INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

A FRAMEWORK TO RE-ORGANIZE DESIGN ACTIVITIES DURING ENGINEERING CHANGE PROCESS

Mohamed Zied Ouertani[1], Khadidja Grebici[2] and Lilia Gzara[1]
[1]CRAN – Nancy University, France
[2]G-SCOP – INP Grenoble, France

ABSTRACT
Design has increasingly become a collaborative endeavour carried out by multiple actors with diverse kinds of expertise. Due to the multi-actors interaction conflicts emerge during the design progress. Hence, a critical element of collaborative design is to manage conflict and particularly the impacts once they are resolved. Indeed, the conflict resolution comes up with a solution which often implies modifications on the product and the process organisation. This paper deals with this problematic of changes impact on conflict management process. It quantifies key issues with regards to concurrent engineering that enables us to better manage the design process. Strategies to overlap coupled activities are proposed based on the dependencies between the handled data during the design process. Furthermore, a framework to optimise re-organising design activities is proposed.

Keywords: Conflict management, data dependencies, impact assessment, overlapping strategies

1 INTRODUCTION
It is characteristic of collaborative engineering design that precedence relationships among design activities contain information flow conflicts. Indeed, due to multi-actors interaction, conflicts can emerge from disagreements between designers about proposed designs. Therefore, a critical element of collaborative design would be conflict resolution. In a collaborative design context, conflicts occur when at least two incompatible design commitments are made, or when a design party has a negative critique of another design party's actions [1]. The conflict management process could be perceived as the succession of five phases: Conflict detection, Conflict resolution team identification and formation, Negotiation management, Solution generation and Solution impact assessment. Current conflict management approaches in collaborative design focus on *Conflict detection* [1] [2] [3], *Negotiation management* and *Solution generation* phases [4] [5] [6].
In a previous work [7], we proposed the DEPNET (product **D**ata d**EP**endencies **NET**work identification and qualification) methodology to tackle the *Conflict resolution team identification and formation* phase. This methodology addresses the problematic of identifying the actors to be involved in the negotiation process to resolve the conflict. Based on a process traceability system, the DEPNET methodology consists of identifying and qualifying the dependencies between the data handled during the design process execution. This leads to the construction of a data dependencies network which allows identifying the actors to be involved in the conflict resolution process. Indeed, each data is carried out by an activity. This activity has a responsible to execute it. Consequently, once a data is identified, the actor responsible for its realisation is identified.
Concerning the *Solution impact assessment* phase once the conflict is resolved, it has not been tackled on the reviewed works. Indeed, the selected solution often implies the modification of one or more input data of the activity where the conflict has emerged, and thus, generating a cascade of modifications on the already produced data. Consequently, these data have to be redefined. This implies re-executing the various design activities responsible on the elaboration of these product data and also adjusting the process still in the pipeline. Accordingly, strategies are to be proposed to coordinate re-execution of the concerned activities.
Hence, this paper purposes' is to come up with strategies to coordinate the activities re-execution. In order to do so, we based ourselves on the data dependencies network, already built thanks to the

DEPNET methodology, to identify the impacted data to be redefined, as well as the correspondent activities to be re-executed. Then, overlapping strategies are proposed in order to re-organise the identified. Furthermore, a model to optimise the activities overlapping is devised.
In addition to the Introduction, this paper consists of five more sections. Section 2 presents the data dependencies network constructs. Then, section 3 describes the mechanisms to assess the solution impact on product data, while the overlapping strategies to re-organise the design process are described in §4. Section 5 focuses on development of the optimisation framework. It discusses the relationships existing between the set of parameters of the framework. Conclusions are summarised in section 6.

2 DATA DEPENDENCIES NETWORK CONSTRUCTS

The data dependencies network is used to represent the qualitative dependencies among design data. It is an oriented graph composed of nodes and arcs:
The "nodes" correspond to the product data handled during the design process and leading to the elaboration of the data source of conflict, those data on which the source of conflict depends. These product data can be of several types such as structural, functional, behavioural and geometrical. They correspond to the various descriptions of the product, elaborated by designers during the development process, such as geometrical parameters, CAD drawings, stress analysis reports, drawing lists and parts lists. For instance, the figure 1 below illustrates the data dependencies network (extracted with the DEPNET tool) associated to the design process of a Flexible Assembly System (FAS) (see [8] for more details about the FAS case study).

