16-20 AUGUST 2021, GOTHENBURG, SWEDEN

MAINTAINING CONSISTENCY ACROSS DESIGN DESCRIPTIONS IN ENGINEERING PRODUCT DEVELOPMENT

McKay, Alison;
Rice, Hugh P.;
Chau, Hau Hing;
de Pennington, Alan

University of Leeds

ABSTRACT

The success of engineering product development depends on the effective communication of design descriptions in formats that suit the needs and capabilities of all stakeholders involved in the delivery to market and through-life support of products. Configuration management is a core design process to ensure the consistency of the technical data package, i.e., the collection of design descriptions needed to support the development, manufacture and operation of a given product. Bills of Materials (BoMs) are critical parts of the technical data package because they act as integrators: adapting detailed design descriptions to suit the needs of particular downstream processes. The ability to reconfigure BoMs while maintaining internal consistency of the technical data package (where all BoM configurations are complete and compatible with each other) is a major challenge. In this paper, we introduce research exploring computational tools that could support engineers in manipulating BoMs while also maintaining the internal consistency of the technical data package.

Keywords: Product architecture, Design informatics, Information management, Product structuring

Contact:
McKay, Alison
University of Leeds
Mechanical Engineering
United Kingdom
a.mckay@leeds.ac.uk

Cite this article: McKay, A., Rice, H. P., Chau, H. H., de Pennington, A. (2021) ‘Maintaining Consistency across Design Descriptions in Engineering Product Development’, in *Proceedings of the International Conference on Engineering Design (ICED21),* Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.460

1 INTRODUCTION

The success of engineering product development projects, measured in terms of time to market, cost and product quality, depends on the efficient and effective communication of design descriptions in formats that suit the requirements and capabilities of the wide range of engineering functions, processes and organisations involved in the delivery of products to markets and their subsequent through-life support. For complex, long-lived, engineered products, the technical data package brings together the collection of design descriptions that form the product definition. Given the range of systems, people and organisations involved in the development of such products, these descriptions are inevitably heterogeneous, for example, in terms of their purposes, intended users, information content, and meta-data. Many of these design descriptions have embedded within them one or more product structures that specify relationships between parts of the product. For example, geometric models typically include two types of product structure: one a network structure of geometric constraints that specify physical relationships between parts, and the other a hierarchical structure of part-whole relationships between component parts and the sub-assemblies into which they are grouped for particular purposes. These hierarchical structures are a form of Bill of Materials (BoM) and are critical parts of the technical data package because they act as integrators: adapting detailed design descriptions to suit the needs of particular downstream processes.

In this paper we report progress on the establishment of a new generation of engineering design tools that support the reconfiguration of BoMs while maintaining the internal consistency of the technical data package (where all BoM configurations are complete and compatible with each other). We begin, in Section 2, with a review of literature on computer support for configuration management processes, theoretical foundations for the definition of BoMs and their transformation to support downstream processes. This is followed, in Sections 3 and 4, with an overview of our research methodology and the case study used in this paper. Results of applying a software prototype to the case study are presented in Section 5 followed by a discussion of results and areas for further work.

2 BACKGROUND

Product data and lifecycle management systems (from here on collectively referred to as PLM systems) support configuration management processes. We begin this section, in 2.1, with a review of literature on configuration management and supporting PLM systems. One or more BoMs underpin PLM system functionality and the workflows that they use to deliver their core functionality, namely, to ensure that all members of a product development team gain access to the right information at the right time. In Section 2.2, we review theoretical foundations for the definition of BoMs and, given the focus of this paper, in Section 2.3 we review literature of the transformation of BoMs.

2.1 Configuration management in engineering design

Configuration management processes typically serve to support three purposes with an overarching goal of ensuring the conformity of design descriptions through the entire product lifecycle.

- Managing change: where the goal is to ensure that the correct versions of the parts of a product are combined to form the final design (Jarratt et al. 2011);
- Managing product variety: where the goal is to ensure that the correct variants of the parts of a product family are combined to form a final product that reflects the options and features selected by a customer (Johannesson et al. 2017; Raja and Isaksson 2019); and
- Enhancing downstream processes: where the goal is to provide design descriptions tailored to support the needs of specific lifecycle processes such as manufacturing or maintenance and repair (Stonebraker 1996; Zhou et al. 2018; McKay et al. 2019).

In practice, all three functionalities are needed because all designs (variant or not) are subject to change and downstream processes benefit from design descriptions that are structured to suit the task in hand and whose information content is minimal, complete and correct for the intended purpose. All three functionalities are supported, to some extent, by current PLM systems (Singh et al., 2020). However, despite its importance, the discipline of engineering configuration management has emerged primarily from developments in PLM systems which initially addressed issues in the management of the proliferation of design descriptions arising from the introduction of Computer Aided Engineering technologies from the 1980s (McMahon 2016). The focus of this paper is on the third of these purposes, i.e., the reconfiguration of BoMs to tailor design descriptions for particular purposes.

