [17; 21; 22] or in other resources. In what follows, the interpretation of ABD from ZF perspective and vice versa is given. We describe and explain the meaning of each ZF column in terms of archetypes and archetype patterns and each ZF row in terms of software triptych.

The general picture of ZF with archetypes and archetype patterns is illustrated in Fig.1. In explanations we are using the ZF rules described by Zachman in 2003 [22].

3.1 Each Column is a Generic Model of Archetype Pattern

Each column of the ZF describes *"single, independent phenomena*. These independent phenomena are Things (What?), Processes (How?), Locations (Where?), People (Who?), Time (When?) and Motivation (Why?).

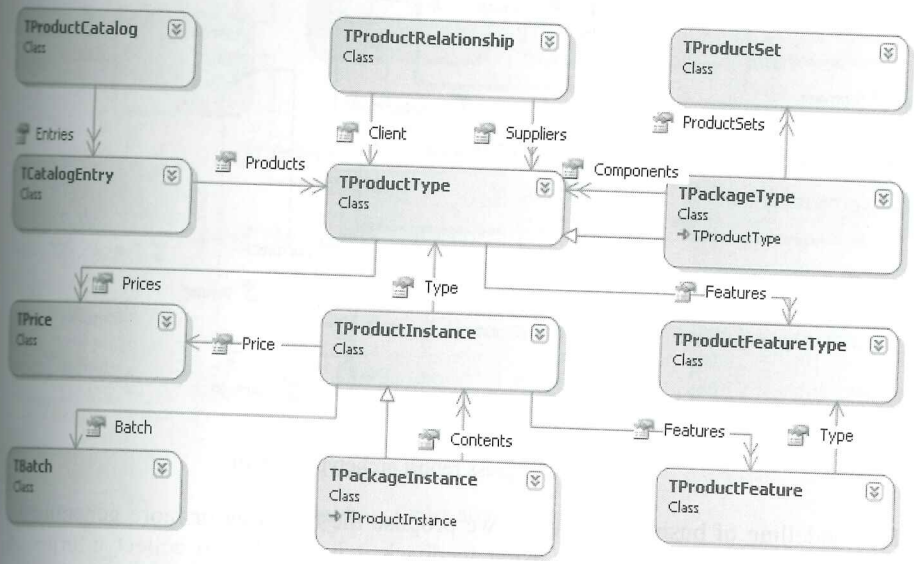

Fig. 2. Product archetype pattern abstraction

The basic models in the ZF based on business archetypes and business archetype patterns can be as follows.

Column 1 (What, Things) describes how products (either goods or services) are related to each other. Examples of product relations are "produced by using", "produced from", "is component of", "belongs to", "upgradable to", "substituted by", "complemented by", "compatible with", "incompatible with", etc. Product features

and product relationships can be modelled by *product archetype pattern* (Fig.2) similar to one described in [8]. Two other archetype patterns we need when describing products are *quantity archetype pattern* and *rule archetype pattern*.

Column 2 (How, Processes) describes business processes. Naive examples of business processes are "buying", "selling", "producing", also "planning", "servicing", "controlling", "reporting", "transporting", "communication", and so on.

