INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

DESIGN 2010

DEVELOPMENT AND EVALUATION OF A TOOL TO ESTIMATE THE IMPACT OF DESIGN CHANGE

N. Ahmad, D. C. Wynn and P. J. Clarkson

Keywords: information structure framework (ISF), change process

1. Introduction

Engineering change can occur at all stages of the product development and delivery process. It can take the form of change in the requirements for the product, change in the functionality, change in the subsystem(s) / component(s), iteration during detail design, or correction of errors discovered after delivery to the customer. A change occurring during any stage of product development may have varying influence and this influence on the design process depends, in part, on how early the change occurred. The fact that a change also has its own lifecycle, from change initiation through the effects on the direct / indirect connections within each design stage to the effects on the following stages (especially the detailed design process), makes it difficult to estimate its impact.

Due to the ever-present nature of engineering changes throughout product development, there is a need for a method to help assess the impact of change across the different stages and domains of the design process. However, existing change management techniques such as the Change Prediction Method (CPM) [Clarkson *et al*. 2004] are currently limited to a single domain of the design process. For instance, the CPM can investigate the impact of a change to one component of a product on other components, but currently it is not able to estimate the impact of the change on the design process, or to estimate the impact when change is initiated in multiple components or in requirements.

On the other hand, there are design process modelling frameworks that can handle iteration in design, such as the Applied Signposting Method [Wynn *et al.* 2006]. These approaches can assist in understanding the rework in the design process, but these frameworks are not able to identify the rework generated due to an engineering change. The House of Quality (HoQ) is one method that can be used to map customer requests to product attributes which are linked to product components [Hauser and Clausing, 1988]. However, it does not include the impact of change on the detailed design process or help assess the duration of that process. In summary, despite an increase in the understanding of changes there is still need to provide support for designers to:

- Manage changes across different stages and domains of the design process;
- Estimate the impact of change when it is initiated in multiple components;
- Evaluate changes which originate in different design process domains in terms of the activities required to implement them and all the knock-on rework.

This paper presents and evaluates a method to represent design process information which can be used to estimate the impact of change on the design process. Section 3 introduces the proposed Information Structure Framework (ISF) and its implementation. Section 4 illustrates the use of ISF in managing a change process through and describes the implementation of the approach in a support tool. In section 5 a preliminary evaluation of the modelling approach is undertaken based on a case study of changes in a real product. Section 6 summarises the contribution of the paper, outlines directions for further work in this area and concludes.

2. Background

Before introducing the Information Structure Framework developed in this paper it is important to establish the need for information to be captured in the framework. The ultimate objective is to capture the chain of dependencies which link a requirement change to the design activities which must be reworked.

During any product development project customers often modify requirements or request new requirements during the design process, resulting in changes to the product. This is supported by a survey conducted by Deubzer et al., who report that requirement changes constitute 44% of all change requests [Deubzer et al. 2006]. Adding the requirements domain will thus help in tracing change from one of its main sources (requirements) to the rest of the product. Boersting et al. discussed the importance of the relationship between requirements and functions to detect linkages between components of the product [Boersting et al., 2008]. Requirement traceability systems are well-established in many organisations and they are used to track changes initiated by a change in requirements. These requirement traceability methods have a common principle by which they connect the requirements to the functions of the product (and then functions are connected to the rest of the product). In the literature functions have been used to make product models but their role in managing change processes has not been explored in depth. Pahl and Beitz (1995) suggest that building a function structure allows a clear definition of subsystems of the product. Therefore we believe that including the function domain in the information structuring approach may help in increasing the understanding not only of the subsystems but also of the relationship between different components/subsystems of the product.

As mentioned earlier, change management methods look mainly at the impact of change on the subsystems of the product. We know of no method which explicitly traces changes from their source to the detailed design process except the method of Gartner et al. [Gartner et al. 2008]; however, their work only discusses the relation between components and activities. In order to evaluate the true impact of a change on time, resources or cost, it is necessary to evaluate the impact on the detailed design process (in terms of the redesign effort required to implement the change), which requires modelling the relationship between components and the design parameters in a detailed design process. In the Information Structure Framework, the detailed design process domain consists of a set of design parameters and activities. The connections between parameters and activities can be used to identify the activities that need rework, by assuming that that any change in a design parameter will require rework in the immediate downstream activities. The rework caused in a task due to changes in the input parameters might potentially lead to propagation of rework to activities further downstream in the workflow. These cross-domain relationships – requirements-to-functions, functions-to-components, components-to-design parameters and design parameters-to-activities – are the basis of the Information Structure Framework presented in the following section.

3. Information Structure Framework (ISF)

An important assumption of this research is that the product architecture and the architecture of the design process are both relatively stable; thus, Information Structure Framework models comprising the relations outlined above can be constructed once and re-used to evaluate many change requests. It is well-reported in literature that this assumption holds for complex adaptive products such as aero-engines, where the product architecture remains similar across product generations and the design process is well-structured e.g., [Eckert et al. 2004]. We take a layered approach to structure the information required to manage a change process. Each layer on its own is a data repository for information describing one of four domains in the design process. The layers also contain links within themselves and to other layers. The information is structured into four different layers: 1) Requirement layer; 2) Functional layer; 3) Component layer; and 4) Detailed design process layer. Figure 1 shows the overview of the ISF and the interconnections between these layers, where 1 → * shows a one-to-many relationship. Each layer is discussed in detail below. Figure 2 shows an example ISF model for a real product, the AUTOBELL (a microcontroller-based product described in detail in Section 5). This will be used as an illustrative example in forthcoming sections.

