INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

DESIGN 2008

DYNAMIC AND CONCEPTUAL DMU

F.J. Gerhardt and M. Eigner

Keywords: digital mock-up, DMU, conceptual CAE, Simulation

1. Introduction

Ever increasing product complexity and the growing call for shorter Time-to-Market (TTM) require a very early, comprehensive and reliable knowledge concerning the product. This is especially true when considering mechatronical products. While a designer's natural design would evolve top-down, from a coarse perception to detailed solutions, today's CAD-Systems mainly serve the bottom-up integration process.

Related to our most recent research efforts in suiting CAD-Systems for application in the early phases of development, we examine collateral ways of respectively enhancing the Digital Mock-Up (DMU). Today's DMU is a tessellated representation of the product along with associated methods of analysis for validation of design. Usually, these methods reflect a static examination, such as Collision-Detection, Sectioning, or a highlighted comparison of multiple variants, and are applied in the later development phases, using detailed geometry. Depending on analysis results, CAD geometry must possibly be adjusted or rectified. This is a throw-it-over-the-wall iterative process, as depicted in Figure 1. On this note, geometrical information coming from E-CAD interfaces is imported into M-CAD systems. Other than that, we find toolset environments for M-CAD, E-CAD, and CASE to be coexistent, but not interacting in the sense of a mechatronical simulation.

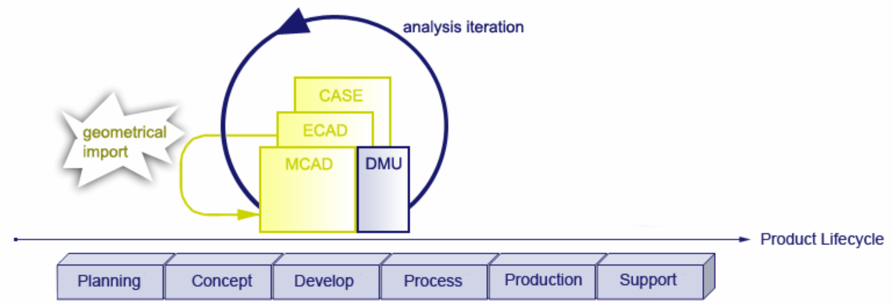

Figure 1. Static Digital Mock-Up today

Concerning control systems and embedded software in general, we would like to force a shift of paradigm, going from late analysis driven, domain-specific and coined by Hardware-in-the-Loop to a holistic, early integrated Software-in-the-Loop course of action. Whether we speak of production systems, automotive subsystems, or every-day equipment, such as autofocus cameras, so far no approach for simulating mechatronical systems as a whole is to be found, especially in the early phases of product development.

In this paper we contribute to the idea of a dynamically growing DMU based on the Unigraphics Solutions (UGS) Jupiter Tessellation (JT) data format, which is to serve as a basis for a multi-disciplinary analysis. The foundation of our work is the concept of using continuously growing geometry, beginning in the early phases of development, be it only in form of placeholders. While our research does not restrict itself to the JT data format, but studies various neutral formats instead, we have found JT to be suitable for our concept.

Mechatronical analysis longs for research on cross-domain descriptions, and ways to especially unite information present in the conceptual phases. Further, speaking of CAE - driven development requires thorough thought on the very integration of simulation into predominating processes. These matters are subject to related and future work. In this paper we mainly discuss a surrounding architectural point of view, targeting a neutral format technology as a base for the so called Dynamic DMU, in this case using JT.

Section 2 discusses work targeting similar and related subjects, before Section 3 presents the idea of our Dynamic DMU. We explain the concept, give an example of a mechatronical and early simulation, and further illustrate why we have chosen JT for the underlying foundation. Further, Section 4 presents an overview of an architectural system environment associated with the concept of a Dynamic DMU, before Section 5 concludes with a summary of our work. Finally, Section 6 discusses ongoing research and future work on related matters.

2. Related work

Examination of ways to more effectively integrate simulation into the product development process is a related subject of interest, dealt with in work as early as e.g. [AS98]. [AN04] describes a more recent work on a requirement-driven product development process. Much research on this matter is restrained to mechanical simulation though.

