Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090240

Computer Science Approach to Structural Meta Software Engineering and Operational Meta Software Engineering

Durgesh Samadhiya[*] *and Wen-Chih Chang*[*]

Department of Technology Management, Chung Hua University, Taiwan

Received: 11 Jun. 2014, Revised: 9 Sep. 2014, Accepted: 11 Sep. 2014
Published online: 1 Mar. 2015

Abstract: Information systems are being used for more and more applications are becoming more complicated and expensive, while at the same hand todays software systems demand increasing tremendously for sophisticated software engineering processes. However, there are a multitude of different development processes and techniques that each have various advantages and disadvantages, some of which relate to the problem domain or the development context. Computer software development processes have to pass from many faces to complete the development. There are many ways to solve a single problem in software development. Sometimes, in Structural engineering, developer is not able to decide which process will suit to that particular problem or in other words we can say that selecting a good process is a big issue in Structural Process Software engineering. The solution of such kind of problem can be found in the work to be done, and the task to be performed by the Operational process rather than structure of a process. This paper introduces the notion of process operationality and proposes method architecture to represent this operationality. Thus, Structural Process Engineering (SPE) becomes Operational Process Engineering (OPE).

Keywords: Process Engineering, Structural Method, process functionality

1 Introduction

Now a days Information System is the base of many activities of this real world. As the requirement increasing complexity of these information system based systems increasing. On the other hand development time is reducing and new processes arriving every moment. As a consequence, the traditional rigid IS engineering process are inadequate to provide the necessary support in new IS developments. New methods, more flexible and better adaptable to the situation of every IS development project, must be constructed. Process engineering in the ”field of information systems is the discipline to construct new processes from existing processes” [1] that focuses on ”the design, construction and evaluation of process, techniques and support tools for information system development [2]. Numerous development processes, based on the variety of paradigms, have been proposed over the years. Of these very few were successfully applied in the development of computer based systems.

Since their introduction various life cycle models and specific supporting techniques have played an important role in building software systems [3]. More recently the topic of software processes have received increased attention in software community. A software design approach called Evolution of Software Processes, is based on the emerging view that software processes - like software - also need to be evolved lest they become obsolete [4,5]. The aim of the evolution is to fulfill the needs of the people who perform the process and the developmental and organizational goals to be achieved. Another recent software design paradigm that can be seen as a generalization of software process evolution is process engineering. While there is a great overlapping of process engineering and process evolution activities, there are also some important divergence - in short process evolution is oriented more toward the improvement of existing processes and process engineering more toward the construction of new methods or processes.

Ralyte suggests that process engineering is facilitated if the goal of the process can be determined and raises some questions [6]:
(a) How can assurance be provided that the process to be enhanced, extended, or restricted is a good candidate process?
(b) What are the chances that at the process engineering intention stage, the process shall have to be discarded

[*] Corresponding author e-mail: samadhiya.durgesh@gmail.com, earnest@chu.edu.tw

© 2015 NSP
Natural Sciences Publishing Cor.

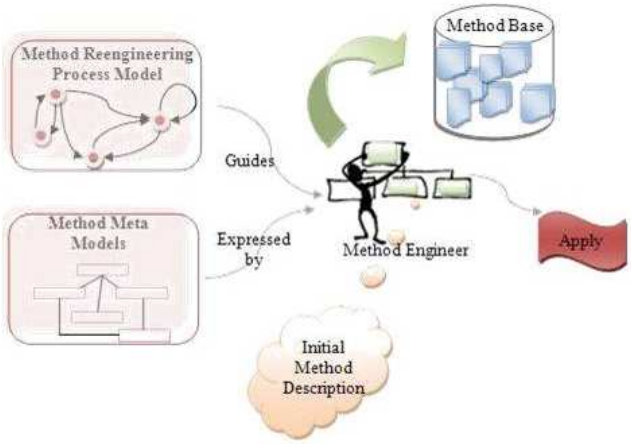

Fig. 1: Sample picture caption.

because its adaptation is very difficult?
(c) Should not some more exploratory work be done before committing to setting up process adaptation intentions?

The solution of these answers is Structural Meta Software Engineering but still new problem arise that no process is best in all the structures

Figure 1 defining a process reengineering process model that provides guidelines to reengineer an existing information system development process into reusable process. Figure 1 summaries our process reengineering approach. In this paper Section 1 is Introduction of the theme, section 2 and 3 explain the brief terminology of Structural Meta Software Engineering and Operational Meta Software Engineering, section 4 represents the Motivation and contribution of this paper, section 5 shows the preliminary result of this contribution and finally conclude in section 6 of this paper.

