



**HAL**  
open science

# Modeling Process Platforms based on An Object-Oriented Visual Diagrammatic Modeling Language

Linda Zhang

► **To cite this version:**

Linda Zhang. Modeling Process Platforms based on An Object-Oriented Visual Diagrammatic Modeling Language. *International Journal of Production Research*, 2009, 47 (16), pp.4413-4435. 10.1080/00207540801950144 . hal-00513029

**HAL Id: hal-00513029**

**<https://hal.science/hal-00513029>**

Submitted on 1 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**Modeling Process Platforms based on An Object-Oriented Visual Diagrammatic Modeling Language**

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2007-IJPR-0717.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	05-Jan-2008
Complete List of Authors:	Zhang, Linda; University of Groningen, Operations
Keywords:	PRODUCTION MANAGEMENT, PRODUCTION MODELLING
Keywords (user):	Process platform, high variety production



# MODELING PROCESS PLATFORMS BASE ON AN OBJECT-ORIENTED VISUAL DIAGRAMMATIC MODELING LANGUAGE

Lianfeng Zhang<sup>\*†</sup>

<sup>†</sup>Department of Operations, University of Groningen,  
Landleven 5, 9747 AD Groningen, The Netherlands

## ABSTRACT

Process platforms have been recognized as a promising means of dealing with product variety while achieving a near mass production efficiency. To assist practitioners to better understand, implement and use process platforms, this study addresses the underlying logic of coping with the challenges in high variety production by adopting process platforms. Accordingly, this paper proposes to model process platforms with focus on the application processes. In view of the significance of dynamic modeling and visualization in shedding light on the logic of any processes, this study introduces a visual diagrammatic modeling language based on object-oriented (OO) techniques, named as OOVDML. With the graphical notations, uniquely shaped symbols, syntax and semantics, control mechanisms and arrangement rules, the OOVDML not only captures the logic of process platform's application but also provides a visualization of their behaviors in a holistic view. Moreover, incorporating OO modeling allows readers to focus on their own interests. This study approaches to modeling process platform's application with respect to activities pertaining to customer order processing, engineering change control and production job planning. Also reported is an industrial example of electronics products. The results of the case study not only show the suitability of the OOVDML but also shed light on the dynamic behaviors of process platforms.

**Key Words:** *Process platform, high variety production, diagrammatic modeling, object-oriented methods, visualization.*

---

\* Corresponding author. Email: [L.Zhang@rug.nl](mailto:L.Zhang@rug.nl)

## 1. INTRODUCTION

Manufacturing environments nowadays are characterized by a high variety of customized products, often coupled with small quantities and short delivery lead-times. To survive the resulting intense global competition by pleasing their customers, manufacturing companies struggle to provide quickly high product variety at low costs. The key for companies to achieve efficiency in producing large numbers of customized products lies in an ability to maintain the resulting high variety production to be as stable as possible (Schierholt, 2001; Williams et al., 2007; Zhang, 2007). In this respect, process configuration contributes to manufacturing stability by generating similar processes for part families (Schierholt, 2001). It is an alternative of computer-aided process planning per se.

Similar as the process configuration thinking, a concept of process platforms has been recognized as a promising means for companies to achieve a near mass production efficiency by managing high variety production, wherein the complex products consisting of assemblies and parts are involved (Zhang, 2007). The rationale is to plan and utilize similar, yet optimal, production processes as these existing on shop floors to fulfill diverse customized products. Current research efforts have approached process platforms from several aspects (Zhang, 2007), including conceptual formulation (Jiao et al., 2007a), structural representation (Zhang et al., 2007), construction (Jiao et al., 2007b) and identification of mapping relationships within a process platform (Jiao et al., 2008). Besides the insight provided by the above works, a good understanding of the underlying logic of coping with difficulties in high variety production by adopting process platforms is necessary for companies to design, develop and apply process platforms.

With an attempt to assist practitioners to better understand, implement and use process platforms, this study addresses the underlying logic of process platform's behaviors. In view of the significance of dynamic modeling and visualization in shedding light on the

1  
2  
3 logic of any processes, this paper, accordingly, proposes to model process platforms with  
4 focus on the application processes, i.e., dynamic modeling. Along with the complexities in  
5  
6 fulfilling high product variety, the resulting difficulties in modeling process platform's  
7  
8 application behaviors have been recognized, as elaborated below.  
9  
10  
11

### 12 **1.1. Difficulties in modeling process platforms**

13  
14  
15 Modeling process platforms intends to visualize their application behaviors in high  
16 variety production in a holistic view. The diverse customized products along with the  
17 resulting large number of constituent items impose many complexities in production. For  
18 example, many personal, activities, data, information, etc. are involved in different phases  
19 of production. The same data may be manipulated by many different activities, which, in  
20 their turn, may manipulate other data as well. Personal may carry out different activities  
21 and activities may be carried out by different personal. Activities also have mutual  
22 dependencies. They may access the same data at the same time or they may be ordered in  
23 complex ways. In this regard, modeling process platforms is expected to capture the  
24 inherent complexities in one dimension rather than many dimensions, e.g., activities,  
25 processes, personal. Describing the different types of system elements in more than one  
26 dimension incurs difficulties in providing readers with an overall picture of process  
27 platform's application in a holistic view.  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44

45  
46 Second, process platform modeling is to shed light on the underlying logic of coping  
47 with difficulties by adopting process platforms in high variety production. Incorporating  
48 process platforms when fulfilling various products has brought about many additional  
49 events, activities, information and data in production. Moreover, it is not uncommon that  
50 any processes involve a number of activities that are concurrent, dependent and parallel.  
51 Because of these activities/events and their complicated relationships, reasoning about  
52 process platform's application behaviors is important for the interested readers to gain an  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3 insight in the logic. Therefore, the process platform models to be constructed must be able  
4  
5 to reveal the underlying reasoning of process platform's application.  
6  
7

8 Third, in a company, people at different levels view the same thing from different  
9  
10 aspects and have different focuses. This is also true for process platform's application. For  
11  
12 example, while management people may care about the functional areas involved and their  
13  
14 relationships, production personal are more concerned with the activity details in his/her  
15  
16 own area. In accordance with the different views, the process platform models to be  
17  
18 constructed should present the corresponding application processes at different levels of  
19  
20 abstraction, which allows different readers to pay attention to their own interested areas.  
21  
22  
23

24  
25 Furthermore, the ultimate goal of process platform modeling is to help practitioners  
26  
27 better understand and grasp the essence of process platform's application. Thus, it raises  
28  
29 the importance in an unambiguous modeling. In other words, process platform models  
30  
31 should be constructed to allow different readers to interpret the corresponding application  
32  
33 uniquely and in an exactly same way. Besides, the models are expected to provide an easy-  
34  
35 understandability and readability.  
36  
37