2.2 The description of BoMs

BoMs are core elements of engineering design and PLM systems (Kashkoush and ElMaraghy 2016). Current approaches to the definition of BoMs tend to use proprietary formats in CAD and associated PLM systems. Typically, these are associated with part geometries and interface formats such as STEP, where BoMs are, in essence, defined as collections of product definitions and relationships between them. More standalone BoM management systems, without links to specific CAD packages, such as Bitzlist (www.bitzlist.com), Kimonex (kimonex.com) and CIIVA (ciiva.com), tend to use tabular formats to represent BoMs. The primary purpose of these systems is to support change management and part tracking processes. These are two aspects of wider design configuration processes which involve working with multiple BoMs that support a range of lifecycle activities. For example, the torch case study used in this paper (see Section 4) includes a BoM that supports assembly processes, where the axis for assembly is perpendicular to the central axis of the torch, and another for a bulb change process, where the axis for disassembly is along the central axis of the torch. In this way, the primary goal for the management of BoMs in the context of engineering change is to maintain the consistency of the BoMs in a given technical data package. Elements of the first BoM in the life of a product surface early in the design process when design requirements are defined and key parts identified (McKay et al. 2001; Park and Kremer 2019). These early design decisions inform the development of product structures in CAD models and so multiple BoMs are created. BoM reconciliation is the process where BoMs related to a given product are aligned with each other (Kashkoush and ElMaraghy 2014). One way to achieve this is through a process where an initial BoM is transformed into another. For example, Zhou et al. (2018) provide an approach for transforming an as-designed engineering BoM into a service BoM. However, as a design develops, and suppliers design key parts and sub-systems, it becomes less feasible to derive all BoMs from a common source. In addition, definitions of new BoMs tend to come in a range of, often proprietary, CAD and other formats. Simons (2000) specifies properties of BoMs that apply regardless of the format of their definitions. These are summarised in Table 1. In this paper we explore how these properties may be exploited in design configuration tools intended to support the reconciliation of multiple BoMs.

Table 1: Properties of Simons (2000) & Dement et al.'s (2001) theory of parts and wholes

Property	**Definition**
Proper part	*a* is a proper part of *b* iff a is part of *b* and *b* is not part of *a*
Overlap	*a* overlaps *b* if some *c* is part of *a* and *b*
Reflexivity	If *a* exists then *a* is part of itself
Existence	If *a* is part of *b* then *a* and *b* exist
Transitivity	If *a* is part of *b* and *b* is part of *c* then *a* is part of *c*
Supplementarity	If *a* is part of *b* and *b* is not part of *a* then some *c* is a proper part of *b* and does not overlap *a*

BoM reconciliation can be regarded as a form of data integration and, therefore, can learn from previous work on database schema integration. Batini et al. (1986) provide a four stage integration process that here is applied to BoMs.

1. Pre-integration: BoMs to be reconciled are identified and a reconciliation strategy selected.
2. Comparison of BoMs: BoMs are analysed to identify conflicts, similarities and interrelationships.
3. Conforming: conflicts are resolved and interrelationships between BoMs elaborated.
4. Merging and Restructuring: a single structure, on which all BoMs can be superimposed, is created.

The goal of Batini et al. (1986) is a single schema that is minimal, complete and understandable. In this research the goal, rather than establishing a single common BoM that, inevitably, cannot be minimal for all users, is to establish an underlying model, for us a Boolean lattice, into which all BoMs can be embedded. The focus of this paper lies in the third stage of Batini et al. (1986) where, in line with Zhou et al. (2018), interrelationships between BoMs are characterised by transformations between them.

2.3 The transformation of BoMs

The transformation operations needed to restructure the BoMs within design descriptions for different purposes while also maintaining the validity of the technical data package and related BoMs are used here explore interrelationships between BoMs. Zhou et al. (2018) discuss operations needed to transform a design BoM into a service BoM and McKay et al. (2019) introduce an application of lattice theory to support BoM transformation processes in general. A third source, Dement et al. (2001), applies theories of part-whole relationships from Simons (2000) to wider configuration management processes. Figure 1 summarises BoM transformation operations available in the software prototype introduced later in this paper and is based on the number of tiers in a BoM structure that it operates on. For a single node in a BoM structure, the only available operation is *articulation* which splits a part into two or more parts. An example of this could be where a part is designed as a single part but then split into two halves, e.g., to support a manufacturing process. There are four operations that can be applied to a two-tier BoM structure. *Factorisation* combines sub-parts to form a single part (i.e., the reverse of *articulation*), *shuffling* moves sibling parts but does not change the part-whole structure, *consolidation* combines selected parts of the parent part and *deconsolidation* adds a new sub-part to the parent part. Finally, for three tier BoMs, *assembly* inserts a new sub-assembly that is formed from existing parts and *adoption* changes the parentage of parts. It can be seen that there is some overlap between these operations, e.g., *consolidation* can be implemented using a *factorisation* operation on selected sub-parts, but the need for this range of operations was established by experimenting with case study BoMs. These operations are compared with those of Zhou et al. (2018), Dement et al. (2001) and McKay et al. (2019) in Table 2.

