Methods to Analyse the Impact of Changes in Complex Engineering Systems

Y. Lemmens[1] and M Guenov[2]
Cranfield University, Cranfield, MK43 0AL, UK

A. Rutka[3]
Airbus SAS, 1 rd point Maurice Bellonte, 31707, Blagnac, France

P. Coleman[4]
Airbus UK, New Filton House, Bristol, BS99 7AR, UK

T. Schmidt-Schäffer[5]
EADS Innovation Works, Munich, 81663, Germany

[Abstract] Changes occur throughout the lifecycle of complex products such as aircraft. Therefore, it is important to be able to analyse the impact (or consequences) of change from different points of view e.g. requirements, physical or functional product architecture, design processes or activities, organisation. The methods described in this paper aim to facilitate decision-making by improving the visibility and shared understanding of the interdependencies that exist within and between such viewpoints. Change Propagation Analysis (CPA) algorithms have been implemented in a prototype software environment to allow the modelling and visualisation of multiple dependencies across multiple information domains, including lifecycle considerations. Results from testing the CPA algorithms are illustrated, using a verification dependency model. Further validation of the methods is on-going using a variety of case studies from the aerospace industry.

Abbreviations:

EC - Engineering Change
CPA - Change Propagation Analysis
ToC - Type of Change
LoC - Level of Change
DMM – Domain Mapping Matrix
DSM – Design Structure Matrix

I. Introduction

Complex products are comprised of a large number of tightly integrated components, assemblies and systems resulting in extensive logical and physical interdependences between the constituent parts [5]. Thus a change to one item of a system is highly likely to result in a change to another item, which in turn can propagate further [1]. It is widely acknowledged [4] that change propagation analyses (CPA) are necessary for predicting and simulating the impact of a change, in order to improve the capacity to manage time, cost, resources and quality. Due to the globalisation and fragmentation of the aerospace and other industrial sectors there appears to be a particular need for a more integrated and shared CPA approach within organisations and across their supply chains [6, 7].

---

[1] PhD Student, Aerospace Engineering Department, Bldg.83.
[2] Professor, Aerospace Engineering Department, Bldg.83.
[3] Digital Mock Up & A3456-GMF Manager, Development & Serial Programmes.
[4] Engineering Capability Projects, Architecture & Integration Centre, Building 09A.
[5] Dept. LG-SE1

In this context the overall aim of our research has been to develop an approach to support decision-making in the EC processes that enables a shared view between all stakeholders, leading to a better understanding of the collaborative and concurrent environment.

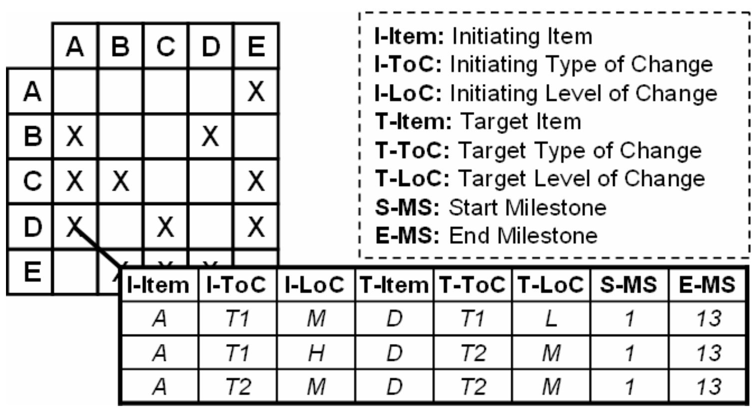
In our previous publication [8], a dependency model was described that enabled the elicitation of relations between different viewpoints or domains of the overall engineering system. These also allowed a more complete definition of the relations by also considering the type of change and magnitude or level of change. In this paper, novel methods are presented which have been developed with the aim to optimally exploit these dependency models. The particular objective is to improve the propagation analysis by considering the appropriate level of detail, limit redundant information and organise the propagation between different information domains.

The next section begins with a brief description of the dependency model that is used in the CPA Methodology. It sets the context for the description of the proposed propagation control method that is the main subject of section II. Section III describes the verification results from testing the methodology. Finally conclusions are drawn and further work outlined.

II. Proposed CPA Methodology

A. Dependency Model

The dependency model, as described in our previous publication [8], can represent several viewpoints of the engineering system, for example, requirements, product architecture, design processes or activities. Each viewpoint or domain has information items associated with it. A key attribute of an item is the milestone in the product lifecycle at which the item will be frozen, i.e. that it should not be changed from this milestone onwards. Secondly, the model also contains dependency information that describes the links or relations between two items. In our model, a relation defines the Type of Change (ToC) and Level of Change (LoC) of the affected item for a specific ToC and LoC of an initiating item. Also, multiple relations can be defined between two items for modelling different impacts for various initiating LoC's and ToC's. Figure 1 depicts a Design Structure Matrix (DSM) [9] representing a dependency model of a single domain. An 'X' in a cell indicates that there exist one or more relations from the initiating item in the column to the target item in the row. The inset table shows the definition of three dependencies from item 'A' to item 'D'.



