INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

NUCLEOCYCLE – THE BUILDING BLOCK OF PROCESS MODELS

A. Sadlauer, P. Hehenberger and K. Zeman

Keywords: process models, engineering design organisation, engineering micro-management

1. Introduction and related work

Engineering design projects exist in various forms with respect to the discipline, duration, cost and necessary effort. Some projects only require a small amount of time, money and labour. Others might span over years, require an enormous amount of money and occupy a horde of engineers. Such different projects on a general scale need different methods, procedures and planning. Consequently process models (in other disciplines referred to with different names, e.g. in software engineering as Software Development Lifecycle Models [Ruparelia 2010] are available in various forms, for small-scale to large-scale projects. Ruparelia [Ruparelia 2010] explains these process models as "a conceptual framework or process that considers the structure of the stages involved in the development of an application from its initial feasibility study through to its deployment in the field and maintenance". Academia (e.g. [Gericke 2011], [Maier 2011] and [Wynn 2005]) discusses the advantages and disadvantages of these different design methodologies and process models. Major aspects in this discussion are commonalities and differences among design approaches, especially considering multi-disciplinary approaches. In addition to the many advantages for the usage of process models (e.g. according to [Gericke 2011] rationalise creative work, limit oblivion of important aspects, simplify planning, communication between disciplines etc.), they also improve the communication between the different hierarchical levels. Nevertheless, the acceptance of these approaches in industry varies significantly. Potentially this has to do with the majority of process models being too general [Wynn 2005]. In software engineering, process models are established and are a crucial part of research and development. In contrast, in mechanical engineering, the distress for companies often is not great enough to introduce methodical development. Their currently applied approach in their opinion is successful enough.

One reason for this probably is the effort needed for the implementation of specific design methodologies or process models. Large-scale methodologies (e.g. "Integrierte Produktentwicklung" [Ehrlenspiel 2007]) possibly require a complete restructuring of the product development department whereas smaller-scale methodologies (e.g. Rapid Application Development, such as SCRUM or other Agile methods [Ruparelia 2010]) lack the range necessary for the overall design of complex systems. Certain process models need special training and the manuals span over thousands of pages. Particularly for the executive agents (engineer or designer, Kerzner [Kerzner 2007] also refers to them as executive employees), this is not practical. This frequently results in the dangerous situation of having specialists with the sole function of applying the specific method. To be successful, however, all the involved parties have to be familiar with the method.

Even when a methodical development approach is accepted, obstacles for engineering design are manifold. Through literature study, a community survey and personal experience, Maier and Störrle [Maier 2011] were able to list six challenges with respective characteristics for engineering design:

development, collaboration, products and services, formality, pragmatics and flexibility. Process models to a certain degree are able to address these challenges.

Gericke and Blessing [Gericke 2011] provide an exhaustive literature review of design methodologies and process models across disciplines (with focus on engineering and architecture). While highlighting the differences between process models and the different opinions in academia, they also point out, that similarities of process models across disciplines are that "…they have a generic core of stages…" and that "…they propose a stepwise, iterative process". In their opinion, the differences might not be caused by the disciplines, but by the differing characteristics of the particular products. One example for such a design approach is the "Integrierte Produkterstellung" of Ehrlenspiel [Ehrlenspiel 2007]. At the core of the problem solving process he places the human being as a problem solver and suggests the TOTE (Test, Observe, Test, Exit) model as a micro-cycle for individual problem solving. The sequential arrangement of such micro-cycles leads to a procedure cycle ("Vorgehenszyklus") which can be used to form a large-scale design process model. This TOTE model represents a link between the agent and a process model.

Ehrlenspiel [Ehrlenspiel 2007] addresses the different tasks of product development within his "Integrierte Produkterstellung" with different scale cycle models. The V-Model [VDI2206 2003] also distinguishes between a Macro- and a Micro-cycle. Sadlauer et al. [Sadlauer 2012] propose the usage of different process models for different scale design tasks in a similar way. Depending on the task size, the hierarchical level of the project and how it is managed, certain process models are more appropriate than others are.

In this paper, the idea of a cycle on the agent level as a building block to shape larger, more complex, process models (cycles) is extended to a micro-management methodology between the executing agent and the first level of decision makers (project managers and line managers). The project manager, equipped with building blocks at the agent level, should be able to build every desired process model, irrespective the discipline. The agent has a clear structure that provides guidance to solve/execute individual tasks. This micro-management methodology relieves the agent of the additional organisational burdens, that have increased significantly in the last decades [Kerzner 2001] and allows concentration on the actual engineering activity, while at the same time being practical and easy to implement.

