

The Simulator; Another, Elementary Problem Frame?

Ian K Bray
Bournemouth University
ibray@bournemouth.ac.uk

Karl Cox
CSE, University of New South Wales,
National ICT Australia
karlc@cse.unsw.edu.au

Abstract

Problem framing is recognised as an important new technique for describing the problem context in order to identify and then apply the most appropriate modelling and description techniques to developing different kinds of software system. Jackson has proposed five distinct problem frames and variants on them. However, Jackson makes no claim that this is an exhaustive list. In order to better model certain problem domains, an addition to the published set of elementary problem frames is proposed. We present the Simulator problem frame and describe through example why the Simulator is not solely a variant of the other frames and should be considered a problem frame in its own right.

1. Introduction

Since being proposed by Michael Jackson [1], problem frames have been recognised as having great significance for the development of software systems. As David Garlan observes [5], "Understanding and using problem frames will likely become an essential skill of all good software system designers".

The approach is grounded upon fitting problem frames to the given problem domain; the closer the fit, the greater the advantage that may accrue.

Jackson originally identified five elementary problem frames; workpiece, control, information, connection and transformation. As has been illustrated in several works (for example, [2], [3]) more complex problems can, generally, be fitted to composite frames composed of two or more interacting, elementary frames. Whilst Jackson explicitly made no claim to have identified all the elementary frames, as far as is known, to date, no others have been added. However, to better fit a particular class of problem, an additional frame is now proposed.

2. Problem frames and Requirements practice

Methodologists often suggest that general methods are better than specific and the "one size fits all" notion is superficially attractive. However, it is flawed; as

Jackson [1] puts it, "the value of a method is inversely proportional to its generality". Use cases, for instance, might well prove useless for describing the functionality of a simulacrum.

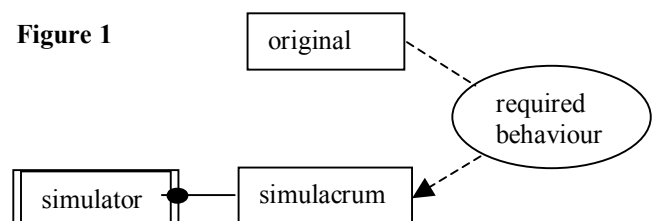
Nonetheless, enormous effort gone into trying to wrap one "standard" approach around all problems and the uptake of general methods is widespread. Meanwhile, little effort has gone into assessing the effectiveness of applying specific problem frames to particular problems and there is a currently lack of published evidence. In the only empirical study conducted on problem frames thus far, Phalp and Cox [8] show that it is relatively straightforward to determine single problem frames but less easy to determine the frames (and their respective weightings) for multi-frame problems. There are, as far as we are aware, no published industrial case studies, of the application of problem frames. However, given a free choice of methods, many of our students adopt this approach in their final individual and group projects. (Indeed, one such project sowed the seed for this paper.)

3. The simulator problem

This class of problem is characterized by the construction of an artifact (a realised domain) which exhibits behaviour that, to a required degree, mimics the inherent behaviour of some dynamic, real world original.

The simulation bears some resemblance to an external reality but this is established only through the requirements; there is no communication (no phenomena are shared) between the original and the simulation during operation. Even so, the original should not be ignored, it is central to the problem and investigation and description of the original would be a vital element of the requirements engineering.

The problem frame may be pictured as in figure 1.



The original is a “real world” domain. It is suggested that the original must be dynamic (or it would have no behaviour to simulate) but that is the only constraint¹. In particular, it may be controllable or autonomous. The required behaviour will, clearly, be based upon the inherent behaviour of the original but (as illustrated in the following examples) it is likely that there will be approximations or simplifications.

The simulacrum is the realized domain which behaves in accordance with the required behaviour. It is an embodiment of the selected event responses and exists only within the simulator, the configured machine that brings about the desired behaviour of the simulacrum.

The simulator frame is likely to be used in conjunction with other frames. In particular, although it could be trivial, an information system may well be required in order to view the state of the simulacrum. (This is akin to the simple workpiece frame requiring an information system to provide feedback [5], page 98.) It is also possible for the inherent behaviour of the simulacrum to be controlled by an associated control frame. However, these associated frames are optional; the simulator can stand alone and this prohibits its classification as a partial frame.

4. Examples

Consider a simulation of a planetary system. The original has certain properties (mass and orbital period of bodies etc.) which can be detailed in a problem domain description. The requirements will state with what degree of fidelity the simulation should mimic the original. There will inevitably be some inexactitude and one might, for example, elect to treat the bodies as point masses and constrain movement to a single plane.

Once implemented, the simulation should comply with the requirements. However, during its operation, the model does not communicate (share phenomena) with the original and at any time, the state of the model may be very different from that of the original.

A second example concerns the simulation of ocean waves. The problem domain description would detail such matters as the propagation and decay of actual waves. The requirement would specify how closely the simulation must mimic these characteristics and the implementation should comply with those requirements.