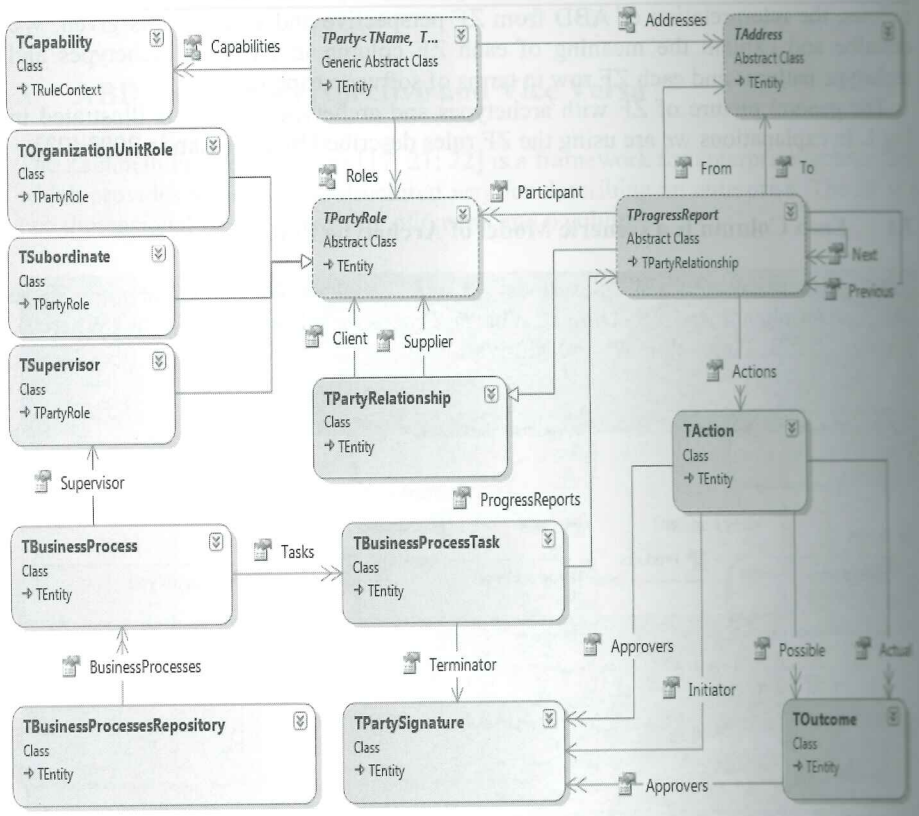

Fig. 3. Progress report archetype pattern

For modelling of business processes we propose the reporting or more generally the feedback archetype pattern. By using feedback it is possible to collect information about the processes. Like movie emulates the dynamics of reality by showing just sequences of static pictures, the feedback (progress report) emulates the business process dynamics by static "pictures" about the business process. Based on *party relationship archetype pattern* we have designed a *progress report archetype pattern* (Fig. 3). Each *progress report* (possibly *from* address *to* address) is the party relationship where *subordinate* reports (the role of person) to *supervisor* (the role of person) or to some *organization unit role* (accounting department for example). More than one *participant* can be related with business progress report. *Business process* tracks all *business process tasks* (sequence of progress reports). *Action* represents

something that can happen after progress report. Action has possible and actual *outcomes*. Examples of business processes in laboratory are "sample collection", "sample determination", "sample processing", and so on.

Column 3 (Where, Location) describes the structure of the organisation in terms of organization units and the roles of each organization unit. Like with persons, where we strongly separate roles (patient, client, mother...) from persons, we also separate roles strongly from organization units. For each role the responsibilities (assigned, mandatory and optional) and requirements for responsibilities also conditions of satisfaction from the business processes (column 2) perspective should be analyzed and modelled. After the roles are specified, the organization structure will be modelled by using the terms of organization units (sub-companies, divisions, departments, teams, etc) and by assigning these roles to designed organization units.