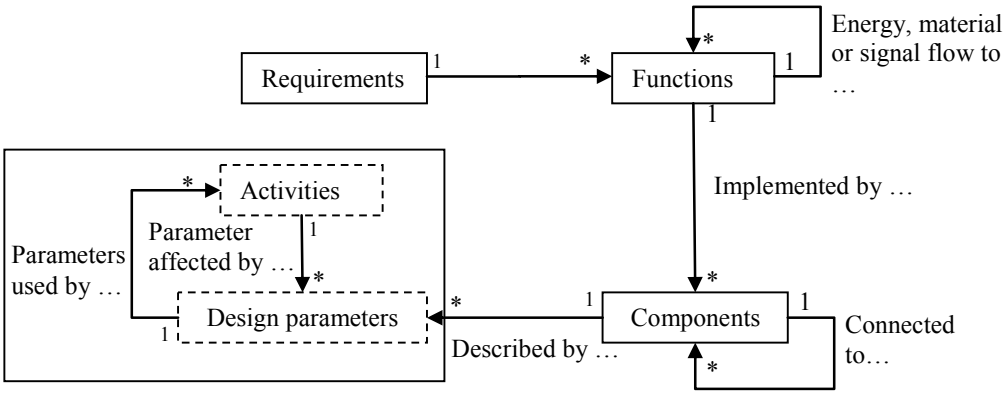

Figure 1. Overview of the Information Structure Framework (ISF)

3.1 Requirement layer

The requirements serve as a channel of communication between customers and the product development team. Figure 1 shows the connections a requirement will have with the function domain in the ISF.

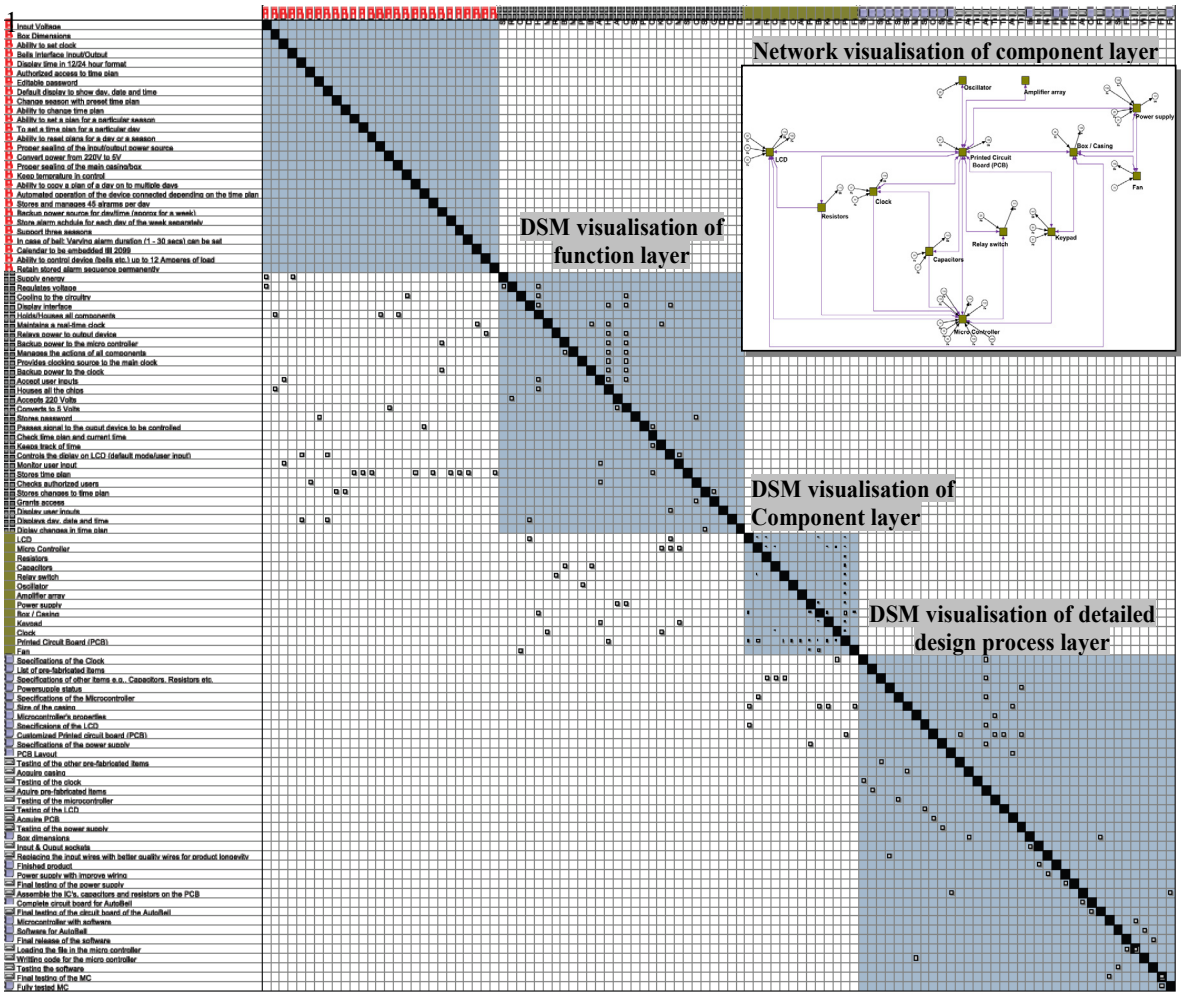

Figure 2. An example ISF Model for the AUTOBELL

3.2 Function layer

A function can have three types of connections in the ISF: 1) from requirements; 2) to other functions within the function layer (through flows of energy, material and/or signals); and 3) to implementing

components in the component layer (see Figure 1). This allows identification of: the functions affected by any change initiated in requirements; the change propagation to other functions through flows, which can be used to identify the indirectly affected components; and the components affected directly by a change in a function.

