

A Product Data Management Integrated Product Configuration and Assembly Process

Planning Environment

J.X. Gao and N.W. Bowland

School of Industrial and Manufacturing Science

Cranfield University

Cranfield, Bedford, MK43 0AL

Email: james.gao@cranfield.ac.uk

Abstract

This paper reports part of the results of a three year project carried out at Cranfield University in collaboration with companies in the automotive and aerospace industry, aiming to develop new methodologies for Computer-Aided Assembly Process Planning (CAAPP). CAAPP is not a mature technology, despite the managerial and technical advantages offered by its implementation. This paper describes the limiting factors in industrial acceptance of CAAPP and discusses methodologies by which these factors can be eliminated. This is done primarily by the integration of CAAPP functionality with a Product Data Management (PDM) tool, providing a data control framework and a high-level data structure to form the basis of planning. The paper also details a system based on the described approach. The underlying object-oriented model developed to facilitate automated assembly process planning is described in detail. Data control methodologies are presented to aid retrieval, storage and creation. Methods for the definition of liaisons, mating features are discussed. The methodologies used for constraint identification, process sequence generation and sequence representation within the system are described. Multiple sequences are represented by means of a tree structure, similar to the AND/OR graph representation. The sequence is navigable and can be used downstream sequence evaluation and simulation or to generate scheduling data.

Keywords: Computer-aided assembly process planning, product data management, CAD/CAM integration

1. Introduction

1.1 Assembly planning problems

Despite considerable research effort over the past two decades, Computer-Aided Assembly Process Planning (CAAPP) has remained an immature technology. The authors suggest that this is in part due to certain management and technical barriers that are not addressed even by state-of-the-art solutions to the CAAPP problem. Firstly, there is little opportunity for upstream or downstream automation of clerical or technical tasks, such as data retrieval and plan rendition or dissemination. Previous assembly planning systems have been developed in isolation from other software tools. This results in a highly unusable and ineffective solution for the industrial user. Secondly, there is no easy, intuitive method to manage the complexity of the planning problem. It is recognised that the assembly planning problem is extremely complex. Therefore, simplifications must be made, whilst maintaining the usefulness of the system. The two most complex aspects of the assembly planning problem can be seen as the generation of fully general vector trajectories and the simultaneous generation of all possible product configurations. This system, like many others, considers only translations along and rotations about the positive and negative major axes. Unlike other systems, this one accepts a simplified product model, defined in an existing data management tool, providing a powerful and consistent method of product sub-division. Thirdly, CAAPP systems are too sequence driven in that they do not consider process selection, a mainstay of manufacturing process planning. A simple, chronological list of components is at too low a level to be of any use to industrial users. It does not facilitate downstream analysis of the assembly

tasks and, more importantly, does not allow process information to influence sequence generation. These barriers have proven to be limiting factors in the industrial acceptance of CAAPP as a mature technology. This paper describes a methodology for furthering the state-of-the-art in assembly planning through various measures to tackle these issues. The implementation of a CAAPP system, entitled MAPS (Manufacturing Assembly Process Sequencer), based on the developed methodology is also described.

1.2 Assembly planning principles

This section describes both the works upon which the developed methodology is based and contemporary developments from the field at large. The generation of feasible assembly sequences by the utilisation of Computer-Aided Design (CAD) data and a series of reasoning mechanisms has been shown to be possible. A simplified principle based on considering only orthogonal trajectories has emerged since the matrix representations introduced by Dini and Santochi [1]. They described that the key aspect of the assembly planning problem was the representation of the geometrical and technical relationships between the components in a manner easily managed by a computer. Thus, an assembly was modelled by means of three matrices, i.e., an interference matrix, a contact matrix and a connection matrix. One of each of these matrices is used along the three major axes, resulting in a total of nine matrices. Each matrix, for an n -component assembly, is a square matrix of order n , formed by n row and column elements, $e_1, e_2, \dots, e_n$. Thus, the various relationships between all the components can be encoded. Despite being superseded by more advanced methods, this methodology is still used as a basis for CAAPP [2,3]. This methodology has been extended [4] into a rather more compact, binary function form through combination with the representation of Homem de Mello [5]. More importantly, the binary function values (used to indicate contact and interference etc.) were determined using CAD data and sequences determined by expert systems or heuristic methods. A binary representation is limited, as the

assembly model requires extensive translation for human understanding. In response to this limitation, feature-based representations were developed [6,7]. Models of this type are far more intuitive without losing computational efficiency. Furthermore, this representation integrates well with both object-oriented software development and knowledge-based methods of reasoning. Feature-based modelling provides good opportunities for more efficient and effective constraint extraction, liaison validation, process selection, sequence evaluation and structure visualisation. This work, therefore, proposes to use a combination of the matrix representation and a simplified feature-based methodology, developed in an object-oriented fashion.

