INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

INTERFACE ERRORS – AN ISSUE IN COMPLEX VIRTUAL PRODUCT CREATION

Rainer Stark[1, 2], Henryk Stöckert[1]
(1) Berlin Institute of Technology (2) Fraunhofer-Institute for Production Systems and Design Technology

ABSTRACT

Growing complexity in virtual product creation is, amongst other reasons, caused by increasing numbers of process steps and the associated increase of process links. Therefore, conformity regarding formal process requirements becomes more and more relevant. Further decrease of vertical integration, international and interdisciplinary cooperation, outsourcing approaches and enhanced concurrent engineering fortify this evolution. A large but limited number of root causes around this complex of problems has been identified and is focused on the interface between process steps. Whereas business administration research already recognized interfaces as heavy risk for errors, industrial product creation and engineering research has not yet reached this viewpoint. This is why process interface requirements are often ill described and therefore requirement compliance is still insufficient. According cognitions are mainly based on analyses in the automotive and aviation industries.
Developed solution approaches shall improve process creation chains concerning internal requirements in context of growing functional specialization of product developers and further increasing division of labor.

Keywords: Collaborative engineering, Process Interface, Interface Error, Virtual Product Creation, socio-technical system, internal requirements

1 INTRODUCTION

Virtual product creation (VPC) is the combination of product development and production processes, regarding all computer-aided technologies and addresses the necessary technological components as well as processes and methods to enable successful product creation in this digital environment.
Increasing numbers of computer-aided process steps add more and more regulative challenges to the work of product developers, aside issues of constructive development itself. The goal still is to fully manage the entire product lifecycle digitally, from the first flash of inspiration until complete reuse or recycling. This leads to growing complexity of the product creation process chain with growing numbers of single working tasks. Reasons are further division of labor and individual software tools for almost every single process step. Exemplary tasks and according IT-systems are shown in Table 1.

Table 1: Connected Tasks and Systems in Virtual Product Creation

Action	System
Design	Computer Aided Design (CAD)
Analysis	Computer Aided Engineering (CAE)
Validation	Digital Mockup (DMU)
Managing Release Status	Product Data Management (PDM)
Exchanging Product Data	Product Data Management (PDM)
Collaboration	Collaborative Engineering Tools
Feasibility Checks	Computer Aided Manufacturing (CAM)

The very relevant field of working psychology and organizational psychology has only little relevance in today's engineering practice, which leads to working conditions that force the product developer to act wrongly even despite knowing better. Error prevention on product creation process interfaces has to be aware of technical surrounding conditions as well as of socio-technical influences. Both issues are addressed in this paper. The aim of this paper is to present analytical results from research in the described context and to introduce a general solution approach. The status-quo analysis is based on industrial experience of the authors, mainly in aerospace, automotive and plant engineering industries, as well as on literature research. Relevant findings have been adapted from all kinds of research areas to the problems in collaborative engineering.

2 INTERFACES IN VIRTUAL PRODUCT CREATION

Even if the term collaboration is heavily used in these days, most of today's development processes are mainly cooperative processes with significant error potential. The relevant difference is that in cooperative engineering, work is not done at the same time with the same material. On the contrary, this is the case in collaborative engineering [1]. Consequently, there has to be a handover of partly completed work to the process successor who has to continue working with the preliminary results. A similar case is information handover to parallel processes without ending a task. These handover situations are called interfaces. In this context, interfaces shall not be interpreted technically but procedural only, even if digital products always have to use technical interfaces to move from process to process. Generally, an interface describes a system boundary where interchange with the environment is possible in the form of information, energy or material [2]. In entrepreneurial context, interfaces can be hierarchical or heterarchical. Hierarchical interfaces are located between different levels of power and authority, like between supervisors and subordinates. Interfaces on equal levels of power and authority are called heterarchical and are considerably more problematic since every party is dependent on the willingness to cooperate with other parties [3]. Unfortunately, everyday business is mainly based on heterarchical cooperation.

Interfaces in product creation can be distinguished in five cooperation directions. Cooperators from the viewpoint of the original equipment manufacturer (OEM) can be:

- Colleagues at the OEM in the same department with the same functional role
- Colleagues at the OEM in the same department with a different functional role
- Colleagues at the OEM in a different department
- External workforces at the OEM
- Employees of supplier or partner companies (subcontracted or independent)

The more system boundaries have to be crossed, the more obstacles have to be overcome. In case of company networks, development partner companies shall also be treated as suppliers, since the barriers of different company cultures and spatial as well as infrastructure separation are comparable. shows the increasing number of barricades between process successor and predecessor with increasing distance between cooperators. Differentiations between in-house employees and external workforces are neglected in the present problem analysis.