3.3 Component layer

A component in the ISF can have three types of connections: 1) to the functions that a component implements, since a change to a component may mean that its functions have to be changed; 2) to other components; and 3) to the detailed design process layer where a change in a component modifies the value of a design parameter resulting in rework of downstream activities. The second type of connection is adapted from the Change Prediction Method (CPM) [Clarkson et al. 2004]. These connections help to identify: the components directly affected by any change initiated in either requirements or functions of the product and the resulting change propagation to other components through linkages; and the design parameters changed which will cause direct rework in activities.

3.4 Detailed design process layer

The detailed design process is modelled as a network of activities each of which acts on input design parameters and produces other design parameters as output. Each component's design is broken down into a set of such design parameters. The relationship between parameters and components is such that if there is a change in a component it may only affect some of its design parameters, but changing a design parameter will always cause change in the component it defines. This relationship helps to identify the activities required to implement a change and hence avoid overlooking necessary work. Adding the detailed design process layer to the traceability model is not the only way to estimate the cost of change, as it can also be done through looking at the design cost of the affected components [Keller 2007]. On the other hand, activity-based costing is more useful as it can also be used to estimate the time and resources required to complete rework. In the ISF, each design activity can have two types of connections: 1) inputs from the design parameters which feed into the activity, such that the activity must be performed each time any of them is changed, and 2) output to design parameters that are updated when the activity is performed. This helps to identify: the direct rework required in the activities to implement a parameter change, and the indirect rework in the downstream activities to implement a change.

4. Change management support approach

Once an ISF model of a product has been constructed, it allows systematic analysis of the impact of a change request (*request for change in requirements*). Figure 3 below outlines the steps which are executed to trace the impact of the change request through all the subsequent domains of function, component and detailed design process:

1. **Identify the function(s) subject to change.** The first step of the method is to identify which functions of the product will have to change to implement a given requirement change. In building the ISF model, requirements were linked to the functions which are responsible for fulfilling them, so such function identification is straightforward when following these links. Designers can select or deselect function(s) based on their experience.
2. **Identify the component(s) subject to change.** The next step of the method is to select the components of the design that will be redesigned to implement the change in the requirements. The ISF model maps each function to one or more subsystem(s)/component(s) as explained in the previous section. So, the component(s) subject to change will be the one(s) implementing the affected function(s)/sub-function(s) found in step 1. Designers can select or deselect components based on their experience and judgement of how the specific function change will be implemented in the physical domain. The component(s) identified in this step are termed 'initiating components' as they may cause propagated changes to other components of the product.

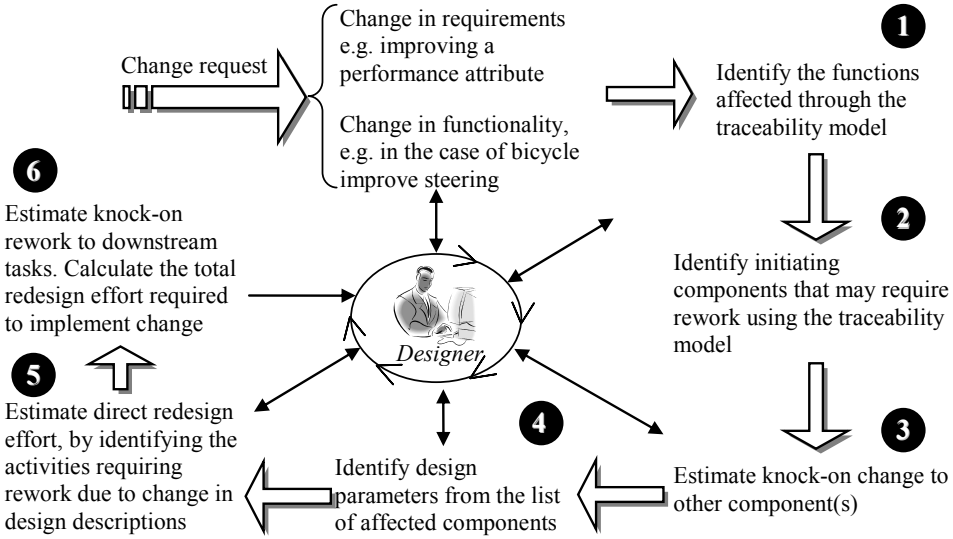

Figure 3. Overview of the proposed approach

Estimate knock-on change to other components. This step identifies the set of product components that will require redesign due to change propagating from the components identified in the previous step. Knock-on changes to other components can be identified using linkages within the component layer. The Change Prediction Method (CPM) is used to calculate the likelihood of change propagation between components, where change is initiated by the component(s) identified in Step 2. However, CPM can only give the likelihood of change propagation to a component from a single initiating component, whereas in reality there may be multiple initiating components. , hence a need to calculated a combined value of likelihood from these components to the rest of the components (a simple heuristic to calculate this combined value is explained in section 4.1).

3. **Identify design parameters which must be modified**. This step identifies the design parameters which must be modified to change the product components which were identified in Step 3. The detailed design process layer consists of activities and parameters which are inputs to activities. These parameters may potentially be modified whenever there is a change in a component.
4. **Estimate direct redesign effort.** The list of design parameters identified in Step 4 is used to obtain the list of directly affected activities. The relationship between a design parameter and an activity in the detailed design process layer is such that each activity, for completion, requires input in the form of parameter(s) and produces output in the form of parameter(s). The direct redesign effort will be the rework required to complete all the activities which have the modified design parameter(s) as inputs.
5. **Estimate knock-on rework.** Information flows in the process model are used to identify knock-on rework. This is based on the assumption that a change to the design parameter(s) which are input(s) to an activity will propagate to cause change to the design parameter(s) which are output(s) from the activity. In the case where the downstream activities affected by these design parameters are not yet executed then this will not be considered as rework. Propagation of rework through activities in the detailed design process is analogous to propagation of change through elements of the product. The amount of resources and time required to complete all activities identified could then be estimated to determine the cost and duration of implementing the change.