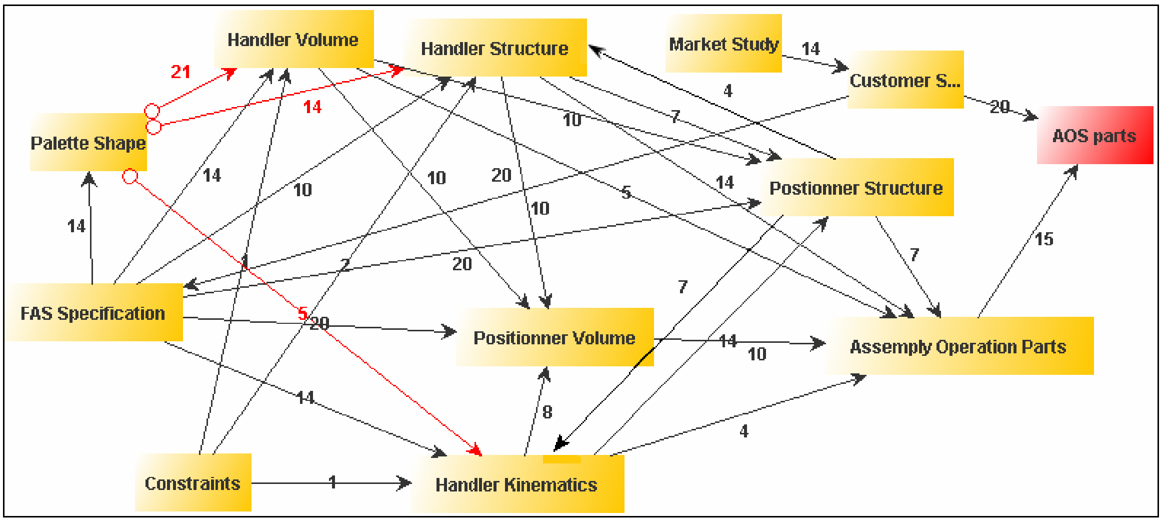

Figure 1. Data dependencies network associated to the FAS design process

The "arcs" correspond to the dependency relationships between the various nodes identified in the network (data), such as the arc linking the "handler volume" and "positionner volume" in the figure 1 above. In a context of collaborative design, dependency between two data could be on *forward* or *feedback* direction. Forward dependent data are those that require input from other activities, but not themselves, such as the arcs starting from "handler volume" (to "positionner structure", "assembly operation parts", and "positionner volume"). Feedback dependent data are those that need inputs from other activities including themselves, such as the arcs linking the "positionner structure" to the "handler structure" in the figure 1 above. The feedback links are to be considered since they are a source of rework and thus are resources consuming and time consuming. Thus, two data are said to be dependent in the case of forward dependency or interdependent in the case of feedback dependency.
The dependency relationships in this network are quantified with a dependency degree (1) which is an aggregation of the three attributes: variability, sensitivity and completeness. The dependency degrees are represented as weights of the arcs in the data dependency network (figure1).

$$Dependency\ Degree = Completeness * (1 + (Variability * Sensitivity)) \qquad (1)$$

The attributes completeness, variability and sensitivity are valued by actors when they capture their activities in the design process traceability system. In practice, these attributes are not always easy to define quantitatively. Therefore, using structured expert interviews, qualitative inputs are developed to provide insights on how to evaluate them. The *Completeness* **(C)** expresses the suitability of an input data to the creation of an output data (**0** Weak, **1** Not Vital, **2** Vital and **3** Extremely Vital). The *Variability* **(V)** describes the likelihood that an output data provided by upstream activity would change after being initially released (**0** Not Variable, **1** Low Variability, **2** Moderate Variability and **3** High Variability). The *Sensitivity* **(S)** describes how sensitive the completion of an output data is to changes or modifications of an input data. It expresses the degree to which work is changed as the result of absorbing transferred data (**0** Not Sensitive, **1** Low Sensitivity, **2** Moderate Sensitivity and **3** High Sensitivity).

As completeness, variability and sensitivity are valued with numerical values (0, 1, 2 and 3), the resultant range value of the dependency degree is an integer between 0 and 30 (cf. Figure 1), whereas {0, 1, 2, 3, 4} denotes a weak dependency and a low risk of rework, {5, 6, 7, 8, 9, 10, 12, 14} describes a moderate dependency and a moderate risk of rework and {15, 20, 21, 30} denotes a high dependency and a high risk of rework.

Based on the data dependency network and thanks to a set of queries on the database storing the process execution instances, it is possible to identify for each identified data, the activity responsible on its elaboration as well as the actor performing this activity. In the next section, we detail how to evaluate the impact of a selected solution once the conflict is resolved.

3. SOLUTION IMPACT ON PRODUCT DATA

The technical solution selected through the negotiation and resolution phase corresponds to the change of one or more product data involved in the design process leading to the elaboration of the data source of conflict. The data to change is then a part of the data dependency network presented in Section 2. These changes correspond to modifications in dimensions, fits, forms, functions, materials, etc. of products or components already released that could be used by other designers to perform their respective activities. Consequently, the engineering change caused by the solution release induces a series of downstream changes. Hence, assessing the selected solution impact returns with propagating the impact of those data changes through the data dependency network and thus on the organisation of the responsible activities. For example, let consider the sub-network of data dependencies in Figure 2. In the case of a conflict occurring on the piece of data “assembly operation parts”, supposes that the solution retained after resolving that conflict consists of changing the value of the piece of data “handler kinematics”. Consequently, according to the data dependency network, all data linked to “handler kinematics” with a forward dependency relationship (arc starting from “handler kinematics” such as “Positionner volume”) must be changed. Then, all data depending on the “positionner volume” and the “positionner structure” must be changed as well, such as the piece of data “assembly operation parts” (depending on the “positionner structure”). Therefore, activities responsible of the elaboration of “assembly operation parts”, “positionner structure”, “positionner volume” and “handler kinematics” have to be re-executed. In this example, the total number of nodes depending on the conflict solution (“handler kinematics”) is low (three nodes) and re-execution of the design process is simple.

