Maintainability and safety indicators at design stage for mechanical products

A. Coulibaly a,*, R. Houssin a,b, B. Mutel a

a Laboratoire du Génie de la Conception (LGECO), INSA Strasbourg, 24, Bd de la Victoire, 67084 Strasbourg, France
b Louis Pasteur University (Strasbourg), IPST-ERT Enroulement, 15-17 rue de Maréchal lefèvre, 67100 Strasbourg, France

Received 28 May 2007; received in revised form 1 December 2007; accepted 17 December 2007

Abstract

With the industrial products competition, reliability, maintainability and safety are key characteristics for availability improvement. This is mainly true in industries like automotive, aeronautic, or NC machines tools. Therefore, it is crucial to predict these characteristics as soon as possible before the manufacturing starts. This paper presents an approach to provide indicators for maintainability and safety prediction at early stage of design. The assessment procedure uses the product CAD 3D model and an associated semantic matrix gathering information on the product components criticality and reliability. Using this information we calculate indicators for the product maintainability and safety. An academic application is developed to illustrate our approach and point out the interaction between maintainability and safety constraints to determine suitable solution. In conclusion, we state on possible extensions of this approach for evaluating other product lifecycle characteristics and we give guidelines on the implementation aspects.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Maintainability; Safety; Reliability; Product behavior; Semantic modeling

1. Introduction

The usual product design approaches like financial, market or productivity approaches come to their limits. So firms are looking for new control to generate competition advantage. Environmental aspects and lifecycle constraints must be now considered in industrial production. This implies the study of different product behaviors along its lifecycle: including utilization, preventive maintenance, repairing, recyclability, wastes management at the end of life. But there is a lack of efficient tools for behavioral performance prediction of these characteristics at early design stage. This paper proposes an approach for product maintainability and safety prediction based on the behavioral performance assessment framework outlined in [1].

Section 2 presents some research works related to product modeling and behavioral assessment. We present in Section 3 the product semantic modeling procedure, the semantic matrix elaboration and the behavioral indicators assessment. Section 4 describes the maintainability assessment procedure. Section 5 does the same for safety assessment. An application is presented in Section 6 to show how to determine maintainability and safety indicators for different alternative design solutions.

Then, we discuss about the interaction between maintainability and safety constraints for design solutions validation. In conclusion, we mention possible extensions of the approach to other lifecycle characteristics assessment and guidelines are given on implementation aspects.

2. Related works

These last years some investigations have been to consider lifecycle constraints at the design stage. Gero [24] has presented a generic product model FBS (function, behavior and structure) concept, for capturing product data and other process information into a global and multi-views model. However, this model does not specifically focus on product behavioral performance evaluation. Other different industrial tools using virtual reality systems [13,17] and collaborative design environment [14] are proposed to verify product functionalities and to analyze its maintainability. However,
usually such systems are used late after the product is entirely designed. In addition, these systems are expensive and often difficult to use. Other investigations mentioned in [11,16,18], present contributions on the integration of maintainability criteria in the product design process. These works give some rules to apply for the design for maintainability.

Users’ safety constraints are often integrated late in the complex systems design process. In the literature, some methods are proposed by the safety Standards and applied in companies [25]. In these methods, the designer treats the safety problem after the product functional design. This late integration is mainly due to the lack of tools and methods to assist designer in his work. So, classically, the designer proceeds as follows:

- first of all, the designer defines solutions to fulfill the technical functions as defined in the functional specifications;
- then a solution is defined, the designer tries to make the system safer and if possible easier to be maintained when failures will happen. However, at this level, after the system is designed, it becomes more expensive to make it safer or maintainable.

Another method presented in [6] proposes to integrate as soon as possible the safety in design process.

Contrary to the above approaches, we consider maintainability and safety as required functions and not as constraints. In our approach, from the beginning of design process, we manage to build technical solutions that assure the product maintainability and safety. We also focus on cases where the user intervenes in dangerous zones to do some maintenance operation (e.g., near a burning component).

3. Product model for behavioral assessment

3.1. Behavioral assessment framework

Our behavioral modeling work aims to provide a generic approach for evaluation of product performance at different situations of its lifecycle [30].

Fig. 1 presents the main steps performed in the behavioral assessment framework with maintainability and safety applications.

(a) The first step begins from an initial product solution represented by the corresponding CAD 3D assembly model. Such model specifies the geometry and dimensions of the different components and their topological interrelationships. Then the CAD software provides the corresponding product part-list that gives a structural decomposition into main components.

(b) The second step upgrades the CAD product model with additional semantic data related to components and to assembly links characteristics. These characteristics inform about the difficulty or the easiness to remove the links and about dangerous phenomena that could be generated by this product solution.

(c) At the third step, a set of components that are bind together by different types of fasteners are defined [15]. Depending on the assessment objective, the decomposition may be structural, functional or trading. Here the structural decomposition is adopted in order to characterize the components interconnections.

So the product is represented by \( n \) different components \( C_i \), interconnected with assembly links as shown in the graph, Fig. 2. In this figure, the assembly link between a couple of components \( C_i, C_j \) is noted \( V(i,j) \) and takes values defining the kind of assembly link used to assemble the two components. \( V(i,j) \) could be screwing, clipping, welding, contact, etc. \( V(i,j) \) is a structural attribute that can be captured from the CAD model. This attribute does not inform about functional link between components and contains no information about the relative movement at the links. The designer must provide these semantic characteristics.

In the case of a product with a huge number of components, the assembly graph becomes too dense and difficult to read. Therefore, we propose a semantic matrix...
At the fourth step, we build the product semantic matrix. In this matrix an n-components product is described by its components $C_i$ (with $i = 1 \ldots n$). The diagonal terms $P_i$ represent components properties (material, surface status, heat treatment...). These properties inform about the potential risk of wear occurrence in case of relative movement between components. The number of occurrences of component $C_i$ is $nb_i$. The assembly type between two components $C_i$ and $C_j$ is noted $L(i,j)$. Compared to $V(i,j)$ used in Fig. 2, $L(i,j)$ has a broader definition of assembly link notion. It is not limited to information about the structural links between components but includes the notion of functional link and information about possible relative movement between the two components. $L(i,j)$ is a semantic vector expressed as: $L(i,j) = (V(i,j), f, m)$.

where $V(i,j)$ is the link type as defined for Fig. 2; $f$ is a boolean parameter informing about the existence or not of a functional link between components $C_i$ and $C_j$.

- $f = 1$, if there is a functional link
- $f = 0$, if there is none;

$m$ is also a boolean parameter:

- $m = 0$, if the two components are embedded with no relative movement
- $m = 1$, if there is a possible translation or rotation between components

For example, $L(i,j) = \langle contact, 1, 0 \rangle$ means that $C_i$ and $C_j$ are in contact, with a functional link and there is no relative movement between them.

A relative movement between components may implies a wear of the components. Then the affected components must be replaced more or less frequently.

For indicator evaluation, we associate to each link type a waiting time required. $V(i,j)$ takes a high value if the link removal is more difficult as for welding.

In Table 1 we mention an estimation of waiting factors related to the time required for the disassembly depending on the relative level of difficulty for different assembly techniques [10].

A part from assembly links we also considered additional semantic data concerning non-graphic characteristics. So, for each component $C_i$ we define its specific properties ($P_i$), functional criticality ($K_i$) and reliability ($R_i$).

Depending on the functional importance of the components, the designer must estimate the functional criticality levels. In Fig. 2, such critical components are in dotted line. The valuation of the criticality level may vary from 0 for non-critical components up to values depending on the system robustness when the component fails. For instance, a housing is a critical component to satisfy user safety specifications. So without this housing, the system could not be used. Here, we consider that a component is either critical with value 1 or not critical with value 0.

The components individual reliabilities are important input characteristics to locate weak elements that constitute potential failure origins.

At the fifth step, we proceed to our behavioral assessment procedure by proposing maintainability and safety indicators detailed in Section 4. Before calculating these indicators, we state assumptions for the behavioral assessment in the following paragraph.

### 3.2. Assumptions for behavioral assessment

In the maintainability assessment procedure, reliability data are used as input of the product semantic matrix. We assume that the product has an acceptable level of reliability. This condition can be easily verified by calculating the product global reliability, $R(t)$, using Eq. (1) where $R_i(t)$ is the component $C_i$ reliability. $R_i(t)$ is defined as the probability that this component performs its function in a given operating...
conditions over its lifetime. For a product with independent components, the global reliability is expressed by:

\[ R(t) = \prod_{i=1}^{n} R_i(t) \]  

(1)

\( R(t) \) is specified in the requirements document and it must be better than the minimum admissible reliability threshold level fixed by the customer, \( R_0 \). Therefore, in order to achieve this reliability level all \( R_i(t) \) must be greater than \( R_0 \).

Then, we perform the behavioral assessment only for critical components with criticality level greater than \( K_0 \) and with reliability \( R_i(t) \) less than the minimum admissible level \( R_0 \).

Therefore, instead of determining maintainability indicator for all components, we only consider critical and non-reliable components. This defines a subset of components to consider in assessment procedure,

\[ C_{\text{assessment}} = \{ C_i \text{ with } K_i \geq K_0 \text{ and } R_i(t) \leq R_0 \} \]  

(2)

Our work is based on the hypotheses that components reliability can be determined in the case of standard components using different existing Reliability Database for industrial equipments. Such databases are used for advising designers on the choice of components \([7,13,21]\). For new components for which no statistical data are available, simulation approaches mentioned in Appendix A may be used \([10]\).

In the next section, we define the product maintainability criteria and propose a method to determine associated indicators.

4. Maintainability assessment

4.1. Maintainability criteria

The maintainability is commonly defined as the characteristics of equipment design and installation that provides the ability for this equipment to be repaired easily and efficiently. From the user point of view, maintainability refers to the aspects of a product that increase its serviceability and reparability, decreases the cost-effectiveness of maintenance, and ensures that the product meets the requirements for its intended use \([19,20]\). For high-integrated products consisting of mechanical parts, electronic devices and software the maintainability assessment must take into account all these different aspects. Here, we only consider the case of basic mechanical products with no electronic equipments and no software. So, the maintainability depends on the complexity of the structure: i.e. the geometry of parts and how components are assembled.

Depending on what activities are considered, maintainability can be affected by actions beginning from the failure detection and those concerning diagnostic, reparation and test. Maintainability depends on all criteria that may affect the main maintenance steps with different actions to be carried out to bring the product back to its functional status.

Different criteria that may affect product maintainability have been identified in \([2]\). Table 2 presents these different criteria into two categories.

Intrinsic criteria are those depending on the product structure configuration. Contextual criteria depend on the maintenance context including human resources, equipments, or working conditions.

Here, we focus on intrinsic criteria that are the most important for alternative design solutions comparison.

To assess the maintainability we consider the time required to detect components or subsystems failures and to repair the

<table>
<thead>
<tr>
<th>Components</th>
<th>C_1</th>
<th>C_2</th>
<th>C_3</th>
<th>C_4</th>
<th>C_5</th>
<th>Criticality</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_1</td>
<td>P_1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>C_2</td>
<td>P_2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C_i</td>
<td>LinkType ((C_i, C_j) = Li)</td>
<td>LinkType ((C_i, C_j) = Li)</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>K_i</td>
<td>R_i</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>C_n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P_n</td>
<td>K_n</td>
</tr>
</tbody>
</table>

\[ \text{Number links} d_1, d_2, \ldots, d_n, d_n, K_n(\text{threshold}), R_n(\text{threshold}) \]

Fig. 3. Product semantic matrix.

