A function-behaviour-structure framework for the lifecycle of an artefact

Ruth Stalker
Computing, Lancaster University
Lancaster, UK, LA1 4YR
01524 593116
ruth@comp.lancs.ac.uk

ABSTRACT
The design prototypes schema is extended to become a function-behaviour-structure artefact lifecycle schema. This extended schema provides a framework in which the artefact, the changes it undergoes during its life, and its interaction with the artefact user aid the design engineer. The potential of the framework is explored by discussing the support it offers for function reformulation, serendipitous, creative and innovative design, and the avoidance of design fixation.

Keywords
function, behaviour, structure, design, user, artefact, lifecycle

INTRODUCTION
In [1] a schema called design prototypes was proposed in order to consider the elaboration of artefact models in terms of function, behaviour, and structure. It provided support for initiating and continuing the design act whilst distinguishing between routine, creative, and innovative design. The schema has been adopted and applied in many areas. Pre-project planning, knowledge-based systems, and design by analogy [2,3,4] are three examples of its employment. Additions to the schema, e.g. context [5] and situatedness [6], have been proposed in order to enrich the knowledge required to perform design. It has been observed that the need to reformulate an artefact's function and behaviour diminishes throughout the design process but does not disappear [7].

In this paper the design prototype schema is extended to consider the whole lifecycle of an artefact, The artefact is tracked from birth to death through time (A FUNCTION-BEHAVIOUR-STRUCTURE ARTEFACT LIFECYCLE). When an artefact physically exists the context in which it is situated is easily identified and may affect its function. Processes such as monitoring and intervention examine how the artefact behaves and functions. This can be fed back into the process of function reformulation.

Comparison of artefact behaviour and purpose through time may enable design engineers to avoid functional fixation when performing reformulation. Observation of multiple functions of one artefact, particularly in a user's hands, allows the design engineer to find new ways of focusing on the design task. Thus, the space of possible functions is exploited (COGNITIVE APPLICATIONS). To observe users employing an artefact it is necessary to bring the user into the FBS framework as significant differences between the intended and actual purpose (or function) of the artefact can occur. Users make serendipitous discoveries of how a given artefact may be used or how a variation of the artefact can lead to new designs. Motivations for artefact misemployment may be necessity, fashion, ignorance and evolution. Thus, the evaluation of user satisfaction and examination of how an artefact is used can serendipitously lead to creative and innovative designs. Such design is supported by the extended schema processes, for example repair/retrofit and dismantlement (recycling). Again results are fed back into the process of design reformulation (CREATIVE ENCOURAGEMENT).

A FUNCTION-BEHAVIOUR-STRUCTURE ARTEFACT LIFECYCLE
The original design prototype schema has been changed in order to corresponds to the somewhat longer term design process favoured in this paper. These changes are the incorporation of time into the schema and a construction transformation that represents the design description, documentation and fabrication stages in the lifecycle of a given artefact. Using this augmented design prototype the paper presents the extended function-behaviour-structure (FBS) artefact lifecycle schema and descriptions of the additional tasks are given.

The design prototype schema
In [1] a schema was proposed in order to consider a designed artefact prototype in terms of function, behaviour, and structure. In Figure 1 the schema is augmented in

order to represent temporal aspects using $_t$. $_0$ and $_{t^0}$ indicate time before the prototype artefact physically exists. $_{t^n}$ indicates the many iterations the artefact design may have gone through before the design is settled upon and becomes an artefact. The construction transformation represents the design description, documentation and fabrication stages in the lifecycle of a given artefact. Iterations are not shown in the diagram.

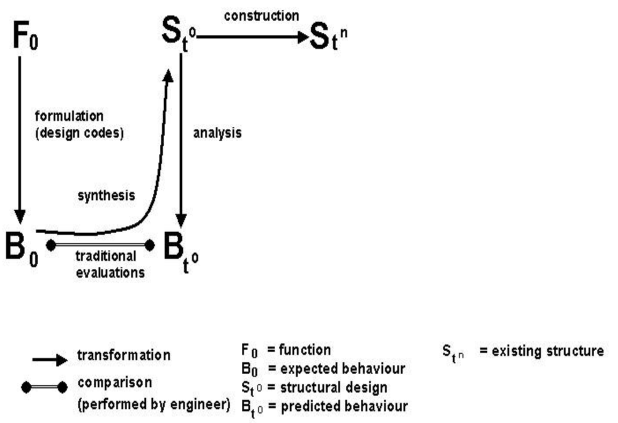

Figure 1: Information is classified into function, behaviour, and structure. Tasks, such as synthesis, analysis and construction transform this information from one classification to another (after [1]).