I-Item	I-ToC	I-LoC	T-Item	T-ToC	T-LoC	S-MS	E-MS
A	T1	M	D	T1	L	1	13
A	T1	H	D	T2	M	1	13
A	T2	M	D	T2	M	1	13

Figure 1: Example of a dependency definition in a single domain model

These dependency models will evolve as the overall design representation matures and the corresponding engineering organisation changes throughout the product lifecycle. The models can also be used as a baseline for new versions of the product.

B. Types of Propagation

Since dependencies are defined between the ToC and LoC of the initiating item and affected ToC and LoC of the target item, a change propagation analysis can take into account the affected ToCs and LoCs of the impacted items during the propagation. However, there are occasions when it can be beneficial to ignore the ToC or/and LoC of the dependencies in order to achieve a wider propagation and hence to identify more potentially affected items. Consequently, 4 types of propagations are considered. Each of these propagation types are described in more detail below and an example of a propagation tree is depicted based on a set of dependencies listed in the Appendix.

1. SIMPLE PROPAGATION ANALYSIS

The 'Simple' Propagation Analysis only considers relations at item level. This means that at every propagation level, all dependencies are taken into account for every initiating item. Hence, only the item name needs to be selected as the initiating condition for the analysis. Consequently, the propagation is not influenced by the defined initiating and target ToCs and LoCs of the relations. The propagation tree below illustrates the result of this type of analysis with 'A' as initiating item (using the example dependencies listed in Appendix A) and for two propagation levels.

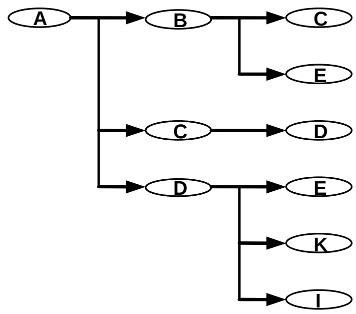


Figure 2: Simple Propagation Analysis

This type of propagation analysis can be used to identify the maximum possible extent of the EC impact, particularly in case where the considered ToCs do not match the modelled ToCs or the latter are not completely independent. Consequently, this type of propagation analysis identifies a rapidly increasing number of possible impacts. For example, in the case of propagation analysis in a Physical Architecture domain, all items could eventually be identified as possibly impacted.

2. DETAILED PROPAGATION ANALYSIS WITH TOC ONLY

This type of propagation analysis only considers the ToCs of the relations between items, but ignores the LoCs. Therefore, together with the initiating item, also an initiating ToC needs to be selected. Consequently, all affected ToCs become initiating ToCs for the following propagation level. The propagation tree below illustrates this propagation type with 'A' as initiating item and 'Blue' as initiating ToC for two propagation levels.

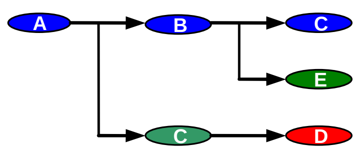


Figure 3: 'ToC-only' Propagation Analysis

This type of analysis simulates a propagation behaviour where a change of an item will always affect another item as long as there is a dependencies between the relevant ToC regardless of the magnitude of the change. Consequently, as this propagation analysis continues, different ToCs will be identified for new and already identified items. This can lead to a propagation behaviour described as an 'avalanche' or 'blossom' effect [2], which is an uncontrolled propagation where a large part of the product is affected due to the initiating change.

3. DETAILED PROPAGATION ANALYSIS WITH LOC ONLY

This type of CPA considers only the LoCs of the relations between items but ignores the ToCs. Therefore, together with the initiating item, also an initiating LoC needs to be selected. Consequently, all affected LoCs become initiating LoC for the following propagation level. The propagation tree below illustrates this propagation types with 'A' as initiating item and 'Medium' (M) as initiating LoC for two propagation levels.

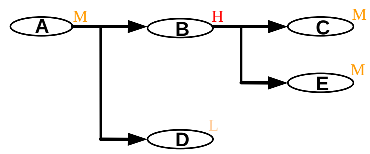


Figure 4: 'LoC only' Propagation Analysis

In this case, a specified level of change is required for the affected items in order for the propagation to continue. However, as the ToCs of the affected items are ignored, this is considered a more theoretical type of propagation analysis, for which the practical use still needs to be further investigated.

4. DETAILED PROPAGATION ANALYSIS WITH TOC AND LOC

For this type of propagation analysis, the CPA should consider the ToC and LoC of every affected item and take these into account to identify the relations for the following propagation level. Hence, this type of analysis requires a ToC or a LoC to be selected as initiating conditions. The propagation tree below illustrates this propagation types with ‘A’ as initiating item and ‘Medium’ (M) as initiating LoC for two propagation levels.

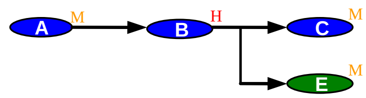


Figure 5: Detailed Propagation Analysis

Similar to the previous propagation type, this last type also requires a specified level of change for the affected items in order for the propagation to continue but now also for the relevant ToC. This implies that changes are absorbed by some items when the LoC of the affected items is lower than required in order to affect more items. Consequently, a more controlled change propagation behaviour is attained where the extent of the EC impact is more constrained. This is comparable to the ‘ripple’ propagation behaviour, which has been described in literature [2].