## 38 **1.2. Strategy for solution**

39  
40 To cope with the modeling difficulties, this study puts forward an OOVDML (object-  
41  
42 oriented (OO) visual diagrammatic modeling language) by integrating the principles of a  
43  
44 number of well-defined modeling techniques.  
45  
46  
47

48 Graphical notations and diagrams are incorporated in the OOVDML to provide a  
49  
50 visualization of process platform's application in a holistic view. Besides graphical  
51  
52 representations, textual representations in the natural language are introduced. Textual  
53  
54 representations are able to capture the complicated data, information, personal, activities  
55  
56 and their relationships and further map them in one dimension. In the OOVDML, they are  
57  
58 used to model the detailed activities, involved personal, inputs and outputs. They are also  
59  
60

1  
2  
3 used to denote the names and attributes of graphical notations defined. To deal with the  
4  
5 issues regarding different levels of abstractions, the class concept in OO modeling is  
6  
7 incorporated. Both the class attributes and/or detailed class operations can be hidden, when  
8  
9 necessary. As a result, the readers are able to focus on their own interests. Moreover, a  
10  
11 number of arrangement rules and control mechanisms along with predicate formulas are  
12  
13 defined in the OOVDML to capture and model the reasoning of process platform's  
14  
15 application behaviors.  
16  
17

18  
19  
20 Rather than all processes in high variety production, this study approaches process  
21  
22 platform modeling with focus on activities pertaining to production job planning,  
23  
24 engineering change control and customer order processing. The reason is that these  
25  
26 processes are fundamental to efficient high variety production.  
27  
28

29  
30 The rest of the paper is structured as follows: The work regarding high variety  
31  
32 production management and dynamic modeling languages is present in Section 2,  
33  
34 following which the fundamentals of process platforms are introduced in Section 3. The  
35  
36 OOVDML is detailed in Section 4. Section 5 reports an industrial example involving  
37  
38 vibration motors for mobile phones. Section 6 concludes the paper by discussing  
39  
40 advantages and disadvantages of the proposed OOVDML and by identifying future  
41  
42 research.  
43  
44

## 45 46 **2. RELATED WORK**

47

48  
49 In the past several decades, a large body of literature has been reported to assist  
50  
51 companies to provide various customized products as expected by customers. Such  
52  
53 research efforts have been made mainly in the areas of product platform development  
54  
55 (Simpson, 2004), product family design (Sanderson and Uzumeri, 1995), outsourcing  
56  
57 (Harland et al., 2005) and supply chain management (Min and Zhou, 2002). While the  
58  
59 resulting methodologies/system prototypes/frameworks, to some degree, help companies  
60

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  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

achieve design efficiency and effectiveness and selection of proper supply chain partners, they lack an explicit consideration of issues regarding production of high product variety based on companies existing manufacturing resources (Lu and Botha, 2006). Schierholt (2001) introduces the concept of process configuration, which combines the principles of product configuration and process planning, to deal with manufacturing process generation. Williams et al. (2007) put forward process parameter platforms for developing manufacturing processes by taking into account the non-uniform market demand. In essence, both process configuration and process parameter platforms address process development for parts. Zhang (2007) presents process platforms to assist companies to achieve a near mass production efficiency in high variety production. Unlike process configuration and process parameter platforms, process platforms intend to facilitate production process generation for complicated end-products, wherein both parts and assemblies are involved.

The conceptual models of process platforms are formulated with respect to basic constructs, definitions, relationships and functionalities using set theory and OO techniques (Jiao et al., 2007a). Also discussed are process platform's concept implications: generic variety representation, generic structures and generic planning. The large number of different types of product items (including end-products) and the corresponding process elements involved in a process platform have been identified and described using unified modeling language (Zhang et al., 2007). To assist companies to benefit from the past production practice, Jiao et al. (2007b) put forward a data mining methodology to identify and form process families in relation to product families. Similarly, to facilitate configuration rule construction in process platforms, an approach based on association rule mining is discussed to obtain mapping relationships between product and process variety from large volumes of existing production data (Jiao et al., 2008).

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  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

Dynamic modeling is to model the relevant processes involving objects being studied. As the core of process modeling, modeling techniques or languages have been investigated. There are two classes of process modeling languages using formal notations: visual diagrammatic process modeling languages and programming process modeling languages. Since the programming process modeling languages (Vernadat, 1993; Sutton et al., 1990) have strong limitations in readability and understandability, most process modeling has employed diagrams. Diagrammatic representation that combines graphic notations and natural language-like strings as the formalism can facilitate readers' understanding.

The major diagrammatic process modeling languages can be summarized as simple process diagrams, advanced process diagrams and functional description diagrams. The simple process diagrams, such as traditional flow charts, material flow diagrams (Baudin, 1990) and acyclic networks (Hajdu, 1997), are simple, easy to use and understand. However, they are unable to express complicated execution relationships and have limited capacity in representing behavioral aspect of a process. With an attempt to include more information about processes, advanced process diagrams, e.g., process flow entity diagrams (Grabowski et al., 1996), event-driven process chains (Zukunft and Rump, 1996), IDEF3 (<http://www.idef.com/Home.htm>), event diagrams (Martin and Odell, 1992), Petri nets (Peterson, 1981), role activity diagrams (Ould, 1995), have been developed. Compared with simple process diagrams, the advanced process diagrams can capture more information about processes from, e.g., organizational perspective, behavioral perspective. Nevertheless, most of such advanced process diagrams are unable to handle the exceptional and non-deterministic execution relationships. Furthermore, they lack the capability to represent processes from the informational and functional perspectives. While functional description diagrams, including IDEF0 (<http://www.idef.com/Home.htm>), object flow diagrams (Martin and Odell, 1992) and use case diagrams (Jacobson, 1992),

1  
2  
3 are well-defined mechanisms to represent functional, informational and organizational  
4 perspectives of processes, they are unable to express the behavioral perspective.  
5  
6

7  
8 OO modeling notions and methods (Rumbaugh et al., 1991; Booch, 1994; Martin and  
9 Odell, 1996; Harmon, 1998) are powerful for dealing with modeling complexity in the real  
10 world and building a model in a comprehensive, expressive, understandable and structured  
11 formalism. The notions and technique are believed to provide an innovative and effective  
12 way of modeling a variety of production activities in relation to diverse products as well.  
13 Recognizing the limitations of the diagrammatic process modeling languages and the  
14 power of OO notations and methods, Ma (1999) puts forward a language including a  
15 customer process flow diagram and an entity representation diagram to model service  
16 product design. With reference to his work, the OOVDML is devised in this study to  
17 model process platform's application processes.  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31

