INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

MODIFIED SAPPHIRE MODEL AS A FRAMEWORK FOR PRODUCT LIFECYCLE MANAGEMENT

G. McSorley, C. Fortin and G. Huet

Keywords: product lifecycle management, product development, information management

1. Background

Product Lifecycle Management (PLM) tools focus on both the capturing and sharing of information across the product lifecycle, in an attempt to make relevant information available to stakeholders when they need it and in a format and structure that is applicable to their work. While implementation of a true PLM strategy in industry has been achieved with varying success, there remains work to be done in determining the most appropriate model for the representation of product lifecycle information [Brissaud and Tichkiewitch 2001], [Huet et al. 2011]. Previous studies have revealed that much of the research into PLM remains focused on the design and manufacturing stages of the lifecycle, with few advances made regarding other stages of the life of a product. When considering testing and in-service activities, the need to properly understand and track product behaviour (in terms of functionality) is brought to the forefront [McSorley et al. 2009], [Toche et al. 2010].

While all PLM systems in one way or another consider the fact that the product, and related information, is evolving over time, temporal and behavioural aspects are not always considered in an explicit manner. As a result, the facilitation of the sharing of information by relating it to a robust representation of the evolution of the product throughout its entire lifecycle has still not been achieved. However, several researchers have suggested frameworks which could potentially pave the way towards this goal. In the next section, a brief overview will be given of PLM strategies for the extended lifecycle, with a specific focus on research which has considered temporal and behavioural aspects of PLM.

2. PLM strategies for the extended product lifecycle

Muller, Muschiol and Stark [2012], in studying the development of steam turbines, identify the need for information to be shared between the various engineering disciplines involved. Responsibility for the product is split across Order Engineering and Service Engineering. Arguing that multiple databases can prevent the efficient sharing of information, the authors propose the use of a single Product Data Management (PDM) system to manage all product related information spanning the life of the turbine. In contrast to traditional information management strategies, they propose that the information structure should primarily reflect the needs of service engineers, as they play a larger role than design engineers in ensuring the satisfactory performance of long lifecycle products. A weakness of this system, however, is the difficulty in implementing strict, consistent document control within a single database when separate processes in order and service engineering can affect the same documents.

Zhang, Hu, Xu and Zhang [2012] have proposed a multi-layer framework using XML to create documents representing the complete lifecycle knowledge surrounding a product. The researchers use a more traditional division of lifecycle stages than Muller et al. [2012], and include six stages during which information is created, beginning with product planning and ending with energy recovery and planning. The information generated at each stage gives rise to knowledge classes which form the basis for an ontology. Information gathered during a case study is then structured using XML, taking into consideration the content, structure and representation used in the source documents. This structuring aims to facilitate the accessibility of the information through a search and exploration interface. Initial results of the case study indicate an increased capacity to share information throughout the lifecycle. The proposed framework is found to be particularly useful for the communication of in-service and usage information to designers.

Lee and Suh [2009] describe a framework, termed Ubiquitous Product Lifecycle Support, based on multiple databases being used to stored information collected throughout the lifecycle. The use of multiple databases recognizes the variety of processes and information management strategies necessary when several organizations using different tools are responsible for collecting the information. By creating and using well defined ontologies for each area of information collection, interface agent software can transfer information between systems in a consistent fashion. The information is then made available through a single, overarching interface known as the Ubiquitous Product Lifecycle Information Highway. However, the details of how the interface agents complete the complex task of translating information across ontologies and combining all information at a single level remain unclear.

As previously mentioned, it is also necessary to consider how a PLM system can reflect the evolution of a product and its behaviour over time, and hence changes to the information being collected. To this end, certain researchers have begun explicitly developing models for information sharing that take into account these temporal and behavioural dimensions. Horvath and Rudas [2012] have developed Coordinated Request Based Product Modelling (CRPM), which aims to allow designers to search for information based on the intended product behaviour and context, as opposed to focusing solely on parts and assemblies.