1.3 PDM environment for assembly planning

Product Data Management (PDM) systems are tools that help engineers and others manage data and product lifecycle processes. PDM systems keep track of the huge amounts of information required for and generated during a product development lifecycle [8]. They ensure live data is up-to-date and old data is archived correctly, provide security measures and allow data to be properly tracked and audited. Workflow and process management allows an enterprise to define the ways in which changes may be made to product data, providing methods to automatically route data and audit approval decisions. Other benefits of PDM include product structure management, part classification and widespread dissemination of data through translation or rendition. The integration of CAAPP and data management technology can provide an excellent means of providing solutions to some of the problems described above. PDM tools often provide methods by which other software systems may use their resources. This project utilises the Motiva DesignGroup, a mid-range, Windows-based PDM system to provide a data control framework on which the CAAPP system is based. The DesignGroup development kit consists of various ActiveX controls and type libraries that give developers access to the majority of the system's resources. This type of component-based integration provides a quick and effective method of providing

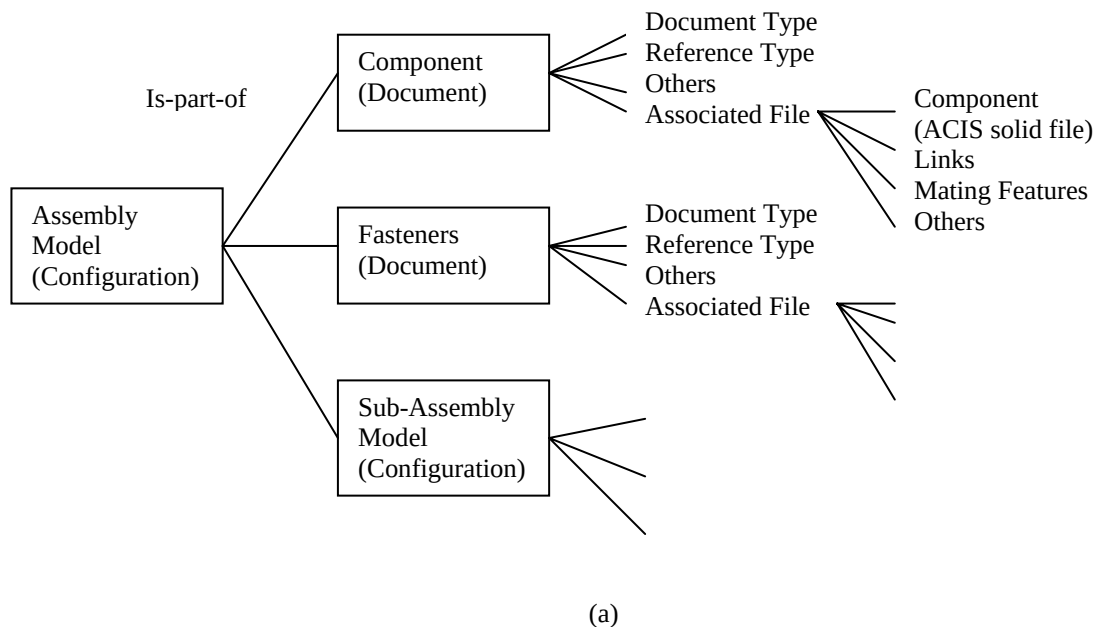
engineering software with high quality data with as little coding as possible. A PDM system can be used as a data portal. For engineering software to be useful and provide meaningful results, the data it uses must be complete, correct and up-to-date. The collection of the relevant data is essentially a clerical task and is, therefore, wasted time for skilled engineers. Integration of CAAPP and PDM provides various methods to simplify the retrieval of relevant data, dramatically cutting down this wasted time. Primarily, this advantage is facilitated by the use of a powerful and consistent interface to allow access to the data within the PDM system vault.

2. The Prototype Assembly Planning System MAPS

2.1 Product configuration management using MAPS

This project differs from other CAAPP systems (most systems discussed above used complicated data structures for assembly modelling) in that it accepts a simplified product model as its input. The assembly model is built in the PDM system and is a hierarchical representation of a product. Note that the hierarchical representation means that inter-component relationships are not described; the model encodes “is-part-of” relationships between a configuration and its members. An assembly is considered as a configuration, consisting of components, fasteners and configurations at a lower-level (sub-assemblies). Each of these three entity types is characterised by a document type in the PDM system and referenced by the configuration to which they belong. Figure 1 shows an example of a simple product model (configuration) used to test the system in the early stage of the project (real example is given in Section 3). Note that an entity is referenced only once, irrespective of how many times it appears within the same level of configuration. The division of a product by this method allows MAPS to differentiate between entities in an assembly without any need to reason on geometrical attributes or other complex data. When assembly

planning data is checked-in or created in the PDM system, the user must assign the correct document type to the file.



Title	Reference Type	Document Type	Revision
Blocks_Config	Hierarchy Member	Assembly Configuration	A
block_square...	Configuration Member	Assembly Component	A
block_hole	Configuration Member	Assembly Component	A
block_pin	Configuration Member	Assembly Component	A
blocks_fixture	Configuration Fixture	Assembly Fixture	A

Properties Revisions References Where Used Process State

(b)

Figure 1: The product model structure (a) and a simple block assembly defined in the PDM system (b)

Each of the entities used to classify an assembly has an associated file. A configuration document is, in fact, a relational database file. This is used to store the property values of the object model used during planning, meaning that the system supports incomplete structure definitions and multiple planning sessions from a single configuration. The database itself consists of four tables; these store data about components, links, mating features and, finally, any ancillary PDM documents associated with the configuration (such as a skinned solid model). A component

document is a prepared ACIS solid model file representing the component geometry. The system prepares the geometry simply by scanning through the component and giving each face a unique tag. This is done by assigning attributes to face entities found in the geometry. This allows MAPS to find relevant faces without having to attach further data to the ACIS file. Once the file is suitably prepared MAPS needs only to copy-out component documents on which it is to work, helping to protect the integrity of working data. A fastener document is also a relational database file. Many CAAPP systems in the past have applied geometrical reasoning techniques to fasteners as well as components. This is ineffective as fastener geometry is often complex, yet their function simple, and they are normally standard parts. In this work a fastener is characterised by its operational parameters rather than its geometry. Both relational data and expert system rules are used in the handling of fasteners within the system. This data is external to the system itself and so can be altered or extended by the user, yet is managed in the PDM system to ensure its quality.