In the following, chapter 2 discusses the connection of analysis, synthesis and decision analysis within a process model. Based on the aspects of chapter 2, chapter 3 introduces a bottom-up approach for process models. After this, chapter 4 explains how to apply this bottom-up approach within a company structure and chapter 5 provides an example for the application of the methodology within a small project. At the end, chapter 6 gives a conclusion and an outlook on future research plans.

2. Process models and decision analysis

When discussing decision taking, it is necessary to define who is taking them. In this context, as already mentioned in the previous chapter, we distinguish between the agent level and the first managerial level. The agent level consists of the executing agents that actually perform the design tasks. The first managerial level represents the lowest stage in the decision hierarchy and comprises of project managers (PM) and line managers (LM). In a traditional company hierarchy, often one individual fulfils these duties, in other company hierarchy structures the relationship between the managers and the agents can be more complicated (see also chapter 4).

Gericke and Blessing [Gericke 2011] state that several authors found that analysis and synthesis emerge together in different disciplines. This is especially the case in new product development, where the available information base is limited. However, even when there is information available (e.g. in the design of variants), individual solutions are often new [Maier 2011]. When mentioning the analysis and synthesis process, authors often neglect the immediate evaluation of findings that even though it mostly happens subconsciously [Ehrlenspiel 2007], is a crucial part of it. Another aspect closely linked to the analysis and synthesis process is the decision analysis. The main goals of the decision analysis are the proper discussion of the results and result preparation that should support the decision and provide additional documentation of the results.

There are several methods regarding the decision analysis mentioned in literature (e.g. [Kerzner 2001] and [Ehrlenspiel 2007]). Although the agent that performed the analysis/synthesis/evaluation cycle might be partial towards certain findings, this agent is still the most appropriate one to perform the decision analysis, since he is most likely the one that has the insight required to document the results and apply the respective decision analysis. Ehrlenspiel [Ehrlenspiel 2007] states there is a difference between evaluation and decision. A decision includes taking the responsibility for it, which often requires courage, whereas an evaluation result is "only" a recommendation others can either follow or not. While there are several methods (e.g. AHP, SWOT) for finding and supporting a decision, it ultimately is a subjective decision [Ehrlenspiel 2007].
There are process models that already include decision taking (e.g. Stage-Gate), but they lack the definition as to who has to take the decision and do not distinguish between agents and managers. Especially in early product development, decisions are mostly made under uncertainty. As Kerzner [Kerzner 2001] mentions, uncertainty, in contrast to risk (with assigned probabilities), leaves the option that there is no single best strategy. Therefore, decision analysis, especially in conceptual design has a very high priority. The following chapter is addressing the combination of problem solving and decision analysis and the relationship between executing agents and managers.

3. Bottom-up approach for process models – The Nucleocycle

As suggested in [Sadlauer 2012], it is possible to represent the design process and the associated documentation using distinct layers (design and documentation layers) that are closely connected. Process models provide a structure or a guideline supporting the execution of extensive projects on the design layer. On this layer, the actual design activities with the respective design tools (CAx-tools, sometimes also referred to as authoring tools) take place. The majority of known process models in engineering are top-down approaches and provide very little information for the lowest, in the design process included hierarchal level. Nevertheless, there are certain guidelines for the executing agent on how to solve/execute a specific task best (e.g. [Ehrlenspiel 2007] and [Wynn 2005]). For the experienced designers such processes are standard procedures and they are filling the gap between the process models and the individual as problem solver. For inexperienced designers it is often difficult to perform a task according to these guidelines, while at same time being burdened with organisational aspects of the task. What complicates the situation additionally is that most process models are rather abstract and according to Wynn and Clarkson [Wynn 2005] too general. The design process, especially on the executing level, requires a substantial amount of creativity and flexibility. Nevertheless, it requires a certain structure (i.e. guidelines) at higher hierarchy and abstraction levels.
The Nucleocycle is supposed to be the integral building block of all process models and in the current state it is based on empiric observations only. It should be implemented on the agent level and include one manageable task that is encapsulated between strategic decisions of the next high hierarchy level. The Nucleocycle should be present on the design layer, which is introduced in [Sadlauer 2012]. The connection to and from the Nucleocycles within the design layer is realized with a control bus. The information transmitted on this control bus includes the detailed definition of the task (e.g. start condition and location of relevant data). A data bus represents the connection between the design layer and the documentation layer. Through this data bus, the agent has to have access to project data stored on the documentation layer. Of course, this requires that all the data (including the rationale) is available there in a structured way. PLM-Systems are one possibility to provide such a structure.
The inspiration for the Nucleocycle was the problem solving cycle [VDI2206 2003] and there specifically the analysis and synthesis step and the analysis and evaluation step. The differences are that the analysis and synthesis step has (subconscious) Evaluation included and the analysis and evaluation step has increased value and is of cyclic nature. The focus is strictly on the aspects with no direct interaction with the superior hierarchy level. All the aspects that require the superior hierarchy (e.g. decision-making and goal formulation) are outsourced. The Nucleocycle (as shown in Figure 1) itself incorporates an analysis/synthesis/evaluation cycle (ASEC) and a subsequent result/evaluation/analysis cycle (REAC).
The ASEC is comprised of the actual solution finding process, whereas the REAC is comprised of the decision analysis, which should support the decision maker in the decision process and provide the