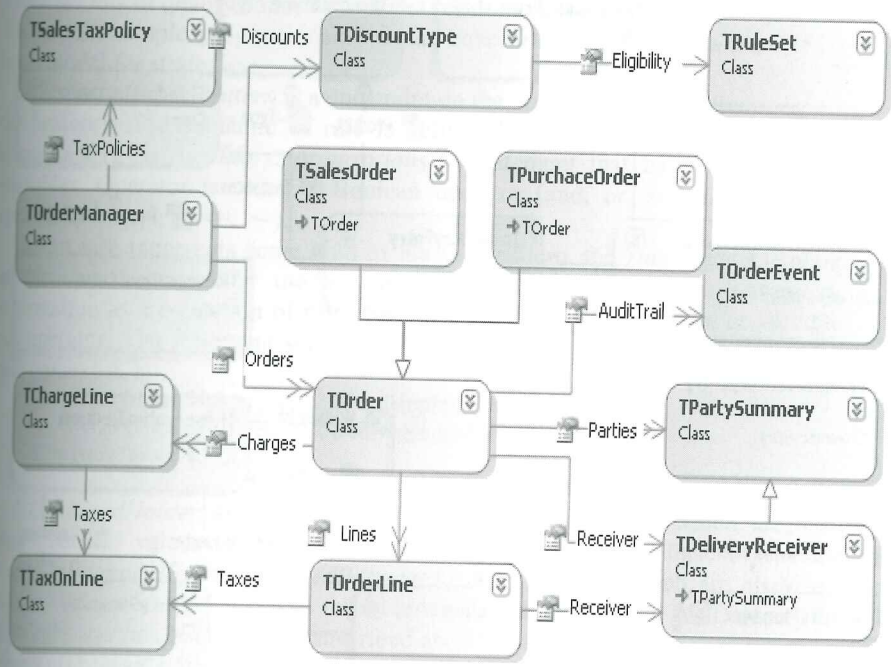

Fig. 4. Order archetype pattern

Each organization unit has the address, where it resides. Examples of organization units in a laboratory are workplaces and work-areas. Examples of organization roles in laboratory are *sample distributor*, *sample measurer*, *sample store* and *laboratory*.

Column 4 (Who, People) describes the people employed by an organization or playing some other roles (customers, suppliers, etc.) related somehow with organization. For fulfilling its assigned responsibilities (column 3), the organization needs employees. We are presenting the employee roles only in the context of roles the organizational units are playing when fulfilling the business processes. Besides the official roles, the people are playing informal roles as well. Sometimes these

informal roles are important and must be taken into account. Examples of a person role in a laboratory are *patient*, *clinician*, *medical technical assistant*. Examples of organization roles which are somehow related to a laboratory but which are not part of the laboratory itself are *hospital*, sick or healthcare *fund*, *supplier* of analyzers and spare parts and etc.

Column 5 (When, Events) describes all the events which are somehow related to the business processes of organization. Examples of events in organization are "plan is ready", "some resource has get to minimal acceptable limit", "new employer is hired", "order from customer", "order is ready for shipment", "payment is received", etc. All such kind of events should be logged and an audit trail should be produced. Events are tightly related to business processes (column 2).

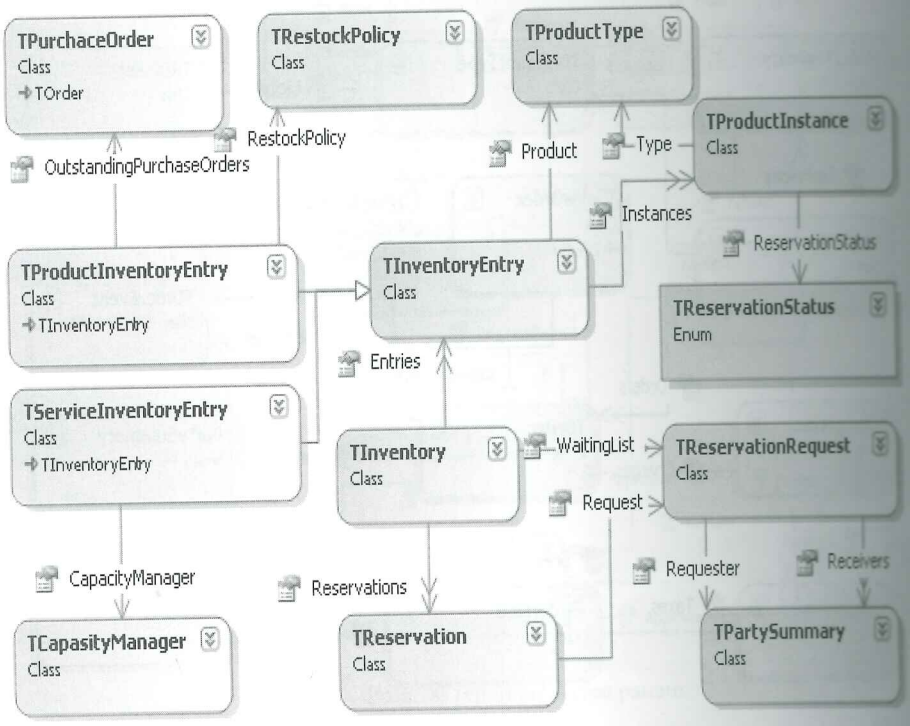

Fig. 5. Inventory archetype pattern

Each business progress report (Fig.3) generates one or more activities (*TActivity*). The progress report activity acts like trigger for event handlers. As the main purpose for event handlers is to log business actions, we model events by *order archetype patterns* and by *inventory archetype pattern*.