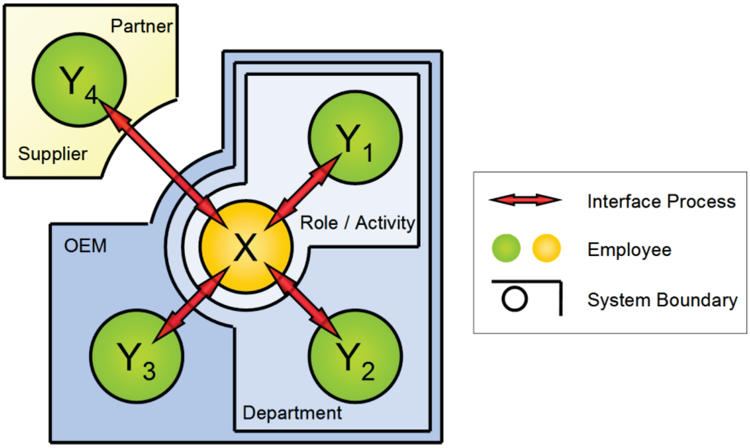

Figure 1: Interfaces in today's virtual product creation

Tasks which require coordinated actions of two individuals in the same department of the same company with the same profession and role (X-Y_1) are affected by interface obstacles, mainly psychological ones. This is exactly the case with cooperating product developers. Cooperation within the department (X-Y_2) is mainly hindered by job or task specific obstacles. Responsibility and socialization barriers become relevant between departments (X-Y_3) and are even stronger between different companies (X-Y_4) with individual company cultures. Striking examples are the different languages between the marketing and the manufacturing department or between companies with strongly different business cultures such as the online auction platform 'ebay' and the automotive OEM 'Volkswagen'. Competitive positions increase in the same order, which reduces unrestricted information flows. Table 2 shows the mentioned accumulating effects at a glance.

Table 2: System boundaries and increasing obstacles to cooperation

Interface type (cf. Figure 1)	**System boundary**	**Obstacles**
$X - Y_1$	• Individual	• Psychological factors
$X - Y_2$	• Individual • Role / Activity	• Job- and task-specialization [4]
$X - Y_3$	• Individual • Role / Activity • Department	• Divisional egoisms • Different socialization • Different professions
$X - Y_4$	• Individual • Role / Activity • Department • Corporation	• Corporate culture • Limited information flow • Obligations of secrecy

Characteristic for interface processes and interface process requirements is the limited duration of the requirements. They are only valid within the product creation process and have absolutely no impact on the product requirements, namely customer requirements.

3 COOPERATIVE AND COLLABORATIVE ENGINEERING

One approach to reduce time to market and costs is to parallelize product development processes and to shift them into earlier phases of the overall process. This is termed frontloading [5] or collaborative virtual engineering [6] and shown in Figure 2.

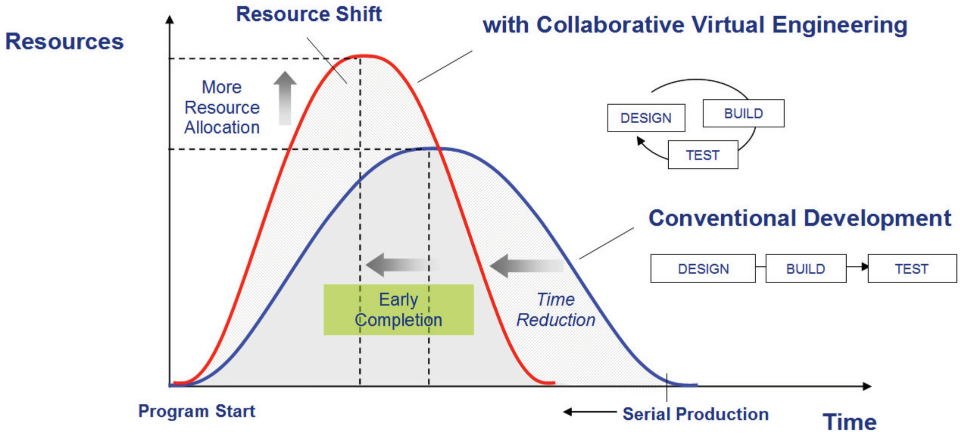

Figure 2: Reducing time to market with frontloading [6]

The illustration assumes a constant integral of both the blue and the red curve, which means equal resource requirements for conventional sequential and for collaborative development. Additional targets connected with this approach are to perform as many tasks as possible “first time right” to achieve early model integrity and, of course, to improve process efficiency. So the request is not only to keep resources equal, but to diminish the integral of the red curve.

Looking closer to organizational processes, there clearly have to be additional expenses in parallelized process runs. Human resources are a discrete measured variable. This means, engineers can solely work on a task or not. Consequently, disposition to tasks is possible in whole and single units only. Therefore the pool of tasks has to be sliced into equal resource intervals, visualized as horizontal lanes in Figure 3. Tasks in the same lane do not necessarily have to be executed by the same resource.