2 Structural Meta Software Engineering

Process engineering (PE) and structural meta software engineering (SMSE) focus on formalizing the use of process for systems development. The broader term, process engineering, is defined as the engineering discipline to design, construct and adapt processes, techniques and tools for systems development, a definition analogous to the IEEE definition of software engineering [7]. In this real world many Information Systems Development processes exist but no method is best in all situations. Structural meta software engineering has been proposed for developing or tailoring information system developing processes for specific structural projects [8]. Structural Meta Software Engineering is directed towards the controlled, formal and computer-assisted construction of structural process out of process fragments [9]. A structural process is an information system engineering process tailored and tuned to a particular structure. Structural processes are engineered in a formal and computer-assisted manner, out of standardized and proven building blocks stored in an electronic data base. These building blocks are called process storage and a process storage is a description of an information system engineering process, or any coherent part thereof [10, 11].

In the introduction of process engineering we discussed the development towards standardized information system engineering processes. Despite various attempts regarding the unified or universal process, it is concluded that there is no process which is best in all situations [12, 13, 14, 15, 16]. To anticipate to this problem, various approaches have been proposed, which are positioned in the so-called Structural Process Spectrum [17]. In spite of the large number of proposals that exist, there is some dissatisfaction with the notion of structure. Bucher (Bucher, Klesse, Kurpjuweit and Winter 2007) is concerned about the poor understanding of the notion of a structure [18] so there is need to find a way to reduce the number of possible situations [19].

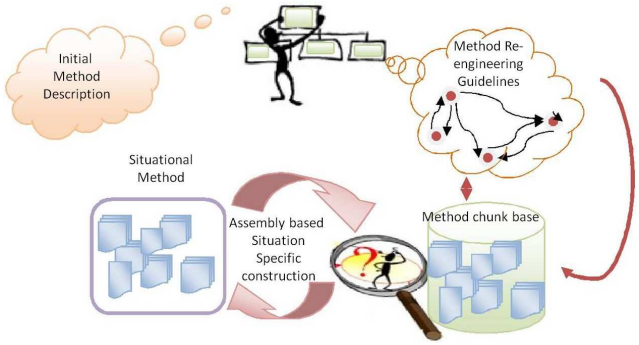

Fig. 2: Sample picture caption.

3 Motivation and Contribution

As already we discussed that, Ralyte suggests that process engineering is facilitated if the intention of the process can be determined and raises some questions [20]:

(a) How can assurance be provided that the process to be enhanced, extended, or restricted is a good best possible process?
(b) What are the chances that at the process engineering intention stage, the process shall have to be discarded

© 2015 NSP
Natural Sciences Publishing Cor.

because its adaptation is very difficult?
(c) Should not some more exploratory work be done before committing to setting up process adaptation goals?

The solution of these questions can be found in the work to be done, the task to be performed rather than structure of a process itself. Every computer software process having a full cycle consisting of

Requirements, Design and Construction Engineering and current state of the art in structural process engineering addresses the construction engineering phase. The other two stages help us in doing further exploratory work referred by Ralyte. At the Design stage we introduce the notion of process operationality and propose process architecture as an abstraction of this operationality. Thus, process engineering becomes Operational Meta Software Engineering (OMSE). Operationally close process architecture is selected, adapted, enhanced, restricted as needed. The task of construction handles the putting together of process features and structuring the process. Thus, we see a difference between structural process engineering and operational process engineering. At last but not least, have to explain about the Requirement Engineering that is upstream to Design Engineering. Here we introduce the notion of a process goal. Once processes with similar goals to the one to be engineered is found, a menu of processes to be adapted, enhanced, restricted is determined. This is further refined in the Design stage where architecture matching occurs. Again, a residue of processes is found and at this stage the architecture of the new process emerges as a set of functions connected together. Finally, this architecture is engineered from building blocks taken from the residue.

It can be notice that progressive selection in the Requirements and Design stages has
(i) Potential that assured the method to be enhanced, extended, or restricted is a good possible process.
(ii) Inappropriate processes, those having dissimilar goals and dissimilar architectures are rejected before the actual construction stage and therefore reduces the possibility of rejection and,
(iii) Enough exploratory work is done before committing to process features.

We can now state the aim of the thesis. We wish to move to goal process engineering so as to explore the context of structural process engineering more fully. As a result, process selection for adaptation shall be more appropriate and assured that the structural process engineering is progressing purposefully. It will considerably reduce the chance of process rejection at later stages. For this task introduce a 3-stage life cycle for goal process engineering.