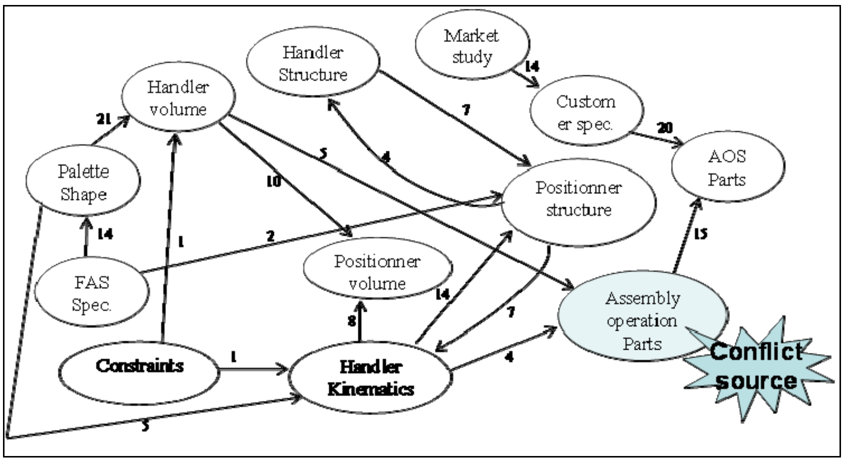

Figure 2. Example of a data dependencies network

In the case of complex products, whereas the corresponding data dependency network is huge, identifying the solution depending nodes and then the design activities to be re-executed may be ad-hoc and costly task. Not all the activities responsible of the identified depending data will be re-executed; such an operation is costly and time consuming. Thus, a concept of *critical Data Dependency Network* is introduced in order to reduce the number of solution depending nodes to consider for design process re-execution. In other words, this consists of eliminating data having low dependency degree and their entire successors among the impacted data[1]. Hence, based on the critical data dependency network, the identification of the activities to be re-executed is performed. Indeed, a set of SQL queries applied to the process execution database allow identifying the activities responsible on the elaboration of the critical data network nodes, the actors performing these activities as well as the input data and output data of each activity. Then, the actor responsible of the change management has at his disposal the set of activities to re-execute with the associated actors and Inputs/Outputs. Based on the input and output data, the order of executing activities can be determined. Indeed, when the output of an activity A corresponds to the input of another activity B, that means that activity A precedes activity B. In this paper, the focus is put on the resulting activities re-execution organisation. The aim of re-organising activities is to minimise the re-execution time by decreasing the probability and the magnitude of iterations in the newly executed process. This can be done by enhancing activities' overlapping and concurrent execution. In Section 4, a set of coordination strategies are proposed based on dependency degree values.

4. DESIGN PROCESS RE-ORGANISATION STARTEGIES

Depending on the dependency condition of the data to be changed (variability, sensitivity and completeness), i.e. depending on the probability of iteration on both feed forward and feedback dependencies between associated activities (those producing and those using the data to be changed), different strategies for re-executing these activities are examined.

Inspired from the coordination strategies developed by [9] and [10], a set of coordination strategies, i.e. ways of ordering activities and diffusing data, are proposed. First, we discuss the case of *dependency*, i.e. data linked with only feed forward dependency. Then, the case of *interdependency* is treated, i.e. data linked with both feed forward and feedback dependencies.

4.1 Dependency case: Only feed forward dependency

This case concerns two activities A and B, respectively producing data D1 and D2 where D1 and D2 are linked by feed forward dependency as shown in Figure 3. In this section, different values of completeness (*Weak, Not Vital, Vital and Extremely Vital.*) are considered; and for each case, depending on variability and sensitivity values, strategies of coordination are suggested.

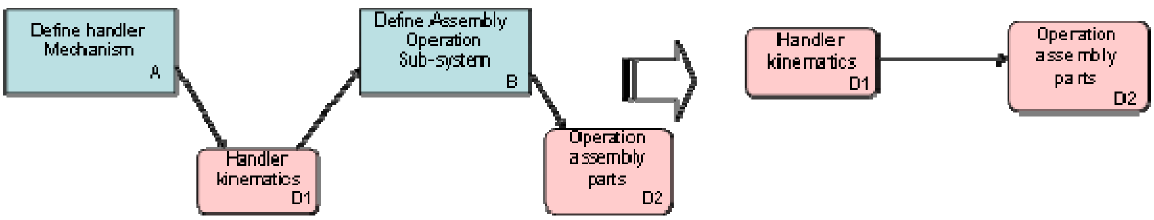

Figure 3. The dependency case

Case 1: if the upstream data could be used, by the downstream activity, given a certain maximum value.