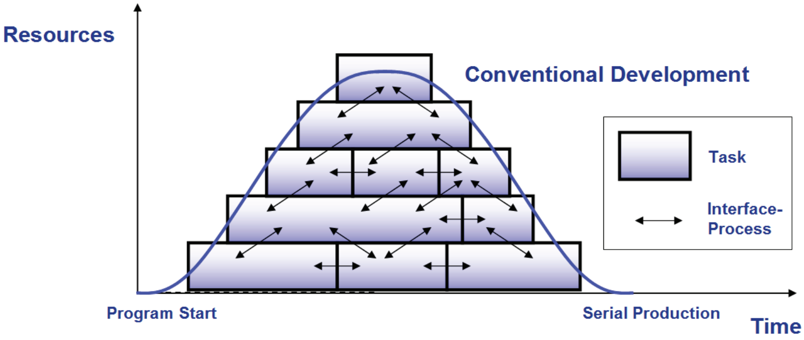

Figure 3: Conventional Development with resource allocation

The crux of every process chain or process net is the interface. An interface is every transition from one task to another and is symbolized in Figure 3 by double arrows which more precisely represent interface processes. The only two purposes of these interface processes are to fully describe the requirements of the receiving to the sending process and to accurately lead over process step results from the sending to the receiving processes. Even if regularly not noticed very well, the number of interface processes can exceed the number of productive processes very easily. Assuming that every process has at least one predecessor and one successor, the number of interface processes is almost equal to the number of product creation processes. This is already the case without any parallelization at all. The more parallelization, specialization and division of labor, the more interface processes occur in the overall process chain, visualized in Figure 4.

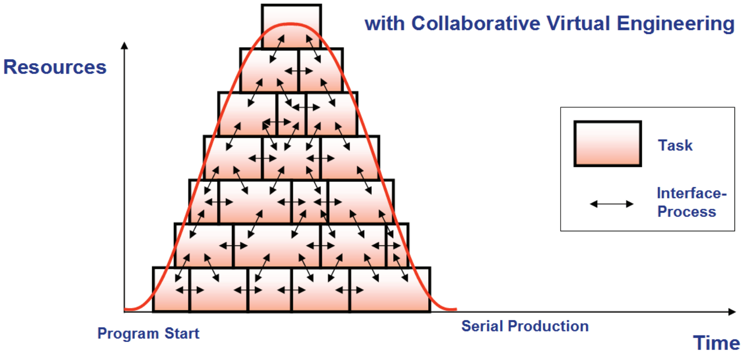

Figure 4: Heavily parallelized and specialized development with resource allocation

Interfaces are prone to error and can be seen as overhead efforts of product creation process steps. This is why they are regarded as idle performance, which does not have positive effects on value creation. However, they are necessary for the overall process to work. Nevertheless, it is worthwhile to reduce overheads as far as possible. The amount of overhead an interface implicates is, statistically, the sum of administration, coordination, start-up time for beginning new activities and possible errors with subsequent error-correction and iteration efforts.

Figure 5 shows why with increasing numbers of interface processes, average process duration cannot stay unaffected, but decreases the overall efficiency. Formerly implicit interface efforts (double arrows in yellow bubbles), in industrial context often called "hidden engineering factory", are explicitly visualized as a time factor within the according task (yellow columns on the lower task).

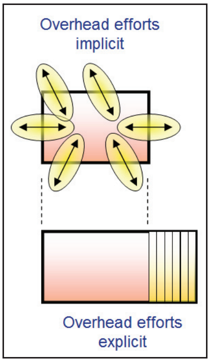

Figure 5: Overhead

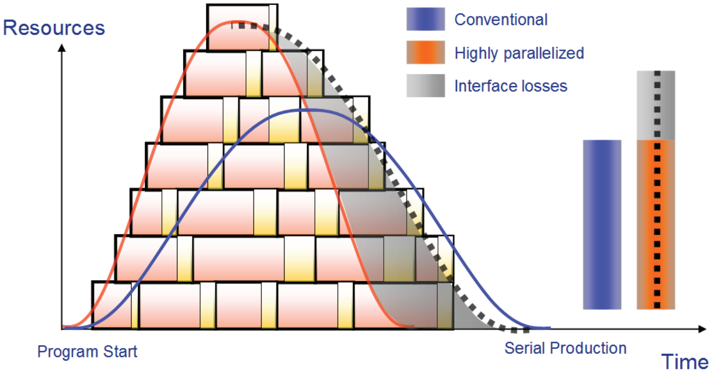

Figure 6: Overall interface losses

Resulting process development times are to be seen in Figure 6. Besides the increased overall process duration, the worst case scenario leads even to a delayed start of serial production, which can be even later than in case of conventional development.

4 ERROR CLASSIFICATION

Error, mistake, failure and defect are just a few terms for nonconformity which are not well-defined or differentiated. This mainly results from the usage of these terms in many different domains like psychology, jurisprudence and philology and in everyday language as well. This weakened exact differentiations over time. As there are manifold meanings of 'error' and also many different terms for different kinds of errors, a classification is necessary for distinctive analysis and for developing error prevention mechanisms in virtual product creation. Figure 7 shows the developed error classification.