C. Limiting the propagation tree visualisation

During a CPA, an item can be impacted which has already been impacted at a preceding level. This would result in the repetition of the same knock-on impacts. This in turn could lead to propagation loops and consequently to an infinite propagation. The repeated impacts add no new information to the propagation results and increase unnecessarily the size of the propagation tree. The loop detection works in two ways. First, the propagation does not continue for items that have been impacted at previous propagation levels with the same ToC or/and LoC depending on the type of propagation that is used. In the case of a Simple Propagation Analysis, there are no impacted ToCs and LoCs and hence not taken into account. Second, in the case where the same items with the same ToC and/or LoC are impacted multiple times at the same propagation level, the propagation should only be continued for one of those impacts.

Both types of loop detection are illustrated in Figure 6 based on the example of a simple propagation analysis shown in Figure 2. First, the propagation is not continued beyond the impact on ‘C’ at the second propagation level as ‘C’ has already been impacted on the first level and the same for item ‘D’ (purple arrows). Secondly, ‘E’ is impacted twice at the second propagation level. Only for one impact is the propagation continued (blue arrow).

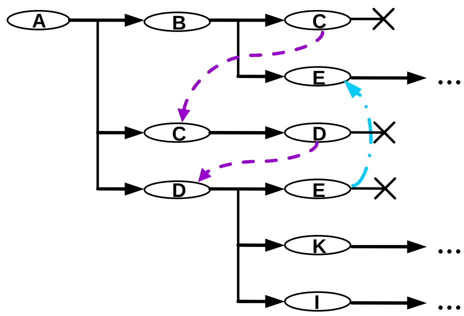


Figure 6: Loop Detection

In parallel, the status of an item is taken into account during the propagation analysis, i.e. that the propagation does not continue for affected items when their status is frozen for the specified milestone. Additionally, a dependency has a lifecycle range for which it is valid. Also this range is also taken into account, i.e. that only those relations are considered for which the specified milestone is within their defined lifecycle range.

D. LoC propagation management

Unlike ToCs, LoCs are not completely independent because these are graded and therefore can be ordered. Consequently, this order can be taken into account during the propagation to identify potentially more relevant impacts and to minimise duplicate impacts. Thus, the propagation and loop detection can be extended with three additional rules for Detail Propagations which consider LoCs. Each of these rules is illustrated with a propagation analysis example based on the dependencies listed in the Appendix.

1. RULE I

Beside the selected or identified initiating LoC, also all lower LoCs need to be taken into account. The rationale is that in case a low LoC of an item has an impact on an item, a higher level of change should also identify this impact. The example below shows this as ‘K’ which impacts both ‘A’ and ‘J’ while the LoC of ‘K’ was High but only a Medium LoC was required to impact ‘J’.

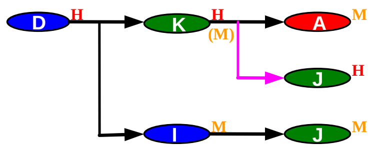


Figure 7: LoC Management Rule 1

2. RULE II

The propagation should be stopped when an item is impacted which has been impacted before at a higher level (hence not only the same LoC). This is because the impact of the new (lower) identified LoC should already be checked for the previous impact with a higher LoC as a result of the previous rule. This is illustrated in the figure below where the propagation at the second level for ‘K’ is stopped because the newly identified LoC (Medium) of ‘K’ had been included at first propagation level with ‘High’ ‘K’ as initiating condition.

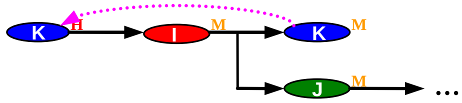


Figure 8: LoC Management Rule 2

3. RULE III

If an item is impacted at multiple LoCs at the same propagation level the propagation should continue only for the impact with the highest LoC. However, due to the first rule, also all lower LoCs should be included for that impact. This is illustrated below where although ‘J’ is identified at the second propagation level as ‘High’ and

‘Medium’, the propagation only continues for the highest LoC of ‘J’ (‘High’) but includes also impacts for lower LoCs (‘Medium’).

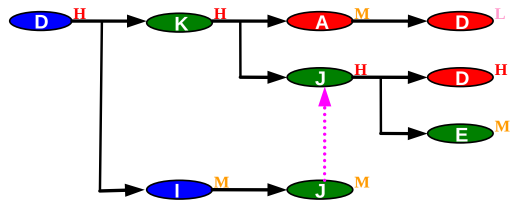


Figure 9: LoC Management Rule 3

E. Inter-domain propagation

As described above, the CPA methodology supports the elicitation of dependencies between items of different types of information organised in domains. All the dependencies, i.e. the dependencies between items of the same domain and between items of different domains, can be visualised in an Inter-domain DSM. This DSM is composed of a set of matrices. Single domain DSMs with the dependencies between the items of the same domain are included along the diagonal. Additionally, Domain Mapping Matrices (DMMs) [3] are used to define dependencies between items in 2 different domains. All the DSMs and DMMs for all included domains form a square Inter-domain DSM. An example of an inter-domain DSM for 3 domains in depicted in Figure 10.