Xiao, Xudong, Li and Guanghong [2010] have approached PLM using a 4D view model, with the views including geometry, task, virtual prototyping and lifecycle. This is in order to differentiate between the static structural and the dynamic behavioural aspects of the virtual prototype. Each phase of the lifecycle is considered from the geometry, task and virtual prototype views, while the lifecycle view allows for the correlation of the first three. In essence, the lifecycle view is a temporal view of the product lifecycle. However, this model is only extended to the point where the physical product is put into service. It is left to be determined how the management of in-service information would be considered.

Vosgien, Nguyen Van, Jankovic, Eynard and Bocquet [2012] also take a particular interest in the simulation stage of the product lifecycle, analysing the behaviour of product interfaces and defining a standard library to be reused when developing and testing new products. Again, it would be very interesting to see how this information concerning the behaviour of the virtual prototype could be used in conjunction within in-service information in order to provide a rich view of product behaviour throughout the lifecycle.

3. The SAPPhIRE Model

The goal of the SAPPhIRE model is to represent the causality of a system, of which previous models only provide partial views [Chakrabarti et al. 2005]. The model was first developed in order to provide a richer representation, at various levels of granularity, of the relationship between the function, behaviour and structure of a system than previous models, such as those of Hubka [1982], Gero and Kannengiesser [2004], and Umeda, Ishii, Yoshioka, Shimomura and Tomiyama [1996].While several definitions for function and behaviour have been developed in the literature, those used by Chakrabarti [2009] when developing the SAPPhIRE model will be used here. In this case, function is considered to be intentional, and occurring at a higher level of abstraction than behaviour, the latter evolving naturally from the structure of the system and related physical laws, and being the way in which the function is achieved. The structure itself consists of the elements and interfaces that form the system and its surrounding environment. As will be seen in the model, the behaviour, and hence function, of the system is brought about through one or multiple changes of state which occur due to the specific structure of the system and the physical laws which come to bear upon it. The model is composed of seven constructs, which are presented in Figure 1 alongside an example provided for illustrative purpose. The constructs are defined in Table 1.