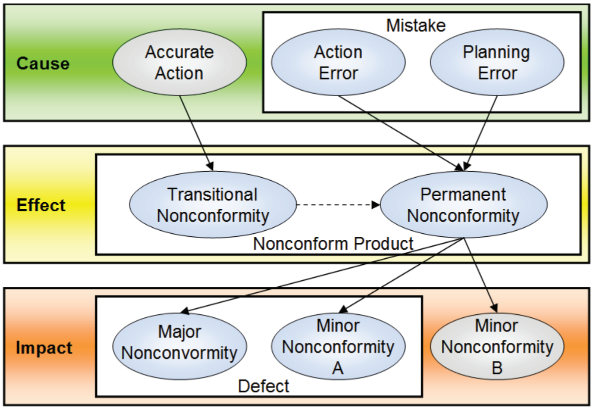

Figure 7: Error Classification

The ISO 9000 certification standard describes nonconformity as the non-fulfilment of a requirement. The reason for nonconformities is based on precedent actions, which can be accurate actions or mistakes. The mistake again can be of two types. It can be an action error, which is also called 'slip', and means an erroneous action. The actor did not want to do the action in this way. He rather had the right plan to fulfill the action, but somehow could not manage to do so. It was the right plan with the wrong effect. Or it can be a planning error, where the action was exactly as planned and predicted, but the plan was wrong. So it was a wrong plan with the right effect concerning the plan, but not fulfilling the initial requirement anyway. Both action errors and planning errors lead to permanent nonconformities. As Figure 7 shows, accurate actions can also lead to non-compliant results.

An example for nonconformity in virtual product creation is a component with drill-holes in a wrong position. Besides this permanent nonconformity, which is an error in the second it is created, nonconformities can also be of transitional nature. A transitional nonconformity would be the same component with a missing drill-hole. Transitional errors are characteristic for transitional states in processes. In the next step, the transitional error can become a permanent one, if the drill-hole is set in the wrong position. Or it can be neutralized by completing the process and setting the missing drill-hole in the correct position. Almost every product creation process needs transitional nonconformities to eventually achieve accurate results. Transitional errors are a serious issue for automatic error testing, since computer algorithms are not able to see this well-planned transitional state as necessary and therefore can only check the error status at the end of the process.

Most interesting is the impact of the mentioned actions and effects on the final result, i.e. the product. Regarding the severity of the impact, it can be a major nonconformity or one of two classes of minor nonconformity. Minor nonconformities of type B (cf. Figure 7) do not have any negative impact on the usability of the product and are therefore removed in Figure 8, which shows the eventual financial impact. They do no harm to anybody. An example for minor conformities of type B would be a wrong font size in digital product model annotations. Minor nonconformities of type A have little impact on the product's usability. This could be a design fault, leading to comfort-reducing errors like squeaking movable components. If an error affects the utility of the product entirely or at least partially, like a missing drill-hole to connect other components, it is classified as major nonconformity. Major nonconformities and minor nonconformities of type A are both termed 'defects'.

Nnonconformities are not a problem per se for the company. Only errors that lead to preventable costs are of relevance for companies. So which errors do in fact lead to preventable costs? Figure 8 illustrates the above discussed error types and resulting financial impacts for companies.

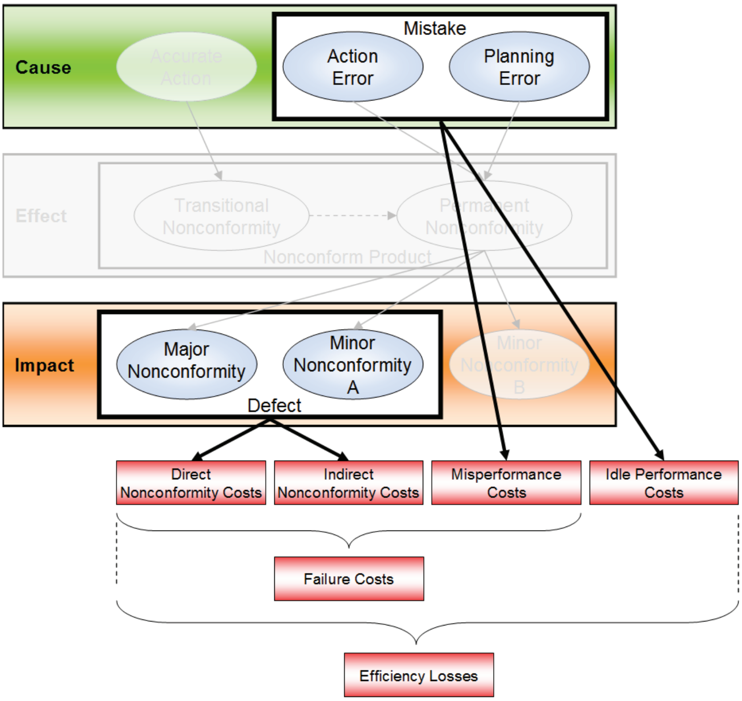

Figure 8: Financial impacts of different error types

For customers, errors are only relevant if they lead to product defects. For the company, other errors can do financial harm, too. Efficiency losses evolve from defects and from mistakes as well. Even if mistakes do not lead to defects, they are to be avoided, since they lead to efficiency losses in every case. Mistakes can either be idle performance, which means that efforts do not lead to negative effects, but not to usable results either. Idle performance is simply a waste of time by performing unnecessary actions. As this waste of time is still working time which has to be paid by the company in labor, infrastructure and operating costs, the result is idle performance costs. Or mistakes can lead to misperformance costs. These actions are not only unnecessary, but they in fact do harm to processes or products as they are unnecessary and they additionally lead to product defects. Product defects themselves lead to nonconformity costs. Direct nonconformity costs include rework, delays, failure analysis, downtimes and rejects. Indirect nonconformity costs are a possibility of direct ones and can occur as costs for correction, losses due to sales reduction, reputation losses, warranty costs, contractual penalties and increased transaction costs in general.