The *order archetype pattern* (Fig.4), records any request to change something in an inventory list of enterprise. Each order is uniquely identified. *Purchase order* and *sales order* are two common order types. By *order events* all order occasions (open, close, cancel, discount, amend, make payment, accept payment, make refund, accept

refund, invoice, order, etc) in the life-cycle of an order will be trailed for auditing purposes. Order has a particular state (initializing, open, closed, cancelled, etc) that constrains what activities (events) can be performed against the order. *Discounts* can be applied to the total price. *Discount type* contains a set of rules that describes the conditions, under which a particular discount may be applied.

The *inventory archetype pattern* (Fig. 5) represents a collection of inventory entities held in the stock by a business. *Product* and *service inventory entries* are representing respectively an inventory entry that holds either a set of product or service instances of one and the same type. The examples of events in laboratory are "new sample has arrived", "sample has tested", "sample test has validated", "new sampling order from clinician", etc. All such kind of events in laboratory should be logged and also audit trailed.

Column 6 (Why, Strategies) describes the strategies in terms of business rules. There are lots of quality control rules, test result validation rules etc. in the laboratory. We use the simple propositional calculus based *rules archetype pattern* [8] as the basic model for strategies.

A *Rule archetype pattern* is a constraint on the operation of the software systems of the business. The semantic of rule is defined by sequence of *rule elements*. Rule elements can be *operators*, *propositions* (a statement that has a truth value) and *variables*. Operator is either a Boolean operator (and, or, xor, not) or quantifier operator (=, !=, <, >, <=, >=).

When *rule* represents some kind of mask or pattern, the *rule context* contains the informational context for the execution of a rule. Rule context represents this information as a collection of rule elements that may be proposition or variable, but not operator. The following sets are examples [8] of simple rule (R) and respective rule context (C).

R = {IsColdCardHolder, IsSilverCardHolder, OR,
CarryOnBaggageKg, AllowedBaggageKg, LESS, AND }
C = {true, false, 4.5, 5.0}

IsColdCardHolder and *IsSiverCardHolder* are propositions which are assigned their actual values from the context C (true and false respectively). *CarryOnBaggageKg* and *AllowedBaggageKg* are variables which are also assigned their actual values (4.5 and 5.0) from context C. OR, LESS and AND are operators. Both rule and rule context are transcribed according to reverse polish notation, which means no brackets are needed and it is possible to evaluate the rule value by simple last in, first out (LIFO) stack.

3.2 Each Row is a Perspective of Triptych Development

The triptych software process – from domain model via requirements to software – has a very simple informal description: before to write software, we have to know requirements; before to know requirements, we have to understand domain; to understand the domain we have to study one. The interpretation on ZF rows in terms of triptych (requirements, domain, and software) development can be as follows:

Row 1 (Conceptual model) is just the glossary (list of things, objects, assets, etc) that defines the scope or boundary of enterprise requirements. For example, the cell

defining the scope for column 4 (people) for clinical laboratory can include terms like *patient*, *clinician*, *medical technical assistant*, *patient representative*, *clinician representative*, *patient hospital number*, and so on. The cell defining the scope for column 1 (things) for clinical laboratory, includes terms like *sample*, *material*, *aliquot*, *analyzer*, *rack*, *container*, *blood*, *urine*, and etc. The cell for defining the scope for column 2 (processes) for clinical laboratory includes terms like *sample request*, *sample collection*, *sample login*, *sample store*, *sample retrieve*, *sample distribution*, *work scheduling*, *sample analysis*, *results verification*, *results correction*, *laboratory management*, *results reporting*, *results interpretation*, etc [23].

Row 2 (Semantic model) is a definition and a model of the actual enterprise requirements. It defines the concepts (terms and facts) the enterprise actually needs.