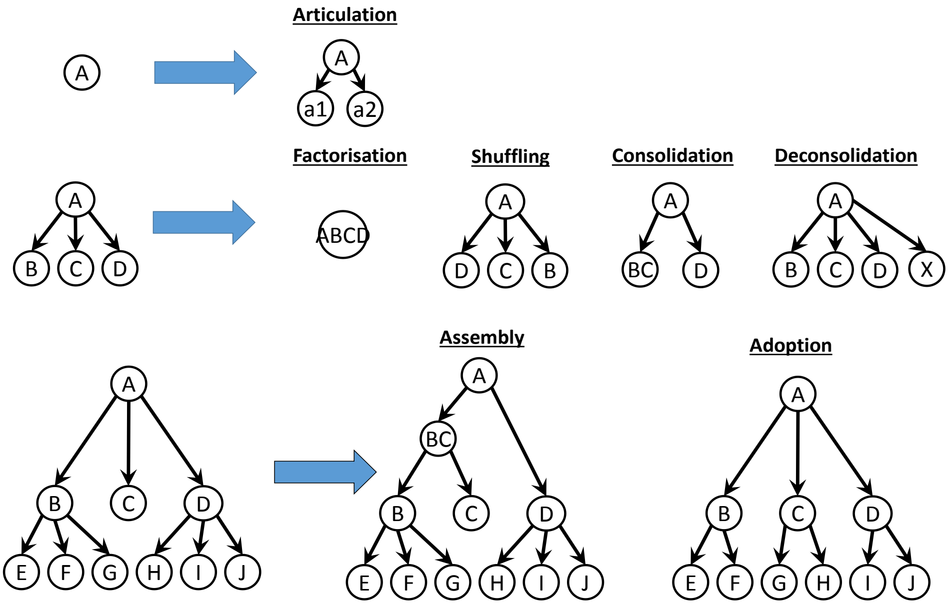


Figure 1: BoM transformation operations

3 APPROACH

The research used an Action Design Research methodology (Sein et al. 2011) that involves the parallel development of theory, design principles and a series of software prototypes that allow the theory and principles to be evaluated with users through case studies. In this section we introduce how we have implemented the BoM transformation operations in these prototypes. This is followed, in Section 4, with the introduction of a case study and, in Section 5, results of using the software prototype to define the BoMs and associated transformation operations from the case study are presented.

Key challenges in supporting the BoM transformation operations lie in (a) establishing what the minimal set of operations is, (b) representing them in a coherent manner, and (c) maintaining the underlying consistency of the technical data package as the new BoMs are created through the transformation process. These three requirements were met by defining the set of transformation operations for the BoM transformation operations given in Table 2 using the adjacency matrices shown in Table 3.

Table 2: Part-based operations in the transformation of BoMs (from BoMA to BoMB)

Operation	**Description**	**Operates on**	Zhou et al. (2018)	Dement et al. (2001)	McKay et al. (2019)
Articulation	A part in BoMA becomes multiple parts in BoMB	A single node, X	Position splitting parts	Articulation: separation of parts	Disaggregation
Factorisation	A part and its sub-parts in BoMA become one part in BoMB (see note 1)	A node, X, and, or in the context of, its sub-tier	Not directly supported	Factorisation OR Consolidation (see Note 2)	Collapse
Shuffling	Reorders siblings on a tier (see Note 3)		Not applicable	Ordering	Drag to new position w.r.t. siblings
Consolidation	A collection of parts in BoMA become one part in BoMB		Not directly supported	Factorisation OR Consolidation (see Note 2)	Collapse
De-consolidation	A part not in BoMA occurs in BoMB (see Note 4)		Addition of sub-parts	Deconsolidation	Disaggregation
Assembly	Creates a new sub-assembly of sub-parts	A node, X, and two tiers above it	Not directly supported	Factorisation OR Consolidation (see Note 2)	Collapse
Adoption	Assigns a new parent to a sub-part		Not directly supported	Not applicable	Adopt
Remove part	A part in BoMA does not appear in BoMB	A node, X, and, or in the context of, its sub-tier	Irrelevant parts deletion	Not applicable	Delete

NOTES
1. In Zhou et al. (2018), this may be achieved by deleting irrelevant parts from an assembly.
2. Factorisation & consolidation have the same structure in Dement et al. (2001) but different meaning
3. The ordering of siblings in a BoM is presentational and has no formal meaning. However, the underlying matrix needs to reflect this to preserve the changed order.
4. Embedding a new BoM in a lattice can do this for assembly parts but not components.

4 CASE STUDY

Figure 2 shows the torch case study used. It includes three BoMs. The one in Figure 2(a) is the torch treated as an individual item, e.g., as it might be supplied to a consumer. Figure 2(b) and (c) show two alternative breakdown structures that typically occur: an assembly BoM for use in manufacturing and a disassembly BoM for changing the bulb. In a product development process, there is a need to be able to transform across such BoMs, for example, from the assembly BoM for use in production to the disassembly BoM that could underpin user instructions on how to change the bulb. The transformation process from the assembly BoM to the disassembly BoM is relatively simple but illustrates how the adjacency matrices are used. In the software, making this transformation is a three stage process: (1) Create new *torch body* assembly; (2) Add l*ower body*, *upper body* and *fasteners* to *torch body* assembly; (3) Aggregate *torch body*.