### 32 **3. FUNDAMENTALS OF A PROCESS PLATFORM**

33  
34 Current practice in design, be it product configuration or platform-based product  
35 family design, leads to the concept of product families. A product family consists of a set  
36 of customized, yet related, products which perform a same basic function. While  
37 customized products belonging to the same family assume a common product structure,  
38 they differ with one another in optional features and functionalities. The design changes  
39 among family members impose necessary variations in converting abstract design  
40 concepts into physical products. On the other hand, the similarity and commonality  
41 inherent in product families, exhibited by similar and/or same raw materials, components,  
42 subassemblies and assemblies, makes it possible for companies to utilize similar  
43 production processes to fulfill product family members. Lu and Botha (2006) point out  
44 that unlike product development, process development has not received too much attention  
45 in today's manufacturing environment because of the technical difficulties and managerial  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3 challenges to be considered. In response to the lack of research in process development,  
4  
5 [Zhang \(2007\)](#) proposes process platforms to assist companies to plan similar, yet optimal,  
6  
7 production processes for product families. The ultimate goal is to help companies achieve  
8  
9 a near mass production efficiency in producing high product variety.  
10  
11

12  
13 A process platform entails a conceptual structure and overall logical organization of  
14  
15 production processes to produce a product family. It provides a generic umbrella to  
16  
17 capture and utilize commonality, within which each new product fulfillment is instantiated  
18  
19 and extended, and thereby anchoring process planning to a common structure. A process  
20  
21 platform contains all data pertaining to the product family, e.g., items, quantity-per,  
22  
23 parent-child relationships, and all production process data of the corresponding process  
24  
25 family, e.g., operations types and precedence, work centers, machines, tools, standard  
26  
27 cycle times, setup activities. With an attempt to include all product and process family data  
28  
29 in a single structure without data redundancy, the concept of generic variety representation  
30  
31 ([van Veen, 1992](#)) is adopted to organize data in process platforms.  
32  
33  
34  
35

36  
37 In a process platform, the specific data pertaining to individual products and  
38  
39 production processes is organized as a generic product structure of the product family and  
40  
41 as a generic process structure of the process family in relation to the product family,  
42  
43 respectively. As a result, the two generic structures are common to all members in product  
44  
45 and process families. Further, they are integrated into a single structure, called generic  
46  
47 product-process structure, which is fundamental to the process platform. The integration is  
48  
49 achieved by adopting the mapping relationships between two sets of family data. Figures 1  
50  
51 and 2 show conceptually the generic product-process structure, the generic product  
52  
53 structure and the corresponding generic process structure, respectively.  
54  
55  
56

57  
58 As shown in the figures, each node in the generic structures is either a generic product  
59  
60 item or a generic process in relation to a generic product item. While a generic product





a class of production activities. The activity attributes common to most production activity classes are identified as follows:

- responsible functions: indicating the functions executing these activities;
- processing time: showing the duration time of an activity;
- frequency of occurrence: describing the frequency of an activity; and
- constraints: concerning with the conditions that need to be met for an occurrence of an activity to start, continue, or stop.

Activity operations describe the behavioral property of a class of production activities, more specifically the processing steps. An activity starts when its operation is triggered; an activity terminates when its operation stops.

The syntax of the production activity names, attributes and operations is defined as follows:

*ActivityName* ::= *Ipa*: *Gerund-phrase* // *Gerund-phrase*: a gerund phrase string

*Ipa* ::= **PA** *IntegerNumber*

*IntegerNumber* ::= 1 | 2 | 3 | ...

*ActivityAttribute* ::= *ActivityAttributeName* = *AttributeValue*

*ActivityAttributeName* ::= *String* // *String*: a string

*AttributeValue* ::= *STF* // *STF*: a written representation in strings, tables, figures, or the combination of them

*ActivityOperation* ::= *ActivityOperationName* {*ActivityScript*}

*ActivityOperationName* ::= *Gerund-phrase* ( ) | *Verb-phrase* ( )

*ActivityScript* ::= *Script* // *Script*: a script written in the natural language

An *Ipa* is a code of a production activity instance of a class. A code is formed by a reserved word **PA** (production activity) and an integer number. A *Gerund-phrase* is a gerund phrase and is used as a part of a name string that can be understood literally. An *ActivityAttributeName* denotes the name of an activity attribute and an *AttributeValue* represents a value of an attribute. An *ActivityOperationName* and an *ActivityScript* define the name and specification of an activity operation, respectively. An



1  
2  
3 Similarly, the adornment in the form of a string is placed near the solid bar and identifies  
4 the *SequentialJunction*. The strings on the tag describe the set of control conditions. When  
5 a *SequentialJunction* connects only two *ArrowLinks*, the tag is allowed to be omitted. In  
6 this case, the default control condition is when the incoming *ArrowLink* receives execution  
7 control flow, the outgoing *ArrowLink* does. The strings are written in predicate formulas.  
8 Such formulas are able to provide a convenient and flexible mode to define the control  
9 conditions in concise propositions. The basic components of strings include two predicates,  
10 three connectives and related semantics conventions as follows:  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21

22 Predicates:

- 23 (1) **signal**( $x$ ): *ArrowLink*  $x$  has an execution control flow; and
- 24 (2) **event**( $e$ ): event  $e$  occurs.

25  
26  
27  
28  
29 Connectives:

- 30 (1)  $\wedge$ : (conjunction) means “and”;
- 31 (2)  $\vee$ : (disjunction) indicates “or”; and
- 32 (3)  $\rightarrow$ : (conditional symbol) means “if... then...”.

33  
34  
35  
36  
37 Related semantics conventions:

- 38 • For *ArrowLink*  $x$  that connects to a *PAC* by the head end, when **signal**( $x$ ) becomes  
39 true, an activity defined by the connected *PAC* will eventually happen, i.e., start.
- 40 • When an activity defined by a *PAC* completes or terminates, **signal**( $x$ ) becomes true  
41 for *ArrowLink*  $x$  connecting to the *PAC* by its tail end.
- 42 • When a *StartPoint* is initiated, **signal**( $x$ ) becomes true for *ArrowLink*  $x$  that connects  
43 to the *StartPoint* by its tail end.
- 44 • For *ArrowLink*  $x$  that connects to an *AchievementPoint*, when **signal**( $x$ ) becomes true,  
45 a process of production activities ends.

46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
With the above definitions and semantics, the OOVDML is able to represent three  
types of activities bearing the three temporal and logical relationships:





- 1  
2  
3  
4  
5  
6  
7  
8  
9
- Its tail end connects to a *SequentialJunction* and the head end connects to another *SequentialJunction*.

R5. For a *SequentialJunction*,

- 10  
11  
12  
13  
14  
15  
16  
17  
18  
19
- there is one or more *ArrowLinks* that connect to the *SequentialJunction* by their respective head ends; and
  - there is one or more *ArrowLinks* that connect to the *SequentialJunction* by their respective tail ends.

R6. For two different *SequentialJunctions*,

- 20  
21  
22  
23  
24  
25  
26  
27
- there exists either only one or no *ArrowLink* that directly connects to the two *SequentialJunctions* by its two ends.