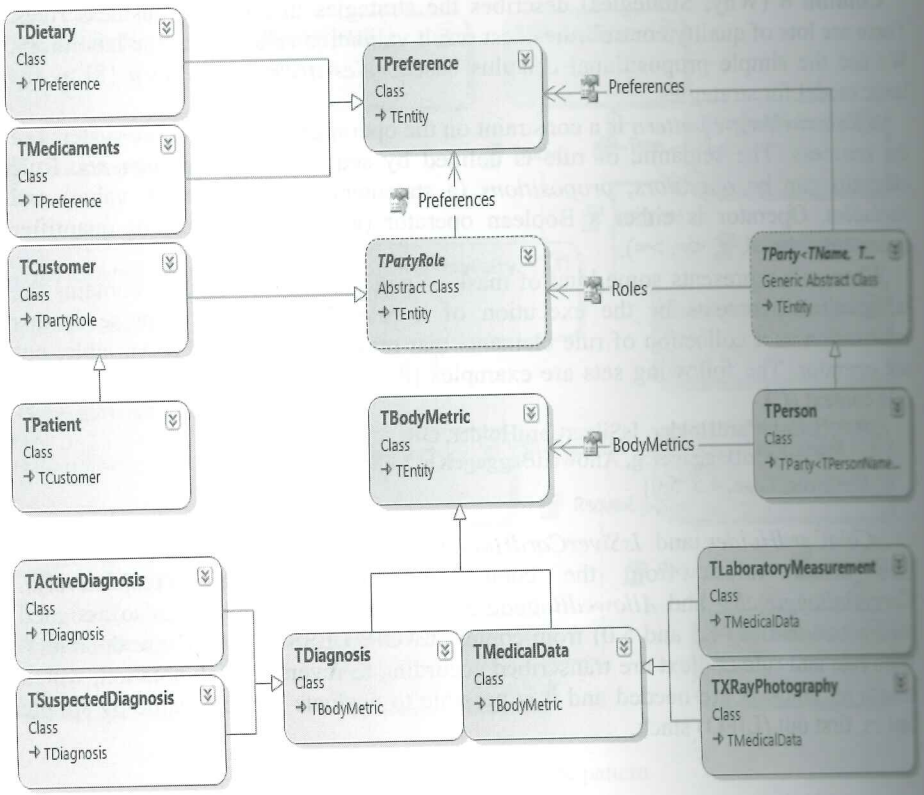

Fig. 6. Patient's Archetype

This can be represented as simple narratives (for instance "patient has a hospital number", "patient has a clinician", etc) or some more formalized (class diagram for example) notation.

Row 3 (Logical model) describes the requirements in terms of domain model. For column 4 this means for example, that "patient is the role for person", "patient hospital number is a registered identifier for person who's current role is patient", etc.

Fig.6 illustrates how patient and patient related features like diagnosis (actual and suspected), medical data (laboratory determinations and etc), dietary restrictions and medicaments can be extended from *party archetype pattern*.

Row 4 (Physical model) is the actual model of the domain. For column 1 (what) this is the definition of *product archetype pattern*; some *party relationship archetype pattern* derivate like progress report for business processes (column 2, how); party and party relationship archetype patterns for enterprise structure (column 3, location, where); party and party relationship archetype patterns for stakeholders (column 4, who); order and inventory archetype patterns for business events (column 5, when); and rule archetype pattern for business rules (column 6, why).

Row 5 (Detailed definition) is the realization of archetype patterns (product, party, party relationship and derivates, order, inventory, quantity and rule) as APIs and as database schemas supporting this pattern under some concrete database engine.

Row 6 (Product) is the software or service ($\mathcal{S}$) which uses the domain model ($\mathcal{D}$) (described according to row 4 and implemented according to row 5) and fulfils the requirements ($\mathcal{R}$) scoped by row 1 and explained by row 2.

4 Methodology and Validation

4.1 Methodology Guidelines

Systematic arrangement of diverse enterprise architecture views and aspects in ABD presented above has been mainly based on experience gained in working with archetype patterns, as well as using ABD in real-life applications. It would be useful to find the basic principles of such an arrangement, to enable its replication and further development with new patterns. We propose the following methodology guidelines for further investigation:

- Use the archetypes and archetype patterns [8] as the meta-model when developing the business domain models;
- Apply a domain analysis methodology based on domain stakeholders and on pragmatically chosen domain facets, such as business processes, supporting technologies, management and organization, etc.;
- Consider the models not only as documentation artefacts, but also as software artefacts;
- Validate the arrangement with software development for real applications;
- Evaluate satisfaction of the ZF rules to assess the arrangement.