5 DIFFERENT ROOT-CAUSES OF INTERFACE ERRORS

Complex systems consist of many interacting components. Variables affect the system as well as they affect other variables. Therefore, complex systems show nonlinear behaviour and effects cannot be tracked back to a single source [7]. The investigated interface processes in virtual product creation gain complexity from the product development process itself, from human behaviour and from social interactivity between human process participants. This leads to a limited number of root-causes for interface errors in virtual product creation. Four categories of error causes have been identified and are clearly distinguishable. Described in detail are the following four influencing factors:

- Domain-specific issues (project size, heavy subcontracting, global development cooperation)
- Technical issues (number of IT-sytems, lacking system compatibility, frequent system changes)
- Organizational issues (inaccurate process descriptions, dysfunctional information policies, information amounts inappropriate to the context, inaccurate surveillance mechanisms) and
- Issues from the perspective of behavioral psychology (role and motivation conflicts, dysfunctional incentive systems).

5.1 Domain-specific issues

Increasing numbers of process interfaces, as described earlier, lead to heavily growing complexity. Besides the already mentioned interface aggrandizement by frontloading and development process parallelization, there are several other causes due to virtual product creation specifics. Most obvious is the increase of project size. Popular examples are recent airliners from Boeing and Airbus with around one million connected CAD-models per aircraft or huge construction projects such as the Burj Dubai skyscraper, Terminal 3 at Beijing Airport and the three gorges dam spanning the Yangtze River. Implicated in the growing project size is the number of contributors numbered in staff members or involved parties, as well as the overall project duration. Product complexity is another characteristic of the growing “size” of projects coming along with the trend to mass customization [8].

Decreased vertical integration is a main target in product development industry and therefore heavily pushed. Vertical integration degrees of 20 percent in case of the compact car smart® in 2004 and around 60 percent at the aircraft company Airbus in 2005 are still not considered as adequate [9]. Especially the loss of executive authority from supervisors to project staff due to subcontracting constellations leads to extended response times and elongation of decision making processes in engineering and organizational questions. Under closer scrutiny, this situation turns out to be a fortified agency dilemma.

Additional issues derive from the global spread of project partners in development cooperation which leads to heavy friction losses due to intercultural differences, language barriers, different core times with the need to synchronize simultaneous working tasks, different systems of units such as the metric and the imperial units system and last but not least, sensitivities concerning the distribution of power.

All these evolutions become noticeable on process interfaces, where requirements for follow-up processes grow rapidly, but are not adequately specified and distributed. Product developers have to take into account every possible connection to many other components in several different layouts (mechanical, electronic, control and software), but are oftentimes not aware of the specifications of those components. In many cases, requirements are not fully specified, not correctly interpreted by product engineers or no adequate help provided by far distant requirement engineers.

5.2 Technical issues

Technological progress of supportive systems has strong influence on company success and competitiveness. Functional assistance is accompanied by the necessity to get familiar with more and more little or large technical helpers. In the terminology of Müller, interaction problems are added to already existing work-related problems [10].

Lacking system compatibilities prevent continuous working processes and lead to media breaks. Data format conversion is a media break as well. This is because the manner of processing data is different in case of different formats. Every media break jeopardizes interface processes as it increases the risk of error. Possible reasons for system incompatibilities are:

- Different creation dates of software applications, e.g. legacy systems
- Different programming paradigms such as functional, modular or object oriented with limited compatibilities for interface communication
- Various software vendors
- Different computer operating systems
- Concealed data models and interface definitions
- Non-standardized proprietary software developments
- Conceptual incompatible system architectures

These reasons may have evolved over time or may have been created intentionally. Political decisions like concealing data models to gain competitive advantage have strong influence on this issue.

Another technical issue is the growing variety of IT-Systems in virtual product. Automotive industries confirm this observation and state that design engineers have to reliably handle up to 15 different software applications. This means, assuming three year system change intervals, users have to learn to handle five new system versions every single year [11]. Especially problematic are legacy system. They are common in the aircraft industry due to long development periods (7 years). Besides technically obsolete and hard to learn, they often do no longer represent latest company processes and workflows.

5.3 Organizational issues

Organizational issues are not limited to the field of virtual product creation but though have strong influence on the discussed situation. Communication problems are the most common and generalizable over different industry branches. The biblical Babylonian language confusion is literally daily occurrence in highly complex organizations and mainly stem from different vocabularies of the affected parties. This again reflects the situation pictured in Figure 1, where higher organizational distance comes along with language barriers and more need for coordination. Implicit incentive systems lead to conflict of interests between different departments which again are a source for communication problems since information cannot be interchanged without self-restriction. The risk to put oneself or one’s department in disadvantage is simply to high.

