A Study of Formal Model of Interactive Behaviour in the Whole System for Implicit Action and Context

Bo Li*, Yanmin Xue

Department of Art and Design, Xi’an University of Technology, Xi’an, Shaanxi, 710054, China
*Corresponding author’s e-mail: ekinshow@sina.com

Abstract. Formal techniques have been used to analyze a variety of properties of interactive behaviour in a number of systems. To have a better understanding of the implicit of the interactive behaviour, it is necessary to find a general method to describe the task. In this study, we proposed a formal model to illustrate the implicit task. We validated the effect of this model by an experiment in the simulated coach bus station by receiving transition information by the holder held device. This paper analyses the whole system, the rooms, the public displays, the mobile devices and the timeliness of information in the system. The model made the interactive process clearer and provided a basic framework to study the related issues of orientation in the interactive system in the future.

1. Introduction

Formal approaches bring rigor and automation to usability engineering. Models capture key features of the interactive behaviour of the design and are then subjected to systematic analysis using verification techniques such as model checking (Clarke, Grumberg and Peled, 1999). The appropriate selection of models and verification techniques is critical to the value of the analysis. Otherwise models can bias the analysis and verification techniques can take the focus away from important safety or business-critical features of the designed interactive system. Two categories of automated verification techniques can be identified. An important challenge if these techniques are to be used effectively and efficiently is that there be generic models of interactive behaviour that can be taken off the shelf by the engineer and instantiated to the particular requirements of the system under design.

The research focuses on formal methods such as statechart diagram as a different sort of model for interactive systems. Instead of modelling the user like cognitive models, these models consider the system and the context for the system. They can then be reasoned about to ensure that they function as they should. This is particularly important in safety critical system, like coach bus control or medical monitors, where the cost of failure is never acceptable.

Although formal techniques are widely used in the analysis of interactive systems there are reasons why an appropriate set of tools, suitably designed to be usable by system engineers, could be of value in the portfolio of techniques used to assess interactive systems. This paper describes the role of formal techniques in modelling and analyzing interactive systems, discusses unfulfilled opportunities and speculates about the removal of barriers to their use. It also presents the opportunities that a clear expression of the problem and systematic analysis techniques may afford.

The model is based on concurrently executing automata that can be synchronized by using labels on the state transitions: a transition can occur in two parallel automata if complementary labels occur on arcs in the two automata and both arcs are enabled at the same stage in the execution of the system.
The approach used Uppaal, which additionally focused on continuous time (Behrmann, David and Larsen, 2004). Models based on these techniques are good for describing behaviour of actions in the interactive system in terms of the actions to which they will lead. Models can be used to analyze the mode characteristics of an interactive device (Gow, Thimbleby and Cairns, 2006). There are also textual approaches to modelling labeled transition systems such as there. For example, SPIN (Holzmann, 2003) captured the model using a process algebra called Promela. Other approaches, for example Alloy (Jackson, 2006) and Modal Action Logic (Campos and Harrison, 2001), captured relations between states described textually in terms of typed attributes of the states.

2. Verification in development

The value of formal (mathematically based) models of interactive behaviour of systems is in providing precise descriptions of features of the interactive behaviour of the design early in the process. The features focus on user action, whether explicit or implicit, and the observable effects that action has in the context. This makes it possible to analyze characteristics of the behaviour of the system before expensive commitments have been made to implementation. In our research, a model approach is taken that share a number of features in common. It is diagrammatic approaches based on finite state model which capture properties of the system in terms of states and transitions between states.

As briefly summarized, analysis explores behaviours of a device. An important feature of the behaviours of interest is that they correspond to envisaged uses of the device. The mobility of the designed hand-held device may have consequences for interaction. The analysis is designed to relate to a number of concerns including, for example, whether these behaviours lead to confusing mode shifts or whether there are situations in typical sequences where it is impossible to recover. As discussed in Campos (1999), the process of establishing usability requirements, modelling systems and checking the models against requirements can be thought of as a cycle consisting of four steps, as represented in Figure 1. These steps may involve discussion between software or system developers on the one hand and human factors specialists on the other. The cycle is as follows:

![Figure 1. The circle of verification in development.](image)

1. Identifying usability requirements: Based on usability guidelines or heuristics or specific concerns derived from a requirements elicitation. They are used to assist the development of a model to be used for the analysis. They involve device descriptions or whole system descriptions – a device might be a mobile phone, the system is the whole system in which the mobile phone is embedded. These requirements are expressed as formal properties.