Function   is a set which refers to requirements and objectives which cannot be directly transformed into a set representing structural description possibilities S_{t^0} without first anticipating desired, or expected behaviour. Therefore, functional objectives are formulated in terms of expected behaviour B_0. The task of synthesis uses expected behaviour B_0 to provide a set of structural descriptions S_{t^0}. Many structural descriptions S_{t^0} may be formulated and iteratively refined. The transformation, or synthesis, of expected behaviour to a structural description S_{t^0} is a difficult task. Structural description S_{t^0} is a geometrical description of the artefact with the topological configuration of types of elements. S_{t^0} may contain material properties, environmental and user effect on the artefact. For example, issues of robustness are represented here.

The iterative process of F_0 to B_0 to S_{t^0} to B_{t^0} and comparison of behaviours B_0, B_{t^0} is performed through the tasks of formulation, synthesis, analysis, and traditional evaluations until a suitable structural design description is decided upon. The task of construction uses the selected structural design description S_{t^0} in order to create an actual, physical, structure S_{t^n}.

The extended FBS lifecycle

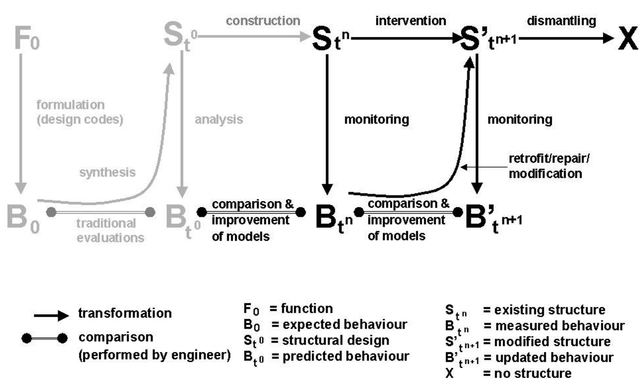

Figure 2: The schema is extended to include construction, monitoring, modification and dismantling. Transformations are iterative (not shown in figure).

The extension of this framework was first proposed in [8] with specific attention paid to the management of knowledge in bridges.

The tasks which differ and are extended from the original framework are considered below and are shown in Figure 2 in black.

Construction

In this extended schema the production of the design description was combined with construction.

The set of S_{t^0} is used in order to create a physical, structure S_{t^n}. Research questions proposed in [1] such as 'how do additional functions appear as a design progresses?' can be answered by using an actual artefact prototype. This approach is used in software development. Requirements engineering (function formulation) often leads to iterative prototyping and models such as Bohem's spiral model [9] are applied to the prototype through its lifecycle in order to assess the need for ad-hoc problem solving which can be viewed as case of function reformulation and its expression [10]. Artefacts which have a long lifecycle, for example bridges, have design documents that are no longer relevant or non-existent when function reformulation is desired. Thus, engineers use other tasks such as comparison and monitoring to understand what functionality and behaviour the artefact embodies. Therefore, the construction transformation embodies the design description and documentation as they may not exist as separate transformations.

Comparison

By comparing the proposed or expected behaviour B_{t^0} and the actual existing behaviour B_{t^n} routine designs can be vastly improved. In a more creative situation B_{t^n} can indicate for future reference what B_{t^0} will give.

Monitoring

The actual structure S_{t^n} is linked to the actual behaviour B_{t^n} by the process of monitoring. This may include sophisticated fibre optic networks for bridges in order to measure current behaviour [11] or it could usability testing of a given artefact with actual users using techniques suggested in [12].

Here it can be seen if add-on or afterthoughts in the design are actually used and focussed upon more readily by the user than the initial design purpose. One example of this is the popularity of SMS text messaging. This was not the original intent of the mobile phone but messaging demands have led to the development of better mobile phone interfaces. The user's perception of an artefact is developed through interaction with the artefact. This may be quite different to the designer's perception of the artefact. Thus, monitoring can identify these differences. The differences may lead to the designer intervening with the structure of the artefact in order to change the artefact's behaviour. Or, feed the monitoring results back into the reformulation process.