4.1 Combining the likelihood of change arising for multiple sources

CPM described by Clarkson et al., [Clarkson et al. 2004], can be extended to cope with multiple initiating components using a simple heuristic. Figure 4 (left) shows the combined likelihood matrix obtained for the example product using CPM, giving the likelihood of change

propagation from each component (considered as a single initiating component) to every other component. To calculate the likelihood of change from two sources propagating to cause change in one component, the heuristic uses a function f of the two relevant single-initiating-component likelihood values, x and y. Any such function must fulfil the following criteria (where x and y can have values between 0 and 1):

$$0 \leq f \leq 1 \quad (1)$$

$$f < x + y \quad (2)$$

$$f \geq max(x,y) \quad (3)$$

$$f(x,y) = f(y,x) \quad (4)$$

$$f(f(x,y),z) = f(x,f(y,z)) \quad (5)$$

The rationale is as follows. Condition (1) ensures that the probability of change in a component remains between 0 and 1. Condition (2) is necessary because when there is change in two components together, the overall likelihood of change may be less than the sum of the two individual values. For instance, it might be feasible to change both components together as it may reduce the possibility of knock-on rework in the other components of the product . However, condition (3) ensures that the overall value is always greater than the individual likelihoods.

Components

1. LCD
2. Microcontroller
3. Resistors
4. Capacitors
5. Relay Switch
6. Oscillator
7. Amplifier array
8. Power supply
9. Box / Casing
10. Keypad
11. PCB
12. Clock
13. Fan

Combined Likelihood Matrix

	1	2	3	4	5	6	7	8	9	10	11	12
1		.64	.43	.24	.11	.11	.11	.34	.26	.25	.54	.22
2	.15		.33	.35	.09	.09	.09	.22	.11	.27	.45	.27
3	.08	.26		.10	.05	.05	.05	.12	.05	.11	.25	.10
4	.08	.26	.10		.05	.05	.05	.12	.05	.11	.25	.10
5	.11	.40	.15	.16		.07	.07	.16	.07	.15	.33	.14
6	.08	.26	.10	.10	.05		.05	.12	.05	.11	.25	.10
7	.08	.26	.10	.10	.05	.05		.12	.05	.11	.25	.10
8	.15	.38	.15	.15	.07	.07	.07		.23	.18	.37	.15
9	.38	.75	.31	.29	.16	.16	.16	.75		.42	.79	.28
10	.15	.40	.16	16	.08	.08	.08	.27	.23		.39	.15
11	.32	.92	.34	.35	.20	.20	.20	.45	.20	.36		.35
12	.10	.31	.12	.35	.06	.06	.06	.15	.07	.13	.31	
13	.26	.55	.23	.22	.11	.11	.11	.58	.62	.30	.55	.21

$f(x,y) = \sqrt{(x^2 + y^2)}$

The combined likelihood is calculated as follows:

1. Set '*x*' to be the likelihood value of the first component(s).
2. Set '*y*' to be the likelihood value of the second component.
3. Use the function *f(x,y)* to get the combined likelihood value.
4. In case of more than two initiating components set 'x' to be the results of *f(x,y)* and set 'y' to be the next value.

Figure 4. Multiple initiating components (Initiating components are highlighted)

There may be many functions that satisfy all these conditions. The function that has been used in this paper is shown in Figure 4. It is not only sufficient to pick a function that will satisfy these conditions but the result should be reflective of the two values being combined. Function 'f' spreads the value across the range of possible values as defined in eq. 1 - 3. However, there is one problem with this function in that it violates condition (1) for some values of x and y. In order to avoid this, the value is reduced to 1 whenever it exceeds 1 so it will satisfy condition (1). If there are more than two values to be combined then the function is applied to the first two values, then the result is combined with the third value using the same approach and so on.

As an example, to calculate the combined likelihood of change propagating to 'Box / Casing' when change is initiated in 'LCD' and 'microcontroller': the likelihood values of 'LCD' (x1 = 0.38) and 'microcontroller' (x2 = 0.75) are combined resulting in xT = 0.84 (the relevant values are circled in Figure 4). If there are more than two initiating components, xT will be combined with the likelihood value of the third component and so on.

4.2 Implementation

A software application was developed to implement this systematic approach for estimating the impact of change The implementation provides interactive options to try different permutations at all the stages of the method, allowing the user to explore their effects. A change is initiated by a change in requirements. Once the changed requirements are selected they are drawn in the drawing panel (Figure 5, top left), then when the 'Next' button is pressed it shows all the functions that may be changed due to a change in this requirement. The energy, material and signal flows between these functions are also drawn (Figure 5, top right). The user has the option to select any functions that may require change. The next stage shows the components implementing the affected functions selected in the previous stage. This is divided into two sub-steps to accommodate the possibility of change propagation in the components. In the first sub-step the components are selected by the user, and subsequently considered the initiating components, as these components may cause propagation of change in other components. In cases of more than one initiating component, the merged likelihood values are calculated using the heuristic discussed in section 3 and shown in a table (Figure 5, bottom right). The components are sorted and presented alongside the estimated change likelihood, at which point, the user must consider this guidance and select those components which they believe will require direct or indirect change in this specific case. Finally the redesign process is generated automatically for this particular change case, showing all the downstream activities and design parameters requiring change (Figure 5, bottom left).