		Domain 1					Domain 2				Domain 3		
		A	B	C	D	E	1	2	3	4	α	β	γ
Domain 1	A					X							
	B			X			X	X			X		X
	C							X		X	X		
	D		X							X		X	
	E			X								X	
Domain 2	1		X					X			X		
	2		X	X								X	
	3						X				X		
	4			X	X				X				X
Domain 3	α		X	X			X		X				
	β				X	X		X					
	γ		X							X			

Figure 10: Example of Inter-domain DSM

In order to control the identification of cross-domain impacts, special attention has been given to inter-domain propagation. The objective is to give the user a high degree of control over the way the propagation continues between domains. The intention is to provide the decision-maker with a bespoke interface which allows him/her to select specific domain combinations wherefore dependencies have to be considered and ignore all other dependencies for a particular change propagation analysis. This interface constructs a Domain Selection DSM of the available domains, representing the Inter-domain DSM discussed above. Hence the cells on the diagonal represent the dependency DSMs while the off-diagonal cells represent the dependency DMMs. The selected cells control the way the propagation is executed. Figure 11 depicts the Domain Selection DSM and Figure 12 shows the selected section of the inter-domain DSM of Figure 10.

Target \ Initiating	Domain 1	Domain 2	Domain 3
Domain 1	☑	☐	☐
Domain 2	☑	☐	☐
Domain 3	☐	☐	☐

Figure 11: Domain Selection DSM

		Domain 1					Domain 2				Domain 3		
		A	B	C	D	E	1	2	3	4	α	β	γ
Domain 1	A	■				X							
	B		■	X			X	X			X		X
	C			■				X		X	X		
	D		X		■					X		X	
	E			X		■						X	
Domain 2	1		X				■	X			X		
	2		X	X				■				X	
	3						X		■		X		
	4			X	X				X	■			X
Domain 3	α		X	X			X		X		■		
	β				X	X		X				■	
	γ		X							X			■

Figure 12: Section of Inter-domain DSM to be considered

In this example, Domain 1 is an initiating domain and the target domains are Domain 1 and Domain 2. This means that the dependencies between the items of Domain 1 and the dependencies from the items in Domain 1 to the items in Domain 2 need to be considered during the propagation analysis. Furthermore, at every propagation level, impacts on items of Domain 1 and Domain 2 will be identified. However, only the affected items from Domain 1 will become the initiating items for the following propagation level. This also means that that the initiating item needs to belong to Domain 1. In general, the initiating item needs to belong to an initiating domain and some of the selected target domains also need to be included as initiating domains in order for the propagation to continue.

III. Verification

A. Verification of Single Domain Propagation

This section provides illustrative results from applying the CPA algorithm discussed above. For the purpose of testing the CPA algorithm for the propagation of changes within one domain, a single domain verification dependency model has been constructed. The dependency DSM is shown in Figure 13 which comprises 10 items ("A" to "J"). The full list with dependency definitions is included in the appendix. The number in the cells indicate the number of relations there exist between the items.

	A	B	C	D	E	F	G	H	I	J
A								1		
B	1						2			
C	1	1					1			
D	1		1							1
E		1		1						1
F					1					
G					2	1				
H				1					1	
I				1	1			1		
J								1	2	

Figure 13: Verification Model for Single Domain Propagation Analysis

The resulting propagation trees from the change propagation analyses using the four types of propagation and taking into account loop detection and LoC management are shown in Figures 14 - 17. Items for which a loop was detected are highlighted. The propagation level of the initial impact is indicated between square brackets. The table on the right hand side of each figure lists the affected ToC and LoC of the impacted item.

<u>1) Simple Propagation Analysis (Figure 14)</u>

A propagation analysis ignoring the defined ToCs and LoCs of the dependencies is performed with item ‘A’ selected as initiating item. The analysis is executed until completely finished due to loop detection. As a result, there are 22 impacts and all 9 remaining item identified as possibly impacted.