Table 1

<table>
<thead>
<tr>
<th>Link type</th>
<th>Required time</th>
</tr>
</thead>
<tbody>
<tr>
<td>No contact</td>
<td>0</td>
</tr>
<tr>
<td>Contact</td>
<td>1</td>
</tr>
<tr>
<td>Clipping</td>
<td>2</td>
</tr>
<tr>
<td>Screwing</td>
<td>3</td>
</tr>
<tr>
<td>Bolting</td>
<td>4</td>
</tr>
<tr>
<td>Housing</td>
<td>7</td>
</tr>
<tr>
<td>Gluing</td>
<td>8</td>
</tr>
<tr>
<td>Hinge</td>
<td>3</td>
</tr>
<tr>
<td>Welding</td>
<td>10</td>
</tr>
</tbody>
</table>
product after failure. Let, $T_{\text{Maint}}$ be this overall time, expressed as follows:

$$T_{\text{Maint}} = T_{\text{FD}} + T_{\text{Diag}} + T_{R} + T_{C}$$

(3)

where $T_{\text{FD}}$ is the failure detection time, $T_{\text{Diag}}$ is the diagnostic time, $T_{R}$ is the repairing time and $T_{C}$ is the control time.

To increase product serviceability, all these different times must be as shorter as possible. In practice, to value the maintainability, the 1010/CCT specification [7] defines the MTTR (mean time to repair) as a most significant criterion. Usually this measure is obtained from statistical data collected over a certain period of product utilization. This is suitable for existing products. However, in the case of new products for which no statistical data exists, the MTTR may be estimated by using simulation methods or any other techniques. The MTTR is the total time required for making diagnostic, reparation or replacement tasks and control.

$$\text{MTTR} = T_{\text{Diag}} + T_{R} + T_{C}$$

(4)

The diagnostic time, $T_{\text{Diag}}$, depends on the type of failure. It can be instantly in the case of breaking of main components. But, it can also take from a few minutes to much longer period if the failure occurred progressively as in the case of parts wear. Thus, in practice, $T_{\text{Diag}}$ cannot be estimated at the design stage. In the other hand the control time $T_{C}$ requires to verify that the product works properly. This time can be very short or imply a long period of settings. Here, this time is not taken into account. So the time devoted to repairing or replacement tasks, $T_{R}$, is usually the most important characteristic that determines product maintainability. $T_{R}$ depends on many criteria like: disassembly/assembly operations to be performed, components Accessibility, components or sub-assemblies maneuverability, reparability and maintenance resources. In this study, we have considered the criteria of disassembly/assembly operations proposed in [8]. In the following sections, we present the approach to predict the maintainability using a module implemented inside conventional CAD systems environment.

### 4.2. Maintainability indicators

We define the intrinsic indicator as the total time required for disassembling all the product components. Let $S_{N}^{k}$ be the optimal disassembly sequence to access to the component $k$, where $N$ is the number of components to remove to reach the target component $k$. Remove$(S_{N}^{k}, i)$ is the time required to remove the component $i$ in the remaining sequence after the $(i−1)$ first components have been already removed. Eq. (5) defines the time required to reach a target component $C_{k}$, and we express the general maintainability indicator for the whole product by (6a):

$$R_{\text{time}}^{k} = \sum_{i=1}^{N} \text{Remove}(s_{N}^{k}, i)$$

(5)

$$I_{M} = \sum_{k=1}^{n} R_{\text{time}}^{k}$$

(6a)

where $n$ is the total number of components or sub-assemblies that the product consists of. For better maintainability, $I_{M}$ must be as smaller as possible.

This general maintainability indicator gives a global idea of the product complexity but it is not significant information for choosing between different design solutions. This general indicator does not consider components criticality. In the case of criticality-based maintainability, the indicator is determined by taking into account only critical components identified in the product. Then we define a criticality-based maintainability indicator $I_{M}^{k}$. This assumes that the product may consist of $N$ components but only a few of them are considered critical. The criticality is defined here as the ability of the system to operate with a certain failure tolerance depending on the components technical functions.

The criticality will be set to 1 for the components that cause the system to stop working when they fail. The level 0 is affected to components with no influence on the system operation. While modeling a product the designer may attribute criticality level to the different parts.

<table>
<thead>
<tr>
<th>Intrinsic criteria</th>
<th>Reparability: ability to be repaired after failure or damage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accessibility: easiness to reach a component inside the assembly</td>
</tr>
<tr>
<td></td>
<td>Assemblability: ability to be assembled from an assembly</td>
</tr>
<tr>
<td></td>
<td>Disassemblability: ability to be removed from an assembly</td>
</tr>
<tr>
<td></td>
<td>Standardization: standard component or equipment</td>
</tr>
<tr>
<td></td>
<td>Interchangeability: ability to be replaced with another component</td>
</tr>
<tr>
<td></td>
<td>Survivability: ability of the product to continue to work after the failure of a considered component</td>
</tr>
<tr>
<td></td>
<td>Redundancy: for components existing in multiple equivalent occurrences</td>
</tr>
<tr>
<td></td>
<td>Competencies: human competencies required to diagnose and to repair</td>
</tr>
<tr>
<td></td>
<td>Tools: maintenance equipments like keys, screwdrivers…</td>
</tr>
<tr>
<td></td>
<td>Logistics: delivery of spare parts, transportation of maintenance team…</td>
</tr>
<tr>
<td></td>
<td>Environment: working conditions like lighting, temperature…</td>
</tr>
<tr>
<td></td>
<td>Detectability: easiness to detect failure and components concerned with</td>
</tr>
<tr>
<td></td>
<td>Testability: ability for a component or a sub-system to be tested…</td>
</tr>
<tr>
<td></td>
<td>Maneuverability: ability for a component or sub-system to be handled</td>
</tr>
<tr>
<td></td>
<td>Auto diagnostic: ability for a system to perform self-testing procedures</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contextual criteria</th>
<th>Accessibility: ability to be assembled from an assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Components or sub-assemblies maneuverability, reparability and maintenance resources</td>
</tr>
<tr>
<td></td>
<td>Human competencies required to diagnose and to repair</td>
</tr>
<tr>
<td></td>
<td>Maintenance equipments like keys, screwdrivers…</td>
</tr>
<tr>
<td></td>
<td>Delivery of spare parts, transportation of maintenance team…</td>
</tr>
<tr>
<td></td>
<td>Working conditions like lighting, temperature…</td>
</tr>
<tr>
<td></td>
<td>Easiness to detect failure and components concerned with</td>
</tr>
<tr>
<td></td>
<td>Ability for a component or a sub-system to be handled</td>
</tr>
<tr>
<td></td>
<td>Ability for a system to perform self-testing procedures</td>
</tr>
<tr>
<td></td>
<td>Component to be replaced with another component</td>
</tr>
<tr>
<td></td>
<td>System to continue to work after the failure of a considered component</td>
</tr>
<tr>
<td></td>
<td>Components existing in multiple equivalent occurrences</td>
</tr>
<tr>
<td></td>
<td>Human competencies required to diagnose and to repair</td>
</tr>
<tr>
<td></td>
<td>Maintenance equipments like keys, screwdrivers…</td>
</tr>
<tr>
<td></td>
<td>Delivery of spare parts, transportation of maintenance team…</td>
</tr>
<tr>
<td></td>
<td>Working conditions like lighting, temperature…</td>
</tr>
<tr>
<td></td>
<td>Easiness to detect failure and components concerned with</td>
</tr>
<tr>
<td></td>
<td>Ability for a component or a sub-system to be handled</td>
</tr>
<tr>
<td></td>
<td>Ability for a system to perform self-testing procedures</td>
</tr>
<tr>
<td></td>
<td>Component to be replaced with another component</td>
</tr>
<tr>
<td></td>
<td>System to continue to work after the failure of a considered component</td>
</tr>
<tr>
<td></td>
<td>Components existing in multiple equivalent occurrences</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2 Maintainability criteria classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic criteria</td>
</tr>
<tr>
<td>Reparability: ability to be repaired after failure or damage</td>
</tr>
<tr>
<td>Accessibility: easiness to reach a component inside the assembly</td>
</tr>
<tr>
<td>Assemblability: ability to be assembled from an assembly</td>
</tr>
<tr>
<td>Disassemblability: ability to be removed from an assembly</td>
</tr>
<tr>
<td>Standardization: standard component or equipment</td>
</tr>
<tr>
<td>Interchangeability: ability to be replaced with another component</td>
</tr>
<tr>
<td>Survivability: ability of the product to continue to work after the failure of a considered component</td>
</tr>
<tr>
<td>Redundancy: for components existing in multiple equivalent occurrences</td>
</tr>
<tr>
<td>Competencies: human competencies required to diagnose and to repair</td>
</tr>
<tr>
<td>Tools: maintenance equipments like keys, screwdrivers…</td>
</tr>
<tr>
<td>Logistics: delivery of spare parts, transportation of maintenance team…</td>
</tr>
<tr>
<td>Environment: working conditions like lighting, temperature…</td>
</tr>
<tr>
<td>Detectability: easiness to detect failure and components concerned with</td>
</tr>
<tr>
<td>Testability: ability for a component or a sub-system to be tested…</td>
</tr>
<tr>
<td>Maneuverability: ability for a component or sub-system to be handled</td>
</tr>
<tr>
<td>Auto diagnostic: ability for a system to perform self-testing procedures</td>
</tr>
</tbody>
</table>

| Contextual criteria                            |
| Accessibility: ability to be assembled from an assembly |
| Components or sub-assemblies maneuverability, reparability and maintenance resources |
| Human competencies required to diagnose and to repair |
| Maintenance equipments like keys, screwdrivers… |
| Delivery of spare parts, transportation of maintenance team… |
| Working conditions like lighting, temperature… |
| Easiness to detect failure and components concerned with |
| Ability for a component or a sub-system to be handled |
| Ability for a system to perform self-testing procedures |
| Component to be replaced with another component |
| System to continue to work after the failure of a considered component |
| Components existing in multiple equivalent occurrences |

Please cite this article in press as: A. Coulibaly, et al., Maintainability and safety indicators at design stage for mechanical products, Comput Industry (Ind) (2008), doi:10.1016/j.compind.2007.12.006
Then, for a given level of criticality $\chi$, the maintainability performance is defined by the following indicator:

$$I_M^* = \sum_{k=1}^{N^*} R_{\text{time}}^k \times w_k$$

(6b)

where $N^* = |\Omega|$, is a number of critical components,
$$\Omega = \{ \text{component} \in \text{product}, \text{criticality} \geq \chi \}$$

and $w_k$ is the criticality level of the component $k$, $w_k \in [0, 1]$. This indicator is used to compare different alternative design solutions.

This maintainability indicator is also influenced by additional contextual criteria including human operator’s competence and related errors, tooling, maintenance conditions (temperature, dusts, lighting, visibility, etc.) [3]. In this paper, only the intrinsic criteria are considered.

5. Safety assessment

5.1. Safety criteria

Safety integration must be in coherence with the application of the strategy of prevention of risks required by safety standards [26]. This strategy is based on three levels [5]:

1. Intrinsic level “Risk reduction by design”.
2. Operational level “Safeguarding”. Using barriers increases safety but decreases accessibility and visibility that are necessary for the user to perform appropriately and as quickly as possible his task. This may lead to decrease the productivity and the system performance [12].
3. “Instruction or information for use” level. This level provides posters of safety instructions to inform users about risks of accident.

From these three levels some safety criteria have been defined in [5,6]. In Table 3, we propose additional intrinsic and contextual criteria related to working conditions.

As we explained for maintainability assessment, we do not estimate safety systematically for all components. From the decomposition made for maintainability assessment, we focus on components that require human interventions in working situation or during maintenance operations.