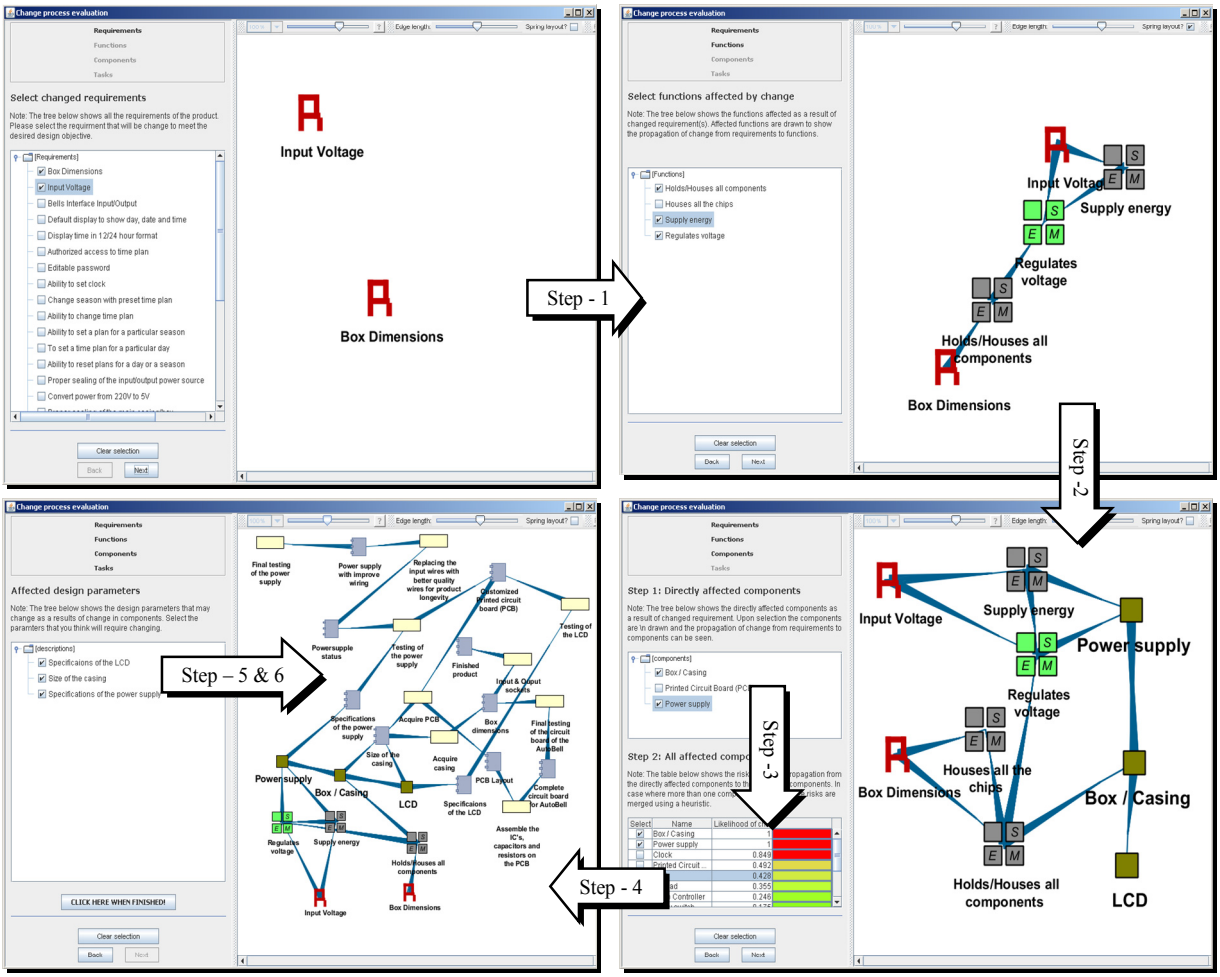

Figure 5. Change management support tool

5. Preliminary evaluation

To evaluate the approach, 10 researchers and research students were given a test case based on a real change example from an electronic product called the AUTOBELL. The objective was to evaluate whether the support tool and knowledge structuring framework allowed subjects with no background knowledge of the product to more quickly and more consistently identify the activities required to implement a change, compared to subjects of a similar background provided with only the product documentation used to build an ISF model of the product.

5.1 Overview of the AUTOBELL

The AUTOBELL is a microcontroller-based device to schedule timings. It is a product of Digital Research Labs (DRL) (www.drl.com.pk). This scheduling device can be used for scheduling in different environments, e.g. shifts in factories or lessons in schools. The AUTOBELL was designed and developed as a product of DRL and not based on customer requirements. As there was no fixed customer, changes were expected depending on the different requirements from different prospective customers. Each change is seen by DRL both as a potential problem and as an opportunity to further develop the product. Figure 6 shows the internal structure of the product (showing some components and few linkages between them). The AUTOBELL operates on 220V but the internal circuitry operates at 5V. So a power supply is used to regulate the voltage to 5V. A microcontroller is the main controlling IC of the system and the software to store the timetable and monitor the time is loaded in it. It constantly checks the time from the real time clock and generates an output signal to the 'relay switch' at the appropriate time.

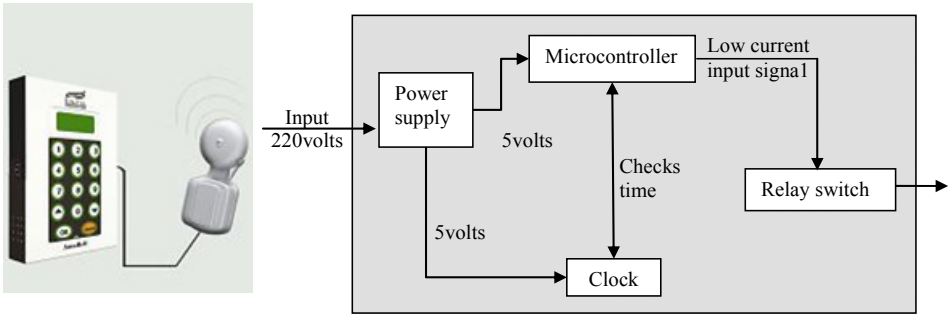

Figure 6. Finished product (left), Internal structure (right)

The AUTOBELL was designed with an initial requirement to allow the users to change the time plan. The time plan includes the time table for the alarms. In the initial design, setting the number of alarms was not considered to be a potential requirement and the AUTOBELL was designed to ring the bell once each time. A request by one of the customers to be able to configure the product to have multiple rings per alarm led to a change in the requirement 'Ability to change time plan'. Apparently this seemed to be a straight forward problem as it did not require any change in components. The change was to be made in the program written in the microcontroller's program memory which will change the functionality of the AUTOBELL. Once the initial changes were made in the software of the microcontroller everything seemingly worked fine. However, during rigorous testing the AUTOBELL began hanging up and not functioning properly.