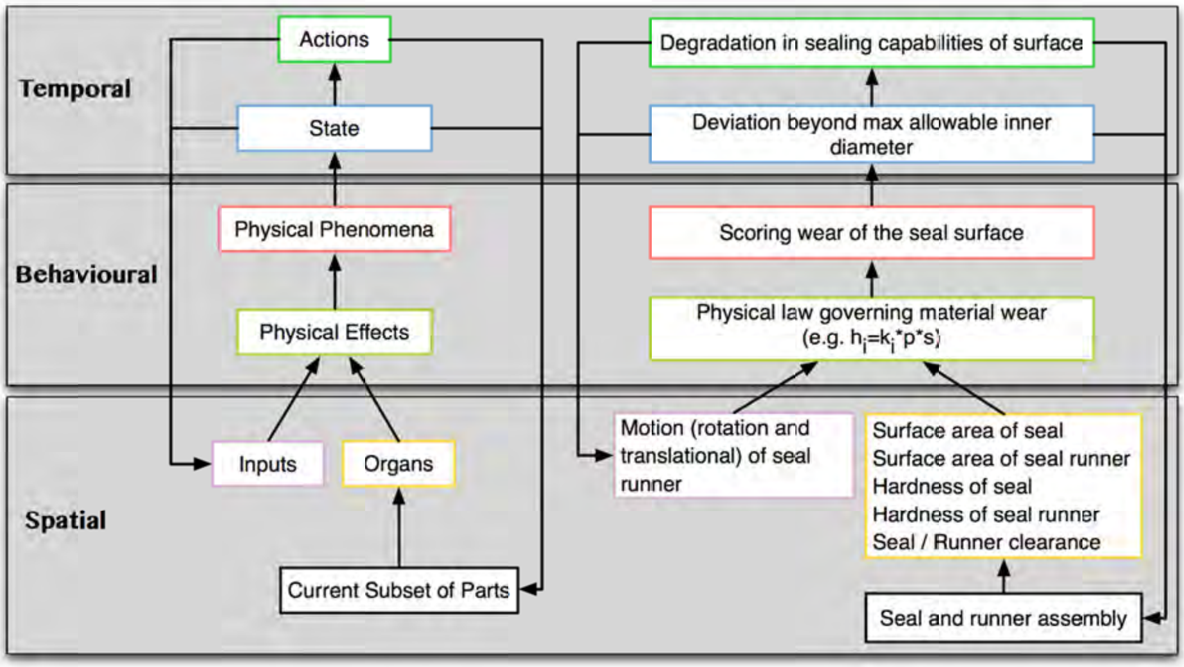

Figure 1. SAPPhiRE Model and example - Based on [Chakrabarti and Srinivasan 2009]

As can be seen from the arrows linking the constructs, the model is not meant to simply indicate a series of activities, rather it is meant as a logical process for understanding a causal chain within a system. Of particular importance are the connections from the change of *state* and the interpreted *actions* to both the *current subset of parts* and the *inputs*. These arrows indicate that the results from the change of *state* or *action* can modify the configuration of a system, giving rise to physical, behavioural and functional changes over time. The model therefore does not only represent the spatial dimension of the product (*parts* and *organs*) and the product behaviour (*physical effects* and *phenomena*), but also the temporal dimension (*inputs,* change of *state, actions*). These characteristics, shown in Figure 1, are important not only from the point of view of offering a rich model of causality, as intended by Chakrabarti, but also from the point of view of the management of product lifecycle information.

The example in Figure 1 illustrates the application of the model to an in-service event. This scenario is based on work completed in conjunction with an industrial partner, however for confidentiality reasons the details of the system have been modified. In this scenario, the system under consideration is a seal and runner assembly mounted on a spinning shaft with the goal of isolating circulating oil from the rest of the product. As can be seen, each construct has been replaced by its appriate element from the system.

Table 1. Constructs of the SAPPhIRE model – Based on Srinivasan and Chakrabarti [2009] & Chakrabarti and Srinivasan [2009]

Construct	Definition
<u>S</u>tate	A property at an instant of time of a system (and environment), that is involved in an interaction. eg: The temperature change in a system.

Action	An abstract description or high level interpretation of a change of state, a changed state, or creation of an input. eg: Temperature drop in a body can be interpreted as cooling of the body
Parts	A set of physical components and interfaces that constitute the system and its environment.. eg: A body surrounded by a medium
Phenomenon	An interaction between a system and its environment. eg: Heat flow from a body to its surroundings
Input	A physical variable that crosses the system boundary, and is essential for an interaction between a system and its environment. eg: A temperature difference which is necessary for heat transfer between a body and its surroundings
ORgan	A set of properties and conditions of a system and its environment required for an interaction between them. eg: Heat transfer from a body through convection requires a fluidic nature of the surrounding medium, surface area of the body and heat transfer coefficient
Effect	A principle of nature that underlies / governs an interaction. eg: Convection law underlies / governs heat transfer between a body and its surroundings

4. The Extended SAPPhIRE Model

In Chakrabarti et al. [2005], a detailed discussion is presented to situate the SAPPhIRE model within the various Function-Behaviour-Structure (FBS) models which have been proposed within the literature. In contrast to the FBS models identified, the goal of the authors was to develop a "richer, encompassing causal description of the functioning of a system" which could accommodate "multiple levels of granularity of structure and behaviour and multiple aspects of the causality". It is not the intent here to enter into a discussion of what constitutes function, behaviour and structure, but rather to demonstrate how the rich causal description afforded by SAPPhIRE can be applied to the context of product lifecycle management.