Standard operation procedures (SOP, not to be mixed up with ‘Start of production’) are an effective approach to coordinate complex processes with multiple process participants and including different areas of expertise. Nevertheless, SOPs are also part of the problem, since they are very difficult to handle if too numerous and often changing. Resembling the “Seven Rights” of Plowman for logistic processes [12], requirements for Standard operating procedures are

- The right standard operating procedure (fitting to the work process)
- In the right amount (all and only relevant SOPs)
- In the right condition (concerning correctness, completeness and usability)
- In the right place (in whatever system the engineer needs to be guided)
- On the right time (i.e. without delays)
- To the right cost (with cost defined as effort for the engineer to find the needed documents)
- For the right consumer (in appropriate level of detail according to the single engineers expertise).

Information policy is another heavy issue in organizational context. Complete information is an academic concept which in reality is not observable very often. Everyday life is characterized by information asymmetry, information incompleteness, information imperfectness, in short, lack of information. Lack of information is also one reason for interface errors, since it forces engineers to make wrong decisions by not knowing better. Information that is not provided or provided too late has been identified by Schömig as the number one determining factor on design quality in product development [13]. Several reasons for the inadequate compliance to specifications, like identified by Fournies [14] and Williams [15], can be assigned to the lack of information category:

- Demands are unknown (different engineers know different requirements)
- Reasons for demands are unknown (Requirements are incomprehensible and therefore hard to meet)
- Demand is seen as erroneous (erroneous demands lead to mistrust and fast refusal of demands)
- Own solution is seen as better solution (better solution is obvious due to misunderstood demand)
- Ways of applying to demands are unknown (nobody knows how to meet the requirement)
- Demands are prioritized improperly (time is running out and less important things are done first)

Problems with lacking information are characteristic for interface processes. This is because demands originate from domains that are not in the area of expertise of the employee. In contrast, demands concerning design issues, come from the engineer's area of expertise, so decisions can be made based on education, experience and common sense. Demands originating from processes on the opposite side of the interface cannot be understood without sufficient additional information on the relevant context. Requirements for finite element method (FEM) simulation, for example, are not self-explanatory and seem very vague to design engineers. Nevertheless, they heavily influence the success of FEM simulations by their way of working. Since design engineers, who are in general the center of competence in their working environment concerning design questions, are trained to question given approaches, systems and designs, there is high risk of misinterpretation in case of information shortage. Not questioning designs and ways of working would be seen as carelessness and is undesired in product development. Insufficient surveillance and improper consequences of surveillance results are another organizational issue which could have been discussed here.

5.4 Behavioral Psychology

As surveillance has strong influence on human behaviour, the just mentioned insufficient surveillance is also an issue concerning behavioral psychology. Already the assumption that working processes or work results are monitored influences human behaviour. Luhmann calls this the "expectation of expectations", [16], which means to expect what is being expecting from oneself. The expectation is considered in one's actions, whether or not surveillance actually takes place. Churchill & Copper [17] and Osterloh [18] confirm this theory particularly in corporate environments. Heavily influential to the expectation of expectations are former experiences with surveillance environments [19], which make long-term consistency a determining factor of success.

Main reason for lacking surveillance activities is, besides financial reasons from the company point of view, the inherent discomfort of surveillance for both the supervisor and the supervised. These feelings originate from inner conflicts [20] in both cases. Employees dither between admitting missing knowledge or capabilities and showing outstanding performance. Admitting fallibility may lead to being educated further on, whereas pretending perfection promises financial and non-financial gratifications. A motivational conflict can be experienced. Supervisors on the contrary experience a role conflict between judging the employee and having a consulting and amicable relationship to their subordinates. Coming to surveillance, the employee shows defensive behaviour, the supervisor avoidant behaviour [20]. Both are better off if surveillance is disregarded. Even trickier, not the real situation causes the behaviour of all participants but the individual and subjective perception of it.

"People do things for which they are rewarded and, conversely, do not do those things for which they are not rewarded" [10]. This is why incentives are another heavy influencing factor on human behaviour. Not only explicitly chosen corporate incentive systems are relevant. Especially those which are in place implicitly as results of conscious or unconscious behaviour or resulting from explicit incentive systems are relevant, too. Expressed in the basic model of motivation psychology, the connection between person (P), environment (E) and behaviour (B) is shown in Lewin's Equation (1) [21]. The Lewin's Equation is a heuristic rather than a mathematical equation.

$$B = f(P, E) \tag{1}$$

Motivational influences on persons (P) can be divided in human basic needs of cultural or biological nature, individual and unconscious implicit motives and consciously chosen, individual explicit motives [22]. The situation consists of incentives that are recognized by the person. Unperceived incentives are noneffective at all. Another constraint to incentives is that only those corresponding to personal motives are effective to this very person. Corresponding motives and incentives are called motive congruent.

A broad variety of the different mentioned incentives surround engineers as well as any other employees. Regarding interface processes, the incentive has extra impact on behaviour, since there is no direct hierarchy between cooperation-partners. Cooperation is therefore dependent on the willingness to cooperate, as already explained in paragraph 2. Effective incentives and the absence of obstacles and misguided incentives are prerequisites for good cooperation. Otherwise, company internal competitive situations will occur and may lead to considerable loss of efficiency in important business processes. The following example, visualized in Figure 9 shows a dysfunctional incentive system.