The dissertation by [CR02] gives an overview of procedural methods associated with mechatronical modelling and provides suggestions targeted at integration of simulation models into PDM systems. In terms of technology, MechaSTEP [ADC99] discusses a STEP-conform data model, comprehending system representations coming not only from mechanical engineering, but from disciplines such as electronics, hydraulics and control engineering as well. While the underlying idea was to eventually replace domain-specific descriptions for simulation, this approach has not yet prevailed. This is due to the fact, that the MechaSTEP format is very complex, further including information ranging from product structure, over process-relevant to geometry. Nevertheless, the existence of a sufficient cross-domain model is a prerequisite to the concept presented in this paper. A function-oriented approach to this foundation is discussed in [EH07].

For related work when considering the conceptual phase, simulation therein is a topic also mentioned in [MS07]. The work presented in this paper gives an overview of practices and thoughts concerning Up-front CAE simulation. It deals with the so-called AutoSIM [AU07] project, with an exclusive focus on the automotive industry.

3. Dynamic DMU

In contrast to today's DMU being applied for analysis in the later phases, we want it to accompany product development much earlier. In terms of mechanics, even when geometry is only coarse, a sooner multi-disciplinary simulation would allow derivation of constraints on components yet to be developed.

3.1 Dynamic DMU: the concept

Conform to the natural top-down body of thought, DMU content and functionality should grow in time, according to the information available in the current phase of development, as illustrated in Figure 2. Hence, we have chosen the name Dynamic DMU.

As depicted, domain overlap is especially high when relating to the Conceptual DMU, which describes the early part of the Dynamic DMU, dealing with product examination during conceptual phases of development. Here already, we long for interaction among various domains, such as mechanics, electronics and computer science, defining constraints on further development using the coarse data already existent at the time. Section 3.2 describes an exemplary scenario, deriving constraints resulting from an early simulation, partially based on coarse geometry. With termination of the conceptual phases, interaction temporarily lessens, as module realization (such as detailed circuit design) remains domain dependent.

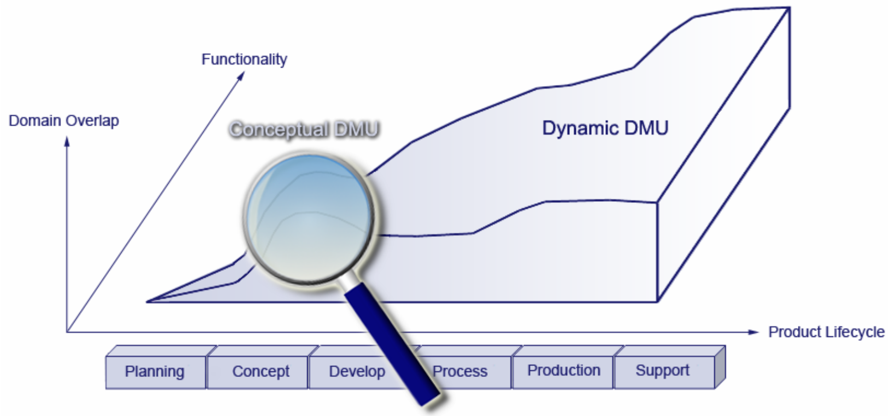

Figure 2. Dynamic DMU

When considering the concept of a Dynamic DMU in order to support the product development process, various challenges are to be faced. As mentioned in Section 2, ongoing research must go into melding various data available in the conceptual phases, be this information a specification in writing or dimensioning, plain text, a set of equations, drafts or represented as coarse geometry. These types of data must somehow be brought into unity, in order to then define a mechatronical simulation based on a cross-domain description.

The mechanical domain is the driving force of the DMU, based on some data format that is able to hold geometrical. In our case, this format must be able to represent dynamically evolving data. For this reason, we have chosen to utilize JT, as described in Section 3.3.

3.2 Exemplary mechatronical and early simulation

The conceptual phase serves its main purpose in forming a coarse image of the product. While forming this image, first insight into certain coactions may be of value for various decisions yet to be made, or mechatronical components yet to be developed.

Consider the development of a vehicle series' windshield wiper subsystem. Body packaging data as a result from early design might result in a tolerated area for two piercing points. Without knowledge of a detailed geometry, dimensions for the vehicle's windshield may already be defined.