4.2 Testing the Approach with the Laboratory Information Management Software Factory

Validity of our approach has been confirmed through archetypes based development of Laboratory Information Management System (LIMS) software factory. Development of the various LIMS artefacts was made more systematic and explicit

by using the arrangement provided by the Zachman Framework with Archetypes and Archetype Patterns.

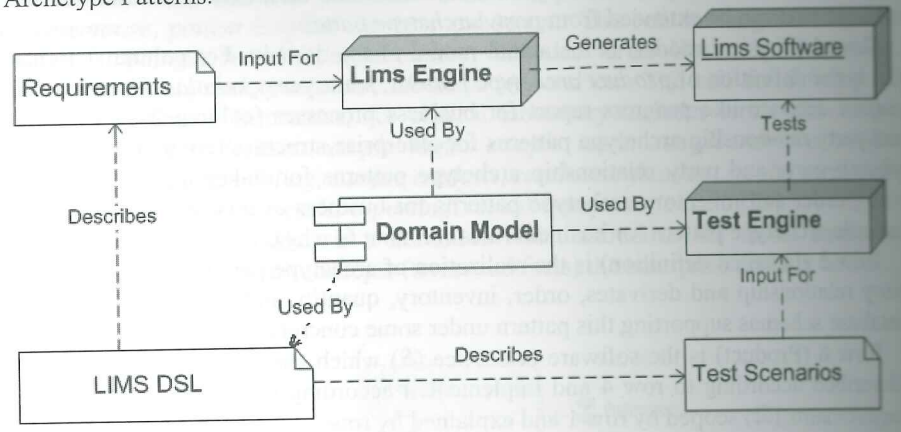

Fig. 7. Architecture of LIMS software factory

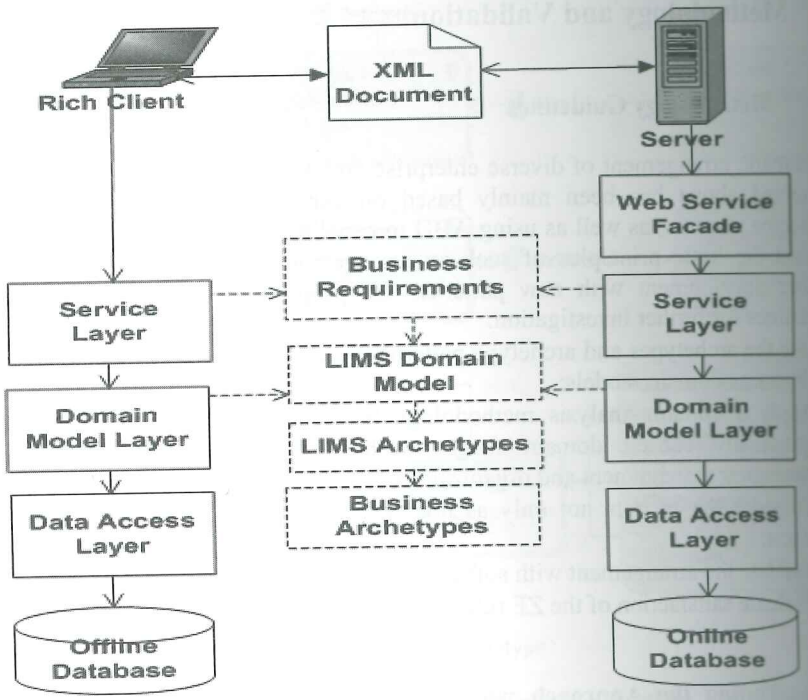

Fig. 8. Architecture of LIMS software

Based on the domain model of the clinical laboratory (implemented as API) the LIMS software factory architecture (Fig.7) consists of LIMS DSL (domain specific language), LIMS Engine, and Tests Engine.

With the LIMS DSL it should be possible to describe and modify the requirements for particular LIMS software. According to these requirements the LIMS Engine must generate the LIMS software and according to test scenarios (acceptance tests, generated also by LIMS DSL), the Tests Engine should validate the requirements according to the domain model and verify that the LIMS software meets requirements.