## 5. INDUSTRIAL EXAMPLE

28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

The industrial example adopted is the production of vibration motors for mobile phones in an electronics company. While vibration motors for mobile phones are not so complicated compared with others, the degree of their product complexity allows the modeling of process platform's application by reflecting real world situations as more as possible. Due to the frequent design changes to mobile phones, motors must be customized to match the requirements of these diverse mobile phones. In spite of the fact that such individual motors have certain items in common, they have their distinct design specifications, thus imposing different production requirements. Different from the mass production environment in the past, the current manufacturing environment in the company is characterized by diverse customized motors, low production volumes, short delivery lead-times and increasingly reduced costs. Together with the limited manufacturing resources, these characteristics complicate the company's production from every aspect: planning, scheduling, execution, etc. Taking production process planning as an example, the production process for producing a standardized motor in the mass





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  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

claimed by Tavcar and Duhovnik (2005), companies' ability to manage engineering changes efficiently reflects their agility. Failure to control ECs and manage the corresponding ECC process inevitably leads to the production of low quality (sometimes even unwanted) products, higher production costs and delayed order deliveries.

Since the company's production involves a high variety of customized motors coupled with small quantities and short delivery lead times, ECs become more serious due to the various design changes. With the process platform, the company can manage ECs effectively, through defining both BOMs and BOMfrOs based on different values of variety parameters in accordance with design changes.

For any EC proposal, the systematic analysis of possible implementations of the change, i.e., new item design, is required. In their previous production, such analysis was time and effort-consuming. With the process platform, the company can perform the analysis effectively. First, the generic product and process structures eliminate the time to analyze the costly and/or technically infeasible product items to implement the proposal. The reason is that the two generic structures provide all the technically and economically feasible designs for both motors and the constituent items. Consequently, the process platform confines the analysis of infeasible implementation by considering the company's existing design and manufacturing capabilities. Second, the simultaneous derivation of BOMs and the corresponding BOMfrOs in response to possible design proposals enables multiple functions in the company to assess concurrently the impacts with respect to time and cost of changes on activities to be carried out in their own functional areas.

The combination of the functional requirements in the order in Table 1 necessitate new bracket assembly. Thus the company has performed ECC on the process platform, as shown in Figure 11. After processing the order and conducting engineering changes, the motor in terms of BOM is obtained, as shown in Table 2.





1  
2  
3 While the above three processes have their different focus areas, they have certain  
4 overlap due to the adoption of the process platform in Figure 9. The common sub-process  
5 is circled by red double lines in the three processes.  
6  
7  
8  
9

## 10 11 6. CONCLUSIONS

12  
13 In view of the significance of process platforms for managing production of high  
14 product variety in manufacturing environments nowadays, this study proposed to model  
15 process platforms with focus on the application processes. Accordingly, the OOVDML  
16 was put forward to cope with the modeling difficulties. The industrial example has  
17 revealed several advantages of the OOVDML for modeling process platform's application.  
18 First, the diagrammatic representation visualizes an overall picture of process platform's  
19 application process in a holistic view. This, in turn, provides a starting point for companies  
20 to understand, analyze and further improve the process in consideration. Second, the well-  
21 defined syntax and semantics offer a thorough understanding and rigorous interpretation of  
22 activities, personal, their relationships, etc. in process platform's application processes.  
23 Consequently, in conjunction with the diagrammatic representation they enable companies  
24 to be aware of the impacts of any changes to activities, personal and their relationships on  
25 the application processes. Third, the OO modeling techniques along with the texts allow  
26 the application processes to be documented easily at different levels of granularity, which  
27 provides different people with their own interests.  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48

49 Nevertheless, the OOVDML has its limitations. With focus on the modeling of process  
50 platform's application processes, the OOVDML does not pay attention to a process  
51 platform itself. Consequently, it cannot shed light on the variety of constituent elements  
52 and their complicated relationships inherent in a process platform. This may provide an  
53 opportunity to extend this study. In future research, a comprehensive modeling  
54 language/formalism may be developed based on several well-defined modeling techniques.  
55  
56  
57  
58  
59  
60

1  
2  
3 Such a comprehensive modeling language should be able to capture and model both  
4 process platforms and their application processes. The resulting models can, thus, be  
5 expected to reflect the interactions between process platforms and their application, based  
6 on which companies can make improvements on process platforms and/or application  
7 processes eventually.  
8  
9

10  
11  
12  
13  
14  
15 Another avenue for directing future research would be to develop a computer-aided  
16 system incorporating the OOVDML. Companies' legacy systems may be integrated with  
17 such a system so that the system is able to capture real-time data and information, thus  
18 providing more accurate information for companies to make right decisions in adding,  
19 removing, modifying activities in the application processes.  
20  
21  
22  
23  
24  
25

## 26 27 28 **ACKNOWLEDGEMENTS**

29  
30 The author would like to thank two anonymous reviewers and the editor for their  
31 insightful and constructive comments on the earlier version of this paper.  
32  
33  
34

## 35 36 **REFERENCES**

- 37  
38 Baudin M., *Manufacturing Systems Analysis*. 1990, Yourdon Press Computing Series.  
39  
40 Booch G., *Object-Oriented Analysis and Design*. 1994, Benjamin/Cummings Publishing  
41 Company, Inc.  
42  
43 Grabowski H., Furrer M., Renner D. and Schmid C., Implementation of information  
44 system supporting engineering process based on World Wide Web. In: Scholz-Reiter,  
45 B., Stickel, E. (Eds.), 1994, *Business Process Modeling*, Springer, Berlin.  
46  
47  
48  
49  
50  
51  
52 Hajdu M., *Networks Scheduling Techniques for Construction Project Management*. 1997,  
53 Kluwer Academic Publishers, Dordrecht.  
54  
55  
56  
57 Harland C., Knight L., Lamming R. and Walker H., Outsourcing: Assessing the risks and benefits  
58 for organizations, sectors and nations. *International Journal of Operations & Production*  
59 *Management*, 2005, **25**, pp. 831-850.  
60

1  
2  
3 Harmon P., *Understanding UML: The Developer's Guide: With a Web-based Application*  
4  
5 *in Java*. 1998, Morgan Kaufmann Publishers, San Francisco.

6  
7  
8 <http://www.idef.com/Home.htm>  
9

10 Jacobson I., *The Object-Oriented Software Engineering*. 1992, Addison-Wesley  
11  
12 Publishing Company.

13  
14  
15 Jiao J., Zhang L. and Pokharel S., Process platform planning for variety coordination from  
16  
17 design to production in mass customization manufacturing. *IEEE Transactions on*  
18  
19 *Engineering Management*, 2007a, **54**, pp. 112-129.

20  
21  
22 Jiao J., Zhang L., Pokharel S. and He Z., Identifying generic routings for product families  
23  
24 based on text mining and tree matching. *Decision Support Systems*, 2007b, **43**, pp.  
25  
26 866-883.

27  
28  
29 Jiao J., Zhang L., Zhang Y. and Pokharel S., Association rule mining for product and  
30  
31 process variety mapping. *International Journal of Computer Integrated Manufacturing*,  
32  
33 2008, **21**, pp. 111-124.