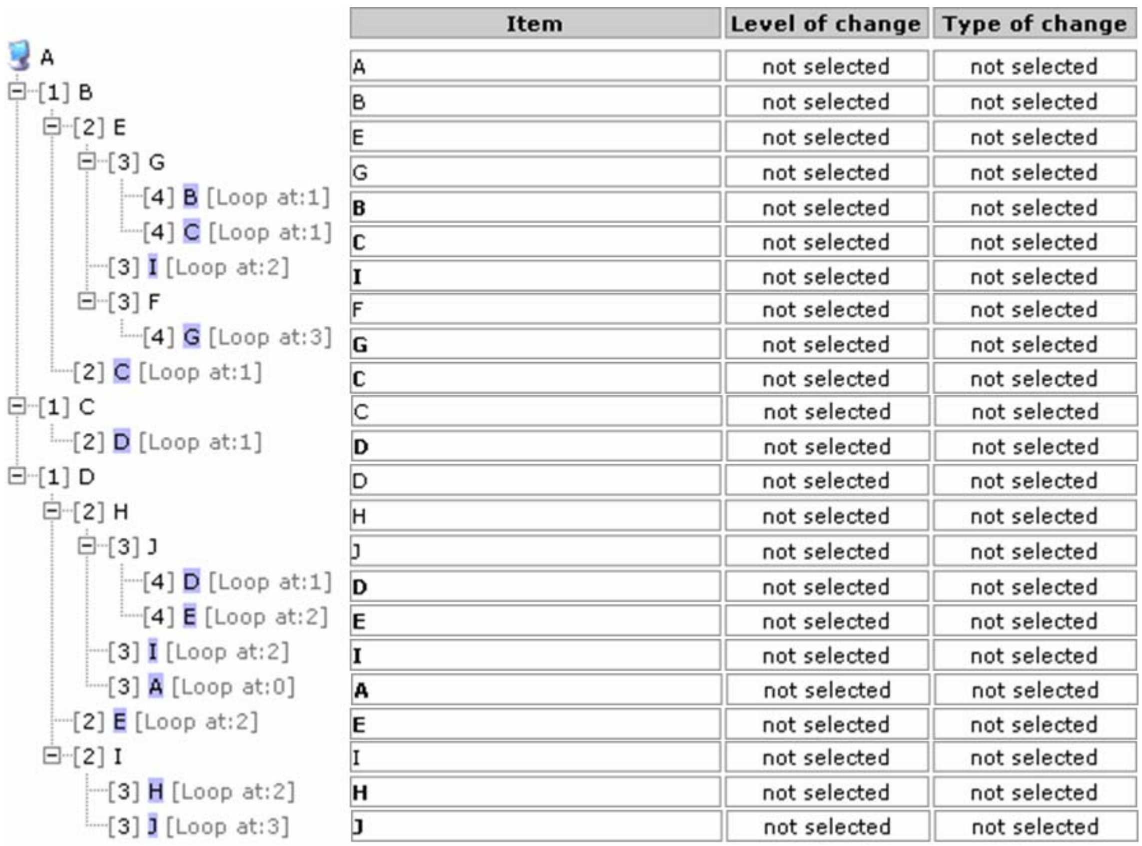
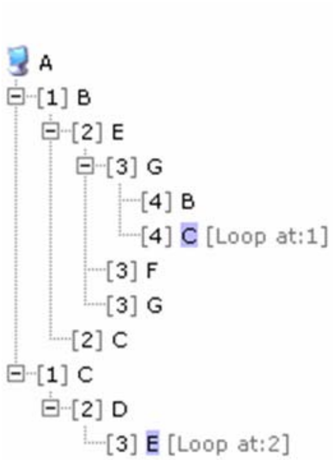


Figure 14: Verification Results for Simple Propagation Analysis

<u>2) Detailed Propagation Analysis with ToC only (Figure 15)</u>

In this analysis, again item ‘A’ is selected as initiating item together with ‘Blue’ as initiating type of change. The analysis is performed until no more items are affected or the propagation is terminated due to loop detection. 6 items are identified as possibly affected.

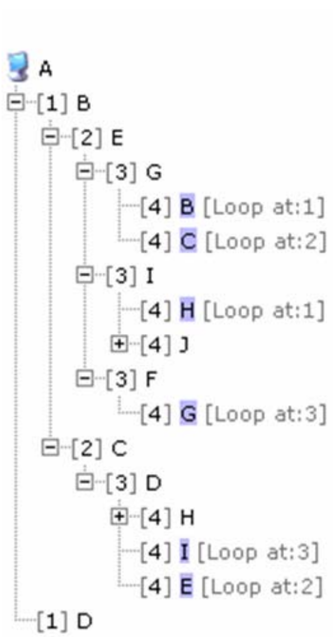


Item	Level of change	Type of change
A	not selected	Blue
B	not selected	Blue
E	not selected	Green
G	not selected	Green
B	not selected	Red
C	not selected	Green
F	not selected	Green
G	not selected	Blue
C	not selected	Blue
C	not selected	Green
D	not selected	Red
E	not selected	Green

Figure 15: Verification Results for 'Toc-only' Propagation Analysis

<u>3) Detailed propagation Analysis with LoC only (Figure 16)</u>

In this analysis, item ‘A’ is selected as initiating item together with ‘Medium as initiating level of change. The analysis is performed until the fourth propagation level or the propagation is terminated due to loop detection. All items are again identified as possibly affected.