Setting the piercing points as well as the wipers' working angles, and interpreting the windshield as a plane geometry without curvature would result in the possibility of running a very first simulation. The gain of such a simulation might be a set of constraints and minimal lengths concerning the wipers, in order to ascertain a minimal percentage of the windshield being wiped.

Coming to a second scenario for early simulation, there exist various ways in the arrangement and handling of both wipers. Going away from a linkage driven design, we might request decoupled, software-driven wiper controls. Given the wipers' lengths, their asynchronous movement must be adjusted in the sense of not nudging each other. Here already we speak of a mechatronical simulation.

When the windshield and wiper geometry becomes more detailed, our simulation possibilities expand in terms of mechanics. Maximum contact pressure is computable by determination of the pivot torque. This pressure results in a friction to be overcome by the responsible electric motor, which implies a required hinge moment thereof.

3.3 Dynamic DMU: the underlying JT technology

As mentioned in Section 3.1, we suggest the JT data format as a foundation for dynamically evolving data based on geometry. Without going into too much detail, Figure 3 sketchily depicts the setup of a JT file.

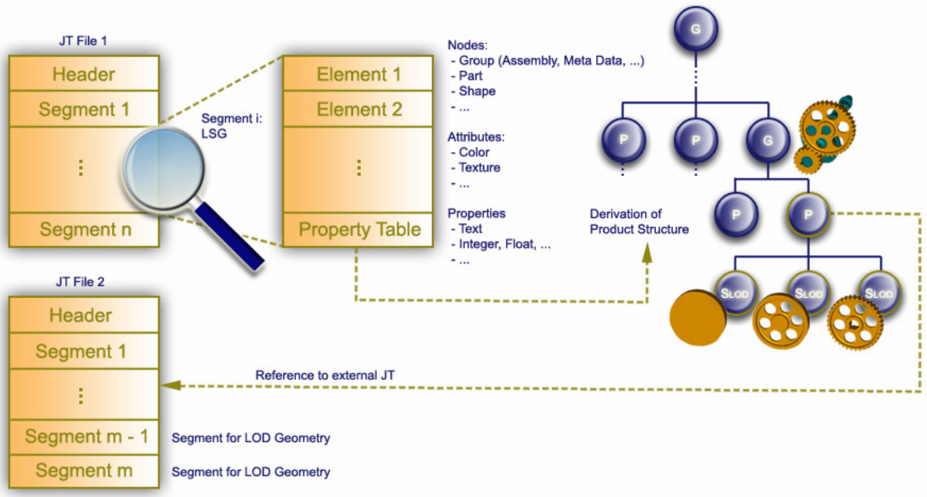

Figure 3. Sketchy illustration of the JT file's setup

In terms of structure, a JT file is subdivided into a set of segments, each holding certain information. One segment, the so called logical scenegraph (LSG) [JT06], stores a list of elements, each element defining a node, an attribute or a property. Nodes, in turn, are further classified into being e.g. a group (G), a part (P), a level-of-detail shape (SLOD), or else. The sum of all elements basically defines a product structure. Even though properties are merely described by basic data types, such as integers, floats or character strings, providing the means to store as many as needed makes JT a rather flexible format. We call JT a semi-intelligent data format, for its possibility to link not only tessellated and exact (B-Rep, NURBS) geometry, but also Meta –and PMI data to any scenegraph node. The file may hold associations to original CAD information, allowing extraction of further properties whenever needed, in terms of a simulation for example.

JT is suited for our dynamic concept for two reasons:

1. *JT Partitioning*
   Generally, JT allows partitioning a model into multiple physical JT files using so called partition nodes. While the original intention lies in being able to store a separate JT file per part in an assembly, we wish to use these external references to allow the dynamic insertion of new information as the Dynamic DMU's data pool grows, without having to create a whole new DMU model every time.

2. *JT Level-of-detail shapes*
   The origin of using LOD shapes lies in providing the means to visualize more or less data, depending on frame rate specifications. With today's resources, it would be impossible to e.g. depict and simulate a whole vehicle using a very detailed resolution, with an interactive frame rate. In our case, however, we can abuse this feature combined with partitioning, in order to insert growing-resolutions of one and the same part, little by little further along in the product development process.