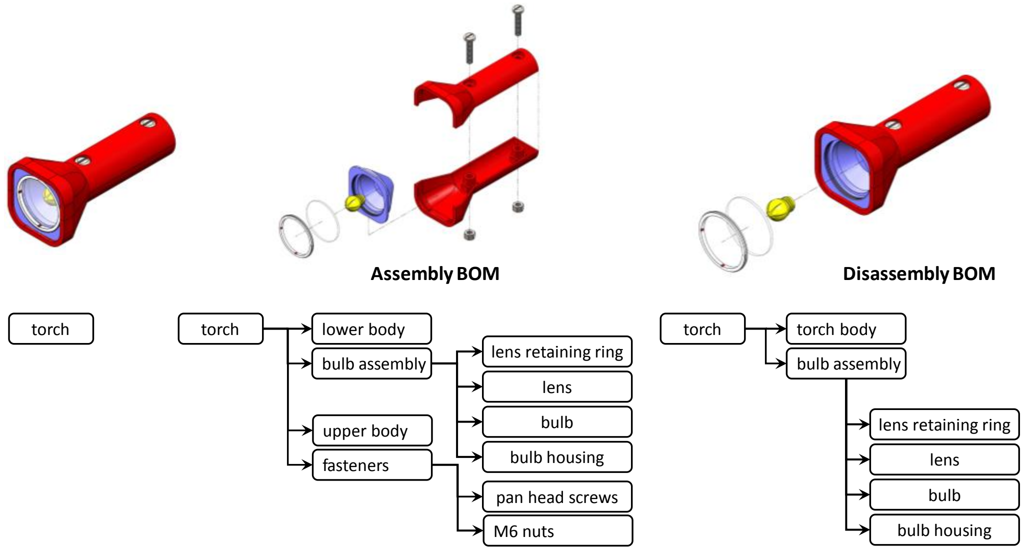
5 RESULTS

The research has resulted in a software prototype that allows users to define and work with multiple BoMs. A screenshot of the prototype is shown in Figure 3. Key areas of the interface are labelled and described in more detail here. The user imports a BoM from a STEP (AP214) file. The BoM is displayed in three views: (a) the parts view in the form of an indented list; (b) the 3D view as rendered CAD objects; and (c) the lattice view as a Hasse diagram. The user can select an item (i.e. a part or sub-assembly) in any of the views, whereupon it is highlighted in every other view; any item whose checkbox is activated in the parts view is also represented as a toggle button in the selector view (d), which provides another method for selection. The user can then modify the BoM structure using the operations specified in Table 3.

Table 3: BoM transformation operations in prototype represented in adjacency matrices (Note: In the matrices, 1 means column header is parent to the part in the row)

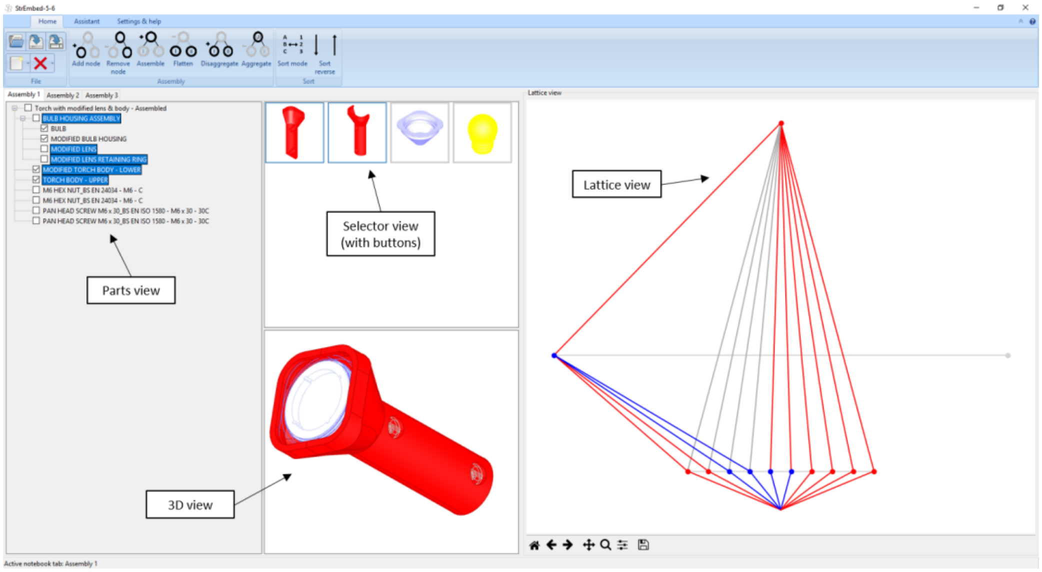
Operation		**Operation**	
Multiple parts in BoMA become one part in BoMB	$\left[\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 0 & 0 & 0 \\ B & 1 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 \\ D & 1 & 0 & 0 & 0 \end{array}\right] \rightarrow [ABCD]$	A part in BoMA becomes multiple parts in BoMB	$[\mathrm{A}] \rightarrow \left[\begin{array}{cccc} & \mathrm{A} & \mathrm{a1} & \mathrm{a2} \\ \mathrm{A} & 0 & 0 & 0 \\ \mathrm{a1} & 1 & 0 & 0 \\ \mathrm{a2} & 1 & 0 & 0 \end{array}\right]$
Shuffling siblings	$\left[\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 0 & 0 & 0 \\ B & 1 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 \\ D & 1 & 0 & 0 & 0 \end{array}\right] \rightarrow \left[\begin{array}{ccccc} & A & D & C & B \\ A & 0 & 0 & 0 & 0 \\ D & 1 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 \\ B & 1 & 0 & 0 & 0 \end{array}\right]$	New sub-assembly	$\left[\begin{array}{cccc} & A & B & C \\ A & 0 & 0 & 0 \\ B & 1 & 0 & 0 \\ C & 1 & 0 & 0 \end{array}\right] \rightarrow \left[\begin{array}{ccccc} & A & BC & B & C \\ A & 0 & 0 & 0 & 0 \\ BC & 1 & 0 & 0 & 0 \\ B & 0 & 1 & 0 & 0 \\ C & 0 & 1 & 0 & 0 \end{array}\right]$
A part in BoMA does not appear in BoMB	$\left[\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 0 & 0 & 0 \\ B & 1 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 \\ D & 1 & 0 & 0 & 0 \end{array}\right] \rightarrow \left[\begin{array}{cccc} & A & B & D \\ A & 0 & 0 & 0 \\ B & 1 & 0 & 0 \\ D & 1 & 0 & 0 \end{array}\right]$	Adoption (shown here only for Part G)	$\left[\begin{array}{ccccc} & A & B & C & G \\ A & 0 & 0 & 0 & 0 \\ B & 1 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 \\ G & 0 & 1 & 0 & 0 \end{array}\right] \rightarrow \left[\begin{array}{ccccc} & A & B & C & G \\ A & 0 & 0 & 0 & 0 \\ B & 1 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 \\ G & 0 & 0 & 1 & 0 \end{array}\right]$
A part not in BoMA occurs in BoMB	$\left[\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 0 & 0 & 0 \\ B & 1 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 \\ D & 1 & 0 & 0 & 0 \end{array}\right] \rightarrow \left[\begin{array}{cccccc} & A & B & C & D & X \\ A & 0 & 0 & 0 & 0 & 0 \\ B & 1 & 0 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 & 0 \\ D & 1 & 0 & 0 & 0 & 0 \\ X & 1 & 0 & 0 & 0 & 0 \end{array}\right]$		