The product model used as input to the CAAPP system provides an intuitive method of subdividing a product into smaller configurations. This can be utilised as a complexity management tool. Plans are generated for each configuration and merged at a later time to produce a full plan. Allowing engineers to divide the product into units defined by them reduces the problem of combinatorial explosion by eliminating a large number of theoretical combinations that need not be considered. Using the PDM system to define configurations is a good solution, as this kind of configuration management is common in that environment. The configuration of the document types and relationships (within the PDM system) can be simplified, to meet MAPS basic requirements, or complex modelling the business rules and process standards used enterprise-wide. Using MAPS on certain entities in the PDM system does not prevent them from being part of workflow activities or other advanced data management techniques; thus, the system can be said to be truly PDM-integrated.

2.2 CAD Integration

In order to provide the user with high quality visualisation tools, MAPS has an integrated CAD viewer based on the ACIS solid modelling system. The ACIS kernel can now be obtained as a component (BuildingBlox), providing excellent scope for quickly and effectively integrating solid modelling technology into engineering software applications. MAPS uses this CAD technology for a number of tasks. Firstly, users may preview the solid model file associated to a component document. If mating or constraint feature definitions exist for a specific component in a specific configuration these may also be viewed. Secondly, the user is expected to select contact surfaces for parts in liaison. This project uses a simplified assembly feature representation to classify mating entities. A feature is defined as a component surface (or set of surfaces) that has some relationship (contact, offset, etc.) to a surface (or set of surfaces) on another component, as shown at the bottom-right corner of figure 2. Surface data for the selected entities is extracted and used to identify it against a collection of library features. During planning, MAPS derives the local freedom of a component with respect to each of its related components. This is done by taking a pair of related parts in their assembled configuration, subjecting one of the pair to a set of translations and rotations and conducting volumetric Boolean operations to reveal any component interference. This technique, although exhaustive, does not take an unreasonable amount of time. This gives the Local Freedom Matrix (LFM), $\mathbf{F}_{1,2}$, which encodes in binary form the freedom of the two parts with respect to each other, holding component 1 in a fixed position and translating component 2. Each element of $\mathbf{F}_{1,2}$ represents the freedom of component 2 under a translation along (of user-defined magnitude d) or a rotation about (of user-defined angle ω) a major axis. The form of $\mathbf{F}_{1,2}$ is as shown in equation 1. It can be seen that $\mathbf{F}_{2,1}$ can easily be obtained through a simple transformation on $\mathbf{F}_{1,2}$.

$$\mathbf{F}_{1,2} = \begin{bmatrix} f_{2,1}dx & f_{2,1}\overline{dx} & f_{2,1}\omega x & f_{2,1}\overline{\omega x} \\ f_{2,1}dy & f_{2,1}\overline{dy} & f_{2,1}\omega y & f_{2,1}\overline{\omega y} \\ f_{2,1}dz & f_{2,1}\overline{dz} & f_{2,1}\omega z & f_{2,1}\overline{\omega z} \end{bmatrix} \quad [1]$$

Where $f_{2,1}dx$ and $f_{2,1}\overline{dx}$ are the translation of component 2 with respect to component 1 along the +x direction and -x direction respectively. And $f_{2,1}\omega x$ and $f_{2,1}\overline{\omega x}$ are the rotation of component 2 with respect to component 1 about the +x and -x direction respectively. Other elements in equation 1 can be explained similarly.

It can be seen from equation 1 that a link can be assigned an LFM to describe the way in which it effects the local freedom of the components associated with it. The Resultant Local Freedom Matrix (RLFM) of a component, e.g., $\mathbf{F}_1$, based on all of its liaisons with other n components, can be derived through conducting a Boolean “and” operation on each of its LFMs, as shown in equation 2.

$$\mathbf{F}_1 = \mathbf{F}_{1,a} \wedge \mathbf{F}_{1,b} \wedge \dots \wedge \mathbf{F}_{1,n} \quad [2]$$

During planning, the RLFMs are recalculated each time the status of the structure changes, providing a method of locating candidate components for removal at any stage of planning. This is the backbone of the trajectory planning functionality of MAPS. Simple constraints, such as floor definitions and fixtures, are implemented as mating features that eliminate degrees-of-freedom in the LFM of constrained components. Therefore, to define the bottom surface of a component to be on the floor, a mating feature is created that eliminates freedom for translations along the -z axis. This can be extended to cover no translations in any direction except z and no rotations to model a base component that remains largely immovable during assembly. The user is free to define their own constraint features and store them for future use. These resources provide a necessary and sufficient resource for modelling an assembly from which it is possible to devise

multiple disassembly sequences. From these disassembly sequences assembly sequences may be derived, so called “assembly-by-disassembly”.