34  
35  
36 Kolisch R., Integration of assembly and fabrication for make-to-order production.  
37  
38 *International Journal of Production Economics*, 2000, **68**, pp. 287-306.

39  
40  
41 Lu Q. and Botha B., Process development: a theoretical framework. *International Journal*  
42  
43 *of Production Research*, 2006, **44**, pp. 2977-2996.

44  
45  
46 Ma Q., *An Approach to the Representation of Service Process Design*. PhD dissertation,  
47  
48 1999, Department of Industrial Engineering and Logistics Management, Hong Kong  
49  
50 University of Science and Technology.

51  
52  
53 Martin J. and Odell J., *Object-Oriented Analysis and Design*. 1992, Prentice Hall,  
54  
55 Englewood Cliffs, NJ.

56  
57  
58 Martin J. and Odell J., *Object-Oriented Method — Pragmatic Considerations*. 1996,  
59  
60 Prentice Hall, Englewood Cliffs, NJ.

- 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  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60
- Maull R., Hughes D. and Bennett J., The role of the bill-of-materials as a CAD/CAPM interface and the key importance of engineering change control. *Computing & Control Engineering Journal*, 1992, **3**, pp. 63-70.
- McIntosh K.G., *Engineering Data Management: A Guide to Successful Implementation*, 1995, McGraw-Hill, New York.
- Min H. and Zhou G., Supply chain modeling: Past, present and future. *Computers & Industrial Engineering*, 2002, **43**, pp. 231-249.
- Ould M.A., *Business Processes*. 1995, Wiley, New York.
- Peterson J.L., *Petri Net Theory and The Modeling of Systems*. 1981, Prentice-Hall, Englewood Cliffs, NJ.
- Rumbaugh J., Blaha M., Premerlani W., Eddy F. and Lorensen W., *Object-Oriented Modeling and Design*. 1991, Prentice Hall, Englewood Cliffs, NJ.
- Sanderson S. and Uzumeri M., Managing product families: The case of the Sony Walkman. *Research Policy*, **24**, pp. 761-782.
- Schierholt K., Process configuration: Combining the principles of product configuration and process planning. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2001, **15**, pp. 411-424.
- Simpson T.W., Product platform design and customization: Status and promise. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2004, **18**, pp. 3-20.
- Sutton S., Heimbigner D. and Osterweil L.J., Language constructs for managing change in processes centered environments. *Proceedings of the Fourth SIGSOFT Symposium on Software Development Environments. Software Engineering Notes*, 1990, **15**, 206–217.
- Taal M. and Wortmann J.C., Integrating MRP and finite capacity planning. *Production Planning & Control*, 1997, **8**, pp. 245-254.

- 1  
2  
3 Tavcar J. and Duhovnik J., Engineering change management in individual and mass  
4  
5 production. *Robotics and Computer-Integrated Manufacturing*, 2005, **21**, pp. 205-215.  
6  
7  
8 Terwiesch C. and Loch C.H., Managing the process of engineering change orders: The  
9  
10 case of the climate control system in automobile development. *Journal of Product*  
11  
12 *Innovation Management*, 1999, **16**, pp. 160-172.  
13  
14  
15 van Veen E.A., *Modeling Product Structures by Generic Bills-of-Materials*, 1992, Elsevier,  
16  
17 New York.  
18  
19  
20 Vernadat F., CIMOSA: enterprise modeling and integration using a process based  
21  
22 approach. In: Yoshikawa H. and Goossenaerts J. (Eds.) *Information Infrastructure*  
23  
24 *Systems for Manufacturing*, 1993, North-Holland, Amsterdam.  
25  
26  
27 Williams C.B., Allen J.K., Rosen D.W. and Mistree F., Designing platforms for  
28  
29 customizable products and processes in markets of non-uniform demand. *Current*  
30  
31 *Engineering: Research & Applications*, 2007, **15**, pp. 201-216.  
32  
33  
34 Wortmann J.C., Muntslag D.R. and Timmermans P.J.M., *Customer-Driven Manufacturing*,  
35  
36 1997, Chapman and Hall, London.  
37  
38  
39 Yeh C.H., Production data modeling: An integrated approach. *International Journal of*  
40  
41 *Operations & Production Management*, 1995, **15**, pp. 52-62.  
42  
43  
44 Zhang L., *Process platform based-production configuration for mass customization*. PhD  
45  
46 Dissertation, 2007, Division of Systems and Engineering Management, Nanyang  
47  
48 Technological University, Singapore.  
49  
50  
51 Zhang L., Jiao J. and Helo P., Process platform representation based on unified modeling  
52  
53 language. *International Journal of Production Research*, 2007, **45**, pp. 323-350.  
54  
55  
56 Zukunft O. and Rump F., From business process modeling to workflow management: An  
57  
58 integrated approach. In: Scholz-Reiter B. and Stickel E. (Eds.) *Business Process*  
59  
60 *Modeling*, 1996, Springer, Berlin.

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  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