This contextualization is completed in two ways. First, an extended SAPPhIRE model (Figure 2) will be proposed which more closely reflects the evolution of a product throughout its lifecycle. Following this, the constructs will be discussed in terms of how they can be most appropriately applied to this extended model.

4.1 Modifications to the model structure

As stated in Table 1, in the SAPPhIRE model, *actions* are an interpretation of the change of *state*. This interpretation will necessarily be based on the intention or view point of an outside observer, such as a product user, inspector or designer. This sets *actions* apart from the other constructs which, once the inputs are defined, are not open to interpretation. As a result, a new construct *observation* has been added to the model (Figure 2). This *observation* is the interpretation of the change of state. It does not have a direct effect on the system, but rather the observer must make a conscious evaluation of whether the change of *state* observed requires *action*. If this is the case, the corrective action to the system requires a revision of certain *inputs, parts* or *organs*, which can entail a reconsideration of the relevant *physical effects*. To simplify the representation of the model in Figure 2, this modification of the system is represented by the *Modified Model* construct. This modification to the way *actions* are interpreted does not affect the changes to the system due to a change of state.

In order to represent the fact that the affected *organs* of a model under consideration can form part of the *parts* or *inputs* for the model of a connected system, an additional element has been added which states this explicitly. Finally, in order to differentiate feedback, whether it be towards the *inputs*, *parts* or *organs* of a model, from the forward flow of information or energy, the prior are represented using broken arrows, while the latter are represented by solid arrows.

4.2 SAPPhIRE constructs within a PLM context

While the high level nature of the SAPPhIRE model provides it with the flexibility to represent the various aspects of the product evolution, this same quality leads to the use of construct labels which are not necessarily familiar to the users of PLM tools. Table 2 summarizes the relevant clarifications, and in the following sections several of the more important aspects will be discussed in detail.

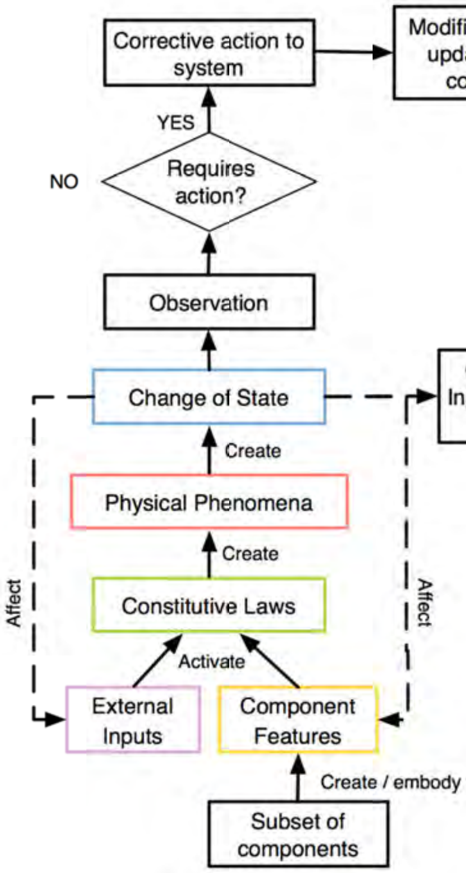

Figure 2. Extended SAPPhIRE Model

Table 2. Clarification of model constructs

Construct	Comments
State	Focus is on the change of state and how the effect of this on the system can be interpreted.
Observation	This is a new construct, representing the interpretation of the change of state which proceeds corrective action.
Action	This is now optional, and depends on the interpretation of the observed change of state.
Parts	Includes assemblies and sub-assemblies, as well as what are typically referred to as parts.
Phenomenon	This is unchanged from the original description.
Input	While the description is the same, it should be specified that these are external inputs.
oRgan	Relevant properties and conditions are typically the geometric features, key characteristics and physical properties of the parts.
Effect	Typically, a physical law will be applied for this construct, rather than an effect.