relevant information. These two cycles are distinct for two specific reasons. First, if necessary, two different individuals can carry out the cycles and second, the decision analysis of the REAC does not influence the solution-finding process in the ASEC. This second aspect counteracts the tendency of engineers to think solution oriented too early in the design process, which limits innovation and creativity. Each of the cycles has five ports. Two control bus ports (in- and output), two data bus ports (which basically are pointers to where the relevant information can be retrieved from and where the generated data has to be stored) and a bidirectional emergency port, where any kind of emergency can be reported between the person in charge (e.g. the project manager) and the agent.

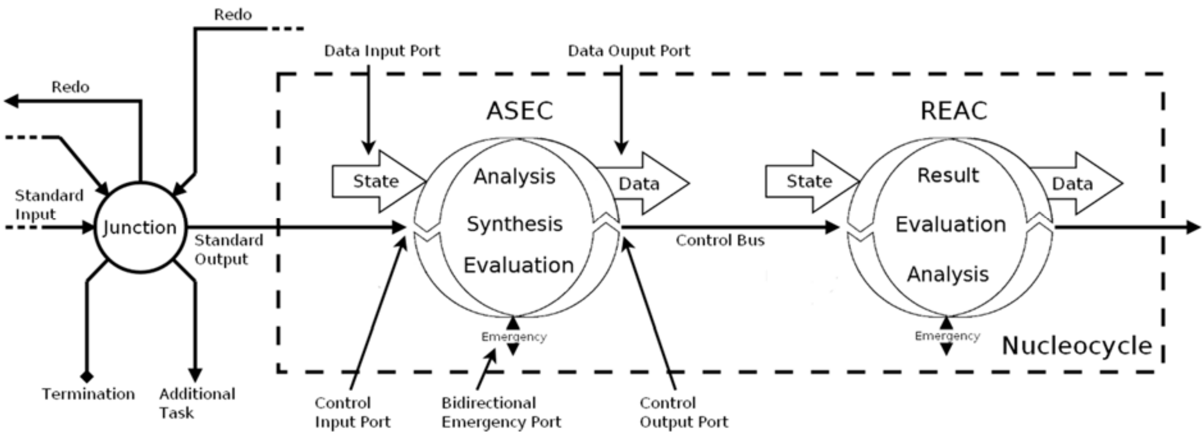

Figure 1. Nucleocycle and Junction

In the following, the Nucleocycle is referenced to only one individual performing the task. However, the task can be assigned to a group as well. The procedure in either case is similar. The agent gets the information that he has to start a certain task over the control bus in the Nucleocycle. He then has to process the ordered task with an ASEC and a subsequent REAC. If the cycle content is too much for one individual or one group to handle, it should be split.

Another, generic element, the so-called "Junction" (also shown in Figure 1 as a circle), comprises all the aspects that concern the hierarchy levels that have decision-making power. These are e.g. the decision making process and the goal definition process which in turn determines the progress continuation and the routing of the control bus. A Junction has a variable number of input and output ports of the control bus. The standard input and output indicate the linear progression of the project. Redo connections indicate iteration loops and a termination exit indicates the end of a (sub-) project.