In the following sections, we consider accessibility and prevention criteria to determine a safety indicator for a given design solution.

5.2. Safety indicator

Using the risks analysis model proposed by Houssin et al. [6] to evaluate at design stage users safety throughout product lifecycle, we define a safety indicator $I_S$ expressed by Eq. (7).

$$I_S = FR_{\text{ris}} \times IR_{\text{ris}}$$

(7)

where

- $FR_{\text{ris}}$ is the factor of risk that indicates the existence of a risk or not,
- $IR_{\text{ris}}$ is the index of risk, which concern its qualification and quantification. For better safety, $I_S$ must be as smaller as possible as we will show in the following section.

5.2.1. Factor of risk ($FR_{\text{ris}}$)

This indicator gives a potential risk, which appears in case of incident or accident. There is user injuries once this indicator has a value different from zero.

<table>
<thead>
<tr>
<th>Table 3 Safety criteria classification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intrinsic criteria</strong></td>
</tr>
<tr>
<td>Dangerous phenomena: any phenomena capable of causing an injury or damage to the health of the user during his or her work in the working situation [25].</td>
</tr>
<tr>
<td>Dangerous zone: any zone inside and/or around a system in which a person is exposed to a risk of hurt or health damage [25].</td>
</tr>
<tr>
<td>Technical solution: technical solutions to realize the technical functions during the design (CAD model)</td>
</tr>
<tr>
<td>Dimensioning: the layout way of technical solution and functional dimensions</td>
</tr>
<tr>
<td>Intervention frequency: number of times intervention in a given period</td>
</tr>
<tr>
<td>Intervention duration: Time necessary to complete human intervention [5]</td>
</tr>
<tr>
<td>Accessibility: We adopt the following definition [23]. Accessibility is the degree of ease with which it is possible to reach a certain location from other locations. This criterion measures the ability of a component to be accessible by user to be used or changed.</td>
</tr>
<tr>
<td>Visibility: measures the ability of a component to be visible by user during while working for production or for the system maintenance.</td>
</tr>
</tbody>
</table>

| Contextual criteria | **Consomable**: consumable materials that is used in the working situation (fuel, chemical product...). |
| Intervention mode: represents the system–user interactions (repairing, sitting-up...) [5]. |
| See Table 2 for the following criteria definitions: |
| **Tooling** |
| **Environment** |
| **Detectability** |
| **Competencies** |
From the definition of the risk given in the safety standards, we propose the following criteria:

\[ FR_{is} = \frac{Ph}{Zo} \times HIn \]  

(8)

where

- \( Ph \) is the dangerous phenomenon or the hazard resulting from technical solution identification. This criterion takes values 1 if there is a dangerous phenomenon (hazard) or value 0 otherwise. In CAD, choosing a solution allows to define the value of this criterion. In Table 4 we present some examples of technical solutions associated to dangerous phenomena and their parameters. More examples are defined in [22]. Thus, for each technical solution already stored designer find the associate dangerous phenomena and get the value of \( Ph \).
- \( Zo \) is a dangerous zone generated by a dangerous phenomenon. For example, in Table 4 the first technical solution has a dangerous phenomenon between the rollers. This dangerous zone is the zone (between rollers) in which user could be subject of a damage resulting from the \( Ph \). Using CAD geometric data the designer should size the zone in which there is a hazard. This criterion takes the value 0 if the zone is not penetrable by man or by one of his organs and 1 otherwise. For example, the zone between two tiny gears in a watch is not penetrable so \( Zo = 0 \).
- \( HIn \) is human intervention in a dangerous zone. This criterion takes the value 1 if there is a human intervention in the identified zone and 0 if not. In fact, this parameter is not so easy to determine. For example, the designer chooses a technical solution with a dangerous phenomena so \( Ph = 1 \) with a very large dangerous zone (e.g. the zone between two rollers of a set-off pressing machine). But if all tasks in this dangerous zone are automated no human intervention is possible there and \( HIn = 0 \).

So, if \( FR_{is} = 0 \) there is no risk and designer could go on his work. If \( FR_{is} = 1 \), designer must try to modify one of the previous parameters (\( Ph, Zo \) or \( HIn \)) to cancel the value of this indicator. If it is not possible, designer must determine IRis to quantify the risk and verify if it is acceptable or not. This acceptability is defined by the standards in function of firm activity and trade aspects.

### 5.2.2. Index of risk (IRis)

The company with regard to its activities defines the acceptability of risk. Therefore, referring to standards data the indicator of risk IRis is defined as follow:

\[ IR_{is} = \frac{Gr}{Ex} \times Pr \times Av \]  

(9)

where

- \( Gr \) is the gravity of risk which depends on the nature (cut, burn, etc.), the level (a finger wound, or a hand cut, etc.) of dangerous phenomena and on the number of persons concerned by this dangerous phenomena (one or many operators). Each company could determinate a scale of values representative of risks in relation with its activities.
- \( Ex \) is the exposure duration and frequency. Every task needs sometimes to be executed one time or may be repeated. This indicator concerns the socio-technical realization of a tasks identified in the dangerous zone.
- \( Pr \) is the probability of dangerous event happening. It is obtained by studying the human reliability [4] during system utilization and the system intrinsic reliability. \( Pr \) is function of user’s competency and knowledge and of system reliability. For example, a component with bad reliability can blow up and generate a dangerous event.
- \( Av \) is the probability of avoiding a dangerous phenomenon and accident or incident. It depends on the how incidents or accidents may happen and on the possible degradations that may occur. It allows estimating the probability that the operator discovers the accident or the incident before it takes place. This requires studying and analyzing the modes of possible technical and human failures.

Once \( I_{S} \) is determined, the designer compares its value with reference values given by his company, or compares alternative solutions.

The integration of these criteria in a CAD system requires the availability of semantic data relating to safety and a list of the dangerous phenomena generated by technical solutions.
drawn in CAD must exist in a database (Table 4). For a new solution, it is necessary to identify the new dangerous phenomenon $P_h$ and save it in the database. Then, designer must determine and define dangerous zone from the geometrical data and verify if user goes in this zone [6]. If the answer is yes, the indicator $FRis$ is equal to one and it is necessary to determine $IRis$ to estimate the risk. A not acceptable value of $IRis$ requires a modification that could concern one or more criteria of (Gr, Ex, Pr and Av) as for example, decreasing the exposure duration or frequency. These data have to enrich the available geometrical and topological the CAD data model. Some data are extracted from standards or they result from designer’s experience.

6. Academic application

We illustrate our approach on an academic example representing a movement transfer system shown in Figs. 4 and 5. We will study this system to outline the interaction between maintainability and safety constraints. The CAD model was constructed using SolidWorks software.

6.1. Alternative design solutions

In this system, the shaft gear and the wheel gear are considered as the most critical components from the maintainability and safety points of view. We ensure that the reliability is better than 95% for these two components by using virtual samples tests method.

To satisfy the specified function which is “to transfer movement”, designer has several alternative technical solutions. From Table 4, for example, designer chooses the technical Solution Version 1 (SV1). This SV1 fulfills functional requirement for movement transfer and presents easiness of maintenance. However, two risks appear: the first one is the risk to wound operators with gears; the second one risk is the projection of the lubricant oil. The designer evaluates $I_S$ for each risk in order to determinate if the solution version 1 respects safety standards or not. If $I_S$ is not acceptable the solution is rejected and the other alternative solutions are evaluated. For each of them, they performances with respect to maintainability and safety are studied. So, we are looking for more safe and more easy to maintain solutions which mean smallest values for $I_S$ and $I_M$.

6.2. Maintainability and safety interaction

For maintainability assessment, we assume that a maximum criticality level is associated to these same components (shaft gear and the wheel gear) so $K_i = 1$ for both. For SV1 solution there is no housing and the maintainability indicator, $I_{M1}$ is the total time required to remove the wheel gear in case of this part is damaged and must be repaired or replaced. To remove the wheel gear, we must successively remove the parts 7, 6, 5, 4 and
finally part 2 as shown in Figure 5. From Table 1, the corresponding removal times are: 3 for part 7 (Nut); 1 for part 6 (Ring), 1 for part 5 (Load), 1 for part 4 (Wheel Rim Spacer) and 1 for part 2 (Wheel gear). Therefore, the total required time to remove the wheel gear is the sum of these removal times:

\[ I_{M1} = 3 + 1 + 1 + 1 + 1 = 7 \]

For safety assessment \((I_S)\) is estimated for (SV1): using Eq. (7). Therefore, \(FR_{I_S1}\) and \(IR_{I_S1}\) are determinate according to Eqs. (8) and (9) and \(Ph, Zo\) and \(HIn\) must be evaluated. We notice that there are two dangerous phenomena: first, between the two gears turning in opposite senses and second the projections of oil.

For the first dangerous phenomena which is the Hazard of destruction between gears, \(Ph = 1\).

\(Zo\) is computed from CAD geometrical data. In the example, the dangerous zone has a penetrable size by a human organ so \(Zo = 1\). As regards \(HIn\), a human intervention could be happened because the SV1 has any housing so \(HIn = 1\). This value could also be determinate by the designer, in analyzing that operator does some task in this zone.

\(FR_{I_S1} = 1\) means that a risk of destruction exists. It is necessary, in this case, to calculate the index of risk \(IR_{I_S1}\) using Eq. (9). Every company according to its own domain and experience determines a significant scale for each of \(Gr, Ex, Pr\) and \(Av\). Here to illustrate the example, we define a scale for each of \(Gr, Ex, Pr\) and \(Av\). Those scales have only a pedagogic meaning [6].

To evaluate the value of \(Gr\) we define a scale from 1 to 10. For the treated example, the gravity is estimated equal to 5 (destruction of one finger between both gears). We suppose that \(Ex\) has its values on a scale form 1 to 2 (for a short duration and weak frequency \(Ex = 1\) and for a long duration and strong frequency \(Ex = 2\)) So for a strong frequency and enough short duration \(Ex = 1.5\).

The parameter \(Pr\) (on a scale form 0.5 to 1.5) could have the value 1.5 because the dangerous zone is accessible by user and a dangerous event is very probable.

At last, we suppose \(Av = 1\) for a not foreseen event. It takes values on a scale from 0.5 (foreseen event) to 1 (sudden event). For SV1 the safety indicator becomes:

\[ I_{S1} = FR_{I_S1} \times IR_{I_S1} = 1 \times 5 \times 1.5 \times 1.5 \times 1 \times 5 = 11.25 \]

If we suppose that company takes \(I_S = 8\) as an acceptable limit of safety indicator the user’s safety is not ensured because \(I_{S1} > I_S\). So, this solution does not satisfy safety standards. Therefore, designer must propose other solution for which index of risk value is smaller. As possible solution, he could add a barrier between the two gears that can prevent the introduction of user fingers in this dangerous zone (Fig. 6).

We named this modified solution (MSV1). Analyzing MSV1, we get \(Ph = 1\) (the same dangerous phenomena remains), \(Zo = 0\) (there is no more penetrable dangerous zone), and \(HIn = 1\) (operator could always interfere in around). Then \(FR_{I_S1} = 0\) and it is not necessary to calculate \(IR_{I_S1}\). In consequence, safety indicator \(I_S = 0\) and the solution is safer.

Now the second dangerous phenomena must be studied. The movement of this system could cause projection of lubrication liquid added after a maintenance operation. This projection is dangerous and could make dirty ground around of the system which implies gliding phenomena. Also depending of the surface state of the gears, some new risks could appear.

Therefore, the MSV1 does not cancel all risks but it deletes one and moves the others to another place. So, neither SV1 nor MSV1 are not acceptable solutions from the safety point of view but we have a good \(I_M\) because of the same time necessary to disassemble/assemble the two gears. The barrier has no influence on this time.