For every value of the variability of the upstream data, the execution of the downstream activity does not require a precise value of this data. Thus, the adopted strategy would be the execution of the involved activities in early overlapping way without any risk of iteration. For instance, from the network illustrated in Figure 1 above, the definition of the "positionner structure" requires a weak definition of the "FAS specification". In fact, the FAS specification allows the definition of the general structure of the designed system and of its functioning. Therefore, lots of alternatives regarding the "positionner structure" could be applied, accordingly to the definition of the "handler

[1] *the methodology proposed to identify the critical data dependency network is not presented in this paper; it will be developed in a future publication.*

mechanism”. In this case, the execution of the activities could be performed in *early overlapping way*, where both activities have to start at the same time.

Case 2: if the upstream data should be used within a certain value range

- *Distributive overlapping*: makes possible to both start downstream activity with preliminary data and to pre-empt later changes in the exchanged upstream data. The impact of overlapping is distributed between upstream and downstream activities. Thus, this strategy is used in the case where the variability of the exchanged data is “Low variability” and the sensitivity of the downstream data to be produced is “Lower sensitivity” or “Moderate sensitivity”. For instance, from the network illustrated in figure 2 above, the definition of the FAS specification is “Low sensitivity” (S=1) to the “Low variability” of the positionner structure (V=1). Indeed, the modification of the “positionner structure” hasn’t got an important incidence on the “FAS specification” as long as the targeted functions of the “positionner mechanism” are fulfilled. Then, the execution of the associated activities could be performed in early overlapping way, where both activities have to start at the same time.
- *Iterative overlapping*: the activities are overlapped by beginning the downstream activity with preliminary data, and incorporating design changes in subsequent downstream iterations. The overlapping is said to be iterative. This strategy is used in the case where the sensitivity of the produced data is “Minor Sensitivity” and the variability of the exchanged upstream data is “Low Variability” or “Moderate Variability”.
- *Pre-emptive overlapping*: When the variability of upstream data is low, the data can be frozen earlier than its normal freeze time. This will allow “High sensitive” downstream data to start earlier with finalised data. In such a case, the exchanged data is to be pre-empted by taking its final value at an earlier point in time. This is called pre-emptive overlapping and would help to reduce design time by starting the downstream activity earlier but with frozen upstream activity data. This is the case of the activities associated with the “assembly operation parts” and the “handler kinematics”, where the sensitivity is “High” (S=3) and the variability is “Low” (V=1) of the assembly operation parts and the handler kinematics, respectively (cf. figure 2). In fact, the “assembly operation parts” is very sensitive to “handler kinematics” modifications since the cinematic of the handler constraints the components so that it would be possible to assembly then later one.
- *Divisive overlapping*: the upstream data is disaggregated into components to see if any of the components have low variability or if transferring any of the components in their preliminary form to the downstream activity is practical. This strategy is used in the case where the variability of the exchanged data is “High Variability” and the sensitivity of the downstream data to be produced is “Moderate Sensitivity”. This is actually the case of the activities associated with the handler structure and the positionner structure as illustrated in figure 2.
- *Activity and dependency link redefinition:* this strategy calls for disaggregating predecessor design data into two or more data fragments that can be released (as incomplete or partial design data) earlier than the planned one-shot release. Consequently, the follower (i.e. downstream) activity is broken into n sub-activities, where each sub-activity uses one upstream data fragment. There is no general rule for the optimal number of upstream partial data pieces required or the number of downstream activity breakdowns. This depends on the specific design situation being analysed and requires a thorough understanding of the design process details. This strategy is used in the case of a “moderate variability” of the upstream data and a “moderate sensitivity” of the downstream data. For instance, the handler volume as well as the assembly operation parts could be broken down into pieces of information. For instance, the “handler volume” information could be spitted into “jacks position on the frame” and into “wholes position on the palette”. Whereas, the “AOS parts” could be broken down to “frame”, “palette”, “pneumatic” and into “electrical connections”. Thus, the related activities could be decomposed into smaller activities and thus the resulting variability/sensitivity is lower. This decomposition decreases the dependency relationships between the “handler volume” and the “AOS parts” by distributing the modification effects on the sub-systems mentioned above (Jacks positions on the frame, wholes position on the palette).
- *Multifunctional team (concurrent execution)*: The basic goal of such a strategy is to guarantee that downstream concerns are considered upstream. This will result in: decreasing variability of a predecessor due to the fact that upstream activity engineer(s) are working closely with

downstream activity engineer(s); and lower sensitivity value of the successor due to the instantaneous feedback accomplished by the multifunctional team. This strategy is used in the case where the sensitivity of the produced data is "Major Sensitivity" and the variability of the exchanged upstream data is "Moderate Variability" or "High Variability".