The problem was cause by the fact that the microcontroller used in the AutoBell has 8 Kbytes of Flash Programmable ROM. Before this change was made the memory was just suitable for every feature offered by the AutoBell. So, when the device was hanging one designer suspected 'low memory' to be the problem and thus it was decided that the microcontroller needed to be changed to an updated version which would have sufficient memory. In practice, changing the microcontroller resulted in changing the PCB, capacitors, resistors, casing and fan. Having the knowledge of the detailed design process alongside the components of the product would have helped to avoid unnecessary rework as various design activities were repeated during this whole exercise because the overall impact of change was not recognised.

5.2 Modelling the AUTOBELL using the ISF

The AUTOBELL design was modelled, using the ISF, by the authors using documentation provided by the manufacturer supplemented by telephone interviews with the AUTOBELL designers. The model was sent to the designers to check for any inconsistencies. Figure 2 shows the resulting ISF model of the AUTOBELL grouped into requirement, function, component and detailed design process layers. The ISF model is shown as an MDM in Figure 2 as it is easier to view the connectivity across different domains but the model is made using the network view (Figure 2 - top right corner shows network visualisation of the component layer).

A list of specifications for the product was used to build the requirement layer, and the links were created between the requirements and the functions of the product. In this case the AUTOBELL requirements were the basis of its function structure, as it falls in the category of an '*original design*'. However, it is important to note that requirements may not always be the basis for the function structure of a product, since in '*adaptive design*' an earlier product may be used as the basis of the function structure. The top-most function in the function structure is 'manages time' and using the requirement specification of the AUTOBELL this was broken down into sub-functions to make the function structure of the product. "*The number of levels of decomposition in the function structure and the number of sub-functions in each level depend on the novelty of the problem"* [Pahl and Beitz 1995]. In this case the function was not decomposed further when each requirement was implemented by at least one function. Each sub-function can have three possible types of flows (inputs/outputs) to other functions: 1) energy flow (blue colour lines); 2) signal flow (green colour lines); and 3) material flow (red colour lines). In the case of the AUTOBELL there were only two types of flow between different functions, energy and signal flows (see Figure 2).

The component layer consists of components and the linkages between them making the construction similar to the component-based DSM by Browning [Browning 2001]. The exercise of building a component layer / product model consists of: 1) decomposing a product into subsystems/components; 2) recording the linkages between these components and 3) estimating the impact and likelihood of a change propagating directly between a pair of components, as in the CPM approach [Clarkson *et al*. 2004]. Likelihood and impact values were estimated during telephone interviews. In the case of AUTOBELL the main components were very clearly defined. However, as a general rule for decomposing products into components, Pimmler and Eppinger [Pimmler and Eppinger 1994] suggest decomposing a product one more level than what is desired as it will enable the ability to cluster the elements at the desired level. It is important to highlight that a balance is required between making the model too complex and omitting the detail necessary to represent a product.

Building the detailed design process layer consists of decomposing the product development process into activities and identifying the input and output information of the activities as 'design parameters'. The resulting ISF model shown in Figure 2 may seem quite complex and knowledge-intensive but most of the information that is represented in the model was already available. The function layer of the product was built from scratch taking six hours to complete. The requirement layer was built using the list of specifications of the product. The links in the component layer were built during the first telephone interviews along with the estimation of likelihood and impact values. The activities in the detailed design process were elicited using the documents provided (the project plan and the production process document) and were validated during the second telephone interview. The time taken to build the requirement, the function, the component and the detailed design process layer was 1 hour, 6 hours, 3 hours and 5 hours respectively, making a total of 15 hours to construct the ISF model for the AUTOBELL.

5.3 Evaluating the approach

Two change cases were developed using information elicited from the AUTOBELL designer.

1. **First case.** The Auto-Bell was designed with an initial requirement that allows the user to store a time plan for three different seasons. In the initial design, varying the number of seasons was not considered to be a potential requirement and the AUTOBELL was designed to support a fixed number of seasons only. One of the customers wanted a product that could

be configured for varying number of seasons between one and four; thus leading to a change in the requirement "Support three seasons".

2. **Second case.** The Auto-Bell was designed with an initial requirement to operate at an input voltage of 220V. One of the customers wanted a product which can be configured to operate at both 110V and 220V; which led to a change in the requirement "Input Voltage".

An experiment was undertaken to test the following hypotheses:

1. The change process evaluation tool allows a user unfamiliar with the product to predict the effect of changes on the design process in less time than a similar user without the tool.
2. The change process evaluation tool allows a user unfamiliar with the product to predict the effect of changes on the design process more consistently than a similar user without the tool.

5.3.1 Methodology

As discussed earlier, the experiment was structured to be performed by two different groups, each containing 5 participants. Group 1 was given the support tool alongside the ISF model for the AUTOBELL and the original product information, whereas Group 2 only had the original product information. This document was originally used to build the ISF model and it consisted of a numbered list of requirements, functions, components, main steps in the detailed design and the production process (which also contained embodiment diagrams) of the AUTOBELL. Apart from this all the subjects were handed another document which included the change cases to be evaluated and the step by step instructions to carry out the experiment. For Group 1 (with the tool) the different screens of the tool were explained so they could run the experiment independently; subjects in this group did not need to record anything as the tool generated a report containing the results of their evaluation. For Group 2 (without the tool), a form was provided with sections to note the functions, components and the design activities affected by a change.Subjects in Group 2 were also asked to record the start and finish times for evaluation of each change case. The results of the preliminary evaluation are presented and discussed below.