In solution version 2 (SV2), a housing component is used to cancel these previous risks. For the first risk between the two gears, the same dangerous phenomenon still exist and \(Ph = 1\), with the housing the dangerous zone is no longer accessible \((Zo = 0)\), due to the necessity of human presence \((HIn = 1)\). So \(FR_{I_S2} = 0\) and we do not need to evaluate \(IR_{I_S2}\) and \(I_{S2}\). In addition, this solution prevents liquid projections to make dirty ground around the system, it does not engender another dangerous phenomena, \(Ph\) for the second risk is equal to 0 and the user safety is assured. So \(I_{S2} = 0\).

From maintainability viewpoint, the MSV1 configuration requires a barrier as an additional part that needs to be fixed to the existing structure using any assembly technique. This will increase the disassembling time. In SV2 configuration, the housing improves the safety, \(I_S\) but it degrades the maintainability, \(I_M\), by adding components and fasteners. In Fig. 7 we illustrate the semantic matrix instantiation for SV2 where the housing is fixed to the support by four screws: the link type is screwing and the associated removal time scale is 3. And the ring is assembled to the shaft by a simple contact and the associated removal time scale is 1 (see Table 1).

For solution SV2, the total required time is the sum of the time for removing the wheel gear determined above added to
the time required to remove the housing. To remove the housing we must remove the four screws used to fix this component: time required for removing one screw × number of screws.

Therefore, the maintainability indicator for this solution is:

\[ I_{M_2} = 7 + (3 \times 4) = 19 \]

To make disassembly easier designer could use the SMED method to decrease the time required to disassemble the housing. So Clipped housing could be used as in SV3. For this third configuration we obtain:

\[ I_{M_3} = \text{necessary time to remove one clip} \]
\[ \times \text{number of clips} (= 4). \text{So } I_{M_3} = 7 + (2 \times 4) = 15. \]

In solution SV4, by replacing the two rear clips by a hinge we improve the product maintainability. Then we determine in the same way \( I_{M_4} = 7 + (2 \times 2) = 11. \) All indicator results are summed up Table 5 referring to the different solutions where the diagonal elements \( P_i \) of semantic matrix (Fig. 7) are assumed to be identical in the four different alternative solutions therefore they have no effect on the comparison.

The designer to decide for the acceptable solutions depending on the requirements uses these results. Then the final choice of a good solution could be the solution with defining a compromise between maintainability and safety aspects. With this evaluation designer may decide to use the version 4. As regards other indicators, this study allows to consider maintainability and safety aspects in the design process. This relative evaluation gives an idea about the product behavior at any step in the design progression.

In the other hand there are two levels of contradiction to solve. First one is between those two indicators. Improving one of them could degrade the other. The second one is between the solutions. If S1 have a very good \( I_{SA} \) and bad \( I_S \) and S2 have a very good \( I_M \) and bad \( I_S \) which solution to choose? The balanced indicators allow having a compromise that could be a bad solution. These points are the subject of our actual study. We can note that, the choice of best decision belongs to multi-criteria problems.

7. Conclusions and future works

In this paper, an approach is presented for maintainability and safety assessment in the design process using CAD model enriched with behavioral semantic data. The approach is aimed to assist designers for taking into account semantic behaviors that are traditionally evaluated after design process using physical tests and/or other Virtual Reality devices.

This approach proposed provides maintainability and safety intrinsic indicators to assist designers for design solutions validation with respect to an admissible performance threshold required by specifications. Also, these indicators could be used to compare alternative solutions. And during design process, the indicators are stored and analyzed to check the solution improvement.

The behavioral assessment procedure presented is a general method that may be used to determine performance indicators for different other lifecycle characteristics like recyclability, environmental impact analysis [29].

The academic application outlined has shown the feasibility of our approach. It also pointed out the interactions between maintainability and safety constraints that require the utilization of multi-criteria approaches to find out the better compromise.

In this study, we have not considered the influence of the context related to maintenance conditions (toolings, logistics, human operators’ competency and reliability). This type of information can be investigated using fuzzy set theory [28].

The product 3D model was built using SolidWorks software and the implementation of the behavioral assessment module is under development in Java language [1].

Appendix A. Reliability determination for components

Many databases are available for industrial components and equipments [7,21,27]. These data can support equipment availability analyses, reliability improvements, maintenance strategies, quantitative risk analyses, and life cycle cost determinations.
In these different cases the components reliabilities are assumed to be constant \((R_i = r_i)\). For reliability prediction at design stage the Weibull distribution is commonly used as expressed in Eq. (A.1) and Fig. A.1.

\[
R(t) = 1 - \lambda (t) = e^{(t - \gamma) / \eta} \\
(A.1)
\]

where:

- \(\lambda\) is the product failure rate;
- \(\beta\) is the shape factor;
- \(\eta\) is the scale factor;
- \(\gamma\) is a geometric parameter, \(\gamma > 0\).

Here, we use this reliability information to detect weak components that may cause failures or accident. However, for brand new components for which no sufficient statistical data are available, the reliability may be estimated using virtual samples tests techniques [10] or accelerated life testing technique where the difficulty is how to reproduce contextual conditions and constraints under which the product will be used [9].

References

Amadou Coulibaly is an associate professor at Graduate School of Science and Technology of Strasbourg (INSA) in the Engineering Design Laboratory. He is a mechanical engineer from Ecole Polytechnique de Thies (Senegal) and doctor in mechanical engineering from Louis Pasteur University of Strasbourg (France). His research interests concern product modeling, knowledge modeling for product behavioral performance prediction at design stage using CAD Systems. His current research industrial applications are focused on reliability and maintainability assessment.

Rémy Houssin is an associate professor at University of Strasbourg and at Graduate School of Science and Technology of Strasbourg (INSA) in the Engineering Design Laboratory. He is a mechanical engineer from mechanical and electrical faculty of the University of Techreen (Syrie) and doctor in industrial engineering from Henri Poincare University of Nancy (France). He works on performance evaluation form design stage, at product modeling and on reconfigurable manufacturing system design. Actu- ally he works on behavioral design approach and focused on availability based safety assessment.

Pr. Bernard Mutel is a Emerite Professor at Institut National des Sciences Appliquées de Strasbourg in the Engineering Design Laboratory. He is an engineer in cybernetics (ISIN), master of science in electrical engineering (IIT) and Doctor ès-Sciences in Automatics (U.M.). His main research interests lie in cellular manufacturing design and new production data analysis technics. Currently he is working on data mining for diagnosis.
Optimal disassembly sequencing strategy using constraint programming approach

Xavier Zwingmann
Centre for Interdisciplinary Research in Rehabilitation and Social Integration (CIRRIS), Québec City, Canada

Daoud Ait-Kadi
Département Génie Mécanique, Centre de recherche sur les Technologies de l’Organisation Réseau (CENTOR), Université Laval, Québec City, Canada, and

Amadou Coulibaly and Bernard Mutel
Laboratoire de Génie de la Conception, INSA de Strasbourg, Strasbourg, France

Abstract
Purpose – The purpose of this paper is to propose a framework to identify all the feasible disassembly sequences for a multi-component product and to find an optimal disassembly sequence, according to specific criteria such as cost, duration, profit, etc.

Design/methodology/approach – Taking into account topological and geometrical constraints of a product structure, an AND/OR disassembly graph is built. Each graph node represents a feasible subassembly. Two nodes \( i \) and \( j \) are connected by an arc \((i, j)\), called a transition, if the subassembly \( j \) can be obtained from the subassembly \( i \) by removing one or several connectors. Constraint programming approach is used to generate the feasible subassemblies and related transitions.

Findings – If a cost \( z_{ij} \) is incurred to perform a transition \((i, j)\), an optimal disassembly sequence can be generated for a given subassembly, using the shortest path algorithm or a linear programming model.

Research limitations/implications – The proposed approach performs very well compared to other approaches published in the literature, even when applied to products requiring parallel disassembly and including a large number of parts.

Practical implications – This approach has been successfully applied to assess the wheelchair maintainability at the design stage and will be implemented in CAD systems. One other application, regarding the disassembly process and total revenue maximization for product recycling, is now under consideration.

Originality/value – Applying constraint programming to efficiently generate the set of the feasible subassemblies constitutes the main contribution in this paper. This process is the hardest step in the disassembly sequencing problem.

Keywords Parts, Assembly, Mechanical components, Automatic programming

Paper type Research paper

The research work reported here was completed thanks to the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the International Council for Canadian Studies (ICCS) and the Technical Aids Program of the Régie de l’assurance maladie du Québec (RAMQ).
**Introduction**

A multi-component product is a system built by a set of components which are bound together by a set of liaisons. If the liaisons are made from detachable fasteners, the product can be split up into subassemblies, or single parts. The objective of the proposed approach is to generate an optimal disassembly sequence for each feasible subassembly.

In this paper, constraint programming approach is used for automatic feasible subassembly and transition generation. Topological and geometrical constraints are among the main constraints considered in the feasible subassembly generation process. Feasible sequences are represented on an AND/OR disassembly graph.

**Methodology for setting a disassembly strategy**

In the first four sections, we will attempt to show, in order, how a product can be modelled, how to generate feasible subassemblies, how to identify feasible transitions and how to build a disassembly graph or its equivalent transition matrix. We will then present the constraint programming models which are new alternatives for generating these elements. We also propose a function for sequence dependent cost in order to reach an optimal disassembly sequence by using Operational Research tools. Figure 1 presents the methodology development.

**Product modelling**

Product modelling is performed by using the connection graph and the precedence relationships first proposed by Bourjault (1984). A simple example (see Figure 1) is used to illustrate different concepts related to product modelling.

**Connection graph**

The connection graph is a representation of the liaisons that bind parts together (see Figure 1). Each node \( p \) of the connection graph materializes a single part and each arc \((p,q)\) represents the liaison between parts \( p \) and \( q \). An equivalent representation is the connection matrix \( M \) with element \( m_{pq} \) equals 1 if there exists a liaison between the parts \( p \) and \( q \), and 0 otherwise. Let \( G \) be a matrix with elements \( g_{pq} (g_{pq} \in \mathbb{R}^+) \) being the required effort or cost incurred to get part \( p \) from part \( q \).

**Precedence relationships**

The precedence relationships consist in Boolean expressions, representing those actions that must be performed prior to the execution of a predefined action. For the example given (see Figure 2(a)), in order to remove part \( A \), parts \( B \) or \( E \) should be...
removed first, leading to the Boolean expression: \( R_B \) or \( R_E \) (\( R_A \). Figure 2(c) gives the remaining precedence relations.

**Generation of feasible subassemblies**

Topological and geometric constraints, respectively coherence and detachability constraints, are used in the feasible subassembly generation process.

A subassembly is said to be coherent if its connection graph is a connected undirected graph (there exists at least one path between every pair of the graph nodes). From the connection graph in Figure 2(b), the subassemblies \( \{B,E\} \), \( \{B,D\} \) and \( \{B,D,E\} \) are not coherent. The short notation \( BE \), \( BD \) and \( BDE \) will be used throughout the paper.

A subassembly is detachable if it can be obtained from the original assembly via disassembly operations only, that is without reassembly and with respect to the precedence relations. For instance, the subassembly \( BCDE \) is not detachable because \( A \) cannot be removed from \( ABCDE \) (\( B \) or \( E \) are still in place therefore the left-hand side in the first precedence relation cannot be true, see Figure 2(c)).