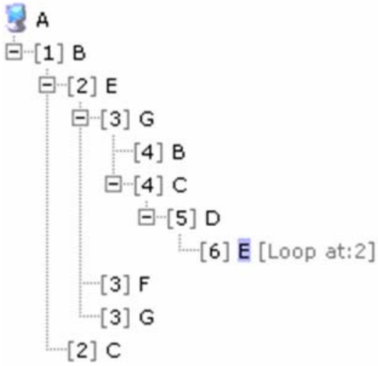


Item	Level of change	Type of change
A	Medium	not selected
B	High	not selected
E	Medium	not selected
G	Medium	not selected
B	Medium	not selected
C	Medium	not selected
I	Medium	not selected
H	Medium	not selected
J	Medium	not selected
F	Medium	not selected
G	Medium	not selected
C	Medium	not selected
D	High	not selected
H	High	not selected
I	Medium	not selected
E	Medium	not selected
D	Low	not selected

Figure 16: Verification Results for 'Loc-only' Propagation Analysis

<u>4) Detailed propagation Analysis with ToC and LoC (Figure 17)</u>

In this analysis, item ‘A’ was selected as initiating item together with ‘Blue’ as initiating ToC and ‘Medium’ as initiating LoC. The analysis is performed until no more items are affected or the propagation is terminated due to loop detection. Six items are identified as possibly affected.



Item	Level of change	Type of change
A	Medium	Blue
B	High	Blue
E	Medium	Green
G	Medium	Green
B	Medium	Red
C	Medium	Green
D	High	Red
E	Medium	Green
F	Medium	Green
G	Medium	Blue
C	Medium	Blue

Figure 17: Verification Results for Detailed Propagation Analysis

B. Verification of Inter-Domain Propagation

As for the verification of the single domain propagation, a bespoke verification model has been developed to test the inter-domain propagation analysis. This model (Figure 18) comprises 18 items, allocated to four domains (indicated by the different background colours for the row and column headings). Three scenarios are used to verify the inter-domain propagation.

	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Req 1	Req 2	Req 3	DP 1	DP 2	DP 3	DP 4	DP 5	DM 1	DM 2	DM 3	DM 4	DM 5
Comp 1									1			1						
Comp 2				1						1		1						
Comp 3	1				1				1				1					
Comp 4			1								1							
Comp 5		1							1				1					
Req 1			1				1	1			1							
Req 2	1				1													
Req 3		1		1		1												
DP 1	1						1						1	1		1		1
DP 2				1			1								1		1	1
DP 3		1			1		1			1		1				1		
DP 4			1					1						1	1	1		
DP 5	1					1			1					1			1	
DM 1									1			1	1					
DM 2										1		1						
DM 3									1		1	1					1	
DM 4										1			1			1		
DM 5									1	1								

Figure 18: Verification Model for Inter-Domain Propagation Analysis

a) Scenario 1

In the first scenario, a component is selected as initiating item. In the Domain Selection DSM, the Components domain is an initiating domain and both the Components and Requirements domains are target domains. Hence, the propagation only continues between the components but at every propagation level, the requirements that are possibly affected by impacted components are identified. Consequently, at every propagation level, the possible affected components and requirements are identified. However, the propagation does not continue for the identified requirements.

target \ init	1	2	3	4
1. Components	☑	☐	☐	☐
2. Requirements	☑	☐	☐	☐
3. Design Parameters	☐	☐	☐	☐
4. Design Models	☐	☐	☐	☐

Figure 19: Scenario 1 - Domain Selection DSM

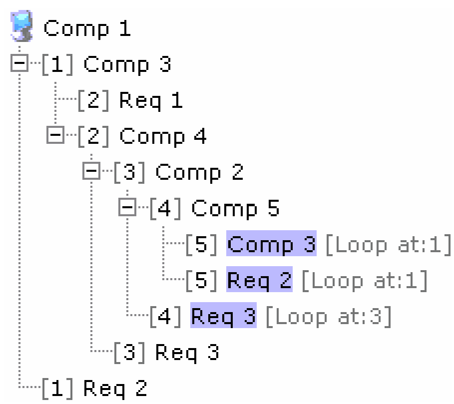


Figure 20: Scenario 1 - Propagation results

b) Scenario 2

In the second scenario, a design parameter is the initiating item. The Design Parameters and Components domains are initiating domains while only the Components domain is a target domain. This means that at the first propagation level, the possibly affected components are identified. In the subsequent propagation levels, the propagation continues only between the components.

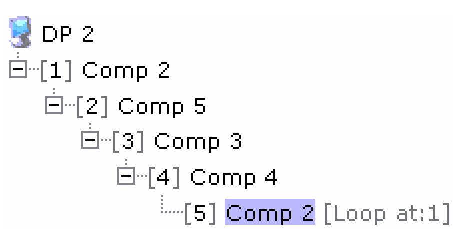


Figure 21: Scenario 2 - Propagation results

target \ init	1	2	3	4
1. Components	☑	☐	☑	☐
2. Requirements	☐	☐	☐	☐
3. Design Parameters	☐	☐	☐	☐
4. Design Models	☐	☐	☐	☐

Figure 22: Scenario 2 - Domain Selection DSM

c) Scenario 3

In this scenario, again a design parameter is used as the initiating item. Now, the Design Parameters and Design Models domains are initiating and target domains. However, the single domain dependencies for both domains are omitted. This means that for the initiating design parameter, only the possible affected Design Models will be identified. Subsequently, for the identified models, the affected design parameters are impacted. The propagation analysis continues accordingly between the design Parameters Domains and the Design Models domain.