4.2.1 Parts

In product design, a part is typically considered a physical object at the lowest hierarchical level of a product structure. However, the SAPPhIRE model uses this term in a more general manner, so that one can consider both a functional or conceptual structure. For example, the *parts* construct could include system interfaces or sub-assemblies, rather than strict physical parts.

4.2.2 Organs

As defined in Table 1, organs are properties and conditions which are necessary for the interaction between the system and its environement. In terms of product design, at the level at which the component will interact with the environment, including other physical products, the conditions would be defined by the geometric features of the components. This is in addition to the physical properties of the system components and the environment. These features would not necessarily be created by the components, but rather are embodied within them.

4.2.3 Phyical effects

With respect to physical effects, it should be clarified that, unlike what one would expect if considering this construct in terms of cause and effect, here the effect is not the end result. While an effect is defined as "something which happens because of a cause", a law is "a basic principle of science or mathematics" [Collin 2003]. In order to understand the physical phenomena implicated in the causal chain, it is the physical law which is vital, and not the effect.

5. The Extended SAPPhIRE as a PLM framework

As discussed in section 1, the ability to represent spatial, behavioural, and temporal dimensions of the product is a requirement for the development of real product lifecycle information and engineering management systems. As the SAPPhIRE model has been shown to address each of these dimensions, and in considering the modified version the model proposed, an argument can be developed for the use of an extended SAPPhIRE model as a framework for this real PLM. This is not intended to supplant the need for research regarding the representation of domain specific information or the evolution of the product structure throughout the lifecycle. Rather, the extended SAPPhIRE model can provide a means through which appropriate models of the physical product can be related to product lifecycle information.

To validate these hypotheses, future work is intended to include the completion of three case studies, each based on a different stage of the product lifecycle: preliminary (or embodiement) design, manufacturing and in-service. The proposed case study scenarios are presented below, however at this stage, they remain conceptual excercises. In al three cases, the product under consideration is a seal mounted on a rotating shaft whose main function is the isolation of the oil circulation system from the rest of the product.

5.1 In-service scenario

Figure 3 (a) provides an example of the use of the extended SAPPhIRE model to represent a causal chain from the in-service phase, using the same example as that in Figure 1, that of the degradation of a seal. As can be seen, there are significant differences in the model compared to Figure 1. Of particular note is the explicit inclusion of the link to the "oil leakage" model, which would be a system level model as opposed to the current subassembly level model. Furthermore, the observation of a degraded seal is not what leads automatically to a change in the physical characteristics, or organs, of the system. Rather, the change of state will lead to a given change. The observation of the change of state (degradation of the seal), on the other hand, will allow an external actor to determine whether further modifications must take place. This more closely represents the reality of the evolution of the product throughout the in-service lifecycle phase. It should also be noted that other relevant information can be related to the constructs. For example, as a decision should be taken regarding whether corrective action should be carried out, information sources detailing the necessary considerations can be associated to the appropriate model constructs. A full representation of how this information can be linked to the extended SAPPhIRE model will be the subject of future work.

5.2 Design scenario

Figure 3 (b) uses the extended SAPPhIRE model to represent a typical activity found within the design stage of the product lifecycle. In this case, the seal wear is simulated by designers rather than observed by inspectors.