The simulation could be used to make predictions about maximum wave heights, to provide test conditions for (simulations of) boats or simply as a

piece of art. In none of these cases, however, does the simulation communicate (share phenomena) with the original or represent any actually occurring state of the original.

Thousands of examples of simulation problems appear in the literature, these include:

- simulation of chemical etching
- simulation of buildings (for earthquake research)
- simulation of vehicles and drivers (as part of a driver training environment)
- simulation of electronic circuits
- simulation of brewery fermentation (as a test rig for a new brewery controller)

5. Differences from other frames

The simulator shares certain characteristics with other elementary frames but has particular features which justify its separate classification. In part, the differences hinge upon the distinction between inherent behaviour and controlled behaviour. The inherent behaviour of a domain is a given. It is not open to negotiation but may be described more or less correctly. A complete description of the inherent behaviour of a domain encompasses all its possible behaviours. Bjorner et al. [4] define this in terms of a basis function which determines the range of its observable variables.

The simulator seeks to approximate the inherent behaviour of the original. If it is possible and desirable to impose a controlled behaviour over the inherent behaviour, this would be the subject of a separate (control) frame.

Jackson [5], page 339, offers a ranking of problem frames in terms of their “depth in the world outside the computer”. This ranking runs from the transformation frame (the shallowest) to the control frame (the deepest) and may be viewed in terms of the degree of binding between the machine and the real world. Within this scheme it is suggested that the simulator problem fits between the workpiece and the information frame; the behaviour of the simulacrum is based upon some real world original (which, therefore, requires study) but during operation, there will be no communication between the simulator and the original.

6. Why isn’t it a workpiece problem?

Both workpiece and simulator contain a realized domain (artifact). In the workpiece case, the realized domain is necessarily inert and it changes subject only to the whim of an editor. The simulacrum can be self modifying and is constrained by requirements that establish, to a greater or lesser degree, some mimicry of an external reality.

¹ There may be some argument as to whether a model of a static domain (for example, a road in a traffic simulation) constitutes a simulation in its own right.

An inert domain may be simulated but, in the absence of any stimuli, will do nothing. It is hard to imagine the circumstances under which the simulation of an inert domain would, of itself, be worthwhile.

An inert domain simulation could, however, have a role as a partial frame. It would embody the event responses that would allow some other machine to exert control over the simulacrum. This approximates to the decomposition of the workpiece problem into two partial frames; a simulator and an “editor”.

However, there remains a difference in that, with the simulator, there does exist an external original which, through the requirements, imposes constraints upon the behaviour.

Furthermore, the simulator has a far greater range; spontaneous and, even, autonomous domains may be simulated. Even in the absence of any external stimuli, such domains may “freewheel” and continually exhibit behaviour. The bodies in the planetary simulation, for example, continue in their orbits without the need for any external stimulation and, in a brewery simulation, a vat of wort will continue to ferment if left alone.

7. Why isn't it an information problem?

An information system provides information about some reality. In order to do this, the information system will frequently contain a model of that reality (what Jackson refers to as the Dynamic Model, Partial Frame [6]). However, in that case, the correspondence between the reality and the model must be maintained (within acceptable limits) during the operation of the information system.

This is not the case with a simulator. At any given time, the state of the simulacrum need have no correspondence with the state of the original. It follows that the simulator frame diagram shows no connection between the original and the simulator; there is no communication, no shared phenomena. For example, a vehicle in a traffic simulation may crash but this certainly does not mean that any original has crashed.

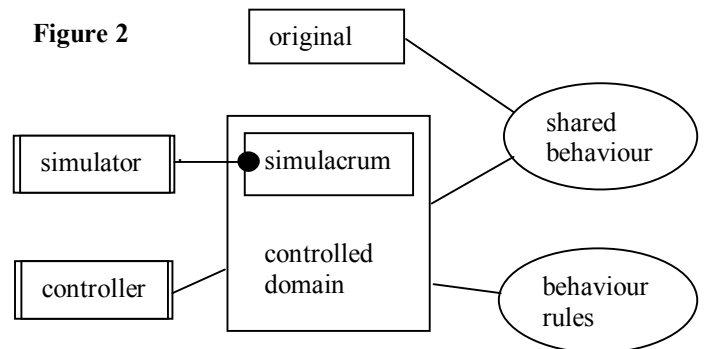
In practical terms, correspondence between the simulacrum and the original is determined only by the requirements; there is no updating of correspondence at run-time. That said, a simulation could form a partial frame in support of an information system *if* its purpose were to help derive reports about the actual state of the original *and* if there were some mechanism for ensuring an ongoing correspondence with the original. But this need not be the case.

8. Why isn't it a control problem?

The control problem is characterised by the machine exerting control over some external reality.