target \ init	1	2	3	4
1. Components	☐	☐	☐	☐
2. Requirements	☐	☐	☐	☐
3. Design Parameters	☐	☐	☐	☑
4. Design Models	☐	☐	☑	☐

Figure 23: Scenario 3 - Domain Selection DSM

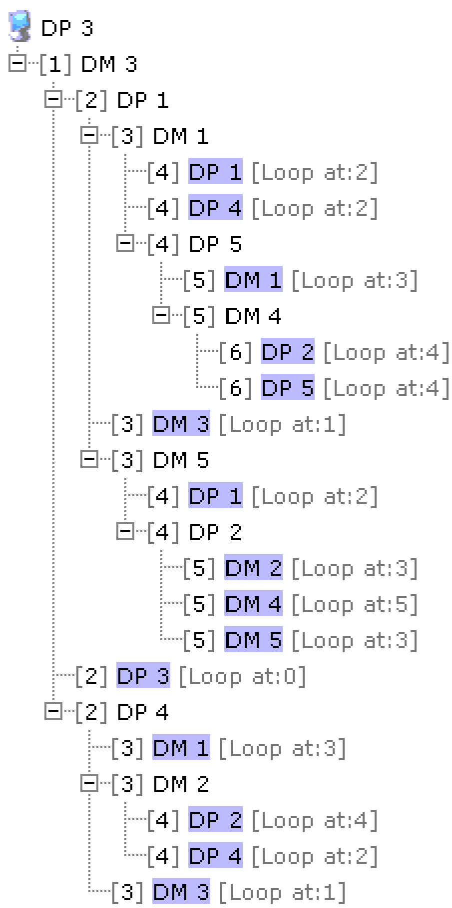


Figure 24: Scenario 3 - Propagation results

IV. Conclusions and further work

The initial evaluation trials indicate that the proposed CPA methods can be very useful, especially with regard to establishing the potential extent of the impact of changes within a complex single domain or across two or more domains. Furthermore, the ability to apply filters and controls such as types of change, levels of change and milestones is also considered very useful, although it was noted that additional tacit knowledge is needed for a proper interpretation of the change impact. Further testing and evaluation of the CPA methods is on-going using a variety of case studies from aircraft development programmes [8]. Several of the domains (or viewpoints) considered are common across the case studies. The case studies also consider the impact of change at different phases of the aircraft development lifecycle. Thus, the major milestones of the development lifecycle can be associated with both the items and the dependency relationships between them. Each case study is running different scenarios to test the CPA control methods described in this paper, within and between different combinations of the domains considered in each model. Finally, the results obtained are being discussed and validated with aircraft architects, engineering discipline experts and programme managers. Future efforts will be directed towards methods assisting the decision maker in trading-off options for localising the impact.

Acknowledgments

The research reported in this paper has been carried out within the VIVACE Integrated Project (AIP3 CT-2003-502917) which is partly sponsored by the Sixth Framework Programme of the European Community (2002-2006) under priority 4 “Aeronautics and Space”. The authors gratefully acknowledge the participation and contribution from Airbus and EADS Corporate Research Centres.

References

[1]Eckert, C., Clarkson, P. J., and Zanker, W., “Change and customisation in complex engineering domains,” *Research in Engineering Design*, Vol. 15, No. 1, 2004, pp. 1-21.

[2]Eckert C M, Zanker W, Clarkson P J (2001) “Aspects of a better understanding of changes”. International Conference on Engineering Design, ICED 01, pp147-154, Glasgow, UK

[3]Danilovic M, Browning T (2004) “A Formal Approach For domain mapping matrices (DMM) to complement Design Structure Matrices (DSM)” Second Cambridge DSM Workshop, Cambridge, UK

[4]Jarratt, T. A. W., Eckert, C. M., Clarkson, P. J., and Schwankl, L., "Product architecture and the propagation of engineering change," *Proceedings of the 7th International Design Conference*, Vol. 1, The Design Society, Glasgow, UK, 2002.

[5]Jarratt, T. A. W., Eckert, C. M., Clarkson, P. J., and Stacey M. K., “Providing an overview during the design of complex products: the development of a product linkage modelling method,” *Proceedings of the 1st International conference on Design Computation and Cognition*, Kluwer Academic Publishers, Dordrecht, Netherlands, 2004.

[6]Ma, S., Song, B., Lu, W. F., Zhu, C. F., “A knowledge-supported system for engineering change impact analysis,” *ASME Design Engineering Technical Conferences*, ASME Publishing, New York, 2003.

[7]Rivière, A., Féru, F., and Tollenaere., M., “Controlling product related engineering changes in the aircraft industry,” *Proceedings of the 14th International Conference on Engineering Design*, Professional Engineering Publishing, Bury St. Edmunds, UK, 2003.

[8]Rutka, A., Guenov, M., Lemmens, Y., Schmidt-Schäffer, T., Coleman, P., and Rivière, A., "Methods for engineering change propagation analysis," *Proceedings of the 25th Congress of the International Council of the Aeronautical Sciences*, ICAS, Stockholm, Sweden, 2006.