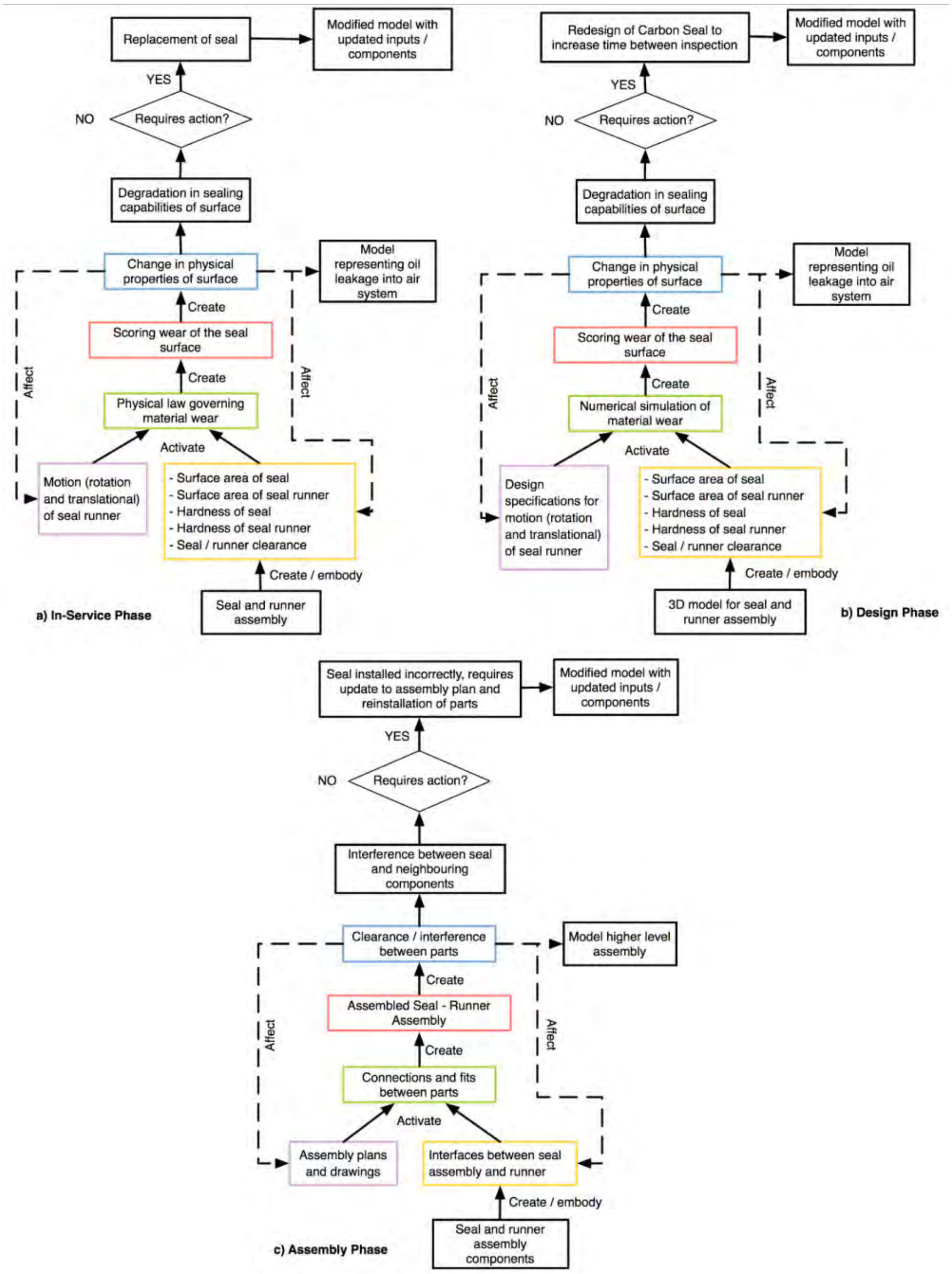

Figure 3. Extended SAPPhIRE model for PLM: a) In-service phase b) Design phase c) Assembly phase

While this is a hypothetical example, not one observed while working with the industrial partner, it is representative of the type of analyses completed during product design.
In this case, some of the elements are familiar, such as the relevant features and the physical phenomena being observed. The major differences are related to the context in which the activities take place, which affects the information sources used to develop the model, as well as the actions taken by outside observers. In this case, the product definition is one version within the multiple iterations necessary when designing a product, and the spatial information is exactly that defined by the designers. While certain aspects can be informed through the study of product in-use information (for example the specific ways in which to model surface wear), the information used remains in the realm of product design. Furthermore, the action taken in this case will be to change the design, rather than to make a change to a physical product.