2.3 Other tools used in the prototype system

An assembly structure is often represented by a simple graph. MAPS uses this technique to both display a structure definition and allow interaction with the underlying object-oriented representation of the assembly. The liaison graph is of a typical form, i.e., a component is represented by a circular node and an arc linking two nodes represents a liaison between two components. Fasteners are represented by a square node, midway along a rigid arc. Note that lower-level configurations are represented similarly to components but in an alternate colour. MAPS uses the diagramming component AddFlow to provide a dynamic method of representing an assembly structure. AddFlow provides creation, modification and analysis techniques for various types of diagram and was found to be an excellent solution to the problem of generating liaison graphs. Figure 2 shows the MAPS interface and its constituents, including a simple liaison graph showing product structure at the left-hand side of the figure. The assembly structure can be altered through interaction with the liaison graph; adding (or deleting) nodes, links and fastener details immediately updates the object-oriented model used to represent an assembly in MAPS. Adding a node to the assembly graph directly corresponds to adding an object to the collection of components. Adding a link, in turn, corresponds to adding an object to the collection of liaisons, and so on throughout the model. In fact, it is the objects themselves that know how to interact with the outside-world, ensuring full homogeneity between the object model and the liaison graph. To facilitate rule-based reasoning, the CLIPS expert system is embedded in MAPS. This allows the system to process rule files and retrieve the results. Figure 3 shows the configuration of tools used in the MAPS development project. Getting and setting CLIPS variables and controlling expert

processes is facilitated by an external Dynamic Link Library (DLL), containing CLIPS user-defined functions and the code to initialise, run and return results from a knowledge-base.

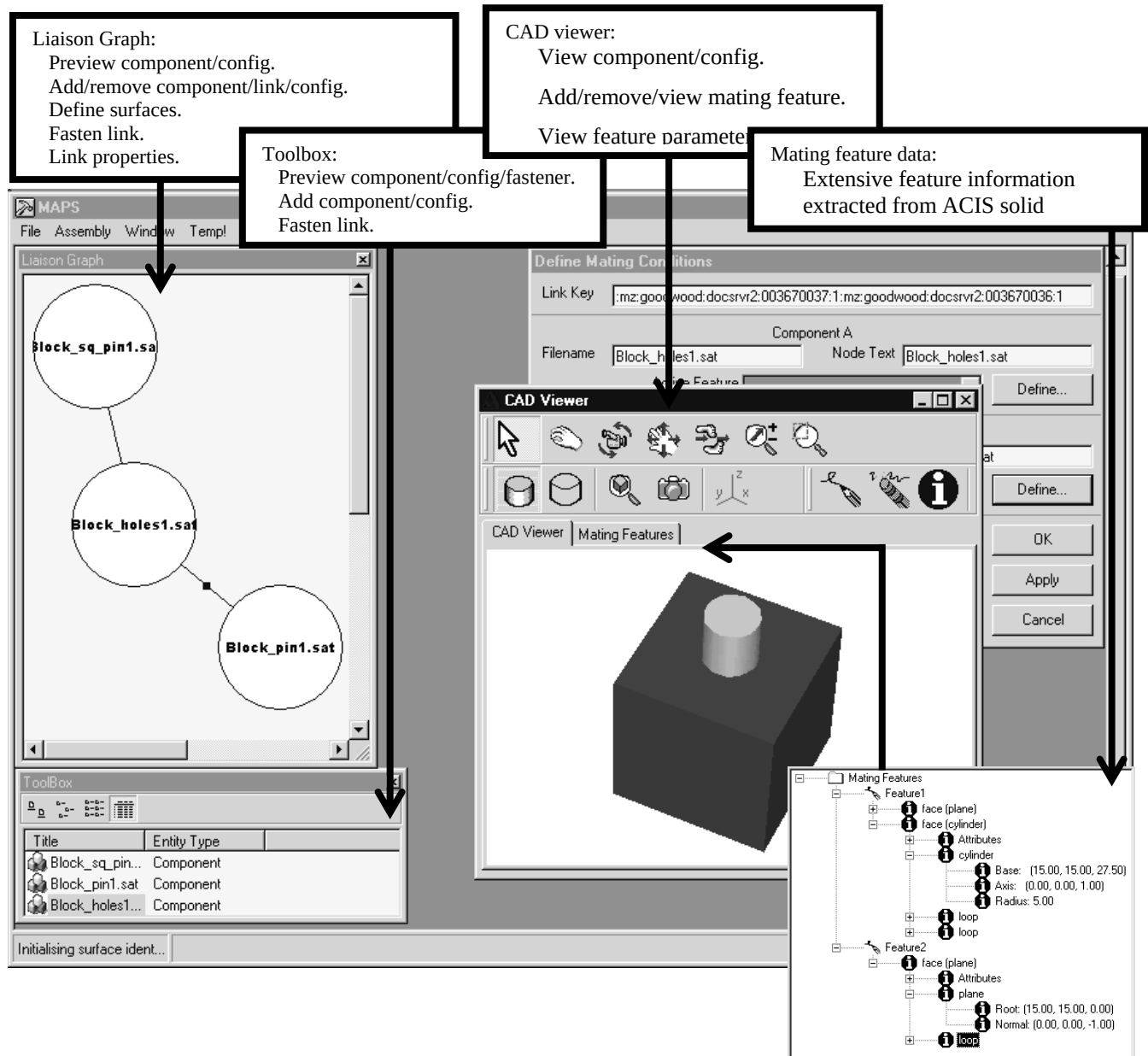


Figure 2. The MAPS interface and its components

As seen in figure 3, MAPS itself is implemented in Visual Basic and C, running under the Windows NT operating system. Motiva DesignGroup PDM System is used to handle product data and related documents. It interacts with Spatial Technology Building Blox (ACIS) to generate

solid models for products and extract feature and constraint information as needed. The Lassalle Technology AddFlow software is used to construct liaison graphs both manually and automatically by the algorithms developed or rules in the CLIPS expert system.