Case 3: if the upstream data is at least "Low Variable", the downstream data is at least "Low Sensitive" and the upstream data is at least "Vital". If the upstream data completeness is "Vital" or "Extremely Vital" then the interest is to avoid as much as possible upstream rework because this induces long iterations. Thus given a low sensitive downstream data, and a moderate or high variability of the upstream data, it is more interesting to realise coordination with less preliminary diffusion as possible; this means to prioritize distributive then iterative overlapping in order to reduce upstream iteration. The best example for this case is the coordination between the "handler kinematics definition" and the "positionner volume definition" (cf. figure 2). However, if the variability of the upstream data is low and the downstream data sensitivity is moderate or major, pre-emptive strategy is more interesting then distributive overlapping in order to reduce design reworks in both upstream and downstream activities.

4.2 Interdependency Case: Feed Forward and Feedback Dependencies

In most product development processes, interdependency between activities is essentially a one-way dependency between an upstream activity and a downstream activity with a feedback dependency from the downstream activity to the upstream activity (cf. Figure 4). When the development process involves interdependent activities, it is divided into two stages: a planned stage and a random iteration stage. The first stage contains only the initial attempts of both activities. The second stage contains all the subsequent design iterations of both activities. There is no confusion on what activity to start first in this case.

The first stage consists of coordination strategy according to forward dependency degree between upstream and downstream (see § 4.1). The second stage represents coordination strategy according to feedback dependency degree between downstream and upstream.

As an illustration, we consider the interdependent activities in Figure 4 (Activity A and Activity B corresponding to the data D1 and data D2 respectively).

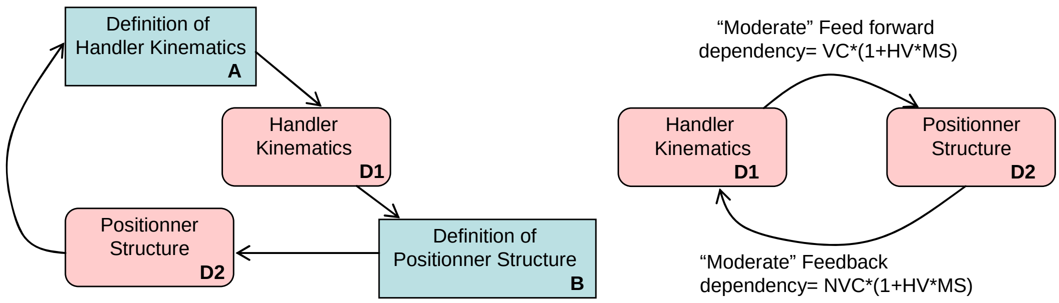

Figure 4. The interdependency case

In this example, feed forward dependency degrees is Moderate (Vital Completeness, High Variability and Moderate Sensitivity) and feedback dependency degree is Moderate as well (NVC, MS and HV). Thus, as the upstream variability is high and the downstream activity is moderate, the coordination strategy between activity A and activity B would be the Divisive overlapping (according to the case 2 of the § 4.1 above). Furthermore, as the downstream completeness is Vital, in order to avoid important upstream iterations, the divisive coordination should be combined with a pre-emptive strategy as shown in figure 5. After that, once the activity A1 is ready to diffuse the handler kinematics data (D1), the second step is launched and iterative overlapping upon the positionner structure (D2) is applied as shown in figure5. The same strategies are proposed for the activity A2 and B2 below.