The Junctions include the entities (persons) that have the power of decision for the respective tasks. While their ultimately should be one person in charge, it is often the case that the results are discussed in a committee with the project leader reaching the decision based on the discussion results. The Junctions can be realized with meetings (e.g. jour fix regarding the project) or a sequential approval process (which in practice often is too complicated and too slow). The decision is made based on the information provided by the executing agent. The more information is available, the smaller is the degree of uncertainty. One threat to successful product development in this situation is a large overhead. With the managerial ladder being too long, the important information might not reach the ultimate decision taker.

It is important to note, that the structure of Nucleocycles is not fixed after being initially set, but should provide a living, adaptable structure. Depending on the individual results, changes of constraints and requirements and modifications of the structure have to be possible. This is especially crucial in the case of development projects (problems) where the goal is not explicitly known or defined.

With the two described elements (Nucleocycle and Junctions) serving as building blocks on the design layer and the control bus as connections between them, it should be possible to form the structures of any desired process model. The task to combine the results of different Nucleocycles can again be addressed with another Nucleocycle. The emergency port provides a possibility of exception case handling as a connection between the executive agent and the reference person on the first management level (e.g. the PM). The PM has the responsibility to plan and direct the project. He has to provide the skeleton of the whole design project (e.g. in the shape of a V-Model [VDI 2206 2003]), set the relevant milestones with Junctions and divide the project into manageable tasks that can be handled within Nucleocycles. The functional leadership has to assign the work force to the individual Nucleocycles [Kerzner 2001].

Regarding strategies on how the network of Nucleocycles and Junctions should be set up, guidelines from business process management are used. Mendling et al. [Mendling et al. 2010] propose Seven Process Model Guidelines (7PMG). While they are developed in the context of business process models, five of them certainly should be considered in product development too. These are: 1) the usage of as few elements as possible, 2) Minimization of routing paths per element, 3) Usage of one starting point and one ending point, 4) Model as structured as possible and 5) Decomposition of a model with more than a certain amount of elements (the number can vary e.g. depending on the experience of the involved personel). During project lifetime, of course, these rules have to be adapted as necessary, but at the beginning, they should be obeyed. The PM should apply the adapted rules of Mendling et al. [Mendling et. al. 2010] accordingly. Each project segment should only include no more than the previously defined maximum number of elements and have one beginning and one end. Segments with more elements should be split. This, however, does not rule out that the project manager has to manage more than one segment of a project. Figure 2 shows an example structure of two design phases (requirements design and functional design) with Nucleocycles and Junctions that are connected via the control bus (including an iteration loop). In contrast to business process models the intention is not to depict all the possible connections, but only the ones that were actually used. In Figure 2 for example the second Junction did not require an iteration loop, therefore it is not depicted.

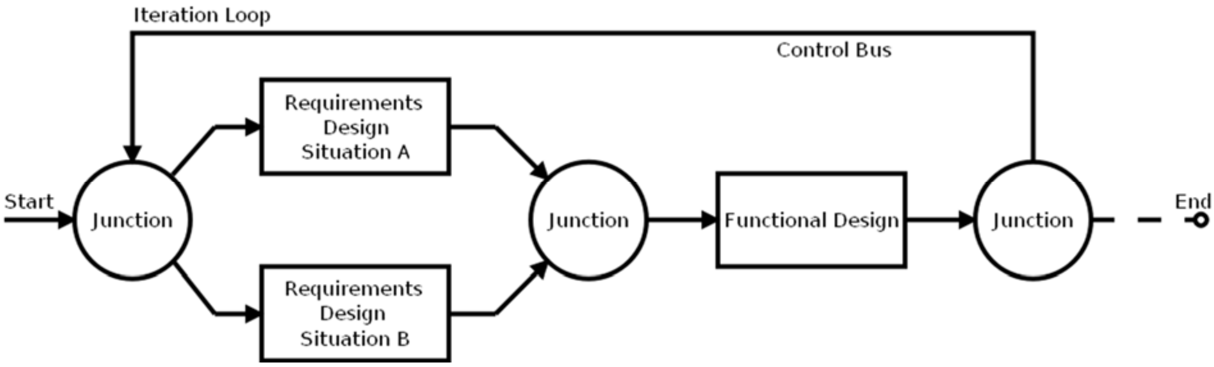

Figure 2. Nucleocycle structure

On the lowest level of project management, only the usage of Nucleocycles, Junctions (for the decisions), and control connections make sense. For the agent, the control bus has to provide the detailed task definition including the following information:

- required detail level
- start condition (e.g. start date)
- end condition (e.g. end date)
- assigned agents (relevant for group assignments)
- relevant contact information
- location of necessary data
- target destination for results.