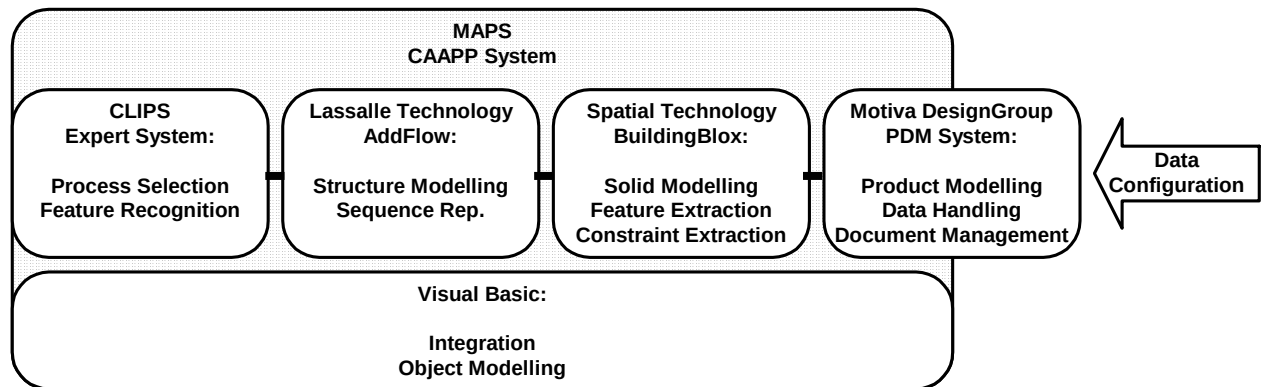


Figure 3. The tools used (and their purposes) in the MAPS development

3. MAPS Operation

This section describes how the object-oriented representation of the assembly is built within MAPS during the planning process. Firstly, the user initiates a planning session by navigating the PDM data vault checking-out the required configuration document. MAPS then resolves the entities referenced to the configuration document and presents the user with the standard DesignGroup vault access dialogue box. This system dialogue provides tools to search a vault (document title, document type, document status, document property and document content variants) and customise the dialogue itself (default settings and look-up locations etc.). Once the required configuration file has been located, checked-out and its relations resolved, another DesignGroup system dialogue box is displayed in which the user may choose which of the resolved references they wish to retrieve (i.e. copy-out to their local machine). Thus, MAPS can

retrieve all of the required data (based on the PDM model) simply by the check-out of the configuration document under consideration.

3.1 The configuration relation object model

The data retrieved is used to generate an object model, which represents the referenced entities. This builds a link between the documents in the PDM system and other object models using a DesignGroup document identification string as a unique key throughout the system. The relation object model is used to build a toolbox (see figure 2), containing all of the components, fasteners and lower-level configurations found for a configuration document. The toolbox's primary purpose is to expose an interface to the relation object model and, hence, the data pertaining to these entities. For example, the user may wish to preview a component or constituent configuration geometry and defined mating features or view fastener parameters. Objects may also be added to the assembly model by using the toolbox (but not removed).

Figure 4 shows a diagram of an example configuration which is a door latch currently being produced by the industrial collaborator ArvinMeritor coded as M2S. In this example, the retention and back plate assembly cells feed the main assembly cell with subassemblies. The main assembly cell feeds the cable and test assembly cell, where the product assembly is completed. As discussed earlier, the product level configuration document resides at the top level of the hierarchy. Figure 5 shows the main line subassembly featuring the retention plate subassembly and the constituent components.