Intervention

Sometimes due to user reaction or artefact behaviour an intervention needs to be made. By monitoring and comparison of the expected and actual behaviours, B_{t^0} and B_{t^n}, areas of redesign – or function reformulation - can be indicated. The task of intervention creates a "new" structure $S'_{t^{n+1}}$. The prime indicates an intervention and t is again indicative of time. Although a new functionality has been added the design engineer and user still must be aware that time has passed and this intervention is either a restriction or addition of functionality. In engineering this could be due to safety, e.g. load restrictions on a bridge. In software development it could be added functionality, e.g. more features in the graphical-user interface. Such interventions have the potential to change the behaviour and performance of the artefact. The intervention stage should be viewed as a secondary construction phase and include sub-transformations of a description of redesign and documentation for future reference.

Retrofit/modification/repair

Sometimes it is insufficient to intervene to add or subtract functionality and it becomes necessary to repair or modify the actual artefact. An updated $S'_{t^{n+1}}$ may only lead to an updated $B'_{t^{n+1}}$ by modifications. This may be viewed as a reformulation of function but on a smaller scale. The FBS schema diagram show the retrofit/repair/modification transformation arrow to be the same as the original synthesis arrow in order to reflect the nature of this transformation. Retrofit/repair/modification is a reformulation of function later in the lifecycle of an artefact. It is redesign of an existing artefact. Interventions and modifications are analysed and monitored in order to measure how the function reformulation reflects the artefact's behaviour and structure.

Dismantling

Finally, an artefact may no longer be of use in its present form and should be dismantled or recycled. Parts of an artefact can be reused. Others must be scrapped. This dismantling transformation can be viewed as reverse engineering. Reverse engineering is a useful teaching tool particularly in user-interface design [13] and if some aspect of an interface (or artefact for that matter) is not clearly understood creative and innovative design proposals may ensue. Good ideas are recycled and evolve over time. Recycling a function from a dysfunctional artefact can lead to a better generation of artefact. Costs often dictate that developing prototypes to throwaway is not a good solution – it is practiced in software engineering [14] - so recycling ideas is a cheaper alternative particularly when dealing with routine design.

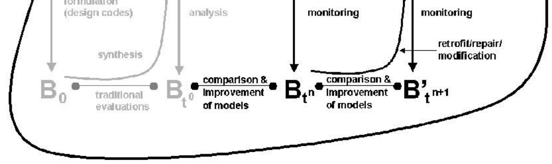

Figure 3: Feedback from various stages enables the reformulation of function stage that may lead to creative and innovative design (Not all feedback is shown).

Figure 3 illustrates how on dismantling the recyclable ideas of the artefact are fed back into the design process. The added functionality or functional parts of the whole which makes up the artefact may be broken down and used as a focus for the functionality of the next artefact prototype.

With the FBS extended schema a framework is given in which cognitive applications such as reflections with respect to time, comparison and artefact architecture can lead to creative encouragement by serendipity, evolution, reinvention and adaptation.

COGNITIVE APPLICATIONS

The use of a framework in which to view an artefact encourages a cognitive approach to artefact development. The designer thinks about the affect of time on an artefact, comparison of snapshots of the artefact through its lifecycle as well as the same artefact in different contexts, and uses the framework to focus and exploit the resources available for better design without becoming fixated.

Time

The addition of time to the framework enables the design engineer to look at the artefact and see if the initial design has been able to produce the behaviour that was expected. Time enables the identification of a design fault (detected early in the lifecycle) or wear and tear in an artefact (detected later in the lifecycle). Both have an affect on the artefact function and behaviour.

The identification of when in an artefact's life it was used for a different purpose or when it fell into disuse may enable the designer to anticipate these moments and plan for them when reformulating design.

By tracking the life and death of a given artefact areas of recycling and monitoring may be improved which would ultimately feed back into the function formulation stage.

One way of using time is to compare snapshots of artefact behaviour throughout the lifecycle in order to trace function reformulation.

Comparison

The comparison of behaviours during the lifecycle, for example B_{t^0} and B_{t^n}, may show discrepancies between expected and actual behaviour. Trying to make these behaviours more similar, by improving the initial design representation of functional requirements, may improve the design process. This idea is used in engineering. Comparison of different behaviours can more easily enable the identification of better ways of modelling artefacts (design solutions) [8].