The control bus has to provide the decision relevant information (e.g. date of completion and location of results) for the Junctions. This control bus communication should be performed in a standardized way (e.g. with software-support). In addition it is crucial, that the rest of the necessary information is stored in the database with well defined access authorization. The currently often applied practice of just sending e-mails with attachments is too error prone.

Figure 3 shows the vision of the authors of how different process models can be used for different (partial) projects as suggested in [Sadlauer 2012] with the addition of the Nucleocycle structure on the executive level (highest granularity). In this figure, the design layer and documentation layer provide the foundation of the project. The V-model is used at the highest project hierarchy, the phases of the V-model are realized with Rapid Application Development (RAD) and the steps of the RAD on the agent level are organized with Nucleocycles and Junctions.

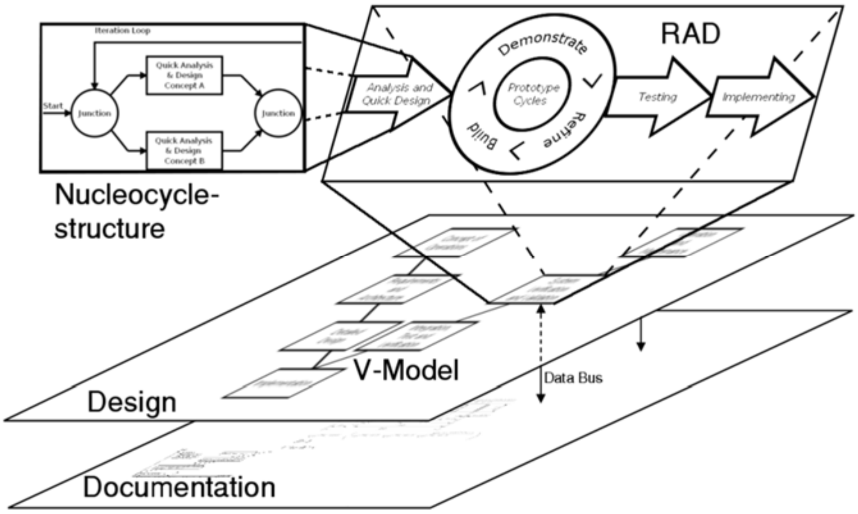

Figure 3. Project segmentation with different process models acc. to [Sadlauer 2012]

4. The Nucleocycle in a matrix organisation

Kerzner [Kerzner 2001, p.113] states, “The matrix organisational form is an attempt to combine the advantages of the pure functional structure and the product organisational structure”. Therefore, the authors chose this organisational structure to exemplify the abilities of the Nucleocycle in a complex organisational situation. Figure 4 shows a simple example of matrix organisation structure within a technical environment with three hierarchical levels. The Chief Technical Officer (CTO) is at the top of the hierarchy and has the ultimate power of decision. Underneath him are a couple of PM and LM having either project or functional responsibility.

The agents actually performing the respective tasks (lowest hierarchical level) are placed at the crossings of the functional and project responsibility (depicted with circles). The PM and LM are responsible for the progress of the project and take intermediate decisions.

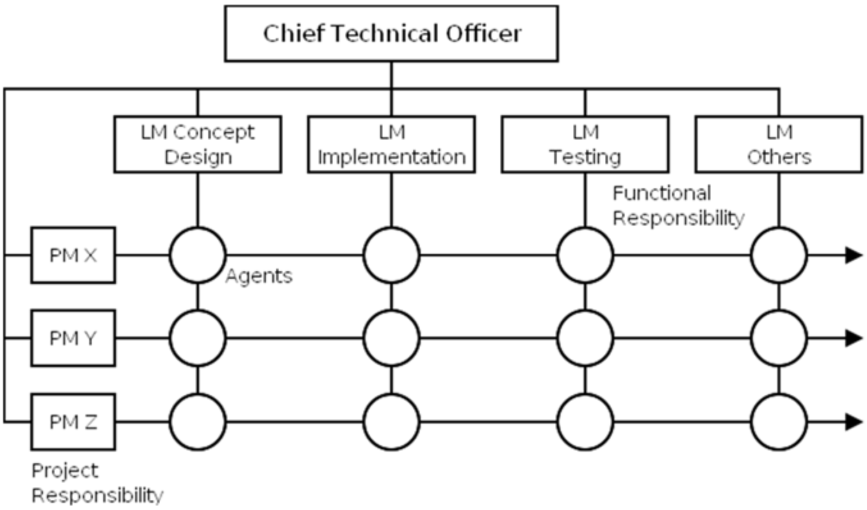

Figure 4. Example structure of a matrix organization in a technical environment modeled after [Kerzner 2001]