In this life cycle, we introduce a process architecture matching phase that corresponds to our view of operational process engineering then the notion of process architecture is explained through a meta-model

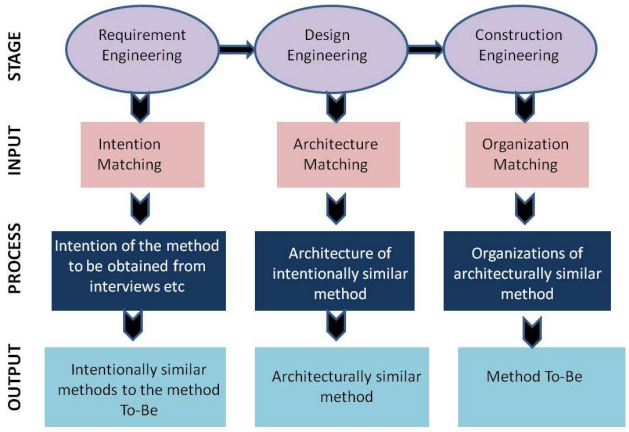

Fig. 3: Sample picture caption.

and a set of operations is defined that enables architecture matching. The two layers Design and Construction Engineering constitute the functional level of process engineering. Once this is developed we expect to put on top of the operational level an intentional level that shall still further raise the abstraction in terms of which process requirements shall be expressed.

4 Preliminary Results

4.1 Process Development Life Cycle

As shown in figure 3, we have developed process development life cycle for development, the Requirements Engineering stage consists of Intention Matching. First, the goal of the process To Be is elicited. The goal matching process uses synonym matching to identify intentionally similar processes that reside in the process storage. These processes become possible methods for the second stage of this cycle.

In the Design Engineering stage, the process engineer retrieves the architecture of each possible process from the process storage. That subset of these components and inter-relationships is selected which best meets the broad operational needs of the process To-Be. Such selections are made from all the possible processes and are synthesized together into the architecture of the desired process. In the Construction stage the architecture is populated with instances of the process features needed in the process.

4.2 Process Architecture

We have defined process architecture as an abstraction of the process that identifies its components and

© 2015 NSP
Natural Sciences Publishing Cor.

inter-relationships to highlight the externally visible operationality of the process. We use the class abstraction as a way of formally defining process architecture as follows:

Process architecture = process — process performs Function F

Process architecture is named and the name reflects the operation performed by the class of processes abstracted in the architecture.

The process architecture meta-model can be summarized as an architecture implemented as process organization and this organization shows the features of the process and their inter-relationships. An architecture can be atomic or complex and architectures can be related to one another by links. These links form a success or predecessor relationship between architectures. Links are labeled by their execution properties, Urgency and Necessity respectively.

4.2.1 Process Architecture Matching Process

In this design, the process engineer retrieves the process architecture of each possible process and these possible processes are obtained from the goal level as shown in figure 3. Match the operational needs of this process with the operational expressions of the candidate processes and make a new desired process architecture. In this process select only those that are useful for the architecture and refuse useless. The following operations have been proposed to do this:

i) Given a named architecture, rename it, ii) Create a new architecture, iii) Delete an existing architecture, iv) Nest, N architectures within another one, v) Un-nest architectures so that a nested architecture becomes visible at a higher level of nesting, vi) Change a link type, vii) Make a sequence of architectures by introducing an edge between them and defining their link type, viii) Eliminate a sequence.

4.3 Operational Process Engineering

Now time to explain the difference between structural Meta Software engineering and Operational Meta Software engineering as proposed. In the fragment based Structural Meta Softeware Engineering proposal [21], we have two fundamental elements

a) product and their structures b) Procedures and their execution order to develop the products.

Product and their structure show that, interest is the structure of products. Similarly, since the structure of a process is largely determined by the order of execution, interest is in process structure. Therefore, we can conclude that structural process engineering is centered around the structural aspects of processes. This focus on engineering the structure of processes de-emphasizes what the process does, what task it is good for. In fact, the determination of whether the process structure can carry out the project task at hand is based on the experience of the process engineer.

Operational process engineering puts process structure subordinate to process operationality. OMSE asks for an explicit determination and representation of process operationality in the form of process architectures. It is only after the architecture has been built that the issue of process structure is to be considered. In this sense, SMSE occupies the, downstream, construction engineering stage of our life cycle.

5 Conclusion and Future Work:

A process organization represents process features and their interconnections, Interest here is in the process concepts, inter-relationships between concepts, constraints, heuristics, guidelines and other such features of a process. It can be seen that process organization represents the structural aspects of processes. Alternatively, it defines the input to be given to a Computer Aided Method Engineering tool to engineer/implement the required process. We have selected the generic process model for representation of process organizations.

In order to finalize the construction-design stage interaction, we are also developing a set of operations for performing organization matching. This will allow us to adapt process organizations determined by architecture matching to our structural needs.

Thereafter we propose to develop the goal level. Process goal refers to the goal that the process fulfils. We shall develop the process goal meta-model and provide a precise definition of a goal. We aim to associate a goal with each process and as for, architecture matching, develop the process goal matching operations. Finally, the link between the goal and architecture levels shall be defined. Thus, the entire life cycle of figure 3 shall be covered. Once tool support is available, we shall experiment with our technique to establish its usefulness.