The employment of the same artefact in different contexts by a user can lead to varied behaviour. The comparison of this behaviour may lead to interesting design observations.

Task Focus

Using the lifecycle of an artefact as opposed to just the design stage may lead to cognitively exploiting user input and design engineer fixation.

User exploitation

When a user interacts with an artefact in an unanticipated way the designer can exploit this by intervening and trying to identify other functions for the artefact. In this way the functional space is explored and exploited [15].

Fixation

Research has shown that looking at typical design solutions for a given problem may lead the designer to become unable to think of alternative solutions [16]. Using the FBS framework to consider an artefact that has been adapted by a user for a different function could encourage the designer to think of multiple-functionality in the initial design phases. Designing artefacts with multiple-functionality may be a solution to design fixation.

CREATIVE ENCOURAGEMENT

Serendipity, reinvention, evolution, and adaptation are ways that an artefact user can inadvertently reformulate function within the FBS schema. Through transformations of intervention, retrofit, monitoring and dismantling the designer can change the structure and behaviour of an artefact to meet reformulated function and perform innovative design.

Serendipity

The process of serendipitous design has been discussed in [17]. Serendipity has been the reason for many artifacts. One example is of the post-it note [18]: A glue was invented by one 3M employee and used by another to keep his bookmarks in place. This eventually led to the post-it. Other happy accidents are the discovery of vulcanised rubber [19], nylon stockings [20] and Velcro.

The tracking of an artefact through its lifecycle can lead to innovative design. The FBS supports serendipitous design in the monitoring and transformation stage particularly when seeing how a user employs an artefact. (Motivations for why a user may use an artefact in a different context are given in the adaptation section.) Serendipity can occur more slowly through the evolution of an artefact over time.

Evolution

Evolutionary design normally brings to mind genetic algorithms and other artificial intelligence techniques. In this paper a simpler definition is given just by looking at the artefact in the framework. The Microsoft operating systems are an example of how software has evolved over time in the way that artefacts do. Questions that won't be addressed in this paper but may be of use are:

- Should the artefact design allow or anticipate for different functionality over time?
- How much does fashion affect the design and evolution of an artefact?
- How do afterthoughts become a major part of artefact design? As in the SMS messaging example given earlier.

Reinvention

Function reformulation can be found without updating or upgrading the artefact. For example, a spoon is primarily an artefact for eating with but viewed through the eyes of some users the spoon becomes a musical instrument. The only intervention that may occur is that the spoon is no longer used for eating but it is stored in a separate area ready for playing. Have there been patents for an improved spoon for better music making? Is this because there is not the market demand for it? Or is it because the spoon is ideal as an eating utensil and a musical instrument at the same time. Again do other motivating factors such as cost, fashion and experience play a role? In order to determine the influence of such factors it is useful to monitor user satisfaction with a given artefact. Research has shown that user satisfaction can determine the success or failure of an artefact [12] or piece of software [21]. At the same time however, motivations such as cost, convenience and fashion can make a user adapt an artefact to the task in hand.

Adaptation

Reasons for adaptation:

- Comfort and convenience. A user may feel comfortable with an artefact and not want to change his style of working.
- Ignorance or laziness. A user may be unaware that the artefact he need already exists.
- Necessity. Cost and time constrain the user's desire to 'shop' around and find the artefact he needs.
- Fashion. A fashion conscious user may be ready to adapt an artefact in order to remain trendy.

To name but a few. It may be worth reflecting on the motivations for a user adapting an artefact in order to focus on function reformulation. In all cases the user is an integral part of the extended reformulated design process throughout the lifecycle of an artefact.

The User

The role of the user in the FBS extended framework can be added in at all stages from the initial functionality discussions which may be expressed through requirements engineering to the monitoring stage where usability engineering and evaluation techniques can indicate where areas for future functionality may lie.

One interesting area is the area of user interpretation of an artefact and a designer's intent as discussed in [12]. How intuitive is any given artefact? How useful is an artefact for a user? What happens if a user can't interact with the artefact as desired? And can't find another use for it? The danger of involving the user too much in the design process is that the user doesn't always know what he wants [22]. Therefore, pulling the user into the FBS framework should be done primarily to see how a user interacts with an artefact once it exists. It is at this point. The point in the diagram where the framework is seen in black and not grey that innovation and creativity can be found.