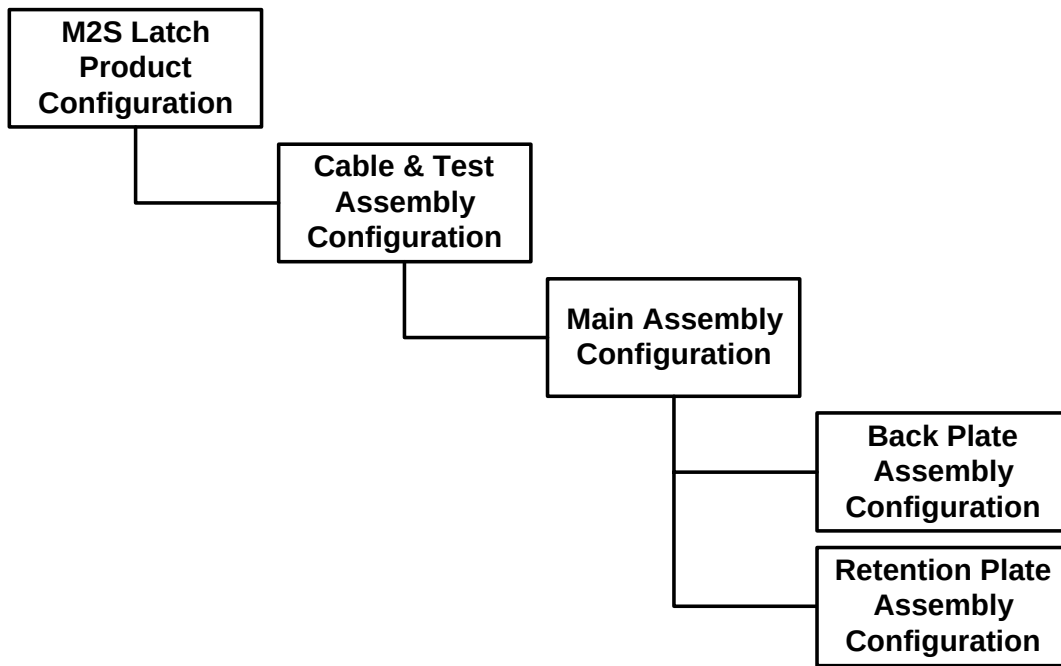
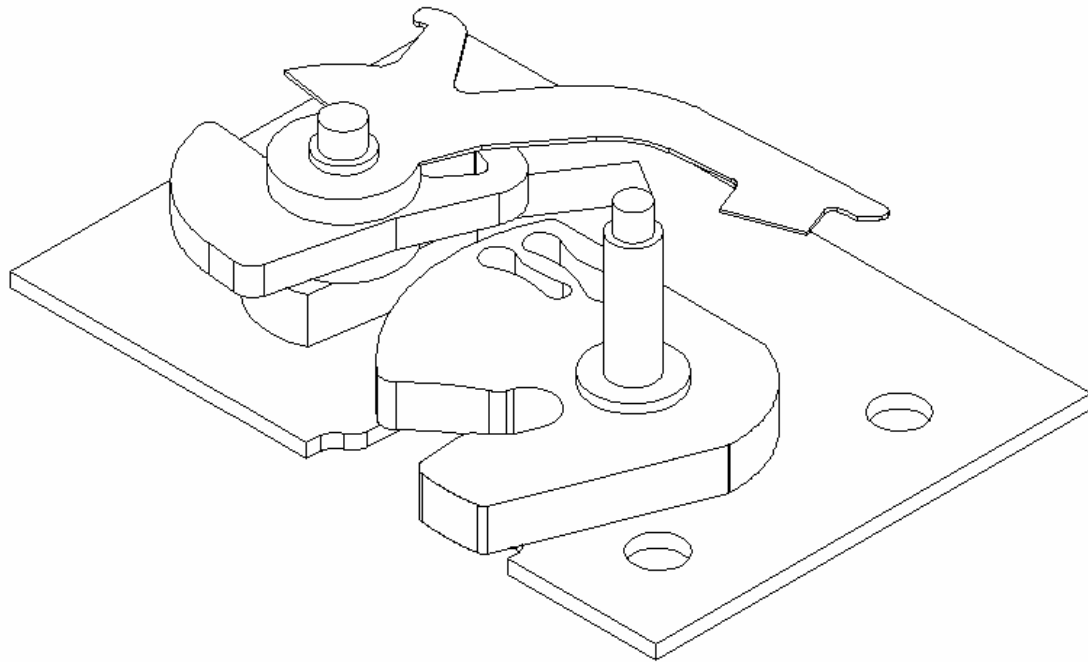


Figure 4: Example of configuration level product model

3.2 The assembly object model

The user is then directed to build the assembly object model by adding toolbox objects and links to the structure via the liaison graph. If previous structure definitions exist, the object model can be spawned from the data contained in the configuration document (and vice versa). The assembly object model is comprised of four main class-and-object entities; the components, their links, the fasteners used and the active mating features. Other objects also exist to hold data about the configuration itself, to conduct various system tasks and to hold an intermediate plan sequence. As described in earlier sections, components, links and fasteners are added to the object model using the toolbox or the liaison graph. Mating features are selected using a CAD tool and written to the object model. The various methods of interacting with the assembly model can be seen in figure 3. Figure 6 shows the resulted liaison graph for the main line subassembly of the example in figure 4 and 5.



Configuration/ Cell	Constituent components
Retention plate sub-assembly	Pawl rivet Claw rivet Claw Retention plate
Main Line	Retention plate sub-assembly Pawl Pawl lifter Outside opening lever

Figure 5: The main line subassembly featuring the retention plate subassembly and the constituent components