As the transformation from assembly to disassembly BoM used in Section 4 requires only one operation, to collect all parts not involved in the bulb change process into a single part, in this next section we use a more complex transformation from the assembly BoM to a production BoM. This involves three operations and better illustrates the operation of the software prototype. The new BoM, in which parts are grouped by whether they are bought-out or not and by manufacturing process, results from flattening the assembly BoM and then creating sub-assemblies of parts. It should be noted that, although represented as assemblies in the BoM, these are merely groupings of parts for particular purposes where there is no need for the parts to be physical assemblies. There are many examples of this in practice, e.g., in the definition of kits of parts for assembly or maintenance processes. The key point to note from Figure 5 is that the parts list and lattice views change with each transformation operation.



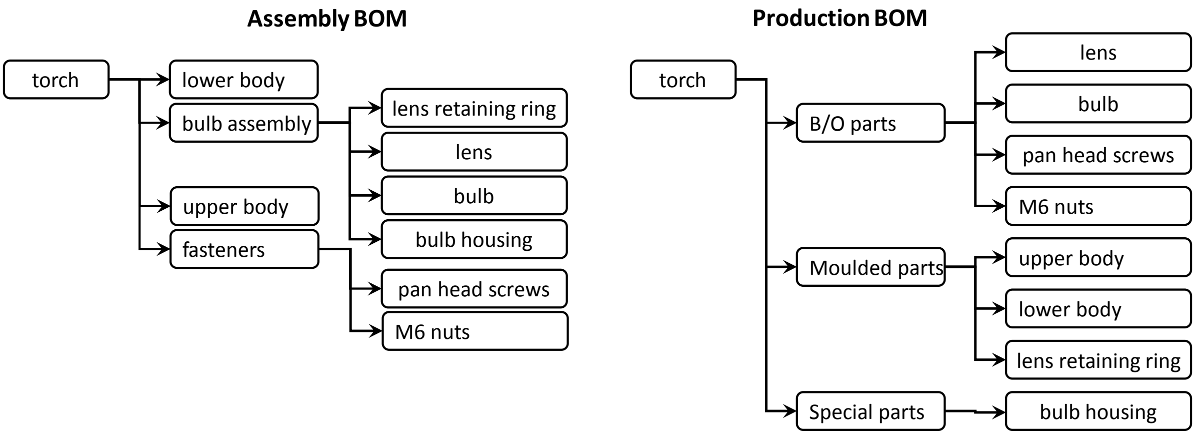
(a) (b) (c)

Figure 2: Case study (NOTE: the positions of the parts in the exploded assemblies are derived from mating conditions in the CAD model and are not reflected in the BoM). (Software is available from https://github.com/paddy-r/StrEmbed-5-5/)



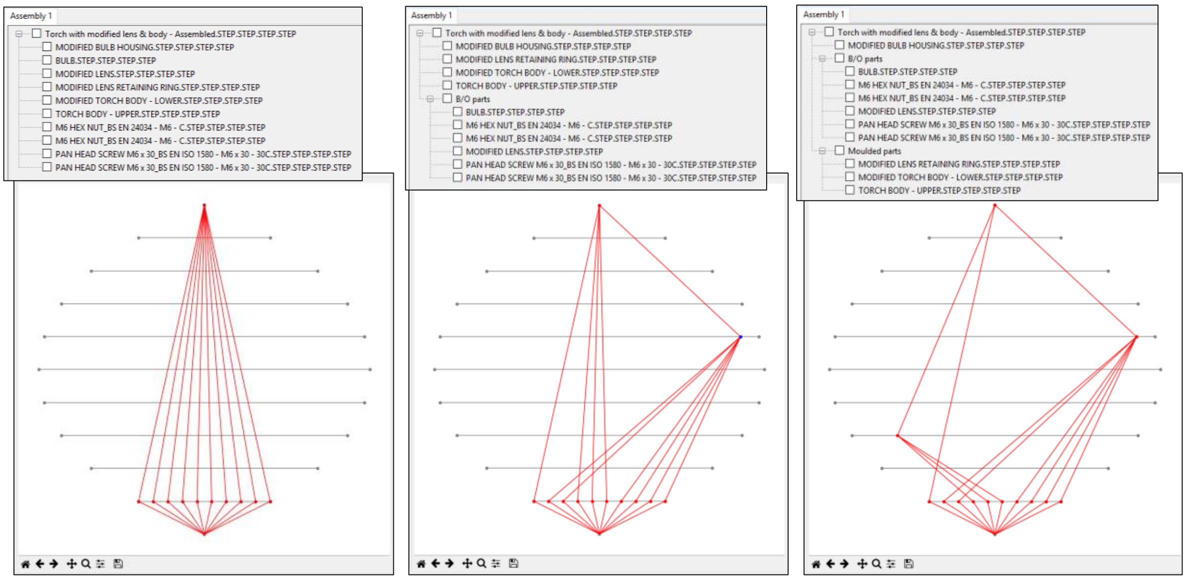
☐ Torch with modified lens & body - Assembled
☐ BULB HOUSING ASSEMBLY
☑ BULB
☑ MODIFIED BULB HOUSING
☐ MODIFIED LENS
☐ MODIFIED LENS RETAINING RING
☑ MODIFIED TORCH BODY - LOWER
☑ TORCH BODY - UPPER
☐ M6 HEX NUT_BS EN 24034 - M6 - C
☐ M6 HEX NUT_BS EN 24034 - M6 - C
☐ PAN HEAD SCREW M6 x 30_BS EN ISO 1580 - M6 x 30 - 30C
☐ PAN HEAD SCREW M6 x 30_BS EN ISO 1580 - M6 x 30 - 30C