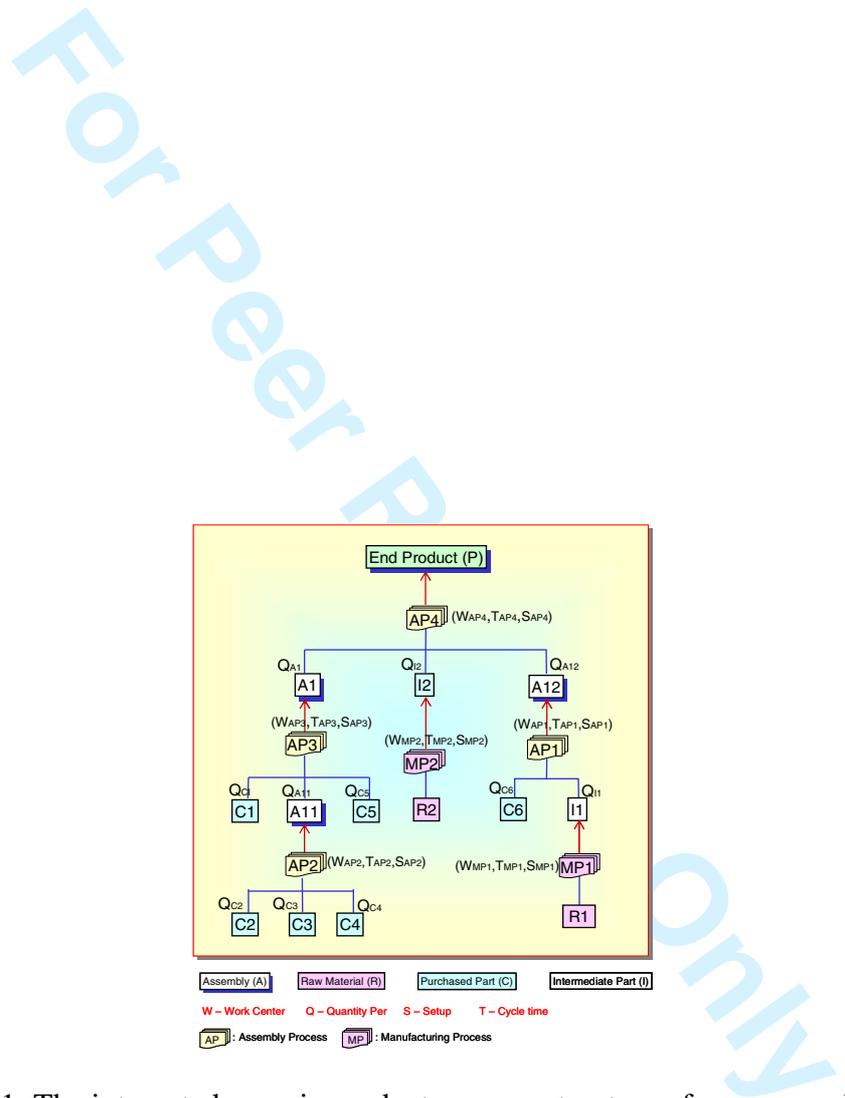


Figure 1: The integrated generic product-process structure of a process platform



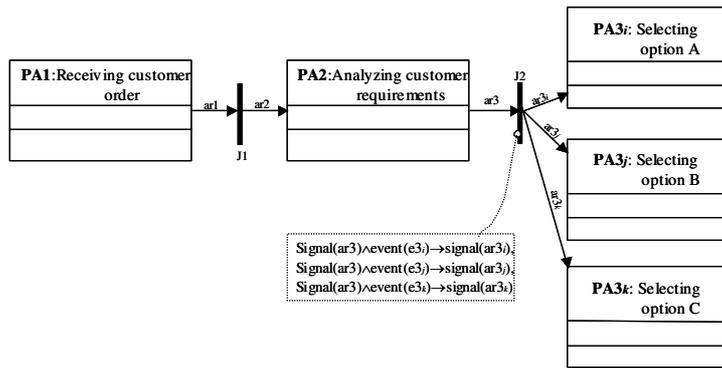


Figure 6: The tag attached to J2



A StartPoint with attribute    An AchievementPoint without attribute

Figure 7: Example notations of StartPoint and AchievementPoint

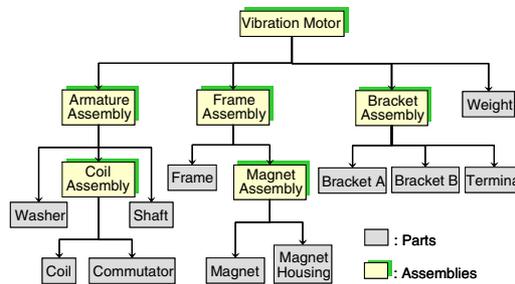


Figure 8: The common product structure of a motor family

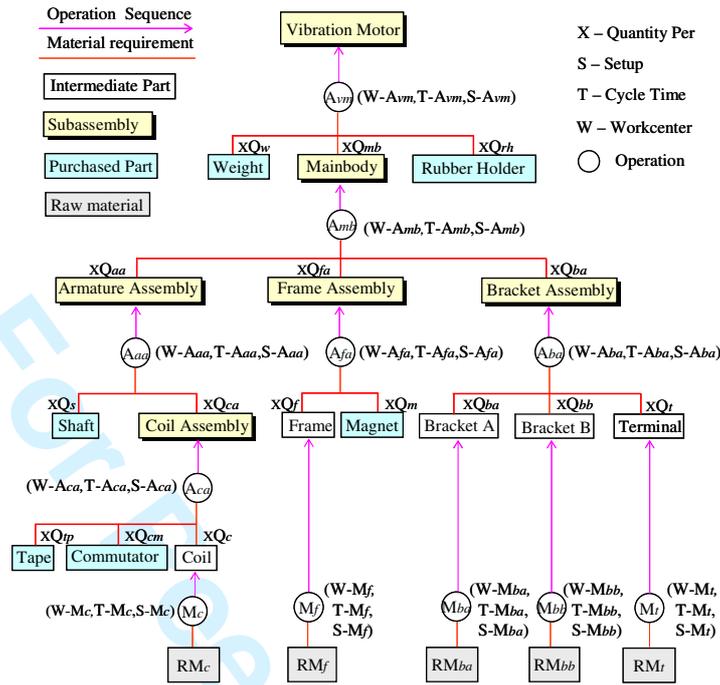


Figure 9: The process platform of the motor family

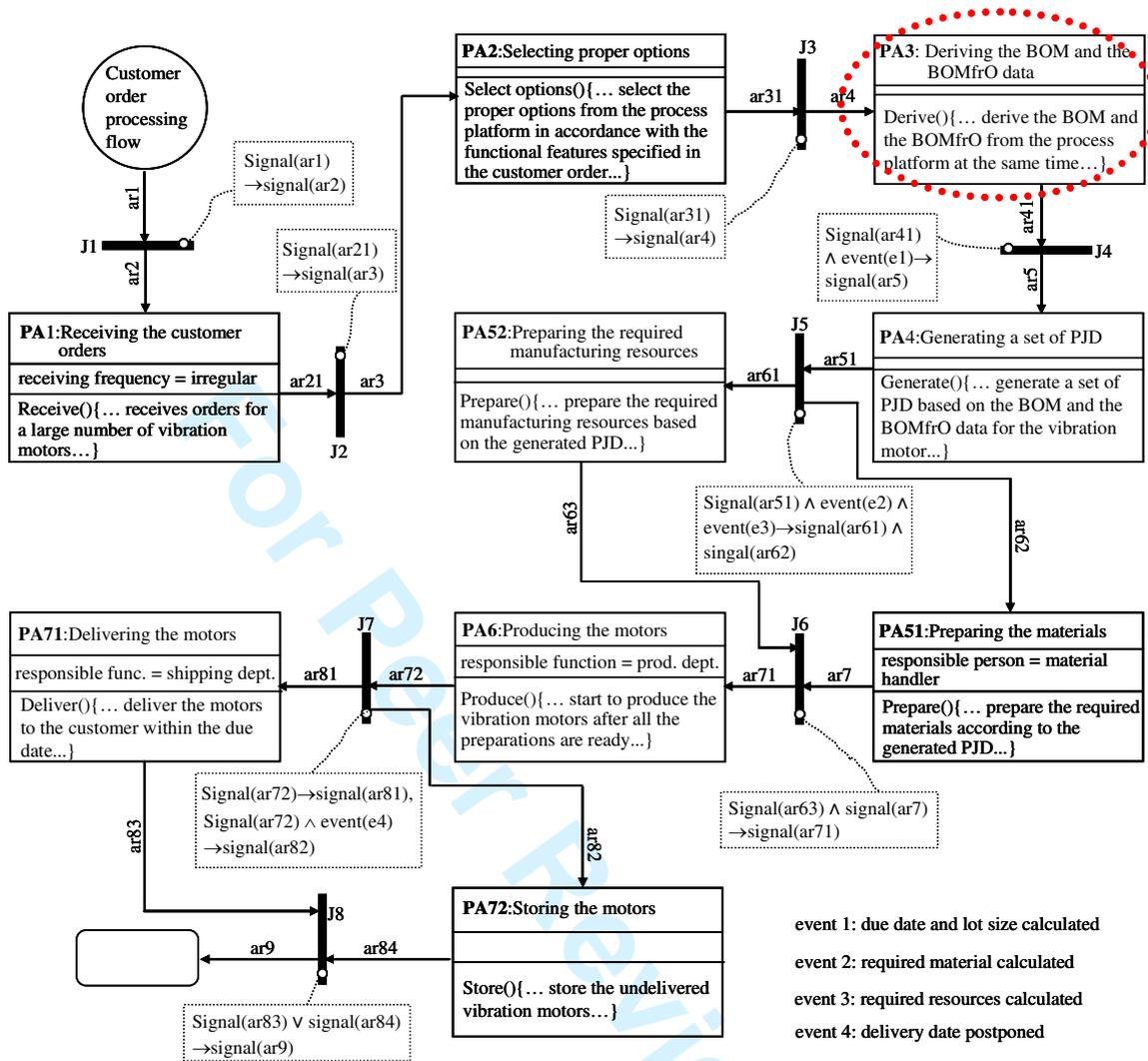


Figure 10: Processing the customer order in Table 1

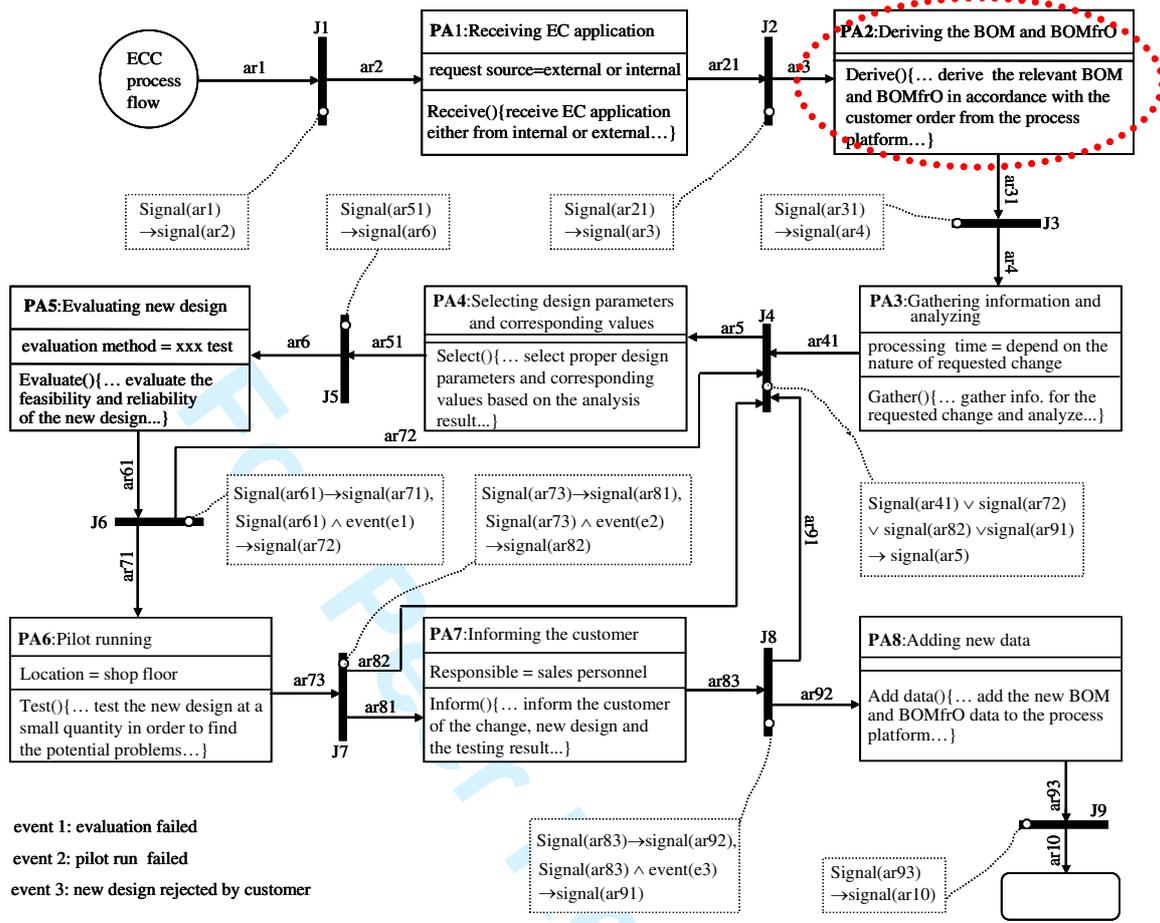


Figure 11: An ECC process in relation to the customer order in Table 1

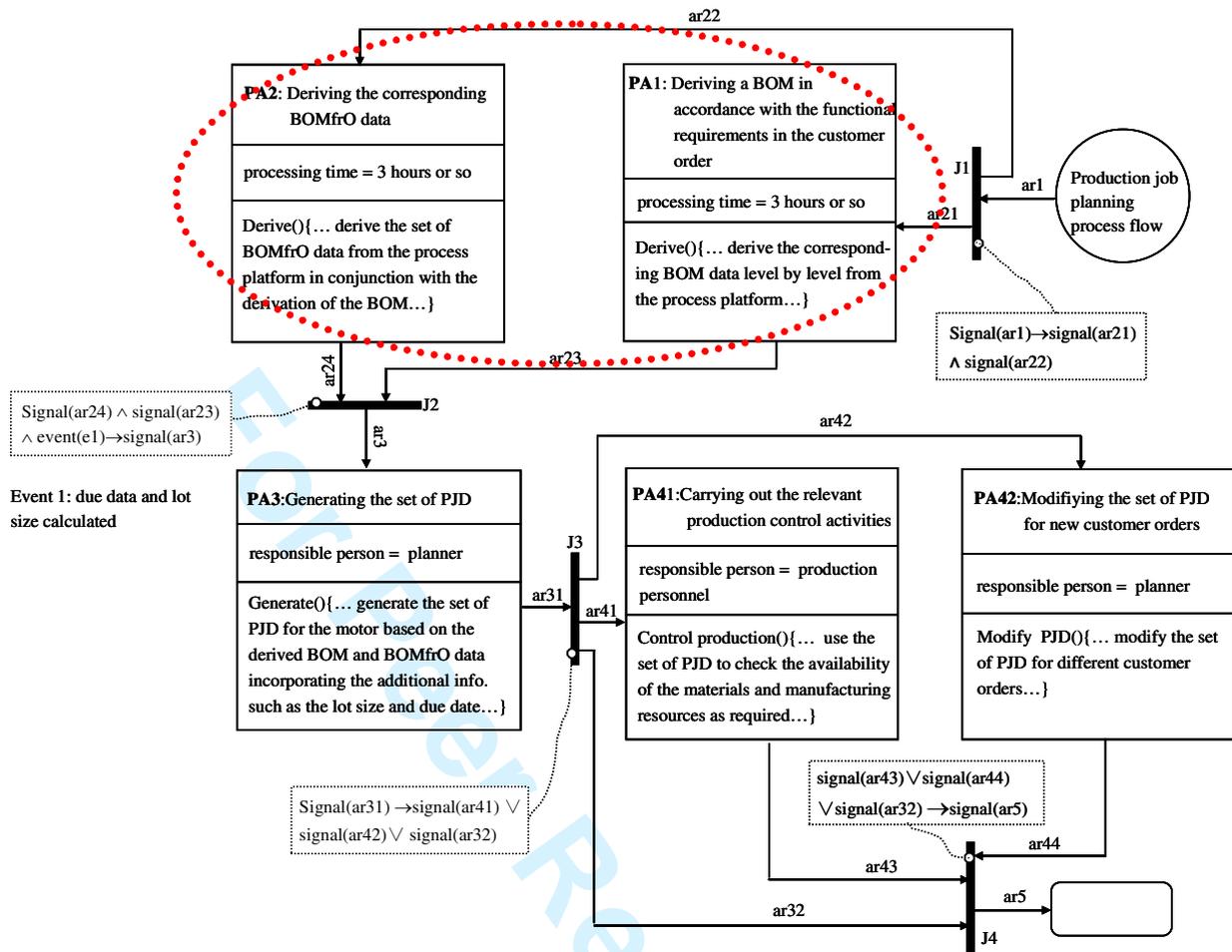


Figure 12: Planning production jobs for the customer order in Table 1

Table 1: A customer order

Order #:xxxxx	Customer Info:xxxxxxx	Volume: 50
Due date:xxxxx	Delivery:xxxxxxx	Description:xxxxxxxxxx
<u>Functional Requirements:</u>		
Rubber Holder. Color = Red		
Weight. Width = 5mm		
Shaft. Diameter = 3mm		
Shaft. Length = 16mm		
Frame. Thickness = 3mm		
Frame. Length = 20mm		
Magnet. Length = 14mm		
Bracket A. Shape = L		
Bracket A. Color = Red		
Bracket A. Width = 6mm		
Bracket B. Shape = L		
Bracket B. Color = Red		
Bracket B. Width = 6mm		