**Generation of feasible transitions**

Given a subassembly \( S \), any transition is said to be feasible, if there exists a set of disassembly actions that leads to two feasible subassemblies \( S_1 \) and \( S_2 \) satisfying the following conditions:

\[
S_1 \cup S_2 = S \tag{1}
\]

\[
S_1 \cap S_2 = \emptyset \tag{2}
\]

\( S_1 \) and \( S_2 \) must be coherent and detachable. \( \tag{3} \)

**Sequence representation on a graph**

A disassembly sequence can be represented by an ordered list of feasible transitions. Since many assembly sequences share common subsequences, attempts have been made to create compact representations, such as the disassembly graph or AND/OR graph (Homem de Mello and Sanderson, 1990). The latter is a directed network where the nodes are the feasible subassemblies and each hyperarc (or V-arc) represents the division of a parent subassembly into two child subassemblies. Hence, disassembly

---

Figure 2.
(a) Basic assembly; (b) connection graph with costs; (c) precedence relations
sequences of the initial product (the source) are represented by the paths that lead to the targeted parts or subassemblies (the leafs).

For a $N$-component system, $2^N - 1$ combinations must be considered to find the feasible subassemblies (Wolter, 1991). In the system depicted in Figure 2(a), 31 combinations must be analysed and only 28 of them are coherent. Because of the geometric constraint (detachability), there remain 18 feasible subassemblies. Note that $ADE$ and $DE$ are feasible only because of the accessibility slot for screw $Z$. The AND/OR graph presented in Figure 3 shows the 18 feasible subassemblies (the nodes) and the 29 hyperarcs representing the feasible transitions for the design with the accessibility slot. From the node $CDE$, the two leaving hyperarcs show two feasible transitions: $1 - C$ AND $DE$; $2 - D$ AND $CE$. The AND/OR graph can also be represented by a transition matrix $W$, where the rows and the columns are associated with the $N_f$ feasible subassemblies and the $N_j$ feasible transitions, respectively. The element $w_{ij}$ is defined as follows:

$$w_{ij} = \begin{cases} -1 & \text{if the transition } j \text{ dismantles the parent subassembly } i \\ +1 & \text{if the transition } j \text{ creates the child subassembly } i \\ 0 & \text{otherwise} \end{cases}$$

Using the matrix $W$, a feasible disassembly sequence can be recognised as an alternated series of columns (transitions) beginning with $(-1)$ and ending with $(+1)$. In one column, there is $(-1)$ in the row $i$ of the parent subassembly $S_i$ (the absence of it means that the subassembly $S_i$ cannot be dismantled) and $(+1)$ in the row of an intermediate child subassembly. The sequence continues with another column which dismantles the latter child until reaching the target.

Figure 4 gives the transition matrix for the illustrative example (Figure 2(a)). Shaded subassemblies and transitions are only feasible with the accessibility slot for the screw $Z$ and the lines represent two different sequence examples to reach part $A$.

If a performance measure (cost, time, effort, entropy, etc.) is associated to each transition (or each column of $W$), optimal disassembly sequences can be generated. Later in this paper, we propose a simplified cost function $z$. 

---

**Figure 3.** AND/OR graph for the system presented in Figure 2(a), with the accessibility slot.
Applied constraint programming for subassemblies and transitions generation

To find all the feasible subassemblies requires a previous decision of which parts to be removed and subsequent ones when to do so. Our approach is based on binary decision variables, associated to parts and defining a subassembly, which have to satisfy the topological and geometrical constraints. The connection matrix $M$ and the precedence relations allow the representation of each constraint as a function of decision variables. Note that, in order to handle feasible subassemblies that can be obtained only by removing several parts simultaneously (parallel disassembly), the precedence relations are enhanced by the clearance relations, introduced in this article to take into account the direction of simultaneous removals.

Numerous studies on disassembly sequencing use heuristics or exact methods such as mathematical programming, graph theory, Petri nets (Lambert, 2003). With exact methods, the feasible subassemblies are often obtained by questioning the user about disassembly actions, which is extensive and time-consuming (Baldwin et al., 1991). In this paper, constraint programming approach is used to efficiently solve the combinatorial problem which aims, first, at finding the feasible subassemblies.

With constraint programming, the approach adopted here, we search for a set of binary variables which satisfies constraints, and avoid hence the run of logic programming steps. The constraint solver modifies the variables, combines and simplifies the constraints, these tasks reducing the search domain, until a solution is found. For this work, OPL Studio and ILOG Solver have been used as constraint programming systems, which allow us to take advantage of existing algorithms (Lustig and Puget, 2001).

Constraint programming approach is also used to generate feasible transitions. In order then to select an optimum disassembly sequence according to maintainability considerations, either the graph theory or a linear programming model are implemented.
Feasible subassemblies identification using constraint programming

$M_S$ is the maximum number of steps required to achieve a disassembly at its full extent; and $P = \{A, B, \ldots\}$ is the set of parts in the product.

$$m_{pq} = \begin{cases} 1 & \text{if there is a connection between the parts } p \text{ and } q \\ 0 & \text{otherwise.} \end{cases}$$

$$b_{i,p} = \begin{cases} 1 & \text{if the part } p \text{ is still present at step } i, \\ 0 & \text{otherwise.} \end{cases}$$

$$d_i = \begin{cases} -\bar{x} & \text{if the parts, removed at step } i, \text{ are extracted to the left}, \\ +\bar{x} & \text{to the right horizontal direction (see Figure 2(a))}, \\ -\bar{y}, +\bar{y}, \text{ etc. etc.} & \end{cases}$$

A subassembly is feasible if there is at least one set of disassembly steps and if all the intermediate subassemblies, generated at steps 1 to $M_S$, satisfy coherence and detachability constraints. For each step, the program determines the parts to be removed by setting the value of the decision variable $b$. If a solution exists, the feasible subassembly state is defined by the final set {$b_{M,p} = 1 \forall p \in P$}.

At step 0, all the parts are present:

$$b_{0,p} = 1, \forall p \in P$$

As soon as a part $p$ is removed, it cannot be reassembled. The constraint (5) prevents a variable which has been set to 0 (at step $i-1$), to be reset to 1 in all subsequent steps ($i, i+1, \ldots$):

$$b_{i,p} \leq b_{i-1,p}, \forall p \in P, \forall i \in [1..M_S].$$

Moreover, each of the subassemblies generated at each step has to be coherent. The corresponding constraint requires the second smallest eigenvalue, of the Laplacian matrix of the connection graph, to be greater than 0 (Weisstein, 2004).

Finally, to remove a part, the detachability constraint should be satisfied. The precedence relations are designed to assess whether it is possible to sequentially detach parts by considering the intermediate subassembly at the previous step. However, they are not suitable to study simultaneous disassembly actions in parallel disassembly, because notations would become recursive. Therefore, we propose the enhanced disassembly relationships, by introducing the clearance relations in addition to the precedence relations. These new relations take into account how to remove a part by considering the current state and by using a new decision variable, the Disconnection Clearance Vector (DCV) $d$, which indicates the direction in which there is a clearance for the first extraction move. As $d$ can only have one value, the program is able to handle the simultaneous removal of several parts, in the same direction. The enhanced disassembly relationships (see Table I) are defined as the combination of the precedence and clearance relations.
According to the notations used in the program and the Boolean formalism, the enhanced disassembly relationship for part \( A \) (the first one in Table I) becomes:

\[
\overline{b}_{i,A} \Rightarrow \left[ \overline{b}_{i-1,B} + \overline{b}_{i-1,E} \right]_{\text{precedence relation}} + \left[ \overline{b}_{i,B} \cdot (d_i = -x) \right] + \left[ \overline{b}_{i,E} \cdot (d_i = +x) \right]_{\text{clearance relation}},
\]

\( \forall i \in [1..M_S] \).

**Feasible transition identification using constraint programming**

\( \Omega \) is the set of feasible subassemblies.

\[
\Delta_{S,p} = \begin{cases} 
1 & \text{if the part } p \text{ belongs to the subassembly } S, \\
0 & \text{otherwise (for each subassembly } S \text{ of } \Omega). 
\end{cases}
\]

\( g_{pq} \) is the effort required to break the connection between the parts \( p \) and \( q \).

\[
t_s = \begin{cases} 
-1 & \text{if the (parent) subassembly } S \text{ is dismantled}, \\
+1 & \text{if the (child) subassembly } S \text{ is created}, \\
0 & \text{otherwise.}
\end{cases}
\]

\( S_1, S_2 \) is 2 child subassemblies of a transition found as feasible; \( S \) is a parent subassembly, such as \( S = S_1 \cup S_2 \); and \( z_{S,S_1} = z_{S,S_2} \) is the cost of the transition which dismantles the parent subassembly \( S \).

Once all the feasible subassemblies (\( \Omega \)) are found, the feasible transitions can be identified and their associated costs calculated. A feasible transition is defined by a parent subassembly and two child subassemblies, satisfying the constraints (1), (2) and (3). The constraint (3) is already satisfied because the subassemblies are taken in \( \Omega \).

For each feasible transition, the status of each subassembly is expressed by the set of the decision variables \( t_s \), defining a column of the transition matrix \( W \). The constraints (1) and (2) are expressed by (7) and (8):

\[
\sum_{S \in \Omega} t_s \Delta_{S,p} = 0, \forall p \in P
\]

\[
\sum_{S \in \Omega} |t_s| = 3 \text{ and } \sum_{S \in \Omega} t_s = 1.
\]

<table>
<thead>
<tr>
<th>Table I. Enhanced disassembly relationships for Figure 2(a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedence relations (at step ( i-1 ))</td>
</tr>
<tr>
<td>------------------------------------------</td>
</tr>
<tr>
<td>([R_B \text{ or } R_E]_{i-1})</td>
</tr>
<tr>
<td>([R_B \text{ or } (R_D \text{ and } R_B)]_{i-1})</td>
</tr>
<tr>
<td>([ (R_A \text{ and } R_C) \text{ or } R_D]_{i-1})</td>
</tr>
</tbody>
</table>
**Function for sequence dependent cost**

In order to prepare the disassembly sequence optimisation, it is necessary to know the costs of the feasible transitions, to be attributed to the AND/OR hyperarcs. For instance, one can choose time (Boks et al., 1996), entropy (Hsu and Lin, 2002) or multi-factor measure such as disassembly effort index (Das et al., 2000), maintainability index (Wani and Gandhi, 1999; Zwingmann, 2005), etc.

In this paper, the transition cost is given by a simple function \( z \in \mathbb{R}^+ \) based on the costs required to break all the connections (see Figure 2(b)) between the parts of two child subassemblies, \( S_1 \) and \( S_2 \). Table II presents the costs of the feasible transitions in the example of Figure 2(a) (the italic ones cannot be completed without the accessibility slot for the screw \( Z \)), according to the cost function (9) and the following scale: \( g_{pq} = 1 \) for a contact between parts \( p \) and \( q \), 2 for a screwed connection and 10 for a welded connection:

\[
z_{S_1,S_2} = \sum_{p \in S_1} \sum_{q \in S_2} g_{pq}
\]  

(9)

|    | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 1  |    | 12 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2  |    |    | 2  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 3  |    |    |    | 4  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 4  |    |    |    |    | 13 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 5  |    |    |    |    |    | 3  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 6  |    |    |    |    |    |    | 4  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 7  |    |    |    |    |    |    |    | 2  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 8  |    |    |    |    |    |    |    |    | 3  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 9  |    |    |    |    |    |    |    |    |    | 11 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 10 |    |    |    |    |    |    |    |    |    |    | 3  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 11 |    |    |    |    |    |    |    |    |    |    |    | 11 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 12 |    |    |    |    |    |    |    |    |    |    |    |    | 5  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 13 |    |    |    |    |    |    |    |    |    |    |    |    |    | 11 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 14 |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 11 |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 15 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 3  |    |    |    |    |    |    |    |    |    |    |    |    |
| 16 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 5  |    |    |    |    |    |    |    |    |    |    |    |
| 17 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 1  |    |    |    |    |    |    |    |    |    |    |
| 18 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 1  |    |    |    |    |    |    |    |    |    |
| 19 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 1  |    |    |    |    |    |    |    |    |
| 20 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 10 |    |    |    |    |    |    |    |
| 21 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 2  |    |    |    |    |    |    |
| 22 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 6  |    |    |    |    |    |
| 23 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 6  |    |    |    |    |
| 24 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 2  |    |    |    |
| 25 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 4  |    |    |
| 26 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 6  |    |
| 27 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 3  |
| 28 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 5  |
| 29 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**Table II.**

Costs of the 29 feasible transitions of the Figure 2 example
**Generation of an optimal disassembly sequence**

Once the transition matrix or the AND/OR graph are generated and cost attributed, the shortest path algorithm, namely the Dijkstra algorithm, can be applied in order to find one sequence to reach the part or subassembly under consideration with the minimum cumulative effort.