It is the duty of the PM to structure and manage the project. It is the duty of the department (or line) manager (LM) to provide the workforce and the functional requirements. The respective line manager has to decide who is performing the task specified by the project manager. The allocation of staff for the project however can be rather difficult, as the specific project has different levels of priorities for the department and the project manager. Limitations in quality as well as in amount of the workforce [Kerzner 2001] complicate the situation additionally. The relationship between the project manager, the line manager and the employee (in this context also referred to as executive agent) depends heavily on the type of project management implemented in the company [Kerzner 2001]. Kerzner [Kerzner 2001] defines four different dimensions for these relationships from the viewpoint of the project manager:

1. What does the PM negotiate for (deliverables, informal/formal information, full time allocation)?
2. From whom does the employee take technical direction (LM, PM or LM, PM)?
3. From whom does the PM receive functional progress (LM, assigned employee through LM, employee)?
4. Who evaluates the employees performance (LM only, LM with input from PM, PM)?

With the Nucleocycle, the technical direction and the functional progress report can be specified in more detail. The technical direction comes from the Junction (which includes the decision takers). Changes to this initial directive can be made through the emergency port with PM or the LM as the contact person. The emergency port represents the interaction between the contact person on the higher hierarchy with the agent in case of exceptions of any kind. The reporting of the functional progress works slightly different compared to the direction, as the executive agent has to document the progress and all the results on the documentation layer. All involved parties should have access to this documentation.

5. Case study example: Conveyor system design with Nucleocycle-structure

In this chapter, we describe a first initial trial project with the purpose to evaluate the Nucleocycle-structure. Therefore the conveyor system example provided in [Sadlauer 2012] is investigated in detail. The goal of our use case was the conceptual development of a conveyor system with the result being a working virtual model in the Mechatronics Concept Designer (MCD). A team of two people performed this first test of the methodology. One person represented the executing agent, the other the first managerial level. Since the goal was a conceptual model of the conveyor system, only the first phases in project development were relevant. Therefore, the preliminary design and detail design phases were not considered.

5.1 Prerequisites

The manager uses the process model of Follmer et al. [Follmer 2011] as a reference framework and the Nucleocycle-structure for the micromanagement and organisation of all the tasks. The initial requirements were available from another project. These included the demand that the conveyor system is modular and fitted in an already existing environment. The initial project plan consisted of one Nucleocycle for each design phase with one Junction in between the phases. Figure 5 depicts this first outline. Between the major design phases, Junctions regulate the approval of prior steps, before initiating the next one. A file structure on a network-drive enabled document management. File names included version numbers to allow for version management. All relevant notification exchange happened via e-mail including the saving position of the relevant files. A work-package-file for each separated task included all the relevant information, such as mentioned in chapter 3.