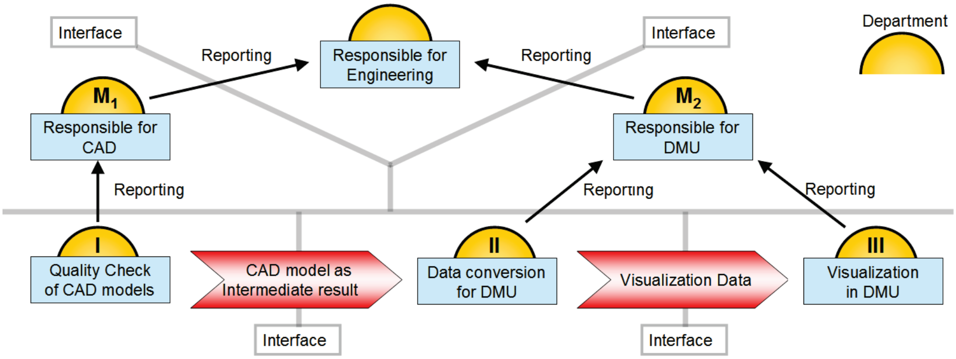

Figure 9: Example for misguided incentives in virtual product creation

Two engineering departments I & II are working on consecutive processes. Department I has to do quality checks on CAD models from data exchange processes with subcontractors, which is preliminary work for department II. Department II is dependent on correct CAD-models to convert them into several data formats for 3D visualization in the Digital Mock up (DMU). Incorrect data can only be converted after time-consuming rework. Both departments are measured by their managers by the number of delivered models. As there is a need for earlier visualization and department I is known as the bottleneck process, they are requested to deliver 20% more CAD models per day. To follow this requirement, the quality checks of department I cannot be as accurate as before any longer. As a consequence Department II gets more inadequate data. Less visualization data can be provided due to the extra effort of corrections.

In the end, only department I has the chance to improve the situation by delivering less models in total but with better quality. But the department has no incentive to do so. The deployed incentive turns out to be a misguided incentive. Despite the aimed increase of CAD model throughput, it actually decreased. Resolving this situation is even more complex, if department I & II are situated in different company locations or they are even in different companies.

6 ERROR PREVENTION APPROACHES AND FURTHER RESEARCH

Interface processes face problems from many root-causes in engineering, technical, organizational, social and psychological context. Error prevention approaches therefore have to address every single relevant problem area. Socio-technical-systems aim at exactly this wide-ranging and complex combination of circumstances in organizational and corporate environments [23]. According to the mentioned main determining factors, approaches to face the interface error issue have to cover exactly those four areas of

- Standard Operating Procedures
- Information policy
- Surveillance mechanisms and
- Incentive systems.

For the variety of necessary changes, a detailed systems analysis and an effective change management with relevant company environmental factors in mind is crucial. Figure 10 shows the general approach and necessary steps to adapt virtual product creation in existing product development companies for best avoidance of interface errors and efficiency losses on process interfaces.

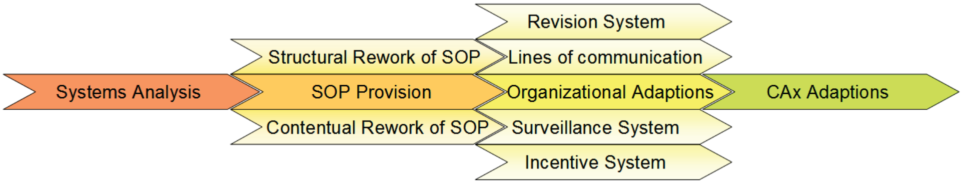

Figure 10: Change Process to avoiding interface errors in virtual product creation

Current innovation research shows that innovation chances are best in applying traditional and well-established methods to other business contexts. Various domains have therefore been reviewed to find solution strategies that are adaptable to the situation in virtual product creation. The following methods have been found usable to analyze, describe and improve virtual product creation:

- Surveillance mechanisms from accounting
- Incentive systems from organizational psychology
- Interface analyses from business administration research
- Behavioral patterns from principal agency theories
- External effects from macroeconomics

Concepts for all shown process steps in Figure 10 have already been developed and are now in the evaluation phase. Additionally, further industry interviews are planned to substantiate gathered information from industrial context and to enlarge the sample size of the previous research activities.