   Figure 4 depicts an exemplary scenario, in which cogwheel geometry evolves over time. The associated material may be redefined as well, information that can dynamically be reattached to the scenegraph likewise.

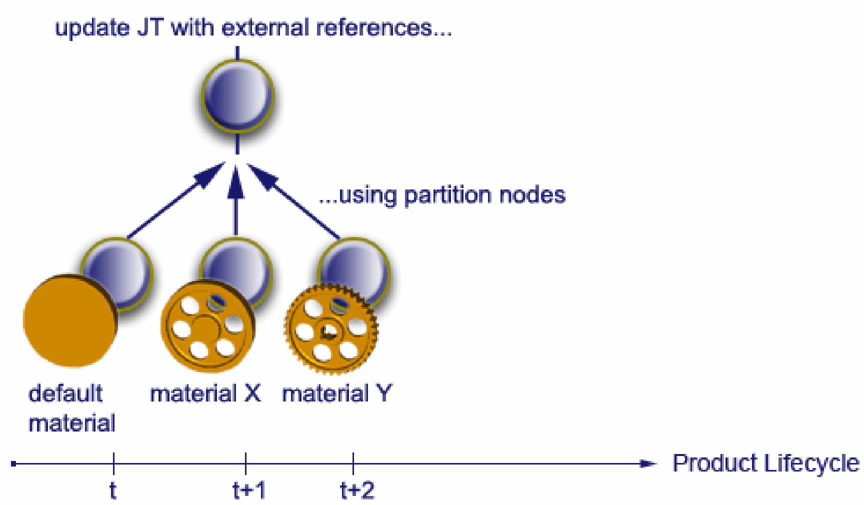

Figure 4. Illustration of an exemplary part geometry evolution. A JT file is updated three times during the product development process, using partition nodes

After we have now discussed the underlying technology, the next Section gives an overview of various systems involved in the realization of the Dynamic DMU concept.

4. The system environment

We have mentioned in Section 2, that the existence of a sufficient cross-domain description is a prerequisite to the concept presented in this paper. We imagine an internal model similar to MechaSTEP [ADC99], but without direct information on geometry, product structure, etc. This sort of information is stored in JT files instead. The definition of a simulation is to be based on block diagrams, as done in tools like Mathworks® Simulink®. A cross-domain model (based on the cross-domain description) should data on the modelled blocks, connections between them, and a linkage to the JT data. Figure 5 depicts the interrelation between a JT based DMU model and a cross-domain description model. After defining a simulation using the cross-domain model, our concept is designed to run the simulation as a whole, visualizing among others the JT geometry. While domain-specific solvers are accessed using application interfaces, it is up to the cross-domain model to transfer resulting data from one interface to the next, completing the mechatronical analysis.

Due to the fact that the Dynamic DMU runs along throughout the product lifecycle, it is linked to existing PDM / PLM structures.