5.3 Assembly scenario

Finally, Figure 3 (c) represents a scenario in which a discrepancy occurs during the assembly of a seal within a product. As can be seen, the model constructs once again exhibit relationships to the design and in-service extended SAPPhIRE models while the particuler elements of the model reflect the current context. During assembly, the main concern is not the performance of the seal, but its physical interfaces with other components and their spatial requirements. When faced with a problem assembling the product, possible actions include modifying the assembly plans or the product itself, which would necessitate a design change.

5.4 Discussion

As can be seen, the same component is traced throughout its lifecycle by means of the extended SAPPhIRE model and there are potential relationships between the elements making up these three models. For example, in the design and in-service models, the wear of the seal is considered, which is a function of the clearance between the seal and the runner. This clearance will also be affected by any interference between the parts, which is considered in the assembly model. By exploiting this relationship between the constructs of each model, it may be possible to facilitate information transfer between stakeholders. In this particular scenario, the spatial requirements used to validate the assembly of a product can be determined through intial design simulations. Later, once a certain number of products are in-service, inspection results regarding wear rates and sealing capability may support a decision to revise the assembly specifications. The inspection results may also be transferred back to designers to update the assumptions used in completing early analyses. Currently, this type of information transfer is difficult to support in an efficient manner, and it is believed that the use of the SAPPhIRE model could be an integral step in resolving this issue.
While the causal chains used for constructing the SAPPhIRE model make up one type of product lifecycle information, the previous examples demonstrate that the value of the model also rests on its power as a framework within which to structure other product related information. For example, the wear experienced by the seal (change of state) will be supported by inspection data collected during in-service activities. The design simulation results will depend on the output of the simulation algorithms. By associating lifecycle phase specicific information with the appropriate constructs, which are present across models, the transfer and reuse of information as well as the identification of connections between different information types could be greatly facilitated.
A caveat, however, is that the information collected must still be considered within the relevant context. For example, the results of in-service inspections should not be mixed with design simulation results without specifying their source and the assumptions made in producing or collecting them.
Further research will be needed to determine the most efficient ways to link these models together. Possible solutions include the results from one lifecycle phase being used directly as elements within subsequent models. Alternatively, robust links could be developed between information collected at one stage and the model constructs of other phases. For example, the previously described scenario where clearances are based not purely on simulations, but also on feedback from product in-use information.

6. Conclusion and future work

Current PLM systems manage information sharing based on the product structure, whether the structure is tailored to designers or other stakeholders in the product lifecycle. These structures remain fundamentally spatially centered, relying on an information structure that cannot easily take into account the behaviour of the product or its transition from one state to another over time. However, the consideration of a product's behaviour and its evolution is necessary in order to meet the requirements of a fully functionning PLM system.

In comparison, the SAPPhIRE model has shown potential to facilitate the management and sharing of not only spatial product information, but also information pertaining to behavioural and temporal elements. While scenarios have been proposed in design, assembly and in-service contexts, it is believed that the model will be just as applicable to other lifecycle phases. By modelling material transformations during manufacturing processes, for example, it would be possible to create a rich representation of manufacturing and production information.

Future work involves the completion of an analysis of in-service and testing information collected during a collaborative project with a leading aerospace manufacturer. By applying the SAPPhIRE model to the representation of the product evolution as well as using it to structure the related information, the objective is to demonstrate both the validity of the model and discover new interrelationships between testing and in-service information. Following this, further studies will be sought to validate the model for other stages of the lifecycle. In this way, it will be possible to examine whether the extended SAPPhIRE model is indeed an appropriate basis for a full PLM system.