Figure 3: BoM editor user interface



(a) (b)

Figure 4: Assembly to production


Assembly 1

- ☐ Torch with modified lens & body - Assembled.STEP.STEP.STEP.STEP
  - ☐ MODIFIED BULB HOUSING.STEP.STEP.STEP.STEP
  - ☐ BULB.STEP.STEP.STEP.STEP
  - ☐ MODIFIED LENS.STEP.STEP.STEP.STEP
  - ☐ MODIFIED LENS RETAINING RING.STEP.STEP.STEP.STEP
  - ☐ MODIFIED TORCH BODY - LOWER.STEP.STEP.STEP.STEP
  - ☐ TORCH BODY - UPPER.STEP.STEP.STEP.STEP
  - ☐ M6 HEX NUT_BS EN 24034 - M6 - C.STEP.STEP.STEP.STEP
  - ☐ M6 HEX NUT_BS EN 24034 - M6 - C.STEP.STEP.STEP.STEP
  - ☐ PAN HEAD SCREW M6 x 30_BS EN ISO 1580 - M6 x 30 - 30C.STEP.STEP.STEP.STEP
  - ☐ PAN HEAD SCREW M6 x 30_BS EN ISO 1580 - M6 x 30 - 30C.STEP.STEP.STEP.STEP

Flatten

Assembly 1

- ☐ Torch with modified lens & body - Assembled.STEP.STEP.STEP.STEP
  - ☐ MODIFIED BULB HOUSING.STEP.STEP.STEP.STEP
  - ☐ MODIFIED LENS RETAINING RING.STEP.STEP.STEP.STEP
  - ☐ MODIFIED TORCH BODY - LOWER.STEP.STEP.STEP.STEP
  - ☐ TORCH BODY - UPPER.STEP.STEP.STEP.STEP
  - ☐ B/O parts
    - ☐ BULB.STEP.STEP.STEP.STEP
    - ☐ M6 HEX NUT_BS EN 24034 - M6 - C.STEP.STEP.STEP.STEP
    - ☐ M6 HEX NUT_BS EN 24034 - M6 - C.STEP.STEP.STEP.STEP
    - ☐ MODIFIED LENS.STEP.STEP.STEP.STEP
    - ☐ PAN HEAD SCREW M6 x 30_BS EN ISO 1580 - M6 x 30 - 30C.STEP.STEP.STEP.STEP
    - ☐ PAN HEAD SCREW M6 x 30_BS EN ISO 1580 - M6 x 30 - 30C.STEP.STEP.STEP.STEP

Add B/O parts

Assembly 1

- ☐ Torch with modified lens & body - Assembled.STEP.STEP.STEP.STEP
  - ☐ MODIFIED BULB HOUSING.STEP.STEP.STEP.STEP
  - ☐ B/O parts
    - ☐ BULB.STEP.STEP.STEP.STEP
    - ☐ M6 HEX NUT_BS EN 24034 - M6 - C.STEP.STEP.STEP.STEP
    - ☐ M6 HEX NUT_BS EN 24034 - M6 - C.STEP.STEP.STEP.STEP
    - ☐ MODIFIED LENS.STEP.STEP.STEP.STEP
    - ☐ PAN HEAD SCREW M6 x 30_BS EN ISO 1580 - M6 x 30 - 30C.STEP.STEP.STEP.STEP
    - ☐ PAN HEAD SCREW M6 x 30_BS EN ISO 1580 - M6 x 30 - 30C.STEP.STEP.STEP.STEP
  - ☐ Moulded parts
    - ☐ MODIFIED LENS RETAINING RING.STEP.STEP.STEP.STEP
    - ☐ MODIFIED TORCH BODY - LOWER.STEP.STEP.STEP.STEP
    - ☐ TORCH BODY - UPPER.STEP.STEP.STEP.STEP

Add moulded part

Figure 5: Transformation from assembly BoM to production BoM

6 DISCUSSION AND FURTHER WORK

Key challenges in ensuring consistency across BoMs lie in (a) establishing the set of operations needed to specify relationships between BoMs, (b) representing these relationships systematically, and (c) maintaining the underlying consistency of the technical data package as new BoMs are defined. Here we report progress on (a) and propose a collection of BoM transformation operations implemented in a software prototype using adjacency matrices. This prototype provides an example of a future design tool that supports the reconfiguration of BoMs by enabling the definition of multiple BoMs, for different purposes, in terms of a common underlying structure. This capability allows the impacts of proposed changes to one BoM on other BoMs, and so associated processes, for the same product to be visualised. The properties of part-whole structures in Table 1 provide the formalism for the BoM transformation operations available in the prototype. Experiments with case studies give us confidence that the proposed set of operations is necessary but further work is needed to confirm that it is complete, correct and minimal. With respect to (b), we define each BoM as a Boolean lattice and transformation operations that specify relationships between pairs of lattices are defined using adjacency matrices. Together, these form foundations on which (c) can be achieved through tools that enable the maintenance of the underlying consistency of the BoMs in a technical data package. In this section we consider three areas for further work that have arisen: issues in the use of the lattice representation, improving understanding of the kinds of product structure that need to be supported and interactions between BoM transformation and shape models that describe the geometries of and physical relationships between parts.