References

[1] F. Harmsen and M. Saeki (1996). ”Comparison of four method engineering languages”. In: Sjaak Brinkkemper et al. (eds.) Proceedings of the IFIP TC8, WG8.1/8.2 working conference on method engineering on Method engineering: principles of method construction and tool support: principles of method construction and tool support. January 1996, Atlanta, Georgia, United States. p.209-231.

[2] Sjaak Brinkkemper, Method engineering: engineering of information systems development methods and tools. Journal of Information and Software Technology, **38**, pp 275-280 (1996).

[3] Royce W W: Managing the Development of Large Software Systems: Concepts and Techniques. Proceedinas WESCON, 1970.

© 2015 NSP
Natural Sciences Publishing Cor.

[4] Bandinelli S., et al.: Modeling and Improving an Industrial Software Process. IEEE Trans. Soft. Engg (1995) 440-454.
[5] Bassili V et. al.: SELs Software Process - Improvement Program. IEEE Software, **12** (1995)83-87.
[6] Ralyt J; Deneckre R., Rolland C., Towards a Generic Model for Situational Method Engineering, Proc. CAiSE 2003, Eder J. amd Missikoff M. (eds.) LNCS 2681, Springer, 95-110.
[7] Brinkkemper, S.: Personal Email Communication to Authors, 29 Sept 2006.
[8] Brinkkemper S, Method engineering: engineering of information systems development methods and tools. Information and Software Technology, 275-280, (1996).
[9] Harmsen F., Situational Method Engineering, Moret Ernst and Young, 1997.
[10] Heym, M., Method Engineering: Specification and Integration of Development Methods for Information Systems (in German: Methoden-Engineering: Spezifikation und Integration von Entwicklungsmethoden fr Informationssysteme), Dissertation, Hochschule St. Gallen, Switzerland, 1993.
[11] Heym, M.,and H. sterle. Computer-aided methodology engineering. In: Information and Software Technology, **35**, pp. 345-354, 1993.
[12] Malouin, J.L., M. Landry, The Mirage of Universal Methods in Systems Design. In: Journal of Applied Systems Analysis, **10**, pp. 47-62, 1983.
[13] Humphrey, W.S., Managing the Software Process, Addison-Wesley, Reading, MA, 1990.
[14] Olle, T.W., J. Hagelstein, I.G. MacDonald, C. Rolland, H.G. Sol, F.J.M. van Assche, A.A. Verrijn-Stuart, Information Systems Methodologies - A Framework for Understanding, 2nd edition, Addison-Wesley, 1991; Vessey, I., and R.L. Glass, Applications-Based Methodologies. In: Information Systems Management, pp. 53-57, Fall 1994.
[15] Purba, S., D. Sawh, B. Shah, How to Manage a Successful Software Project - Methodologies, Techniques, Tools, John Wiley and Sons, New York, 1995.
[16] Parkinson, J., 60 Minute Software Strategies for Accelerating the Information Systems Delivery Process, John Wiley and Sons, New York, 1996.
[17] Harmsen, F., S. Brinkkemper, H. Oei, Situational Method Engineering for Information System Projects. In: Olle, T.W., and A.A. Verrijn Stuart (Eds.), Methods and Associated Tools for the Information Systems Life Cycle, Proceedings of the IFIP WG8.1 Working Conference CRIS94, North-Holland, pp. 169-194, Amsterdam, 1994.
[18] Bucher T., Klesse M., Kurpjuweit S. and Winter R. (2007). Situational Method Engineering On the Differentiation of ”Context” and ”Project Type”.. Situational Method Engineering 2007,33-48.
[19] Brner R, (2010), Applying Situational Method Engineering to the Development of Service Identification Methods, 16th Americas Conference on Information Systems, Lima, Peru, forthcoming. Paper 18. , 1-10.
[20] Ralyt J; Deneckre R., Rolland C., Towards a Generic Model for Situational Method Engineering, Proc. CAiSE 2003, Eder J. and Missikoff M. (eds.) LNCS 2681, Springer, 95-110.
[21] Brinkkemper S., Saeki M., Harmsen F., Assembly Techniques for Method Engineering, Proc. CAiSE 98, Pernici B. Hanos C. (eds.) LNCS 1413, Springer, 381-400.

Durgesh Samadhiya, He completed PhD from Chung Hua University, Taiwan. He completed many projects. He is serving for many International Journals around the worlds. He delivered speeches in many International conferences. His contribution in Software Engineering and Networking is appreciable.

Wen-Chih Chang is an associate professor in the the Department of Information Management at Chung Hua University, Taiwan. His research interests includes e-learning, game-based learning and data analysis.

© 2015 NSP
Natural Sciences Publishing Cor.