5.3.2 Results

Results of the preliminary evaluation are summarised in Figures 7, 8 and 9. Each bar chart in Figure 7 and 8 shows the results of both change cases for all five subjects in a group. The time taken by both groups to evaluate change cases is shown in Figure 7. Generally, it indicates that subjects in both the groups took less time to evaluate the second change case (light pattern bars) because they were more familiar with the product than when they were evaluating the first change case (black bars). Moreover, on average, the time taken by Group 1 to evaluate both change cases is less than Group 2. *The results of this experiment show that the change management support tool allows a user unfamiliar with the product to predict the effect of changes on the design process in less time than a similar user without the tool.*

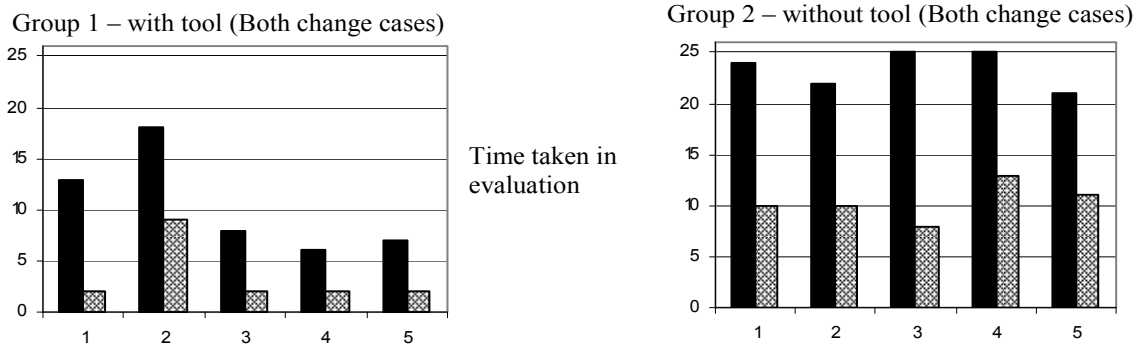

Figure 7. Time taken to evaluate both change cases

Figure 8 (left) shows the activities picked by all subjects in group 1 for the evaluation of the first change case. The activities picked by each user are shown in sets and each set is shown in a different colour. This Euler diagram shows that a total of 12 activities were picked by all the subjects and out of

these 12 activities 10 of the same activities (3, 9, 10, 11, 12, 13, 14, 15, 16, 17) were picked by every subject in this group, which is shown by the area in the figure where all the sets overlap (the Euler diagrams are made using the tool explained in [Wyatt et al., 2009]). This shows the consistency of results obtained by the group using the tool. Whereas, Figure 8 (right) shows the activities picked by all subjects in group 2 for the first change case. Out of a total of 17 activities selected by different users only 4 same activities (12, 13, 14, 15) were selected by everyone in this group.

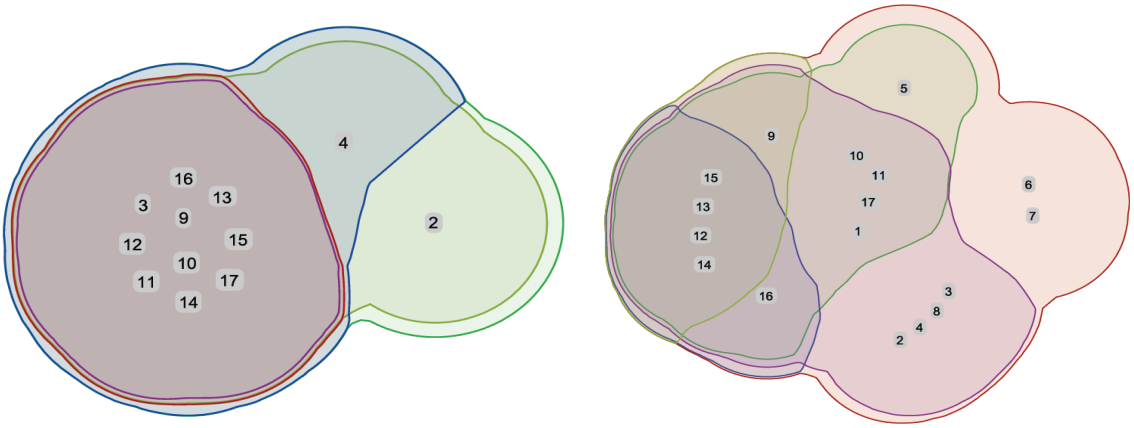

Figure 8. Euler diagrams of activities; Group 1 – with tool (left), Group 2 – without tool (right)

The consistency of the results was further analysed by plotting the total number of functions, components and activities that may require change / re-work picked by both groups, shown in Figure 9. The bar charts clearly indicate more variability in the results of Group 2 than Group 1. The number of functions, components and activities picked by subjects in this Group 2 varies between them, whereas the results obtained by Group 1 are more consistent and importantly the lists of activities are almost identical across the group. An additional benefit is that the dependencies among these activities were also clearly identified as they were picked using the tool. This information could be used to sequence the redesign process, avoiding unnecessary rework. *The results of this experiment show that the change process evaluation tool allows a user unfamiliar with the product to predict the effect of changes on the design process more consistently than a similar user without the tool.*