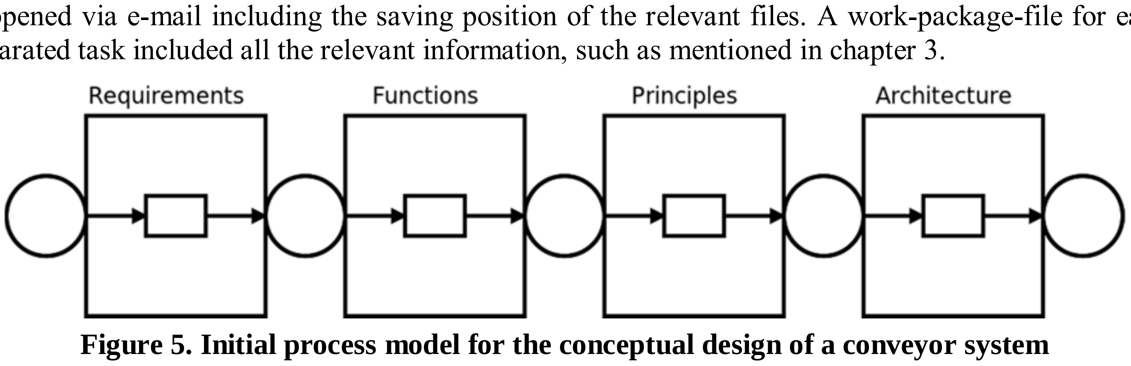

Figure 5. Initial process model for the conceptual design of a conveyor system

5.2 Project progress

After the first initial meeting that included the definition of the project plan and the first work package the project started. Since there was only one executing agent involved, he addressed the work-packages sequentially. The first package consisted of the analysis of the requirements and constraints. The agent had to identify and combine the customer's wishes, legal conditions and technical limitations. In the ASEC, the agent evaluated the legal situation, customer wishes, and technical limitations and translated them into requirements. In the subsequent REAC, the results of the ASEC were analysed and prepared for the decision taker. Even though the work packages were supposedly clear and precise, at the second meeting it was clear that the definition of the task was not narrow enough and the information was too overwhelming. Hence, an iteration of the first work package was necessary with focus just on the functional requirements. This resulted in a new version of the work-file for the first work-package. Since the changes were minor, the second work-package for the functional design was also set already.

In the project management, this resulted in two routing paths from the Junction. One path represented the iteration back to the requirements and the other path the continuation to the functional design. One consequence of iterations is the possibility of different Junctions (decisions regarding different phases in different iterations) being addressed in the same meeting (context, document, ...). In the case of concurrent engineering, this would be the standard. The question how to address several simultaneous action paths at the current state of the Nucleocycle-structure remains unanswered and is a topic of future research.

In between the second and the third meeting, the agent worked on the identification of the functional requirements and then on the functional structure based on these requirements. In the third meeting, the functional structure did not include the differentiation between mass, energy and information flow. Consequently, an additional iteration was necessary. Based on the experience from the previous work-packages, we separated the principle design into two work-packages. One package was for the identification and description of the solution principles, the other one for the realisation of the individual modules in the MCD.

For the architectural design, we applied the same strategy. The agent analysed the path definition in one work-package and then connected the modules in the MCD in another work-package accordingly. For this second architectural design work package, the boundary to the preliminary design becomes blurred. For the preliminary design, however, more details would have to be provided. The agent concluded the project with a final documentation phase. Figure 6 shows the final project plan with the respective meetings represented with numbers at the relevant Junctions and rerouting paths.

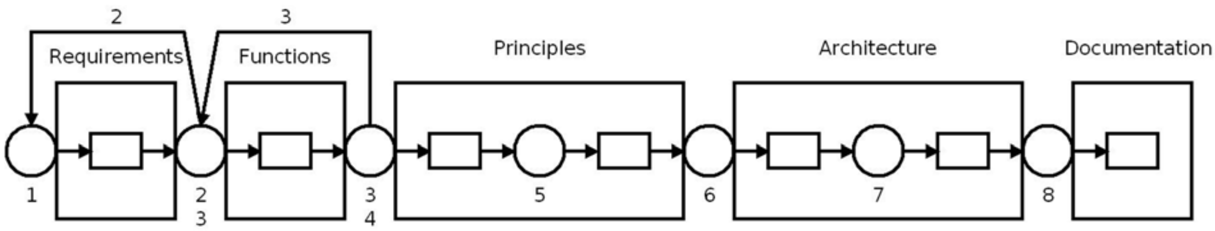

Figure 6. Final project plan

5.3 Results

The project showed certain weaknesses and certain strengths of the Nucleocycle-structure. The Nucleocycle is a possible way to manage the tasks of the executing agent. The methods used within are up to the agent. The quality of the output however does not only depend on the qualification of the agent, but also on the clarity of the task definition. The differentiation between the ASEC and the REAC is fruitful, since especially with growing project size, the information required for decisions needs to be condensed. The REAC is a first filter from the agent (who should be the person best informed) dividing the necessary information from the unnecessary. The Junctions are viable for the rerouting of the task dimension.

There are certain aspects that are not yet fully discussed at the current state. One aspect is the decision management. In the current state, the decision management (allocation of meetings to Junctions and their traceability) is not addressed in a satisfactory manner. The numbering system such as shown in Figure 6, with increasing project size, can cause confusion. Iterations and parallel execution of tasks can result in several Junctions being addressed in one meeting (as exemplified with meeting 3 in Figure 6). Due to this, it is difficult for the PM to keep the overview of the project. A possible solution would be having a time-bar with the meetings aligned chronologically with a depiction of specific meetings that show the connection to the relevant Junctions.