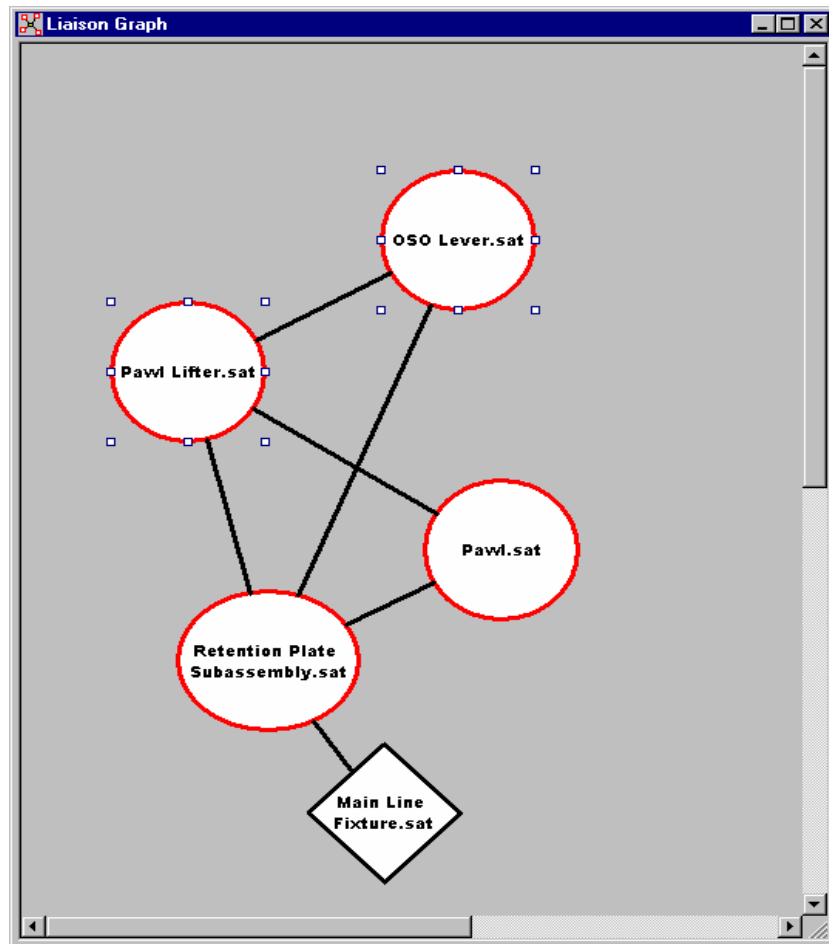


Figure 6: The liaison graph of the main line subassembly of the latch example

3.3 Plan generation

Due to the nature of the system, the sequence generator can be run at anytime and will provide accurate results based on the current state of the object model. When the plan generator is initiated, MAPS batch processes component global freedom (i.e., the system assumes the structure is completely defined in its assembled state). Global freedom is defined by taking each component in turn and determining whether any blocking components lie in the path of a possible disassembly trajectory. This is done by checking each associated link LFM, rather than RLFM, because trajectories impossible in a fully assembled configuration are likely to become possible later. In fact, the Global Freedom Matrix (GFM), GF_n , which determines all the directions which may be

possible from some intermediate configuration can be determined using simple Boolean algebra (see equation 3). The trajectories determined by the GFM for each component are checked and any blocking are recorded. This record can be consulted during sequence generation in order to determine global freedom.

$$GF_1 = F_{1,a} \vee F_{1,b} \vee \dots \vee F_{1,n} \quad [3]$$

Another batch process is initiated to attempt to derive a process type associated to the realisation of each link. This is done by considering the properties of the link and its associated component, mating feature and fastener objects. A link to the CLIPS expert system facilitates process selection by the activation of specified rule files, governed by the code in the CLIPS user-function DLL. CLIPS user-defined functions allow transfer of data from the object model to appropriate CLIPS variables. The work of compiling effective knowledge-bases based on industrial experience is ongoing. The model is now fully defined, containing local and global constraints, feature-based information, fastener details and some process information. The structure is recursively examined to find locally and globally free candidate components. If multiple candidates are found, the configuration is stored to allow back-tracking and the generation of multiple sequences. Process and fastener details can be used to force MAPS to exclusively consider or overlook links based on expert system rules. Geometrical constraints are evaluated by accessing the object model algorithmically.

The sequences are represented by a variant array, stored as a property of the plan object. Each element contains a rendition of the state of the structure, information regarding the link (and its associated mating features and fasteners) that has been removed to form that state and another variant array representing the children of that state. This is a similar representation to the AND/OR (i.e. nodes as configurations, arc as possible transitions) graph but is at a higher level. It is possible to draw an AND/OR graph from this data structure and it contains data regarding process selection. The array is steadily filled with process sequences as the planner runs. As the assembled

configuration part count can be managed by the PDM system it is possible to avoid the combinatorial explosion often encountered at this stage.

When the process sequences are completed (i.e. all permitted paths through configuration space have been followed to full disassembly), MAPS creates a new DesignGroup document object for the intermediate plan. This document object is associated to a relational database file that stores the variant array in tabular form. The object model that controls the display of the intermediate plan can be recreated at any time using the relational data stored in the plan document. Navigation of the intermediate plan is supported within MAPS. A sequence analysis module can be used to view the sequence tree and highlight the possible paths through it. The user can remove specific configurations at this stage should they wish. The user can view the process, fastener and associated component properties of any particular state transition and apply a weight to them. This provides an elementary method of sequence evaluation. Note that this approach is primarily concerned with the acquisition of process sequences and their representation in such a way as to be suitable for downstream analysis. The output is in a recurring format of 7 lines. These lines and the meaning of the associated data is shown in figure 7.