In the case of the simulator, the external reality (the original) is not controlled (not even connected). There could, of course, be an associated control system for which the simulacrum *is the reality*. This could be depicted as below (figure 2), but the simulator and the controller are separate systems. This situation commonly occurs where a simulator is built in order to provide a test environment for a control system (without any risk of accident cause by defective control of the original).

Figure 2



However, a telling fact is that it is possible to simulate an autonomous domain (such as a weather system) which, by definition, cannot be controlled.

Even where control is possible, it is not *necessary* to have any planned control. Simulators for two or more systems could be interconnected such that each provides stimulation for the other and, without any requirements for controlled behaviour, they could be left to interact and their emergent behaviour can simply be observed. Certain simulations of evolving systems fit this pattern.

9. Is the Simulator a lexical domain in disguise?

Jackson [5] describes a lexical domain as a "physical representation of data - that is, of symbolic phenomena" (p84). The lexical domain combines a mix of causal and symbolic phenomena. Because of this necessity for causal phenomena and its need to change the state of data in the domain of interest, we can say that a lexical domain interacts with the problem entity it is representing. However, the simulator does not interact with its original entity in any way. It does not change the state of its real world representation. It only mimics it or simulates behaviours beyond the capability of its original counterpart.

Jackson also presents another point of view that a lexical domain does not need to be causal, but that it is just "a structure of symbolic phenomena" (p84). For the Simulator frame, the lexical domain could be the combination of the simulacrum and the simulator machine. This lexical domain would then represent the original real world domain. However, since the original

domain is, necessarily, dynamic (e.g. controllable or autonomous) and lexical domains are often static representations of data, we argue that the simulator has particular qualities that distinguish it from a typical lexical domain. With the advent of machine learning/genetic algorithms and AI, there is no reason to suppose that the simulator will never go beyond the capabilities and behaviours of the original domain (perhaps to explore ways in which to enhance the original). For instance, in the simulation of fault tolerance of house safety structures under extreme conditions such as an earthquake or a tornado. To await an actual tornado or earthquake to assess the effectiveness of the building materials or architecture would be impracticable.

10. The fantasy problem frame

A further possibility is that, within a simulation, the original is imaginary. This situation occurs with certain games, computer animations and the like.

An imaginary object, for example, a dragon, is endowed with certain behaviour and a machine is constructed that exhibits that behaviour. As before, it is quite possible that there will also be a connected system that provokes the simulation (with or without the aim of evoking a certain desired behaviour).

References

- [1] M. A. Jackson, "Requirements & Specifications: A Lexicon of Software Practice, Principles and Prejudices", Addison Wesley, 1995.
- [2] D. Jackson and M. A. Jackson, "Problem Decomposition for Re-use"; *Software Engineering Journal*, Volume 11, Number 1 pages 19-30, January 1996.
- [3] B. Kovitz, "Practical Software Requirements; A Manual of Content and Style", Manning, 1999.
- [4] D. Bjorner et al., "Michael Jackson's Problem Frames: Towards Methodological Principles of Selecting and Applying Formal Software Development Techniques and Tools"; *ICFEM'97: Intl. Conf. on 'Formal Engineering Methods'*, Hiroshima, Japan, pages 263-271, Los Alamitos, CA, USA, 12-14 November 1997
- [5] M. A. Jackson, "Problem Frames: Analyzing and Structuring Software Development Problems", Addison Wesley, 2001.
- [6] M. A. Jackson, "Problem Analysis Using Small Problem Frames", *South African Computer Journal*, Volume 22, pages 47-60, March 1999
- [7] I. K. Bray, "An Introduction to Requirements Engineering", Addison Wesley, 2002.
- [8] K. Phalp, and K. Cox, "Picking the Right Problem Frame - An Empirical Study", *Empirical Software Engineering Journal*, 2000, 5(3), pp. 215-228.
- [9] ProSim - Software Process Simulation and Modelling International Workshop Series, <http://www.prosim.pdx.edu/prosim2003/index.htm>
- [10] *The Sims* computer games series, Electronic Arts Games.
- [11] NASA, Smart Systems Research Lab, <http://ssrl.arc.nasa.gov/techsim.html>

11. Conclusions and further work

Any development that allows closer fitting of problem frames to problem domains should prove advantageous and the proposed new frame appears to offer the possibility of achieving a better fit for certain types of problem.

Development of case studies will allow further investigation. In particular, the nature of the various domains can be more precisely determined and checklists (along the lines of those provided in [3], page 82 and in [7] page 103) for the investigation and documentation of the problem can be developed.

We are currently (through our students) exploring the efficacy of the problem frame approach and are in the process of beginning new industrial liaisons both in the UK and Australia where we plan to evaluate the problem frames in a variety of different industries. With the rapid growth of simulation modelling, for instance as software process simulation and modelling (e.g. the ProSim workshops [9]), through leisure activities [10] to safety critical assessment, for example, a simulation of docking a Space Shuttle to a space station [11], it is absolutely imperative to be able capture all the necessary domain elements of the problem. For simulation problems, the simulation frame is a definitive place to start.