With respect to the use of the Boolean lattice representation, in producing the software prototype we have identified a number of technical issues that need to be addressed. Some are related to the behaviours of particular operations. For example, in the current version of the software, flattening a BoM removes all structure and transforms the BoM into a flat list of component parts. This could be addressed by incorporating parameters in the software but further work is needed to understand the implications of decisions made when transforming BoMs. Similarly, it is possible within StrEmbed to “delete” parts but the semantics of “deletion”, and consequences for the underlying lattice, need further consideration.

Other issues relate to the use of Boolean lattices to represent BoMs. In the vast majority of cases we have found them to be adequate but there are some exceptions. For example, in Figure 4, although the final operation can be completed (creating ‘special parts’) because the lattice is implemented as a general graph, formally, the underlying lattice requires each node to have at least two parts. Although this is a fictitious example, there are real-world scenarios where we can see this becoming problematic. For example, in some BoMs the same part is used multiple times and the number of parts in the assembly is captured through a quantity attribute on the part-whole relationship rather than through the explicit representation of parts. In other examples it would be necessary to revert to the underlying theories on the properties of BoMs. For example, one could argue that the node called ‘special parts’ is a category of parts rather than a part, which leads to questions surrounding what is and is not a node in a BoM.

While BoMs are important tools for engineers, they are just one kind of product structure. BoMs are hierarchical structures where (directed) part-whole relationships bring together parts to form assemblies or other groupings of parts. Many other kinds of product structure exist that are not hierarchical and where the connections between parts are not part-whole relationships. For example, Park and Kremer’s (2019) topological characterisation of product structures uses undirected relationships. Structures such as those formed from assembly mating relationships are also important. For example, history data from CAD systems and constraints that define mating conditions between parts can capture design rationale that is lost when a CAD model is translated into a STEP file. This is particularly problematic for the articulation operation because a geometric modeller is needed to identify which parts of the shape of the initial part are associated with each of the new parts. However, BoM edits can also affect wider aspects of geometric models because the hierarchical structure of the BoM is often used in CAD systems to ensure the geometric integrity of assembly relationships. For example, CAD systems typically use a hierarchical part-whole structure to enforce the underlying assumption that shapes are of rigid parts and so parts cannot be related to themselves through assembly mating conditions. Some BoM operations, e.g., adoption, explicitly change the hierarchy and so have the potential to invalidate shape models. This could be addressed by closer integration between BoM editing and CAD systems.

We present a software prototype where multiple BoMs for a given design are embedded into a shared lattice. The creation of such a lattice involves mapping common parts across multiple BoMs, i.e., a schema matching problem that, in general, is unresolved. Current solutions depend strongly on the particular application (Rahm and Bernstein 2001; Christen 2012) and so, we focus on an application-specific and user-led process where an aggregate similarity score between parts is determined from: (a) part data (i.e., names) extracted from the STEP file(s), using a Levenshtein distance-based metric (Elmagarmid et al. 2007); and (b) local assembly structure, using heuristics from graph theory (Wills and Meyer 2020). Although not a solution to the general problem, the development of and experimentation with this prototype has allowed us to identify the necessary properties of the underlying lattice representation that is key to ensuring the integrity of a technical data package.

Our long-term goal is to inform the design and development of future configuration management methods and tools, for use by practising engineers, and so improve configuration management processes. A key area for further work lies in improving understanding of how BoMs are used in engineering practice and so requirements for tools that support the reconciliation of BoMs. Preliminary work in this area (through interviews with engineering leaders at four small-medium manufacturing enterprises and three engineers at a FTSE100 manufacturing company) identified BoMs as being critical elements of technical communication. In line with researchers such as Kashkoush and ElMaraghy (2014), limitations in current IT support for the configuration of BoMs were identified. In particular, the practitioners noted that, like shape models, BoMs typically exist in multiple systems/formats, such as PLM systems for design data, ERP systems for manufacturing and asset management systems when products are in use. Although each system has bespoke functionalities, maintaining consistency across multiple systems and data formats was seen as critical in reducing rework and improving product development processes.

This paper relates to the consistency of BoMs that describe a given version of a given design for different purposes. Further work is needed to explore the applicability of the work to wider configuration management processes. For product families, typically defined through so-called “generic BoMs”, an important distinction lies in the fact that elements of such BoMs are placeholders for parts whose presence and specification are determined from customer and other configuration options. Given that these elements are not parts, the underlying representation scheme of the Boolean lattice needs to be reviewed and further configuration operations defined. For the management of change, the presence of the parent-child relationships offers some promise in that they provide a context for parts and so any changes that are applied. However, again, the underlying lattice representation would need adjustment because, although BoMs before and after a change could be represented using the lattice representation, relationships between lattices and meta-data related to change processes would also need to be captured.