Terminal. length = 6mm		
Terminal. Width = 5mm		
Tape. Color = Red		
Tape. Width = 3mm		
Commutator. Thickness = 2mm		

Table 2: The motor specified in the customer order in Table 1

Hierarchy Level	Generic Item	Parameter	Value	Parameter Value	Quantity per
1	Rubber holder	Color	Red	C_R	1
1	Weight	Width	5mm	W_5	1
1	Mainbody				1
.2	Armature Assy				1
.2	Frame Assy				1
.2	Bracket Assy				1
..3	Shaft	Diameter	3mm	D_3	1
..3		Length	16mm	L_16	1
..3	Coil Assy				1
..3	Frame	Thickness	3mm	T_3	1
..3		Length	20mm	L_20	1
..3	Magnet	Length	14mm	L_14	1
..3	Bracket A	Shape	L	S_L	1
..3		Color	Red	C_R	1
..3		Width	6mm	W_6	1
..3	Bracket B	Shape	L	S_L	1
..3		Color	Red	C_R	1
..3		Width	6mm	W_6	1
..3	Terminal	Length	6mm	L_6	1
..3		Width	5mm	W_5	1
...4	Coil	Diameter	1mm	D_1	1
...4	Tape	Width	3mm	W_3	1
...4	Commutator	Thickness	2mm	T_2	1

Table 3: The PJD for producing the motor in Table 2

Job no.	Sequence no.	Operation	Work center	Machine	Tool	Fixture	Labor No./wc	Cycle time (Sec./item x Lot Size)	Component/Material (Quantity)	Product (Lot size)	Component job no.
20	50	Vibration motor assembly	WC-A <sub>vm</sub>	WcaulkingMcI	.....	WwsittingjigI WcaulkingheadI	1	9.25 x 50	Weight (1x50) Rubber Holder (1x50) MainbodyAssy(1x50)	Vibration Motor (1x50)	N/A N/A 19