References

Brissaud, D., Tichkiewitch, S., "Product models for life-cycle", CIRP Annals-Manufacturing Technology, Vol.50, No.1, 2001, pp. 105-108.

Chakrabarti, A., "Sapphire an Approach to Analysis and Synthesis", 17th International Conference on Engineering Design, M. Norell Bergendahl, M. Grimheden, L. Leifer, P. Skogstad & U. Lindemann(Ed.), 2009, pp. 417-428.

Chakrabarti, A., Sarkar, P., Leelavathamma, B., Nataraju, B. S., "A functional representation for aiding biomimetic and artificial inspiration of new ideas", AI EDAM, Vol.19, No.2, 2005, pp. 113-132.

Chakrabarti, A., Srinivasan, V., "Sapphire–an Approach to Analysis and Synthesis", Proceedings of the International Conference on Engineering Design, Paolo Alto, California, 2009, pp. 417-428.

Collin, S. M. H., "Dictionary of Science and Technology", A & C Black, 2003.

Gero, J. S., Kannengiesser, U., "The situated function–behaviour–structure framework", Design Studies, Vol.25, No.4, 2004, pp. 373-391.

Horvath, L., Rudas, I. J., "Requested behavior driven control of product definition", Proceedings of the IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, 2012, pp. 2839-2844.

Hubka, V., "Principles of Engineering Design" (W. E. Eder, Trans.), Butterworth Scientific London, 1982.

Huet, G., Fortin, C., McSorley, G., Toche, B., "The management of manufacturing processes using complementary information structures", Proceedings of the 18th International Conference on Engineering Design (ICED11), Vol. 6, 2011, pp. 183-191.

Lee, B.-E., Suh, S.-H., "An architecture for ubiquitous product life cycle support system and its extension to machine tools with product data model", The International Journal of Advanced Manufacturing Technology, Vol.42, No.5-6, 2009, pp. 606-620.

McSorley, G., Huet, G., Fortin, C., "A preliminary framework for the management of "product in-use" information", Revue d'Ingénierie Numérique Collaborative, Vol.3, No.1, 2009.

Muller, P., Muschiol, M., Stark, R., "PLM-based service data management in steam turbine business", Proceedings of Springer New York, Montreal, QC, Canada, 2012, pp. 170-181.

Srinivasan, V., Chakrabarti, A., "An Empirical Evaluation of Novelty-SAPPhIRE Relationship", Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2009), ASME, San Diego, CA, 2009, pp. 985-994.

Toche, B., Huet, G., McSorley, G., Fortin, C., "A product lifecycle management framework to support the exchange of prototyping and testing information", Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC), Montreal, Canada, 2010.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., Tomiyama, T., "Supporting conceptual design based on the function-behavior-state modeler", AI EDAM, Vol.10, No.04, 1996, pp. 275-288.

Vosgien, T., Nguyen Van, T., Jankovic, M., Eynard, B., Bocquet, J.-C., "Towards model-based system engineering for simulation-based design in product data management systems", Proceedings of the IFIP WG5.1 9th International Conference on Product Lifecycle Management, PLM 2012, L. Rivest, A. Bouras & B. Louchini(Ed.), Springer New York, Montreal, QC, Canada, 2012, pp. 612-622.

Xiao, S., Xudong, C., Li, Z., Guanghong, G., "Modeling framework for product lifecycle information", Simulation Modelling Practice and Theory, Vol.18, No.8, 2010, pp. 1080-1091.

Zhang, D., Hu, D., Xu, Y., Zhang, H., "A framework for design knowledge management and reuse for Product-Service Systems in construction machinery industry", Computers in Industry, Vol.63, No.4, 2012, pp. 328-337.

Mr. Grant McSorley, Ph.D. student
École Polytechnique de Montréal
Department of Mechanical Engineering
2500, chemin de Polytechnique, H3T 1J4, Montreal, Canada
Telephone: 514-340-4711 x4497
Email: grant.mcsorley@polymtl.ca