The optimum sequences can also be obtained via linear programming (LP). We adapted an end-of-life disassembly model from Lambert (2002) for the maintainability paradigm. Originally, the feasible subassemblies i represent potential revenues ri and the transitions j (with cost zj) are chosen by decision variables xj. In our case, a very high revenue is attributed to the part to be maintained and a null value for all the other subassemblies, so that minimizing the disassembly costs (to reach the part considered) means maximizing benefits:

Maximize the net revenue \(\sum_i \sum_j \left( w_{ij} r_i - z_j \right) x_j \) \hspace{1cm} (10)

Subject to \(\sum_j w_{ij} x_j \geq 0 \) and \(x_0 = 1\). \hspace{1cm} (11)

Table III gives the resulting sequences and costs for maintaining each part of the product (see Figure 2(a)). Based on this quantitative analysis, the maintainability of parts A and C is improved by the accessibility slot for screw Z.

Besides, our approach allows us to solve more complex problems, requiring parallel disassembly, such as the polyhedral case (Chen et al., 1997). Recently, this kind of problem has been addressed with a new technique (Lambert, 2006), however, to the best of our knowledge, the entire AND/OR graph for the polyhedral case is not reported in the literature. Based on our approach, 145 feasible subassemblies are found in about ten seconds (see Table IV), as well as 614 transitions (Zwingmann, 2005). For this case, all the calculations took about ten seconds on an Intel Pentium 4 CPU 2 Ghz with 512 Mb of RAM. For a larger case like a wheelchair designed with 29 parts, it can take several hours to found the 385,860 feasible subassemblies and the 8,276,023 feasible transitions. To measure the product complexity, note that the connection graph density is very low (72 arcs versus 392 combinations) and the 29 enhanced disassembly relationships involve a mean of 3.5 parts (from 1 to 13).

Constraint programming reduces efficiently the search space: the feasible subassemblies represent less than 0.072 per cent of the theoretical combinations.

<table>
<thead>
<tr>
<th>Part to maintain</th>
<th>Sequences of transitions (see Figure 4) in the design with the accessibility slot (matrix W&lt;sub&gt;18 *30&lt;/sub&gt;)</th>
<th>Cost</th>
<th>Sequences of transitions (see Figure 4) in the design without the accessibility slot (matrix W&lt;sub&gt;16 *22&lt;/sub&gt;)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0-24-26</td>
<td>8</td>
<td>0-7-15-16</td>
<td>10</td>
</tr>
<tr>
<td>B</td>
<td>0-13</td>
<td>11</td>
<td>0-13</td>
<td>11</td>
</tr>
<tr>
<td>C</td>
<td>0-28-13</td>
<td>16</td>
<td>0-3-4-21</td>
<td>19</td>
</tr>
<tr>
<td>D</td>
<td>0-16</td>
<td>5</td>
<td>0-16</td>
<td>5</td>
</tr>
<tr>
<td>E</td>
<td>0-15-16</td>
<td>8</td>
<td>0-15-16</td>
<td>8</td>
</tr>
</tbody>
</table>

Table III. LP results for the example of Figure 2(a)
Table IV.
The 145 feasible subassemblies of the polyhedral case

<table>
<thead>
<tr>
<th>Parts</th>
<th>Subassemblies</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>ABCDEFGHIJKLMNOP</td>
</tr>
<tr>
<td>12</td>
<td>ADEFGHIJKLMNOP</td>
</tr>
<tr>
<td>11</td>
<td>BCDGHIJKLMNOP</td>
</tr>
<tr>
<td></td>
<td>AEFGHIJKLMNOP</td>
</tr>
<tr>
<td>10</td>
<td>CDGHIJKLMNOP</td>
</tr>
<tr>
<td></td>
<td>BCGHIJKLMNOP</td>
</tr>
<tr>
<td></td>
<td>AFGHIJKLMNOP</td>
</tr>
<tr>
<td></td>
<td>ADEFGHIJKLMNOP</td>
</tr>
<tr>
<td>9</td>
<td>DGHJIKLNM</td>
</tr>
<tr>
<td></td>
<td>BGHIJKLNM</td>
</tr>
<tr>
<td></td>
<td>AEFGJIKLNM</td>
</tr>
<tr>
<td>8</td>
<td>GHIJKLNM</td>
</tr>
<tr>
<td></td>
<td>CDGJKLNM</td>
</tr>
<tr>
<td></td>
<td>BCDHIJMN</td>
</tr>
<tr>
<td></td>
<td>AFGJKLNM</td>
</tr>
<tr>
<td></td>
<td>AEFGJKLNM</td>
</tr>
<tr>
<td>7</td>
<td>DGJKLNM</td>
</tr>
<tr>
<td></td>
<td>CDHIJMN</td>
</tr>
<tr>
<td></td>
<td>BCHIJMN</td>
</tr>
<tr>
<td></td>
<td>BCDJMN</td>
</tr>
<tr>
<td></td>
<td>BCDHJM</td>
</tr>
<tr>
<td></td>
<td>BCDHIJN</td>
</tr>
<tr>
<td></td>
<td>BCDHIJM</td>
</tr>
<tr>
<td></td>
<td>AFGJKLNM</td>
</tr>
<tr>
<td></td>
<td>AEFGJKLNM</td>
</tr>
<tr>
<td>6</td>
<td>GJKLNM</td>
</tr>
<tr>
<td></td>
<td>DHIJMN</td>
</tr>
<tr>
<td></td>
<td>CDIJMN</td>
</tr>
<tr>
<td></td>
<td>CDHIJN</td>
</tr>
<tr>
<td></td>
<td>CDHIJM</td>
</tr>
<tr>
<td></td>
<td>BHIJMN</td>
</tr>
<tr>
<td></td>
<td>BCIJMN</td>
</tr>
<tr>
<td></td>
<td>BCHIJMN</td>
</tr>
<tr>
<td></td>
<td>BCHIJN</td>
</tr>
<tr>
<td></td>
<td>BCHIJMN</td>
</tr>
<tr>
<td></td>
<td>BCDJMN</td>
</tr>
<tr>
<td></td>
<td>BCDIJN</td>
</tr>
<tr>
<td></td>
<td>BCDHJM</td>
</tr>
<tr>
<td></td>
<td>BCDHIN</td>
</tr>
<tr>
<td></td>
<td>BCDHJM</td>
</tr>
<tr>
<td></td>
<td>BCDHIJ</td>
</tr>
<tr>
<td></td>
<td>AFGJKL</td>
</tr>
<tr>
<td></td>
<td>AEFGKLN</td>
</tr>
<tr>
<td>5</td>
<td>HIJMN</td>
</tr>
<tr>
<td></td>
<td>GKLNM</td>
</tr>
<tr>
<td></td>
<td>DJIJMN</td>
</tr>
<tr>
<td></td>
<td>DHIJN</td>
</tr>
<tr>
<td></td>
<td>DHIJM</td>
</tr>
<tr>
<td></td>
<td>CDIJMN</td>
</tr>
<tr>
<td></td>
<td>CDIJN</td>
</tr>
<tr>
<td></td>
<td>CDHIJ</td>
</tr>
<tr>
<td></td>
<td>BIJMN</td>
</tr>
</tbody>
</table>

(continued)
<table>
<thead>
<tr>
<th>JQME 14,1</th>
<th>BHIMN</th>
<th>BHJN</th>
<th>BHIJN</th>
<th>BCJMN</th>
<th>BCIJN</th>
<th>BCHMN</th>
<th>BCHIN</th>
<th>BCHIM</th>
<th>BCHIJ</th>
<th>BCDJN</th>
<th>BCDIJ</th>
<th>BCDHN</th>
<th>BCDHM</th>
<th>BCDHI</th>
<th>AFGKL</th>
<th>AEFGK</th>
</tr>
</thead>
</table>

4 parts (26)

| IJMN | HIMN | HIJN | HIJM | GLMN | GLKM | DJMN | DIJN | DHIJ | CDJN | CDIJ | BJMN | BIJN | BHMN | BHN | BHM | BHI | BCJN | BCIJ | BCHN | BCHM | BCHI | BCDJ | BCDI | BCDH | AFGK |

3 parts (21)

| JMN | JN | HMN | HIN | HIM | HIJ | GLM | GLM | GKL | DJN |

Table IV. (continued)
However, the memory limit slows down the calculations if the product has more than 20 components.

**Conclusion**

In this paper, an efficient and automatic approach for disassembly sequencing generation is proposed. A constraint programming approach is used to determine the set of subassemblies and transitions, including procedures to validate their feasibility. The concepts related to the proposed approach are illustrated through a simple example. The shortest path algorithm or a LP program are implemented to get an optimal disassembly sequence in the context where a cost is incurred for each disassembly action. Other performance criteria such as revenue to be maximized may be used in the disassembly sequence generation.

The proposed approach is compared to many available methods already published in the literature. It performs very well, mainly with respect to feasible disassembly sequence generation process. The product modelling method is quite simple. The approach has been successfully applied to improve both the wheelchair design and their maintenance activities. Other applications are still under consideration. The authors seek to implement the proposed approach in CAD software for maintainability assessment purposes.
References


Corresponding author
Xavier Zwingmann can be contacted at: xavier.zwingmann@centor.ulaval.ca

To purchase reprints of this article please e-mail: reprints@emeraldinsight.com
Or visit our web site for further details: www.emeraldinsight.com/reprints
Product modeling framework for behavioral performance evaluation at design stage

A. Coulibaly, B. Mutel, D. Ait-Kadi

Abstract

In this paper we present a framework for product behavioral performances evaluation during design process. It is based on the product behavioral modeling approach and uses an evaluation method that determines performance indicators for different domains of the product lifecycle. We are mainly focused on semantic and fuzzy domains that are complex to formalize. We show how the product design solutions represented in a CAD system can be analyzed to evaluate the product performances at some specific domains.

After a short state of art on product modeling methods we define a FBS product behavioral modeling approach based on the object-oriented modeling formalism. Here the product is described as a conceptual object with properties and formal criteria evaluation methods for different domains. Then a behavioral performance evaluation procedure is detailed and we outline software architecture for implementation in a CAD system. An application for product maintainability assessment of an industrial trailer illustrates the approach.

Keywords: Product modeling; Behavioral Performance Assessment; Maintainability; CAD systems

1. Introduction

Product Life Cycle Design is a global design approach that aims to control the different factors which affect the product functional and behavioral performance and competitiveness [5, 21, 31]. This evolution takes care of the impact of the product behaviors with respect to domains like: sustainable development, safety, maintainability, reliability or recyclability [24], etc. So there is the need for the product performances evaluation tools at design stage. This evaluation requires additional data including semantic aspects to characterize the product lifecycle properties.