REFERENCES

[1] Lu, S. C-Y, et al. A scientific foundation of collaborative engineering. In *Annals of the CIRP*, 2007, Vol. 56/2, pp. 605-634

[2] Bossel, H. *Modellbildung und Simulation: Konzepte, Verfahren und Modelle zum Verhalten dynamischer Systeme,* 1994 (Vieweg, Braunschweig)

[3] Meckl, R. *Schnittstellenmanagement bei unternehmungsübergreifender Zusammenarbeit. Regensburger Diskussionsbeiträge zur Wirtschaftswissenschaft,* 1994 (University of Regensburg, Regensburg)

[4] Brockhoff, K. *Management organisatorischer Schnittstellen unter besonderer Berücksichtigung der Koordination von Marketingbereichen mit Forschung und Entwicklung; Berichte aus den Sitzungen der Joachim-Jungius-Gesellschaft der Wissenschaften e.V.*, 1994 (Vandenhoeck & Ruprecht, Göttingen)

[5] Thomke, S. and Fujimoto, T. The Effect of "Front-Loading" Problem-Solving on Product Development Performance. *Journal of Product Innovation Management,* 2000, 17(2), 128-142

[6] Dankwort, C.W.; Ovtcharova, J.; Weidlich, R. *A* Concept of Engineering Objects for Collaborative Virtual Engineering: Automotive Case Study. In *Proceedings of the ProSTEP iViP Science Days*, Dresden, 2003, pp. 161-172 (Fraunhofer IRB, Stuttgart)

[7] Richter, K. and J.-M. Rost. *Komplexe Systeme*, 2004 (Fischer-Taschenbuch-Verlag, Frankfurt am Main)

[8] Dean P. R., Tu Y. L., Xue D. A framework for generating product production information for mass customization. In *The International Journal of Advanced Manufacturing Technology,* 2008, Vol. 38, No. 11-12, pp. 1244-1259 (Springer: London)

[9] Geisler, B. Airbus wächst auf 56 000 Mitarbeiter. In *Hamburger Abendblatt*, 2005, 10(15) (Springer Verlag, Hamburg)

[10] Müller, T. Planung und Entwicklung. In *Handbuch Arbeitswissenschaft*, H. Luczak (ed.) 1997, p. XVI (Schäffer-Poeschel, Stuttgart)

[11] Krause, F.-L., et al. Nachhaltigkeit durch virtuelle Produktentwicklung. In *Tagungsband zum XII.*

Internationalen Produktionstechnischen Kolloquium (PTK), Berlin, 2007, pp.191-201 (Berlin)
[12] Plowman, E.G. *Lectures on elements of business logistics*, 1964 (Stanford University - Graduate School of Business, Stanford)
[13] Schöïnig, D. *Fehlerreduzierung durch die Einbindung eines Qualitäts-Informationssystems in die Konstruktion*, 2001 (Vulkan-Verlag, Essen)
[14] Fournies, F.F. *Why employees don't do what they're supposed to do*, 1999 (McGraw-Hill, New York)
[15] Williams, V. *Why Employees Fail to Meet Performance Expectations,* 2005, (Empowerment Publishers)
[16] Luhmann, N. *Soziale Systeme: Grundriß einer allgemeinen Theorie*, 2008 (Suhrkamp, Frankfurt am Main)
[17]Churchill, N.C. and Cooper, W.W. A Field Study of Auditing as a Mechanism for Organizational Control - A Hypothesis. In *International Conference on Operational Research and the Social Sciences,* London, 1966, p. XXXIV (Tavistock: London)
[18] Osterloh, B.W. *Die betriebliche Investitionskontrolle : Probleme der Kontrolle betrieblicher Investitionen unter besonderer Berücksichtigung der Kontrolle der Investitionsplanung*, 1974 (Stuttgart)
[19] Churchill, N.C. and Cooper, W.W. *Effects of auditing records: Individual task accomplishment and organization objectives*, 1964 (Wiley, New York)
[20] Lattmann, C. *Die Leistungsbeurteilung als Führungsmittel*, 1994 (Physica-Verlag, Heidelberg)
[21] Lewin, K., et al. *Feldtheorie in den Sozialwissenschaften*, 1963 (Huber, Bern)
[22] Heckhausen, J. and Heckhausen, H. Motivation und Handeln, 2006 (Springer, Berlin, Heidelberg)
[23] Geels, F.W. *Technological transitions and system innovations: a co-evolutionary and socio-technical analysis*, 2005 (Elgar, Cheltenham)

Contact: Rainer Stark
Berlin Institute of Technology
Institute for Machine Tools and Factory Management
Chair for Industrial Information Technology
10587 Berlin, Germany
Tel.: Int +49 30 39006 243
Email: Rainer.Stark@tu-berlin.de
URL: http://www.iit.tu-berlin.de

Rainer Stark is the Director of the Chair for Industrial Information Technology of the Berlin Institute of Technology and the Director of the Department of Virtual Product Creation of the Fraunhofer Institute for Production Systems and Design Technology. His key domains of research are context relevant digital information modeling, intuitive usability of and functional interaction with virtual prototypes, function driven virtual product creation as well as product development methods and processes.

Contact: Henryk Stöckert
Berlin Institute of Technology
Institute for Machine Tools and Factory Management
Chair in Industrial Information Technology
10587 Berlin, Germany
Tel.: Int +49 30 39006 218
Email: Henryk.Stoeckert@ipk.fhg.de
URL: http://www.iit.tu-berlin.de

Henryk Stöckert is research engineer in the information management department. His education background is industrial engineering and management. He dealt with projects concerning design competence management, collaborative engineering and design information management in scientific and industrial projects. Interface issues in collaborative engineering are his main focus of research and the topic of his ongoing doctorate.