ACKNOWLEDGEMENTS

This research is funded by the UK EPSRC (Grant: EP/S016406/1, "Assuring the quality of design descriptions through the use of design configuration spaces"). Peter Hayward produced the CAD models of the torch case study. Our collaboration with Ian Bentley and Richard Baker (Rolls-Royce plc) on requirements for engineering configuration management systems has been invaluable.

REFERENCES

Batini, C., Lenzerini, M. and Navathe, S.B. (1986) 'A comparative analysis of methodologies for database schema integration', ACM computing surveys (CSUR), 18(4), 323-364, http://dx.doi.org/10.1145/27633.27634.

Christen, P. (2012) 'A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication', IEEE Transactions on Knowledge and Data Engineering, 24(9), 1537-1555, http://dx.doi.org/10.1109/TKDE.2011.127.

Dement, C.W., Mairet, C.E., DeWitt, S.E. and Slusser, R.W. (2001) MEREOS, Final Report (AFRL-ML-WP-TR-2001-4116), Materials and Manufacturing Directorate, 45433–47750, Marietta, GA: Air Force Research Laboratory, Air Force Materiel Command, Wright-Patterson Air Force Base, OH.

Elmagarmid, A.K., Ipeirotis, P.G. and Verykios, V.S. (2007) 'Duplicate Record Detection: A Survey', IEEE Transactions on Knowledge and Data Engineering, 19(1), 1-16, http://dx.doi.org/10.1109/TKDE.2007.250581.

Jarratt, T.A.W., Eckert, C.M., Caldwell, N.H.M. and Clarkson, P.J. (2011) 'Engineering change: an overview and perspective on the literature', Research in Engineering Design, 22(2), 103-124, http://dx.doi.org/10.1007/s00163-010-0097-y.

Johannesson, H., Landahl, J., Levandowski, C. and Raudberget, D. (2017) 'Development of product platforms: Theory and methodology', Concurrent Engineering-Research and Applications, 25(3), 195-211, http://dx.doi.org/10.1177/1063293x17709866.

Kashkoush, M. and ElMaraghy, H. (2014) 'Product design retrieval by matching bills of materials', Journal of Mechanical Design, 136(1), 011002, http://dx.doi.org/10.1115/1.4025489.

Kashkoush, M. and ElMaraghy, H. (2016) 'Product family formation by matching Bill-of-Materials trees', CIRP Journal of Manufacturing Science and Technology, 12, 1-13, http://dx.doi.org/10.1016/j.cirpj.2015.09.004.

McKay, A., Chau, H.H., Earl, C.F., Behera, A.K., de Pennington, A. and Hogg, D.C. (2019) 'A lattice-based approach for navigating design configuration spaces', Advanced Engineering Informatics, 42, 100928, http://dx.doi.org/10.1016/j.aei.2019.100928.

McKay, A., de Pennington, A. and Baxter, J. (2001) 'Requirements management: a representation scheme for product specifications', Computer-Aided Design, 33(7), 511-520, http://dx.doi.org/10.1016/S0010-4485(01)00050-1.

McMahon, C. (2016) 'Design informatics: supporting engineering design processes with information technology', Journal of the Indian Institute of Science, 95(4), 365-378.

Park, K. and Kremer, G.E.O. (2019) 'Topological Characterization of an Evolving Product Structure Network: A Case Study of Generational Smartphone Products', Journal of Mechanical Design, 141(9), http://dx.doi.org/10.1115/1.4042844.

Rahm, E. and Bernstein, P.A. (2001) 'A survey of approaches to automatic schema matching', The VLDB Journal, 10(4), 334-350, http://dx.doi.org/10.1007/s007780100057.

Raja, V. and Isaksson, O. (2019) 'Modelling-Integrated Product Architectures: An Aero Engine Component Example' in Chakrabarti, A., ed., Research into Design for a Connected World, Singapore: Springer Singapore, 847-858, http://dx.doi.org/10.1007/978-981-13-5974-3_73.

Sein, M.K., Henfridsson, O., Purao, S., Rossi, M. and Lindgren, R. (2011) 'Action design research', MIS quarterly, 37-56, http://dx.doi.org/10.2307/23043488.

Singh, S., Misra, S.C., Chan, F.T.S. (2020) 'Establishment of critical success factors for implementation of product lifecycle management systems', International Journal of Production Research, 58:4, 997-1016, https://dx.doi.org/10.1080/00207543.2019.1605227

Simons, P.M. (2000) Parts: A Study in Ontology, Oxford: Clarendon.

Stonebraker, P.W. (1996) 'Restructuring the bill of material for productivity: A strategic evaluation of product configuration', International Journal of Production Economics, 45(1-3), 251-260, http://dx.doi.org/10.1016/0925-5273(95)00147-6.

Wills, P. and Meyer, F.G. (2020) 'Metrics for graph comparison: A practitioner's guide', Plos One, 15(2), e0228728, http://dx.doi.org/10.1371/journal.pone.0228728.

Zheng, Y., Han, D., Ni, Y., Lu, M. (2014) 'Research and application of bottom-up route-based product data conformity inspection approach for civil aircraft'. International Journal of Computer Integrated Manufacturing. 27(6): p. 591-607. https://dx.doi.org/10.1080/0951192X.2013.834464

Zhou, C., Liu, X., Xue, F., Bo, H. and Li, K. (2018) 'Research on static service BOM transformation for complex products', Advanced Engineering Informatics, 36, 146-162, http://dx.doi.org/10.1016/j.aei.2018.02.008.