In traditional CAD systems the design environment provides not only geometric data but also knowledge that may be used for further analysis in design solution. The behavioral domains boundaries are often fuzzy and related knowledge are correlated to contextual conditions of use. So there is a crucial need for suitable product models and efficient evaluation tools that allow behavioral performances assessment.

In this work, we propose a general framework to model a mechanical product and to evaluate such aspects. The challenge is to define a product model including not only functional and structural aspects but also behavioral criteria for all over the product lifecycle [30].

In Section 2, a brief overview of different product modeling methods is presented. In Section 3, we propose a FBS product modeling extension adapted to behavioral analysis based on the object-oriented modeling formalism [7]. The behavioral performance indicators evaluation procedure is detailed in Section 4. In Section 5, we outline software architecture for implementing the method in a CAD system. Section 6 describes an application for maintainability assessment and presents results in the case of an industrial trailer. Conclusion in Section 7 shows the feasibility of the approach and future developments.

2. Product modeling approaches

Different product modeling techniques are proposed in the literature as quoted in [3, 4, 11, 12]. In the following section
we sum up these main approaches, their possibilities and limitations.

2.1. Geometric and topological modeling

Modern CAD systems provide high quality 3D geometric models with a few properties. For validation analysis some parameters and variables can be derived from geometric and topological data available in the CAD systems. Figs. 1 and 2 show respectively the main data managed in a CAD: part model and additional information related to part assemblies. But, for a single part model no semantic data about the component reliability or its environmental impact for instance are available. For an assembly model, no more information is given about the links types (soldering, screwing, welding, etc.) between components, no information about the product maintainability, security, reliability, etc. are available and no information about the product operating conditions (temperature, dusty atmosphere, etc.) are considered. Then traditional simulation modules like finite element method cannot perform the suitable analysis to assess these missing characteristics that obviously have impact on the product performance.

So the traditional CAD models are insufficient and must be upgraded with additional information for behavioral performances evaluation.

2.2. Models using graphs and matrices

For assembly and disassembly plans elaboration, the earlier works for product modeling used graphs.

In these graphs a product is represented by allowing vertices to stand for the components and edges to represent assembly relationships between two components. The interaction from one element to another is captured by an arrow instead of a simple link. The resulting graph is called a digraph that mainly represents the product structure. Such graphical representation is usually hard to read and to analyze in the case huge assemblies. The Design Structure Matrix (DSM) [20], has been introduced as a generic matrix-based modeling method for information flow analysis in design process. Various conventions are used to define the content of the DSM cells. The most common uses of DSM are:

- Component-based representation for system architecting, engineering and design.
- Team-based representation for organizational design, interface management, team integration.
- Activity-based representation for project scheduling, activities sequencing, cycle time reduction.
- Parameter-based representation for low level activity sequencing and process construction.

For behavioral analysis, only the component-based representation is considered. In practice the DSM component-based representation, Fig. 3, is obtained from the product CAD model.
by using data including components features and assembly relationships between components. The “X” marks just indicate relationships between pairs of the product components without any more precision about assembly types.

The product structure is well represented at any desired level of decomposition as well as its different components: single part or sub-assembly. Another major advantage of this matrix representation over digraph is its compactness and ability to provide a systematic mapping among product components. It is clear and easy to read regardless of size of the part list. But both product modeling using digraph or DSM matrices are not intended to capture the product behavioral characteristics.

The next section presents the generic FBS approach and one of its variants, FBS-PPR, developed for product modeling including its associated process and resources.

2.3. FBS generic model

The Function-Behavior-Structure (FBS) modeling approach outlined in [13, 14, 41] is a global framework that describes the product through its functions, structure and behavior, as shown in Fig. 4. For enterprise process modeling this initial model has extended in [22] by the FBS-PPR model (FBS Product-Process-Resources).

The FBS model and its extensions are aimed to integrate all product related data and knowledge over the lifecycle. The FBS approach proceeds through three steps to build the different models.

2.3.1. Functional specifications

This first step deals with the translation of the requirements into functions that the product has to realize and to define the different constraints to be fulfilled. The functional specifications are made using a traditional Functional Analysis diagram. This diagram helps for identifying the product main functions, the constraints to fulfill and the relationships between the product and external elements [26]. These functional specifications are used as the basis for design requirements validation by designers and customers. In a CAD system a product functional performance is currently evaluated through simulations using cinematic and dynamic analysis software. Different other virtual reality systems are also used to test the mechanisms functionalities [17, 27, 34, 35].

2.3.2. Structural model

The second step defines the different components of the product and specifies their geometry, dimensions, topology and other physical properties. The structural model is derived from the functional specifications by matching functions with parts or sub-assemblies that realize each specific function. Note that several functions may be realized by a same set of parts or by a same sub-assembly. In the example of Fig. 5, the component \( C_2 \) is used by the two functions \( F_1 \) and \( F_2 \). Thus each function or group of functions is associated to structural submodel represented with a sub-matrix that consists of components and links involved in the functions.

2.3.3. Behavior model

Finally, the product behaviors are described from the manufacturing to its destruction or recycling. In the standard FBS model the behavior is mainly related to manufacturing processes while other aspects like security, reliability, maintenance and environment for example, are not covered.

In the following section, we propose a modeling approach based on the FBS concept to support behavioral assessment tools in specific lifecycle stages. Our approach intends for evaluation at preliminary design step where the product is partially defined. So in this approach we are not looking for optimal solution to the designer but to inform him if its last proposed solution is better than the previous one referring to the domain criteria.

3. FBS for Behavioral Performance Assessment

3.1. Behavior oriented modeling framework

We decompose the product specifications into classes or domains according to the kinds of requirement which often depend on viewed lifecycle stage. Based on the concept of noise in the design and criteria evaluation defined by [38, 40], the product technical performance and business success depend on two types of factors: the product intrinsic and external factors. The intrinsic factors depend on the product structure: geometric and dimensional tolerances, manufacturing process errors or revisions, material characteristics dispersion, etc. The external factors concern sources of variability that come from outside the product like: temperature and humidity in which the product is used, external loads, dust in the environment, human error, maintenance procedure and planning advises, etc.

From this point of view a product has different behaviors over its lifecycle domains that may be evaluated for performance...
analysis. The challenge is to define for each domain suitable criteria to evaluate the product specific behavior. Such evaluation is usually performed by considering the mechanical resistance, kinematics or dynamics behavior. But some other behavioral performances are much more difficult to evaluate at design stage. For example existing methods for reliability evaluation are statistics-based and require database and knowledge-base of historical data about similar products behavior. For this reason such methods are not directly usable in the case of new product design since no statistical data is available in this context. As shown in Fig. 6, our model is derived from the generic FBS model. It aims to capture data and knowledge about the product behavior concerning a large scope of domains. For behavioral assessment, we need to enrich the product with data and other non-graphical data such as physical properties or assembly links types. The following data may be added to structural model.

### 3.1.1. Assembly types

The product components can be assembled in different ways: welding, soldering, riveting, bolting, screwing, sticking, crimping, etc. Some of this assembly types are removable while other are permanent.

### 3.1.2. Semantic characteristics and material properties of components

Beside components physical and mechanical properties, there are other semantic properties that influence the product behavior. Some of them are listed below [42]:

- the component criticism,
- the reliability,
- the toxicity of the material,
- the recyclability (component reusability, biodegradability),
- the cost of components,
- etc.

The procedure of the object-based modeling is shown in Fig. 7. From the requirement document which defines the product performance objectives the CAD model is built with additional semantic data. Then the product is structurally decomposed into components and sub-assemblies. The pertinence of the structural decomposition level depends on different goals like product functional decomposition, manufacturing and assembling processes or commercial packaging. In the case of functional decomposition, the product is decomposed into functional sub-assemblies that match to functions. If manufacturing aspects are considered, the structural decomposition corresponds to other sub-assemblies. If commercial packaging is considered, the product will be decomposed in such a way that spares are provided easily. For complex products like cars or planes the decomposition depends on all these three factors. So the choice of a suitable decomposition level is a key problem that determines the behavioral performance indicator pertinence. After decomposition the assembly graph is constructed and transformed into a semantic matrix. Finally the object model is elaborated. The DSM semantic matrix and the product object-oriented model elaboration procedures are detailed below.

### 3.2. Building the DSM semantic matrix

The product design solution is assumed to consist of multi-components structure built by a set of components which are bind together by different types of assembly links. If some links consist of detachable fasteners, the product is decomposed into sub-systems, or single parts, by removing the links. The semantic DSM matrix shown in Fig. 8 represents the product CAD model data with additional semantic data concerning non-graphical characteristics that may have quantitative or qualitative values. For example material properties, functional criticality ($K_i$), reliability ($R_i$) are considered in the following matrix. Here a $n$-components product is described by its $C_i$ (with $i = 1, \ldots, n$) components. The link type for every couple of components $(C_i, C_j) = L_k$ is the assembly type between components $C_i$ and $C_j$. $L_k$ takes different values depending on
how two components are assembled. $K_i$ and $R_i$ stand respectively for the functional criticality and reliability associated to component $C_i$ [43]. The functional criticality level is estimated by the designer depending on the relative weight of the different components. $K_0$ is the threshold level of criticality used to identify critical components to be considered for maintainability indicator calculation. $R_0$ is the product global reliability threshold fixed in the requirements specifications [18].

The $P_i$ diagonal elements represent the components semantic properties. So $P_i$ is a semantic vector of various types of information like: material properties, surface state, heat treatments, surface treatments, etc.

Then, for component $C_i$, the attributes vector $P_i$ is defined by: $P_i = \text{Vector (material properties, surface state, surface treatment, heat treatment, density, volume, surface, etc.)}$

So, auxiliary information is captured at two levels: for each component in its semantic vector $P_i$, and for assemblies in the link type describer Link ($i, j$) between each couple of components $i$ and $j$. These information are initialized with default values stored in a database. Then, if necessary, the designer can modify these values to match particular specifications.

To perform the behavioral performances evaluation, we express the FBS product model as a generic object class. For a domain $D$, the criteria are defined as required conditions to ensure an acceptable behavior; criteria are then used to build a qualitative or quantitative indicator that informs the designer on the proposed design solution referring to $D$.

3.3. Building the product behavior oriented model

The product behavior-oriented model building procedure consists of the five following main steps:

(1) The procedure starts by the behavioral domain boundaries definition. These limits are not often so easy to define precisely and are simply expressed in terms of quality that can be appreciated by end-users. This is the case for reliability or availability requirements.

(2) Then behavioral criteria are identified. The identification consists of analyzing all internal and external elements that affect the behavior of a generic product with respect to this domain. The main criteria for a domain are identified by experts.

(3) Then follows the criteria characterization phase. For each criterion the parameters and the variables to be used in formulas or into a reasoning criterion are determined.

(4) The criteria are formalized by expressing them either with formulas or with algorithms.

(5) Finally, a behavioral object-oriented model is elaborated. For a considered domain, this final model defines criteria evaluation methods and specifies how the performance indicator is derived from criteria.

The product is defined as a generic class consisting of functions, structure constructors, criteria and performance indicators evaluation methods.

- According to requirements a product may have one or several intended Functions. These functions are specified in the model declarative description.
- Structure is the physical representation of the product. Different design solutions may be eligible to satisfy the product function(s). Each solution has a physical structure defined by its components with their specific geometry, topology, material properties, and surface characteristics. Different assembling techniques are used to attach these components: screwing, welding, etc.).