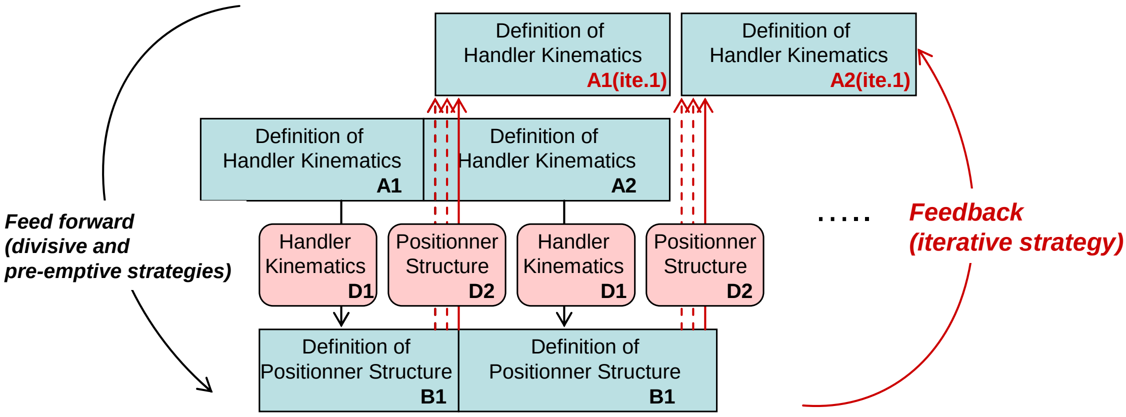

Figure 5. Organisation strategy for independent activities

5. TOWARDS A BETTER DESIGN PROCESS RE-ORGANISATION

Without careful management of the overlapped design activities, the development lead time and effort may increase. This section goes beyond the common recommendation to simply overlap activities as much as possible. Indeed, the earlier we begin overlapping, with less reliable upstream data, the greater the risk of a downstream redesign. Therefore, overlapping strategy policy should be determined. We propose a process optimisation model to support design managers to better re-organise the design process following from conflict resolution. This model is mainly based on the defined uncertainty attributes: completeness, variability and sensitivity (cf. Section 2). We assume that data sensitivity and variability causes design iterations on upstream and downstream activities. Furthermore, data completeness influences the magnitude of these iterations. Hence, functional interaction and overlapping rate measures are studied and taking into account in order to optimise the design process re-organisation.

5.1 Objectives

Concurrent Engineering can be used to either (a) increase the product quality/performance for fixed development time, or (b) reduce the development time and effort without explicit consideration of product quality/performance issues. In this section, we propose a model for improving the understanding of the latter. The goal of this section is to provide insights about optimal strategies to manage dependant activities under uncertainty conditions. The uncertainty conditions depend on upstream data variability and completeness and downstream data sensitivity.

5.2 Parameters for better re-organisation

The uncertainty conditions (variability, sensitivity and completeness), the *interaction rate* and the *overlapping rate* are the main parameters considered in this paper. The *interaction rate* (denoted α in figure 6) is the multidisciplinary interaction rate. It corresponds to the required time of cooperation among the actors associated with the dependent activities. This interaction, also called *functional interaction* in [11], allows the actors to solve together the design problems. The *overlapping rate* (denoted β in the figure 6) corresponds to the overlapping time of the dependant activities [12] [11].

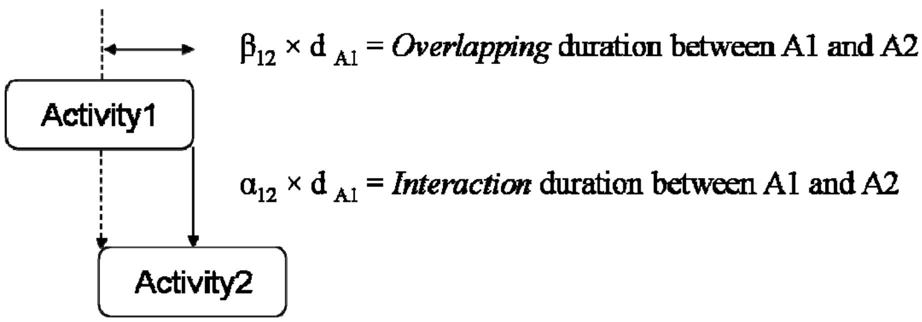

Figure 6. The interaction rate and the overlapping rate related to the dependent activities

Quantitative methods have been proposed in order to provide interaction rates and overlapping rates associated with the dependent activities in the design process [13] [11]. The interaction rates have been estimated from several previous projects. Authors used questionnaires to associate disciplines (and functions) to the different participants to the project. The type and the frequency of the interaction are also associated. Similar questionnaires are used in order to define the rates of overlapping. For example, in [11], to improve the reliability of collected information, the authors used multiple data sources, including interviews, surveys, attendance at project meetings, informal conversations, company documents, and observations of the new product development process.
In [11], scales of arbitrary percentages are associated to each of the both interaction and overlapping rates values:

- *nul* (0%), *low* (25%), *moderate* (50%) and *dedicated interaction* (100%) are different values associated to the interaction rate α,
- *nul* (0%), *low* (33%), *moderate* (66%) and *high* (100%) are different values associated to the overlapping rate β.

5.3 Parameters relationships to optimise the process re-organisation [14]

The focus of the present section is to describe the evolution of the both rates α and β regarding the values of the uncertainty conditions (variability, sensitivity and completeness) of the exchanged information between dependant activities. Estimation of the interaction and the overlapping rates are provided to optimise the design process time and the efforts (person-days).
Research studies such as [13] [15] [11] have determined, through simulation, the effects on development time and effort of changing the amounts of interaction and overlapping for new product development (NPD) project under differing uncertainty conditions. To accomplish these simulations, these studies selected uncertainty condition akin to those in [9], and then developed a stochastic model of the NPD process. These simulations were, afterwards, compared to previous [16] [17]. In our research work, a qualitative extension of these works is undertaken. In the following, the relationship between the parameters described previously is discussed, based on previous results [11]:

1. The increasing of the interaction rate (α) decreases the development time and effort. Indeed, increasing multidisciplinary interaction implies less design iteration and thus less effort. However, up to a certain value of the interaction rate, increasing the interaction leads to increasing the development effort. In fact, the *churn iteration* occurs in upstream activity by the increasing of α. Churn iteration represents redoing an activity by making an informal incremental change. Exceed certain interaction rate; the churn iteration raises the effort and eventually the development time. Thus, it is highly recommend to not exceed a low rate of interaction (25% for instance) in the case of high variability and to not exceed a moderate rate (50% for instance) in the case of low or moderate variability.
2. There is no effect on development time as overlapping rises without interaction (α =0%). Given a null interaction rate, increasing overlapping rate has the beneficial effect of lowering development time by allowing a more partially concurrency execution of the activities. On the other hand, increasing overlapping rate also has the harmful effect of raising both effort and development time due to the repetition of more activities when a design iteration occurs. In addition, a very high probability of design iteration exists. In this case, both of the development time and effort can increase (because of several repetitions). However, the beneficial and harmful development time effects tend to cancel each other out.
3. At positive values of interaction rate (α ≠ 0), the overlapping rate and the uncertainty conditions (variability and sensitivity) interact in their effects on development time. Indeed, in the low uncertainty conditions (blue and white cells in figures 7a and 7b) development time and effort decreases since that such condition implies a low repetition probability as well as a partial concurrency execution of the activities. While in the highest-uncertainty condition, (S=2, 3 and V=2, 3), development time and effort increase slightly. Indeed, increasing β implies high probability of design iteration and, thus, more upstream data changes. It is then recommended to decrease the overlapping rate β in the highest-uncertainty conditions and decrease it in lowest-uncertainty conditions. This result is confirmed by [18] [10] [19]. The overlapping strategies (β≠0) are recommended in the case of low-uncertainty condition, while for highest-uncertainty condition, the sequential or the multifunctional team strategies (β=0) are suggested (cf. Section 4.1).

The results presented above are verified in the case where only variability and sensitivity attributes are taken into account for the uncertainty conditions. In this research paper, this case refers to a *Non Vital* completeness. We remain that the completeness expresses the suitability of an input data to the creation of an output data.

4. Given a Vital or Extremely Vital completeness values, the duration of the upstream iteration could be significant. Indeed, the upstream activity runs into a delay when iteration occurs. Hence, depending on the upstream variability, the completeness plays a major role on iteration incidence. Consequently, a highest-variability and a highest-completeness lead to a major impact on the product development time and effort (figures 7a and 7b). It is recommended in this case to assign lowest-interaction rate (α) to highest-variability. Furthermore, given high upstream activity completeness, the highest the sensitivity of downstream activity is, the more it will be impacted by the upstream activity delay. It is, then, recommended to increase the overlapping rate (β) when downstream sensitivity decreases (figures 7a and 7b).

	V = 1	V = 2	V = 3
S = 1	α = 50% β = 66%	α = 25% β = 66%	α = 25% β = 66% (β = 0%)*
S = 2	α = 100% β = 33%	-	α = 50% β = 0%
S = 3	α = 100% (α = 50%)* β = 33%	α = 50% β = 0%	α = 50% β = 0%

Blue: Low-impact of uncertainty condition
White: Moderate-impact of uncertainty condition
Red: High-impact of uncertainty condition
*** Effort is the performance criteria to optimise**

Figure 7a. Interaction and Overlapping rates: the case of moderate completeness (C = 2)

	V = 1	V = 2	V = 3
S = 1	α = 25% β = 66%	α = 25% β = 66%	α = 25% β = 66% (β = 0%)*
S = 2	α = 100% β = 33%	-	α = 25% β = 0%
S = 3	α = 50% β = 0%	α = 25% β = 0%	α = 25% β = 0%

White: Moderate-impact of uncertainty condition
Red: High-impact of uncertainty condition
*** Effort is the performance criteria to optimise**

Figure 7b. Interaction and Overlapping rates: the case of high completeness (C = 3)

To summarise, a set of conclusion derives from the above discussion. First of all, no matter what the uncertainty condition, the high likelihood to re the overlapped activities adds to effort, at least at higher-overlapping rate, and eliminates the initial development gains. Secondly, given any value of overlapping rate (β), increasing the interaction rate (α) does initially lower effort and development time, but subsequently leads to trade-off between more effort and less development time. The beneficial effects of fewer design versions outweigh the harmful effects of more churn only up to a certain amount of interaction. Thirdly, providing more overlapping rate when some interaction occurs is appropriate in low, but not in moderate and high, uncertainty conditions. A trade-off exists between the negative effects of repeating more activities when design iteration occurs versus the positive effects of typically having somewhat fewer design iteration. The former tend to outweigh the latter, when as under moderate and high uncertainty, design iterations occur relatively frequently.

6. CONCLUSIONS

In this paper, a framework to address the problematic of solution impact assessment, following from the conflict resolution phase, is presented. First of all, based on the DEPNET methodology, the impact data as well as the managing activity are identified. Secondly, strategies to re-organise these activities are proposed based on uncertainty condition (*Completeness, Variability and Sensitivity*). These strategies are partially inspired by studies addressing the overlapping problem by developing approaches to study Concurrent Engineering (CE) process. Finally, a parameter-based framework to better manage the overlapping of interdependent activities is presented. For this purpose, we take into account two more parameters, *the overlapping rate and the interaction rate*. The relationship between these parameters and uncertainty condition is discussed to identify various appropriate types of overlapping.