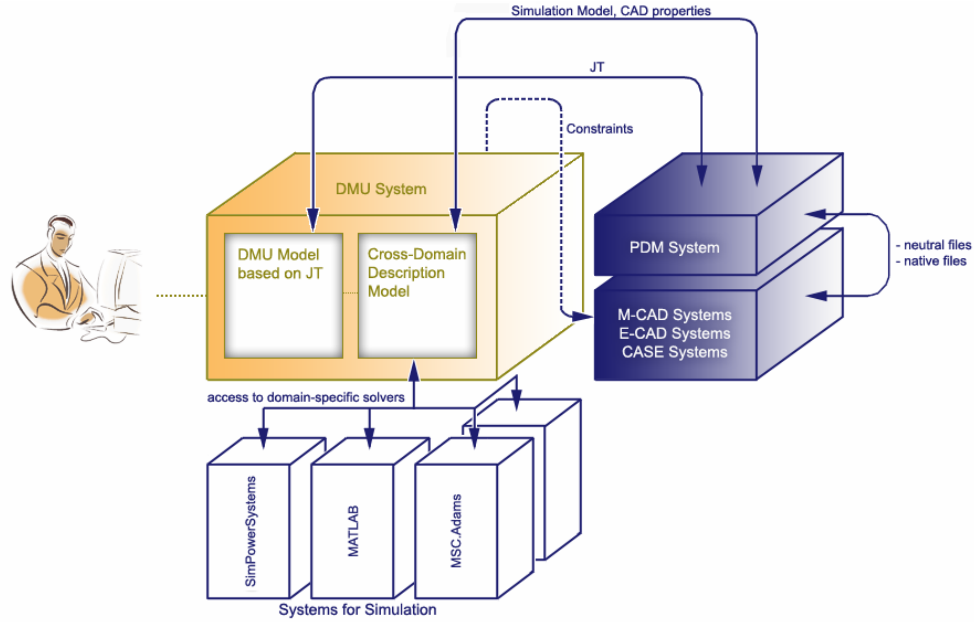

Figure 5. Dynamic DMU system environment, based on a neutral data format, e.g. JT in this case

The origin of a JT file's geometry lies in a native CAD of course. Native and neutral data is administered by a PDM system. The PDM system is used to communicate JT files to and from the DMU system. Using an appropriate interface, there are no technical barriers in doing so.

Once a JT file's contents are in the DMU system, it must be linked to a defined cross-domain description model. Using JT's feature to hold associations to original CAD files allows us to derive further (not stored in the JT directly) properties needed for various simulations, such as mass by using a given material. These properties are then broadcasted to the simulation model stored in the cross-domain description. The simulation model, as well, is saved into the PDM system for later retrieval.

5. Conclusion

We have presented a concept for a dynamically evolving DMU, and have suggested the JT data format as a basis. This format describes a well accepted industrial standard, and is recently attracting more attention. In terms of technology, it allows integration of the DMU into the product development process, starting in the early phases thereof. Even though this paper is coined by JT, our research does not restrain itself to this very data format. We have illustrated our work from an architectural point of view, and provided a basis for further research on mechatronical simulation.

6. Future work

With JT being a very suitable format for visualizing product data, we currently study its application in a whole set of industrial processes, such as the simple presentation of information, the exchange of information, and the DMU of course. We are, at the moment, in an early phase of research.

When considering the possibilities of a data format such as JT, one must certainly deal with its deficiencies as well. When having identified requirements related to an industrial process, we work on a procedural method to map these requirements to structures inside a JT file. While some mappings may be straight forward, other requirements may only be met by thinking outside the box. We study the possibility of extending JT in terms of its contents, maintaining compatibility to standard JT files though. Since our concept does not rely on a specific data format, we consider neutral data formats in general, weighing functionality and usability.

Independent of JT, we would like to expand research on a cross-domain description, linked to a block-diagram modelling approach. A prototypical mapping to existing toolsets and their solvers shall be implemented.

References

[AU07] AutoSIM Project: http://www.autosim.org; cited October 2007

[EH07] Eigner, M.; Hollerith, T.: "Concept for an Integrated Mechatronic Simulation by a Cross domain Function Model", ProSTEP iViP Science Days 2007 Proceedings, Bremen-Germany, pages 146-152, 2007

[MS07] Moser, A.; Schweiger, R.: „Integration of Up-front CAE-Simulation and How to Overcome the Barriers“, ProSTEP iViP Science Days 2007 Proceedings, Bremen-Germany, pages 76-83, 2007

[CR02] Krastel, M.: „Integration multidisziplinärer Simulations- und Berechnungsmodelle in PDM-Systeme“, Dissertation DiK TU Darmstadt, Shaker Verlag, 2002

[ADC99] Anderl, R.; Donges, C.; Krastel, M.: „MechaSTEP – STEP Datenmodelle zur Abbildung mechatronischer Systeme“, Produktdatenjournal (1999), Nr. 5, pages 30-34, 1999

[AS98] Andersson, K.; Sellgren, U.: „MOSAIC – Integrated modeling and simulation of physical behavior in complex systems“, NordDesign '98,Stockholm, Sweden, 1998

[AN04] Andersson, K.: „A requirement-driven product development process“, 7th Workshop on Product Structuring – Product Platform Development, Chalmers University of Technology, Göteborg, 2004

[JT06] JT File Format Reference Version 8.1: http://www.jtopen.com/docs/JT_File_Format_Reference.pdf; cited March 2008

Dipl.-Inf. Florian Gerhardt
University of Kaiserslautern
Gottlieb-Daimler-Str. / Geb. 44
67663 Kaiserslautern, Germany
Emial: gerhardt@mv.uni-kl.de