Line No	Text			Remark
1	Candidate...			Remark line
2	Sequence #: 1			Sequence of the component in the disassembly.
3	Name: Retention Plate.sat			Name of the component (SAT file)
4	Key: :mz:goodwood:docsrvr2:003670175			Key of the SAT file in the PDM system
5	Cell: Retention Plate			Name of the assembly cell (in the configuration)
6	Cell ID: :mz:goodwood:docsrvr2:003670145			Key of the assembly cell in the PDM system.
7	0	0	0	Local freedom matrix.
	0	0	0	
	0	0	0	
	1	0	0	

Figure 7: Output of the MAPS for the retention plate of the latch example

Assembly Process Plan

Control Plan Number	M25 9/2/00	Key Contact's Phone	0121 459 1166	Date (Orig.)	02/11/98
Customer Part Number		Core Team	Dev. DCCC	Cost Engg. Approval Date	
Supplier Part Number	M740773021				
Issue Level	F				
Part Name/Description	M25-S1	Supplier's Approval Date		Cost Quality App. Date	
Supplier Plant	BIRMINGHAM	Other Approval Date		Other Approval Date	
Supplier Code					

Part/Process Number	Process/Method/Operation Description	Machine/Device/Jig/Tools for Mfg	Classifications			Special Class Class	Methods				Resource Plan	
			No	Product	Process		Product/Process Specification/Tolerance	Evaluation Measurement Technique	Sample	Control Method		
101	Assembly Part 1170	Small tools and equipment		Part 1170	part 1170, Class 1000		Product of Part	Probe Settings	1	100	Check with part Probe Yoke	M/C steps Test leads as usual
101	Small tool part 1170	Equipment, machine		Part 1170								

1 of 1 2 of 2 Total: 2 100%

Figure 8: Format of the resultant assembly plan for the latch example generated using both MAPS

This pattern is repeated for each component in the assembly. It gives all the necessary information to realise the assembly/disassembly sequence of the assembly. This output has not been formatted or stored in a database deliberately to allow easy viewing, editing and retrieval. In case the system is to link to any downstream system for more analysis then this output would have to be formatted accordingly.

In the later stage of the project an interactive planning module has been developed within the same environment and thus it is fully integrated with MAPS and shares the all the data managed by the PDM system. The interactive system can use the result of the automatic system (MAPS) as incomplete plans and generate complete assembly plans manually or with support of evaluation methods which can be manually executed. The results of the complete plan using both the automatic system and the interactive system for the latch example have satisfied the company's

requirement and the format is shown in figure 8. In fact the authors do not intend to develop a system that can generate complete assembly plans automatically.

4 Conclusions

The development of prototype modules has shown that the approach discussed in this paper is possible and effective. Initiating planning session takes a very short time and provides the system with correct, up-to-date data without troubling the user. Interaction with the underlying object model through a liaison graph proved to be fast, intuitive and effective. The identification of local and global constraints proved to be fast and accurate (although the system current assumes perfect geometry and does not support tolerance). The planning engine itself works well but the results depend very strongly on the state of the object (i.e. its accuracy) and the quality of the knowledge-based routines that it uses. A case study has been completed with ArvinMeritor for its assembly planning of door latches for motor cars. Further developments are required in the sequence analysis module to allow full interaction with the process sequence tree. Also, gradually refining the object model (and the interface elements to it) as the development continues will increase the model accuracy and effectiveness with respect to issues such as component type, fastener and process/resource modelling. The automatic planning system (i.e., MAPS) proves to be the ideal input to the interactive planning module which will eventually produce the complete assembly plan for real application. The authors do not intend to, and indeed it is impossible to develop a fully automatic assembly system without human intervention. The main benefit of the research is the proof of using the powerful product data management tool for assembly planning which has not been tested by previous researchers.

Acknowledgements

This research is supported in part by EPSRC Research Grant GR/L66236. The support of Matra-British Aerospace Dynamics and Meritor Light Vehicle Systems is also greatly appreciated. Provision of Motiva DesignGroup by Photonic EDC is acknowledged with thanks. Individual thanks also go to Alan Crawford and Ray Muldoon of LSC Group, Andy Gordon of Photonic EDC and Patrick Ollerton of PTC.

References

- 1 **G. Dini, G. and Santochi, M.** Automated sequencing and subassembly detection in assembly planning. *Annals of the CIRP, Vol. 41, No. 1* (1992).
- 2 **Kuo, T. C., Zhang, H. C. and Huang, S. H.** Disassembly analysis for electromechanical products: a graph-based heuristic approach. *International Journal of Production Research, Vol. 38, No. 5* (2000).
- 3 **R.B. Gottipolu, R. B. and Ghosh, K.** Representation and selection of assembly sequences in computer-aided assembly process planning. *International Journal of Production Research, Vol. 35, No. 12* (1997).
- 4 **Zha, X. F., Lim, S. Y. E. and Fok, S. C.** Integrated knowledge-based assembly sequence planning. *International Journal of Advanced Manufacturing Technology, Vol. 14: 50-64* (1998).
- 5 **Homem de Mello, L. S. and Sanderson, A. C.** Automatic generation of mechanical assembly sequences. *Technical report: Carnegie Mellon University* (1988).
- 6 **Pham, D. T. and Dimov, S. S.** A system for automatic extraction of feature-based assembly information. *Proceedings of the Institute of Mechanical Engineers, Vol. 213, Part B* (1999).

- 7 **Wang, E. and Kim, Y. S.** Feature-based assembly mating reasoning. *Journal of Manufacturing Systems, Vol. 18, No. 3* (1999).
- 8 **CIMdata Incorporated.** Product data management: the definition. Fourth Edition (1996).



DSPACE

<https://dspace.org/>

A product data management integrated product configuration and assembly process planning environment.

Gao, James X.; Bowland, N. W.

2002-03

Professional Engineering Publishing

<http://hdl.handle.net/1826/1048>

Gao, J X; Bowland, N W; A product data management integrated product configuration and assembly process planning environment. Proceedings of the Institution of Mechanical Engineers Part B. Engineering Manufacture. Mar 2002, Vol. 216 Issue 3, p407-418.