In the current stage of the project, we have focused on the domain model layer (Fig. 8) and on the possibilities to describe requirements from the concrete laboratory (service layer) by using terminology and features from the domain layer. We also investigate possibilities to automate generation of the UI as well as database schemas based on the domain layer. The structure of current code is following: infrastructure ~ 1000 LOC (lines of code); base archetype patterns ~ 5000 LOC; laboratory domain model ~ 5000 LOC; UI ~ 6000 LOC; ~ 2000 LOC is currently laboratory specific code; the data layer is currently based on flat XML files and realized by using .NET serialization (less than 200 LOC). Additionally approximately the same amount of unit tests. It is developed from scratch by one developer (18 man-months). There have been no software crashes in the laboratory, where the first version has been used from November 2009. We will monitor the second version (April 2010) which has been installed into three different research laboratories with three different requirements.

4.3 Assessment of the approach using the ZF rules

We follow [22] and evaluate the proposed ZF with business archetype patterns through the given rules from [22].

Rule 1: No Additional Rows or Columns

We do not add additional rows or columns to ZF. We have not proposed the modification or special case for ZF. The ZF with business archetypes and business archetype patterns is still the ZP with 6 rows and 6 columns proposed by Zachman. We agree that dimensions of the ZF classification scheme, the perspectives (rows) and the abstractions (columns) are both primitive and comprehensive. Therefore we do not need additional rows or columns to describe business domains and enterprise applications. Instead we have suggested the business archetypes and business archetype patterns based explanations to the long-standing questions of Who, What, When, Where, Why, and How.

Rule 2: Each Column is a Simple Generic Model

Each column of the ZF describes single, independent phenomena within the analytical target. These independent phenomena are Things (What?), Processes (How?), Locations (Where?), People (Who?), Time (When?) and Motivation (Why?). These basic models for ZF with business archetypes and archetype patterns are:

- Product archetype pattern for column 1 to describe things the business is dealing;
- Party relationship archetype patterns for column 2 to describe business process feedbacks;
- Party and party relationship archetype patterns for column 3 to describe internal structure of enterprise;

- Party and party relationship archetype patterns for column 4 to describe people and organizations somehow related with enterprise;
- Order and inventory archetype patterns for column 5 to describe the related with enterprise business processes events; and
- Rule archetype pattern for column 6 to describe business rules.

Rule 3: Level of Detail Is a Function of a Cell

Each cell in ZF with business archetype patterns specializes its column's generic business model from perspective of row. This means, that the level of details is a function of cell and not the function of column. This is because each row is still a different (conceptual, semantic, logical, physical, detail and product) model.

Rule 4: Meta Concept Can Be Classified Into One Cell Only

The ZF with business archetype patterns constitutes the same clean and normalized classification system that ZF has. "*Each of the Columns is unique. Each of the Rows is unique. Therefore, each of the Cells is unique. No meta-concept can be classified into more than one Cell. There is no redundancy.*" [22]. We kept this fundamental factor when interpreting ZF columns in terms of business archetypes and ZF rows in terms of triptych software development. We did not change the logic of the ZF; we only gave a methodological interpretation for ZF columns, rows and cells.

Rule 5: No Diagonal Relationships Between Cells

We have to mention that the terms in the first row (for example product, client, order, etc describing the universe of discourse) are not the same as the same terms for example in third row describing these terms in the language of archetype patterns. Neither the same terms in fifth row where these terms are implemented in some language of programming or being tables, rows or columns in some relation database. The client (human being), in terms of archetypes and archetype patterns (information model) and the realization of this model (bits and bytes) are all totally different things.

There are no diagonal relationships between cells in the ZF with business archetype patterns.

Rule 6: The Same Rows or Columns Names

Although we gave the business archetype patterns based interpretation for columns and the triptych software development based interpretation for rows, we did not change the names nor the meaning of Rows and Columns. So, the logic of the proposed framework is still the logic of ZF. Our contribution is standardization of the Column and Row interpretations. We argue that it is imperative to standardize not only the names and the meanings but also the interpretations if we are ever going to "*advance the state of the art in Enterprise Architecture necessary to accommodate the Information Age Enterprise*" [22].

Rule 7: The Logic is Generic, Recursive

The logic of the ZF with business archetype patterns is generic only in enterprise applications domains, while the ZF itself is universal.

5 Discussions and Conclusions

The ZF with business archetypes and archetype patterns is generic for business (enterprise application) domains and can be used to analyze any business domain like healthcare, transportation, education, manufacturing, etc. It is also recursive. It can be for example used to analyze sub-domains of healthcare domain like hospital, laboratory, general practices and so on.