Let $S_1, S_2, \ldots, S_k$ be the corresponding structures of these solutions. Each structure $S_i$ is determined by design parameters: $P_{i1}, P_{i2}, \ldots, P_{ir}$.

- The Behavioral analysis of a technical solution provides a performance indicator with respect specific constraints imposed by the requirements document. A performance indicator may have a qualitative or quantitative value.

Traditionally, the way of determining the indicator is a method that works on some design parameters and additional information and then returns results to the designer [15].

Considering these three characteristics we specify a product behavior-oriented model by a generic class as follows.
The next section describes how the performance evaluation is carried out using this model.

4. Behavioral performances evaluation

We assume the hypothesis that any behavior satisfaction constraints are specified in the requirements documents before starting the design. In the FBS approach such information is provided either by the designer or by using an external knowledge-based system.

This information indicates how the product behaves and what the acceptability levels to be satisfied are. Depending on the complexity of the rules the different criteria are expressed using different more or less complex formalisms [6,25].

In the trivial cases a criteria is expressed as a comparison of two values. In the complex conditions the evaluation is made through reasoning or by using an algorithm.

Fig. 9 describes this evaluation procedure. As input the procedure uses our object-oriented product model defined above and performs analysis by using criteria evaluation.
corresponding methods. Then each behavior performance indicator is derived and displayed to the designer. The explored solutions are stored and a control loop gives information to the progressive improvement of the design solutions.

In the next section a software architecture is proposed for implementation in CAD systems.

5. Implementation in CAD systems

The integration of the Behavioral Performance Assessment Software (B-PAS) in a CAD system is proposed in Fig. 10. This software consists of three modules:

- The BEHAVIOR Modeler module which captures additional data and behavioral rules to enrich initial CAD model. This module also builds the product semantic matrix. From this matrix the product and its components object classes are generated.
- Using the Description Logic, the COHERENCE module controls the consistency of the different concepts referring to criteria to be satisfied. This is done using a Description Logic semantic analysis as outlined in [10].
- The EVALUATION module determines performance indicators.

The input data are provided from a CAD system using either native format or STEP/ISO-10303 format (STandard for the Exchange of Product data) [1,2,16,32]. The STEP format is nowadays widely used in industry. Its formal description language EXPRESS/Part 11 allows the implementation to be independent from the CAD software [23,28].

Currently the prototype of our system is implemented as an add-on module into SolidWorks CAD system. In this case the model translation consists of extracting the product part list that is provided as input to the assessment module.

In future work we plan to develop a standalone solution which will be independent from the CAD system.

In this purpose the input CAD model requires the use of neutral format like STEP AP-203/AP-214. Then the very tough work will concern components and mating conditions recognition in the STEP file.

The next section shows an application of the approach used for maintainability prediction of a technical solution.

6. Application for maintainability assessment

The maintainability is commonly defined as the characteristics of equipment design and installation that provides the ability for this equipment to be repaired easily and efficiently [9,19,36,37]. From the user point of view, maintainability refers to the aspects of a product that increase its serviceability and reparability, increase the cost-effectiveness of maintenance, and ensure that the product meets the requirements for its intended use. Here, we consider basic mechanical products with no electronic equipments and no software. Then the maintainability depends on the complexity of the structure: i.e. the geometry of parts and how components are assembled [29].

Table 1 lists the main criteria that affect maintainability.

Considering that criteria depend on intrinsic and external factors we define two kinds of evaluation: the intrinsic and contextual. The intrinsic evaluation provides indicators using criteria that depend only on the product structure. The contextual evaluation determines indicators taking into account additional constraints related to external criteria about usage conditions, the required maintenance tools, etc. Here we focus on intrinsic criteria for alternative design solutions comparison.

In practice, to evaluate the maintainability, the 1010/CCT specification defines the MTTR (Mean Time To Repair) as a most significant criterion. It is defined as the total time required for making diagnostic, reparation or replacement and control. For better maintainability the MTTR must be as short as

<table>
<thead>
<tr>
<th>Table 1 Maintainability criteria classification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intrinsic criteria</strong></td>
</tr>
<tr>
<td>Reparability, accessibility, dismountability, assemblability, disassemblability, standardisation, interchangeability, survivability, redondance</td>
</tr>
</tbody>
</table>
possible to minimize the repair times, thereby minimizing the downtime and increasing the product availability.

$$MTTR = T_{\text{Diag}} + T_R + T_C$$

The diagnostic time, $T_{\text{Diag}}$, depends on the type of failure. It can be instantly in the case of break of main components. But it can also take from a few minutes to much longer period if the failure occurred progressively as in the case of parts wear. Thus the $T_{\text{Diag}}$ is practically difficult to be estimated at the design stage. The control time, $T_C$, requires to verify that the product works properly. This time can be very short or can imply a long period of settings. Here, these times is not taken into account. The time devoted to reparation or replacement tasks, $T_R$, is the characteristic commonly used to estimate product maintainability.

$T_R$ depends on many criteria like: disassembly/assembly operations to be performed, components accessibility, components or sub-assemblies manoeuvrability, reparable and maintenance resources. For existing products the MTTR is traditionally computed from statistical data collected over a certain period of product utilization. But while designing new products no statistical data are available and the MTTR must be estimated by simulation methods as implemented in some commercial software like the Relx Maintainability Prediction software [44] or the package edited by Item Software/NSWC 98 [45]. These simulation methods are designed to analyze and calculate component, sub system and system failure rates in accordance with the appropriate standard and use repair time database but they are not connected to any CAD system. Other theoretical approaches quoted by [8] propose rules to apply for the integration of maintainability criteria in the product design process. These methods provide the design for maintainability guidelines but are not intended to evaluate the maintainability performance of an existing solution.

Here we consider the criteria related to disassembly/assembly operations to evaluate the MTTR.

### 6.1. Maintainability indicator definition

At first approximation, the product maintainability indicator is assumed to be the MTTR.

Step 1. **Semantic coherence control.** This preliminary control is necessary to detect inconsistencies in the product behavioral model and in the constraints specifications.

Step 2. **Behavioral analysis.**

- For $i = 1 \text{ to } k$ ($k$: number of components).
  - Search for critical components with $K_i > K_0$ (criticality threshold).
  - Determine for each component $C_i$ the accessibility sequence $\rightarrow a_i$ (component $C_i$ accessibility) using constraints propagation method proposed by Zwingmann et al. [33].

- **Maintainability indicator evaluation** $I_M$ of the product. Determination of MTTR, total required time for dissembling the critical components.

Step 3. **Validation or new design iteration.** If $MTTR < $ acceptable threshold fixed by requirements specifications

Then OK Else: Repeat product redefinition until constraints are fulfilled.

End of procedure.

The main steps of the evaluation algorithm are outlined below for a $k$-components product.

This procedure is refined if more criteria are considered. For instance the evaluation becomes more complex if contextual criteria like maintenance devices or human competence must studied.

### 6.2. Case study for an industrial trailer

In this section an application is presented to analyze the maintainability of a beam type trailer shown in Fig. 11.

This trailer is a manually handling device used to move a charge. A corresponding exploded view is represented in Fig. 12.

Here the structural decomposition is made considering manufacturing process concerning components and sub-assemblies. For detailed behavioral evaluation a deeper structural decomposition must be provided. For example Fig. 13 shows some sub-assemblies exploded views for a second level decomposition.

For illustration, we just analyze the first level evaluation. Fig. 14a represents the corresponding assembly graph. For evaluation Fig. 14b indicates the different assembly link types and their relative weights $q$. Coefficient $q$ is related to the time duration for maintenance operations required for disassembling components.

In Fig. 15 we assume that all components are critical, i.e. all criticality coefficients $K_i \geq K_0$. Fasteners referred to component $C_9$ are considered as accessories used to attach components together.
Fig. 13. CAD 3D model (2nd level decomposition).

Fig. 14. (a) Assembly graph (1st level decomposition). (b) Assembly link types relative weights.

Fig. 15. DSM semantic matrix of a trailer.
Since all components are assumed to be critical, disassembling operations consist of removing all links between components [39]. The MTTR is determined as the sum of the relative weights corresponding to the first level decomposition through components $C_1$ to $C_6$. Then the maintainability performance indicator is: $I_M(S_1) = \text{MTTR} = 36$. This is a relative indicator it must be checked through comparison to other alternative solutions and to the MTTR threshold specified in requirements. Let us consider two other alternative solutions $S_2$ and $S_3$:

- $S_2$ consists of welded solution where all connections are welded; then $I_M(S_2) = 70$.
- $S_3$ is a clipping assembly where all connections are clipped, then $I_M(S_3) = 14$.

This trivial comparison shows that $I_M$ can vary in a relatively large extends even for similar solutions. So this indicator is focused on the product structural architecture and informs the designer if the performance is improved or not.

7. Conclusions and future works

The framework for modeling and evaluation approach presented here is a generic object-oriented approach that attempts to ensure product behavioral performances evaluation during design process. The model presented allows capturing various criteria identified for different domains of the product lifecycle. The evaluation method provides to the designer an indicator that informs on how a design solution fulfills the behavioral constraints for a specific domain.

The integration of behavioral performance evaluation software may be independent of the CAD system native format. The feasibility of our approach is illustrated by an example on maintainability prediction.

Here the evaluation is just bounded to the disassembly constraints related to the product critical components but the approach can easily be extended to any other maintainability criteria like components accessibility, manoeuvrability.

In this application we analyze the intrinsic maintainability but the approach is also well adapted for contextual aspects related to external environment like maintenance tooling and equipments.

The challenge is how to express such criteria and make then usable in practice.

As the approach is generic it can be extended to other domains. We are working on reliability, safety and recyclability.

Acknowledgement

The authors are grateful to Manuest-Concept firm for its technical support concerning the industrial case study presented here.

References


[33] X. Zwiggmann, D. Ait-Kadi, A. Coulbaly, B. Mutel, Disassembly sequencing model for multi-component products, In a special issue on Quality and Maintenance in Industrial Engineering and Systems Management, in press.


Amadou Coulbaly is an assistant professor at Graduate School of Science and Technology of Strasbourg (INSA) in the Engineering Design Laboratory. He is a mechanical engineer from Ecole Polytechnique de Thies (Senegal) and doctor in mechanical engineering from Louis Pasteur University of Strasbourg (France). His research interests concern product modeling, knowledge modeling for product behavioral performance prediction at design stage using CAD systems. His current research industrial applications are focused on reliability and maintainability assessment.

Pr. Bernard Mutel is an emerite professor at Institut National des Sciences Appliquées de Strasbourg in the Engineering Design Laboratory. He is an engineer in cybernetics (ISIN), master of science in electrical engineering (IIT) and doctor é-sciences in automatics (U.M.). His main research interests lie in cellular manufacturing design and new production data analysis technics. Currently he is working on data mining for diagnosis.

Daoud Ait-Kadi is a full professor and the chairman of industrial graduate program in the mechanical engineering department at Laval University. He earned the B.Sc. in Mechanical Engineering in 1973 from Mohammadia School of Engineering (Morocco), the M.Sc. in 1980 and a Ph.D. in 1985 in industrial engineering, computer science and operation research from Ecole Polytechnique of Montreal and University of Montreal. His current research interests include reliability and maintainability modeling and optimisation, performance improvement, spare parts provisioning, life cycle engineering and reverse logistics. He has authored two books, co-authored over 200 scientific papers in journals and conferences. His research has been supported by NCRG and FQRNT (Canada) and industrial funding. Daoud Ait-Kadi is a senior member of IEEE and IIE. He is also a member of Academie Hassan II des Sciences et Techniques of Morocco.