We should note, however, that the results showed in figures 7a and 7b were not computer modelled and simulated. These results are obtained following analyses that we carried out starting from the results of simulations and observations obtained in [13] and [11], and starting from our comprehension of the relationship between the various uncertainty condition attributes. The overlapping and interaction rates, shown in figures 7a and 7b, are only as an indication and represent tendencies rather than exact values. Hence, future work is to develop a computer-based model to simulate a design process, which will contain effort, development time, uncertainty condition, overlapping rate and interaction rate. This is essential to validate the proposed framework.

REFERENCES

[1] Klein M. Conflict Management as Part of an Integrated Exception Handling Approach. AI in Engineering Design Analysis and Manufacturing, Vol. 9, (1995) 259-267.

[2] Zhuang R. Conflict Detection in Web Based Concurrent Engineering Design. At Master Thesis, University of Florida. (1999).

[3] Slimani K., DaSilva C.F., Médini L. and Ghodous P. Conflict mitigation in collaborative design. International Journal of Production Research, Vol. 44(9), 1 May, (2006) 1681-1702.

[4] Cooper S. and Taleb-Bendiab A. CONCENSUS: multi-party negotiation support for conflict resolution in concurrent engineering design. Journal of Intelligent Manufacturing, Vol. 9, (1998) 155–159.

[5] Lu S.C.-Y., Cai J., Burkett W. and Udwadia F. A Methodology for Collaborative Design Process and Conflict Analysis. CIRP Annals, Vol. 49(1), (2000) 69-73.

[6] Rose B., Gzara L. and Lombard M. Towards a formalization of collaboration entities to manage conflicts appearing in cooperative product design, in: *Methods and Tools for Cooperative and Integrated Design*, S Tichkiewitch and D Brissaud, Editors. Kluwer Academic. Printed in The Netherlands. (2004).

[7] Ouertani M.Z., Grebici K., Gzara L., Blanco E. and Rieu D. DEPNET: A Methodology for Identifying and Qualifying Dependencies Between Engineering Data. The future of product development, proceedings of 17th CIRP International Design Seminar. 26-29 March. Berlin, Germany: Springer-Verlag (2007).

[8] Ouertani M.Z., Gzara-Yesilbas L., Lombard M. and Ris G., Managing Data Dependencies to Support Conflict Management, in 16th CIRP International Design Seminar. 2006: Kananaskis, Alberta, Canada.

[9] Krishnan V., Eppinger S.D. and Whitney D.E. A Model-Based Framework for Overlapping Product Development Activities. Management Science, Vol. 43(4), (1997) 437-451.

[10] Loch C.H. and Terwiesch C. Communication and Uncertainty in Concurrent Engineering. Management Science, Vol. 44, (1998) 1032-1048.

[11] Bhuiyan N., Gerwin D. and Thomson V. Simulation of the New Product Development

Process for Performance Improvement. Management Science, Vol. 50(12), (2004) 1690-1703.
[12] Joglekar N.R., Yassine A.A., Eppinger S.D. and Whitney D.E. Performance of Coupled Product Development Activities with a Deadline. Management Science, Vol. 47(12), (2001) 1605-1620.
[13] Swink M., Sandvig J. and Mabert V. Customizing Concurrent Engineering Processes: Five Case Studies. Journal of Product Innovation Management, Vol. 13(3), (1996) 229-244.
[14] Grebici K. La Maturité de l’Information et le Processus de Conception Collaborative. At Thése de doctorat de l'INP Grenoble. (2007).
[15] Bhuiyan N. Dynamic Models of Concurrent Engineering Processes and Performance. At PhD dissertation, McGill University. (2001).
[16] Burton R.M. and Obel B. The validity of computational models in organization science: From model realism to purpose of the model. Computational & Mathematical Organization Theory, Vol. 1(1), (1995) 57-71.
[17] Lee J.-S., Carley K. and Effken J., Validating a Computational Model of Decision-Making using Empirical Data, in North American Association for Computational Social and Organizational Sciences, NAACSOS Conference. 2003: Pittsburgh, PA.
[18] Smith R.P. and Eppinger S.D. Deciding between Sequential and Concurrent Tasks in Engineering Design. Concurrent Engineering : Research and Applications, Vol. 6(1), (1997) 15-25.
[19] Terwiesch C., Loch C.H. and DeMeyer A. Exchanging Preliminary Information in Concurrent Engineering: Alternative Coordination Strategies. Organization Science, Vol. 13(4), (2002) 402 - 419.

Contact: Mohamed-Zied Ouertani
Nancy University
Research Center for Automatic Control (CRAN)
BP 239
54506, Vandoeuvre-lès-Nancy
France
Tel: 00 33 3 83 68 44 30
Fax: 00 33 3 83 68 44 59
e-mail: mohamed-zied.ouertani@cran.uhp-nancy.fr