Our research and development confirms that by studying enterprise domains (transportation, banking, education, healthcare, etc.), by building universal and abstract domain models, and by developing software source artefacts (software kernels, frameworks, engines, DLLs, APIs, DBs, etc) based on these models, it is possible to generate or customize software according to different business requirements.

We have proposed an interpretation of ZF columns in terms of business archetype patterns and an interpretation of ZF rows in terms of triptych software process. We argue that the ZF with business archetype patterns facilitates automated generation of enterprise applications. We are studying such possibilities and experimenting with them at the Leeds Institute of Molecular Medicine, University of Leeds where we are developing laboratory software for Clinical and Biomedical Proteomics Group.

Acknowledgments. This work is supported by Estonian Ministry of Education and research (SF0140013s10); by Estonian Science Foundation (grant 6839); by Tallinn University of Technology (Estonia); by University of Leeds (United Kingdom); by Cancer Research UK.

References

1. Avižienis, A., Laprie, J.-C., Randell, B.: Fundamental Concepts of Dependability. Research Report N01145, LAAS-CNRS, April (2001)
2. Halevy, A.Y.: Why Your Data Won't Mix: Semantic Heterogeneity. Queue, vol. 3, 8, pp. 50-58 (2005)
3. Wache, H.: Solving Semantic Interoperability Conflicts. In: Methodology workshop: Modeling eGovernment entities Methodologies and Experiences under review, Brussels, February (2009)
4. CMMI product team, CMMI for Development, Version 1.2, CMU/SEI-2006-TR-008, Software Engineering Institute (2007)
5. Charette, R.N.: Why Software Fails. IEEE Spectrum. Sept., (2005)
6. Greenfield, J., Short, K.: Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools. OOPSLA (2003)
7. Greenfield, J., Short, K., Cook, S., Kent, S.: Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools . Wiley (2004)
8. Arlow, J., Neustadt, I.: Enterprise Patterns and MDA: Building Better Software With Archetype Patterns and UML. Addisson-Wesly (2003)
9. Piho, G.: Archetype patterns based method of prescribing enterprise software requirements. International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatia, Croatia, (2007)

10. Piho, G., Tepandi, J., Parman M., Perkins, D.: From archetypes-based domain model of clinical laboratory to LIMS software. 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatia, Croatia, May 24-28, accepted paper (2010)
11. Piho, G., Roost, M., Perkins, D., Tepandi, J.: Towards archetypes-based software development. In: International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE 09), December 4 -12, accepted paper, (2009)
12. Piho, G.: Towards Archetypes and Archetype Patterns Based Software Engineering Techniques of Domains, Requirements and Software. In: Nordic workshop and doctoral symposium on dependability and security (NODES), Magnuse, Estonia, pp. 31-36 (2008)
13. Björner, D.: Domain Theory: Practice and Theories (A Discussion of Possible Research Topics). In: 4thInternational Colloquium on Theoretical Aspects of Computing - ICTAC, Macau SAR, China, (2007)
14. Björner, D.: Software Engineering, vol. 1: Abstraction and Modeling, Springer (2006)
15. Björner, D.: Software Engineering, vol. 2: Specifications of Systems and Languages, Springer (2006)
16. Björner, D.: Software Engineering, vol. 3: Domains, Requirements, and Software Design, Springer (2006)
17. Zachman, J. A.: A Framework for Information Systems Architecture. IBM Systems Journal. vol. 26, 3 (1987)
18. Tepandi, J., Piho, G., Liiv, I.: Domain Engineering for Cyber Defense: a Case Study and Implications. In: CCDCOE Conference on Cyber Conflict, Tallinn, accepted paper (2010)
19. Beck, K.: Test-Driven Development: By Example. Addison-Wesley (2003)
20. Evans, E.: Domain-Driven Design: Talking Complexity in the Heart of Software, Addison-Wesley (2004)
21. Zachman,J. A.: The Framework for Enterprise Architecture – Cell Definitions. ZIFA (2003)
22. Zachman, J. A.: The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing. (2003)
23. ASTM, E1578-06 Standard Guide for Laboratory Information Management Systems (LIMS). ASTM International (2006)

Software Process Improvement