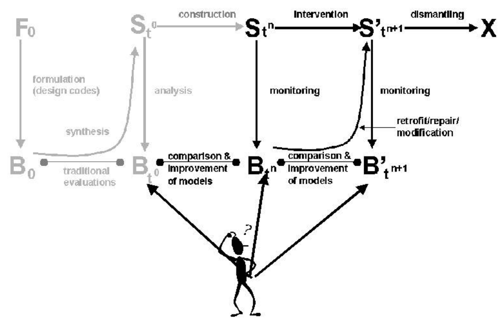

Figure 4. Feedback from the user is included in the FBS framework. Misemployment of an artefact leads to function reformulation, serendipitous and innovative design.

CONCLUSIONS

The extension of the design prototypes schema to become a function-behaviour-structure artefact lifecycle schema provides a framework for long term design. The artefact is tracked through its life and the changes that occur to it can be used to reformulate functionality or change its structure and behaviour. Such transformations are performed serendipitously through evolution, adaptation and misemployment by the user. Monitoring user interaction and satisfaction with an artefact may lead to innovative and creative design and avoidance of design fixation.

ACKNOWLEDGMENTS

The author would like to thank David Brown and Alan Dix for feedback and discussion on this framework.

REFERENCES

1. Gero, J. Design Prototypes: A knowledge representation schema for design *AI Magazine 11*, 4 (Winter 1990), 26-48.
2. Divita, E.; Kunz, J.; Fischer, M. ``Collaborative desktop engineering''. International Meeting on Artificial Intelligence in Structural Engineering, July

26-31, 1998, Centro Stefano Franscini Monte Verita Ascona, Switzerland, 69-85

3. Coyne, R; Rosenman, M; Radford, A; Balahandran, M; Gero, J. *Knowledge-Based Design Systems*, Addison-Wesley, Reading, 1990
4. Balazs, M.E.; Brown, D.C.; A preliminary investigation of design simplication by analogy, in *Proceedings of AID'98*, Kluwer Academic Publishers, 517-534
5. Gero, J. S. and Maher, M. L. (1997) A framework for research in design computing, *in* B. Martens, H. Linzer and A. Voigt (eds), *ECAADE'97*, Osterreichischer Kunst und Kulturverlag, Vienna
6. Gero, J. S. (1998). Conceptual designing as a sequence of situated acts, in I. Smith (ed.), ***Artificial Intelligence in Structural Engineering,***Springer, Berlin, 165-177.
7. Gero J. S. and Kannengiesser, U. The situated function-behaviour-structure framework. 2002 *in progress*.
8. Stalker, R.A.. *Engineer-computer interaction for structural monitoring*, Ph.D. thesis, EPFL, 2000.
9. Bohem, B. *A spiral model of software development and enhancement*. ACM SIGSOFT Software engineering notes, 1986
10. Sommerville, I., Sawyer, P. *Requirements engineering*, John Wiley & Sons, 1998
11. Inaudi D., Vurpillot, S. , Casanova, N., Kronenberg, P. Structural monitoring by curvature analysis using interferometric optic sensors, *Smart Material Structures*, 7, 1998, 199-206
12. Norman, D. *The design of everyday things.* New York: Doubleday/Currency, 1990.
13. Greenberg, S. Teaching Human Computer Interaction to programmers**.** *ACM Interactions*, 3(4), 1996, 62-76
14. Pressman, R., adapted by Ince, D. *Software Engineering: A Practionners Approach, Third Edition, European Edition,* McGraw-Hill, 1994
15. Smithers, T., Towards a knowledge level theory of design process, in *Proceedings of AID' 98*, Kluwer Academic Publishers, 3-21
16. Purcell, T. A. and Gero, J. S. Design and other types of fixation, *Design Studies* **17**(4): 363-383, 1996
17. Guindon, R., Krasner, H., Curtis, B. *Breakdowns and Processes During the Early Activities of Software Design by Professionals*, chapter 5, pages 65--82. Ablex Publishing Corporation. Norwood, NJ, 1987
18. http://www.3m.com/about3m/pioneers/fry.jhtml
19. http://www.invent.org/book/book-text/47.html
20. http://www.pbs.org/wgbh/aso/databank/entries/dt35ny.html
21. Brown, J., Earnshaw, R., Jern, M., Vince, J. Visualisation. Using computer graphics to explore data and present information, *John Wiley*, 1995
22. Nielson J., Usability Engineering, *AP Professional*, Boston, 1994

▪