[9]Steward, D. V., “The Design Structure System: A method for managing the design of complex systems,” *IEEE Transactions on Engineering Management*, Vol. 28, No. 3, 1981, pp. 71-74.

Appendix

A. Dependencies for single domain model

Initiating Item	Initiating LoC	Initiating ToC	Target Item	Target LoC	Target ToC
A	M	Blue	B	H	Blue
A	H	Blue	C	M	Green
A	M	Red	D	L	Red
B	H	Blue	C	M	Blue
B	H	Blue	E	M	Green
C	M	Green	D	H	Red
D	M	Red	E	M	Green
D	H	Blue	K	H	Green
D	H	Blue	I	M	Blue
E	M	Green	F	M	Green
E	M	Green	G	M	Green
E	M	Green	G	M	Blue
E	M	Red	I	M	Blue
F	M	Blue	G	M	Red
G	M	Green	B	M	Red
G	M	Red	B	M	Blue
G	M	Green	C	M	Green
K	H	Green	A	M	Red
K	H	Blue	I	M	Red
K	M	Green	J	H	Green
I	M	Red	K	M	Blue
I	M	Blue	J	M	Green
I	M	Red	J	M	Green
J	M	Green	D	H	Red
J	H	Green	E	M	Green

B. Dependencies for inter-domain model

Initiating Item	Initiating LoC	Initiating ToC	Target Item	Target LoC	Target ToC
Comp 1	Default	Default	Comp 3	Default	Default
Comp 2	Default	Default	Comp 5	Default	Default
Comp 3	Default	Default	Comp 4	Default	Default
Comp 4	Default	Default	Comp 2	Default	Default
Comp 5	Default	Default	Comp 3	Default	Default
Comp 1	Default	Default	Req 2	Default	Default
Comp 2	Default	Default	Req 3	Default	Default
Comp 3	Default	Default	Req 1	Default	Default
Comp 4	Default	Default	Req 3	Default	Default
Comp 5	Default	Default	Req 2	Default	Default
Comp 1	Default	Default	DP 1	Default	Default
Comp 1	Default	Default	DP 5	Default	Default
Comp 2	Default	Default	DP 3	Default	Default
Comp 3	Default	Default	DP 4	Default	Default
Comp 4	Default	Default	DP 2	Default	Default

Comp 5	Default	Default	DP 3	Default	Default
Req 1	Default	Default	Req 3	Default	Default
Req 2	Default	Default	Req 1	Default	Default
Req 3	Default	Default	Req 1	Default	Default
Req 1	Default	Default	DP 5	Default	Default
Req 2	Default	Default	DP 3	Default	Default
Req 2	Default	Default	DP 2	Default	Default
Req 2	Default	Default	DP 1	Default	Default
Req 3	Default	Default	DP 4	Default	Default
DP 1	Default	Default	Comp 1	Default	Default
DP 1	Default	Default	Comp 5	Default	Default
DP 1	Default	Default	Comp 3	Default	Default
DP 2	Default	Default	Comp 2	Default	Default
DP 3	Default	Default	Comp 4	Default	Default
DP 4	Default	Default	Comp 1	Default	Default
DP 4	Default	Default	Comp 2	Default	Default
DP 5	Default	Default	Comp 3	Default	Default
DP 5	Default	Default	Comp 5	Default	Default
DP 3	Default	Default	Req 1	Default	Default
DP 1	Default	Default	DP 5	Default	Default
DP 2	Default	Default	DP 3	Default	Default
DP 4	Default	Default	DP 3	Default	Default
DP 5	Default	Default	DP 1	Default	Default
DP 1	Default	Default	DM 1	Default	Default
DP 1	Default	Default	DM 3	Default	Default
DP 1	Default	Default	DM 5	Default	Default
DP 2	Default	Default	DM 2	Default	Default
DP 2	Default	Default	DM 4	Default	Default
DP 2	Default	Default	DM 5	Default	Default
DP 3	Default	Default	DM 3	Default	Default
DP 4	Default	Default	DM 1	Default	Default
DP 4	Default	Default	DM 2	Default	Default
DP 4	Default	Default	DM 3	Default	Default
DP 5	Default	Default	DM 1	Default	Default
DP 5	Default	Default	DM 4	Default	Default
DM 1	Default	Default	DP 1	Default	Default
DM 1	Default	Default	DP 4	Default	Default
DM 1	Default	Default	DP 5	Default	Default
DM 2	Default	Default	DP 2	Default	Default
DM 2	Default	Default	DP 4	Default	Default
DM 3	Default	Default	DP 1	Default	Default
DM 3	Default	Default	DP 3	Default	Default
DM 3	Default	Default	DP 4	Default	Default
DM 4	Default	Default	DP 2	Default	Default
DM 4	Default	Default	DP 5	Default	Default
DM 5	Default	Default	DP 1	Default	Default
DM 5	Default	Default	DP 2	Default	Default
DM 3	Default	Default	DM 4	Default	Default
DM 4	Default	Default	DM 3	Default	Default