19	40	Mainbody assembly	WC-A <sub>mb</sub>	FcaulkingMcHS AinsertingMcV	.....	Caulking bladeII BracketholderL	1	9.25 x 50	Armature Assy (1x50) Frame Assy (1x50) Bracket Assy (1x50)	Mainbody Assy (1x50)	18 15 13
18	30	Armature assembly	WC-A <sub>aa</sub>	Sinserting and Soldering Mc.	.....	Supporting holderI PalletII	1	5.00 x 50	Coil Assy (1x50) Shaft (1x50)	Armature Assy (1x50)	17
17	20	Coil assembly	WC-A <sub>ca</sub>	.....	.....	GuidingjigIII	1	5.00 x 50	Coil (1x50) Tape (1x50) Commuter (1x50)	Coil Assy (1x50)	16
16	30	Coil fabrication	WC-M <sub>c</sub>	Cwinding McV5	.....	TrayII	1	4.50 x 50	Raw Material (1set x50)	Coil (1x50)	N/A
15	20	Frame assembly	WC-A <sub>fa</sub>	FMpressing Mc.	.....	FholderI MholderVII	1	4.50 x 50	Magnet (1x50) Frame (1x50)	Frame Assy (1x50)	14
14	10	Frame fabrication	WC-M <sub>f</sub>	FstampingMcI	Die	Holder01	1	5.00 x 50	Raw Material (1set x50)	Frame (1x50)	N/A
13	20	Bracket assembly	WC-A <sub>ba</sub>	Bfusing McII	BinsertorV	Bpressing jigAI	1	4.00 x 50	Bracket A (1x50) Bracket B (1x50) Terminal (1x50)	Bracket Assy (1x50)	12
12	10	Bracket A fabrication	WC-M <sub>ba</sub>	BinjectionMc04	BadjustorA	BlocatorA	1	5.25 x 50		Bracket A (1x50)	
	10	Bracket B fabrication	WC-M <sub>bb</sub>	BinjectionMc04	BadjustorA	Bprealignment jigI	1	5.25 x 50	Raw Material (1set x50)	Bracket B (1x50)	N/A
	10	Terminal fabrication	WC-M <sub>t</sub>	TcuttingMcI	DieII	TholderIII	1	5.00 x 50		Terminal (1x50)	