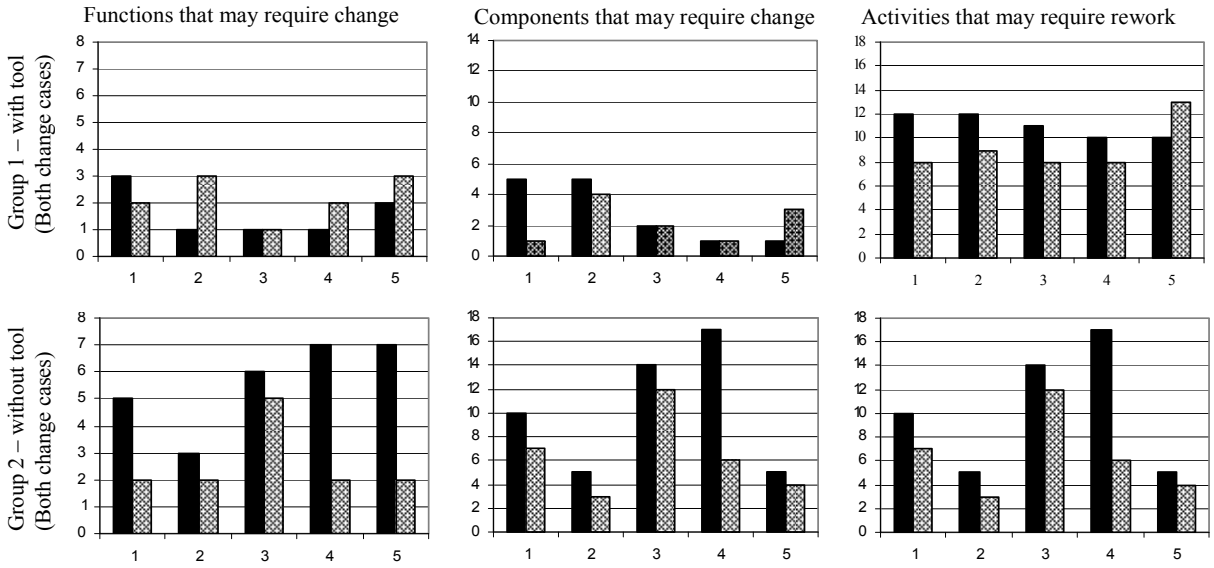

Figure 9. Summarising the results of the preliminary evaluation. X-axes show group numbers, Y-axes show number of elements selected

6. Discussion and conclusion

This paper presents a new way of representing information across different domains of the design process. A design example was used to show how an ISF model should be constructed, and to illustrate the change management approach. The main assumption of the ISF is that the architecture of the product and the design process are both relatively stable; thus, ISF models can be constructed once and re-used to evaluate many change requests. While the method is limited by high initial cost of building an ISF model, it can be reused across multiple product variants and design projects. Tracing a change request across different stages of the design process and knowing the impact on each stage could benefit designers in managing a change process effectively.

The main contributions of this paper are: 1) It identified the need to consider the impact of changes across different stage of design, by estimating the knock-on rework in the detailed design process alongside the more widely discussed knock-on effect of changes to the product, in order to effectively manage change processes. To achieve this, a new modelling framework has been developed which can be used to build cross-domain models; 2) A method and tool to support designers in effectively managing a change process was proposed, based on systematic consideration of the cross-domain model. The model was used in a preliminary evaluation of the support method, which indicates the benefits of using the change management approach. The approach enabled a test group with the tool to achieve more consistent results in less time than another group solving the same problem without the tool; and 3) The need to address the case of change initiation in multiple components was identified. A heuristic to calculate the likelihood of change when change is initiated in multiple components was proposed.

References

Boersting , P., Keller, R., Alink, T., Eckert, C.M., Clarkson, P.J. (2008). "The Relationship Between Functions and Requirements for an Improved Detection of Component Linkages", International Design Conference (DESIGN 08), Dubrovnik – Croatia.

Browning, T.R. (2001). "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions", IEEE Transactions on Engineering Management, Vol.48, pp. 292-306.

Clarkson, P.J., Simons, C.S. and Eckert, C.M. (2004). "Predicting Change Propagation in Complex Design", Journal of Mechanical Design, Vol.126, pp 788-797.

Deubzer, F., Kreimeyer, M., Lindemann, U. (2006). "Exploring Strategies in Change Management – Current Status and Activity Benchmark", International Design Conference (DESIGN 06), Dubrovnik – Croatia.

Eckert, C.M., Clarkson, P.J. and Zanker, W. (2004). "Change and Customization in Complex Engineering Domains", Research in Engineering Design, Vol.15, pp. 1-21.

Gärtner, T., Rohleder, M., & Schlick, C.M. (2008). Simulation of Product Change Effects on the Duration of Development Processes based on the DSM, In 10th International DSM Conference, Stockholm, Sweden, November 2008, pp. 199-201.

Hauser, J.R. and Clausing, D. (1988). "The House of Quality", Harvard Business Review, Vol. 66, pp. 63-73.

Keller, R. (2007). "Predicting Change Propagation: Algorithms, Representations, Software Tools", PhD thesis, University of Cambridge, United Kingdom.

Pahl, G., Beitz, W. (1995). "Engineering design: A systematic approach", 2nd Ed., Springer, London, UK, 1996..

Pimmler, T.U. and S.D. Eppinger (1994). "Integration Analysis of Product Decompositions", ASME Design Theory and Methodology Conference, Minneapolis, MN, USA.

Wyatt, D. F.;Wynn, D. C.;Clarkson, P. J. (2009). "Exploring spaces of system architectures using constraint-based classification and Euler diagrams", Proceedings of the International Design Structure Matrix Conference (DSM '09), Greenville, South Carolina, USA.

Wynn, D.C., Eckert, C.M., Clarkson, P.J. (2006). "Applied Signposting: A Modeling Framework to Support Design Process Improvement", Proceedings of ASME IDETC/CIE 2006, Philadelphia, Pennsylvania, USA.

Naveed Ahmad
The University of Cambridge, Engineering Design Center, Department of Engineering
Trumpington Street, CB2 1PZ, Cambridge, United Kingdom
Email: na315@cam.ac.uk