A further aspect that has yet to be addresses is the differentiation between higher hierarchy levels. For certain decisions, only the next highest hierarchy is required, for others maybe even the highest. This would possibly require a differentiation between the Junctions (e.g. by different colours or sizes).

6. Conclusion and future work

This paper introduced the Nucleocycle as a building block that addresses micromanagement of the engineering design process in the context of process models. The Nucleocycle is an extension of the TOTE-Model of Ehrlenspiel [Ehrlenspiel 2007] achieved by including decision analysis and an additional hierarchical level, which is responsible for decision taking. Each individual task that does not require a decision by a higher hierarchy of a project requires a Nucleocycle. The decision processes take place in the Junctions and a control bus connects the Junctions and Nucleocycles. With these elements, it is possible to build a process model from the bottom up. The example project of a conveyor system planning process gives an insight into the application of the proposed micromanagement with Nucleocycles.

In the small-scale test project, with the Nucleocycle as a guideline for the agent satisfying results were obtained. It however required some effort to follow the separation of the ASEC and the REAC. The amount of documentation required was high, but not necessarily higher than in traditional projects. The application of the Nucleocycle structure revealed certain obstacles and questions that need to be addressed in the future. These aspects of further interest are the handling of exceptions, the application for concurrent engineering, the inclusion of higher hierarchical levels and the decision management. A next step is to structure the current state of the Nucleocycle including the time and decision dimension according to standard scientific research methodology and extend the focus to systems engineering. Afterwards we intend to take the input from the small project and apply the improved Nucleocycle-structure to a larger, preferably industrial project.

Acknowledgement

This work was kindly supported by the Austrian Center of Competence in Mechatronics (ACCM), a K2-Center of the COMET/K2 program, which is aided by funds of the Austrian Republic and the Provincial Government of Upper Austria. The authors thank all involved partners for their support.

References

Ehrlenspiel, K., "Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit", Carl Hanser Verlag, Vienna, Munich, 2007.

Follmer, M., Hehenberger, P., Zeman, K.,"Model-based approach for the reliability prediction of mechatronic systems on the system-level", LNCS Computer Aided Systems Theory - Eurocast 2011, Volume 6928/2012, Moreno-Díaz, R., Quesada-Arencibia, A., Pichler, F. (eds.), Springer-Verlag Berlin Heidelberg, 2012, pp. 105-112.

Gericke, K., Blessing, L., "Comparisons of design Methodologies and process models across disciplines: a literature review", Proceedings of the International Conference on Engineering Design – ICED 2011, Volume 2, Technical University of Denmark, Kopenhagen, 2011, pp. 344-355.

Kerzner, H. "Project Management – a systems approach to planning, scheduling and controlling" 7th Edition, Wiley & Sons Inc., New York, 2001.

Maier, A.M., Störrle, H., "What are the characteristics of engineering design processes?", Proceedings of the International Conference on Engineering Design – ICED 2011, Volume 1, Technical University of Denmark, Kopenhagen, 2011, pp. 188-189.

Mendling, J., Reijers, H.A., van der Aalst, W.M.P., "Seven process modeling guidelines (7PMG)", Information and Software Technology, Volume 52, Issue 2, Elsevier, Amsterdam, 2010, pp. 127-136.

Ruparelia. N. "Software Development Lifecycle Models", Software Engineering Notes, Volume 8, Number 3, Association for computing machinery, 2010, pp. 8-13.

Sadlauer, A., Hehenberger, P., Zeman, K., "Perspectives for the usage of design and modelling languages in mechatronics systems development", Proceedings of TMCE 2012, ed. I. Horváth, A. Albers, M. Behrendt and Z. Rusák, Karlsruhe Institute of Technology, Delft University of Technology, Karlsruhe, 2012, in press.

VDI Guideline 2206, "Design methodology for mechatronic systems", VDI Publishing Group, Düsseldorf, 2003.

Wynn, D., Clarkson, P.J., "Models of Designing", Design process improvement – A review of current practice, Clarkson, P.J., Eckert, C. (eds.), Springer-Verlag London, 2005, pp. 34-59.

DI Alfred Sadlauer
Junior Researcher
Johannes Kepler University, Institute for Computer Aided Methods in Mechanical Engineering
Altenberger Strasse 69, 4040 Linz, Austria
Telephone: +43 732 2468 6564
Telefax: +43 732 2468 6542
Email: alfred.sadlauer@jku.at
URL: http://came.mechatronik.uni-linz.ac.at