2. Building model and formulating properties: Once the initial requirements have been expressed, a model of the interactive device or system and a set of properties are formulated. This step benefits from the participation of someone with specific user or domain expertise. Models capture
assumptions about the user’s view of the device or how users are embedded along with devices inside the system. Properties capture constrains that if broken represent potential failures in the use of device or system. This stage could be achieved by re-using an old model or instantiating a generic model.

(3) Verifying properties: The next stage is to verify whether the properties hold of the device under the constraints imposed by the environment—for example, a property requiring the visibility of actions but only over the paths that relate to expected tasks. Constraints are either expressed in the properties themselves or through additional models capturing relevant characteristics of the environment in which the device is to be considered. The properties could be examples of general property patterns.

(4) Analyzing the results: Finally the consequences of the results are analyzed. The technique used in our research is model checking. The method of analysis involves checking a property. If the answer is positive then, under the assumptions captured by the model, the usability property is true of the system described by the model. If the answer is negative then the reasons why are investigated. A rationale explaining why the property does not hold is provided by the reasoning tool, in this case a trace representing the confluence of conditions in which the property fails. A negative result might indicate that the device model is incomplete. It may therefore be necessary to determine whether all relevant aspects of the device have been considered. Alternatively it might point to a situation where the assumptions made about the user must be refined. For example, it may be necessary to eliminate specific courses of action from the expected user behaviour. Finally, it might be a genuine usability problem.

3. Modeling implicit action and context in the whole system
We provide a model of the system that can be used to check properties for the coach bus station. The model focused on usability issues associated with interaction with a mobile device within a system. Usability issues may also be associated with the characteristics of the whole system, for example the way in which information is presented to users in different locations within a built environment. The issues that are of concern are about whether implicit actions will have desired effects—for example, will a relevant message be sent to a user within a certain time after arrival at a given location.

Many factors affect users of built environments. These include the texture and physical characteristics of the environment and where information displays are situated. They are also affected by whether relevant information is received by them in a timely way. Consider the example of a coach bus station. In each space within the station there is a public display of messages about transit information to passengers that occupy the space at any one time. Each passenger carries a mobile phone and receives messages on this phone that are specifically relevant to their information and location. Action is implicit, in the sense that when a passenger walks into a space the location sensor notes the fact and as a result information may be distributed to the passenger’s mobile device and possibly to the public display if it is the first time someone for a designated coach bus arrives in the space. The design that is modelled adopts a simple set of techniques for deploying information users sufficient to provide a basis for the discussion. Many schemes are feasible for combining public displays with private information, see for example Gilroy et al. (2006). The scheme used here is illustrative of a range that could equally be addressed by the techniques described.

In this case the implicit action occurs when a user moves into a space. The effect of this movement is that a sensor in the space recognizes the presence of the user’s mobile device and triggers the system to provide information that is relevant.

3.1 User requirements for the system

The user concern is translated into the following properties of the model:

**P1** When the passenger moves into a location then the transit status information is presented to the passenger’s hand-held device within some maximum delay time.

**P2** Information on public display should reflect the current state of the system within a maximum delay time.
P3 When the passenger enters a new location, the sensor detects the passenger’s presence and the next message received concerns transit information and updates the passenger’s hand-held device with information relevant to the passenger’s position and stage in the embarkation process.

P4 When the passenger moves into a new location then if the passenger is the first from the coach bus to enter, public displays in the location are updated to include this transit information.

P5 When the last passenger on a particular transit in the location leaves it then the public display is updated to remove the transit information.

3.2 Characteristics of the coach bus station model
Since properties P1 and P2 are timing properties, a model that incorporates time is adopted for this system. The model is split into a number of processes describing:

Sensor: the activity within a room, including the mechanism for sensing the arrival and departure of passengers. This process updates the room-based display to show transit information for those passengers that are in the room.

Passenger: the passenger that receives specific messages relating to transit and location in the coach station. The passenger moves from room to room.

Dispatcher: the centralized dispatcher that sends messages regularly.

The dispatcher is critical to the timing characteristics of the design and is the basis for exploring the system to ensure that timing guarantees are satisfied. It is assumed that the dispatcher is a human machine system. The dispatching of messages involves human intervention at some level and this is the primary source of delay. This exploration is carried out by adjusting the rate and the order of distribution of messages, as well as passenger arrival volumes. The illustration in Figure 2 distributes messages in strict order with a delay between broadcasts specified by the constant workload. The model makes a simple set of assumptions about messages and the way they are tagged for recognition. Tags are associated with coach bus number and location. In practice, of course, the station system would be more elaborate than this. Many of the potential complexities of the model are removed by focusing on the specific requirements with the proviso that checking the properties does not bias the analysis exclusively to a certain class of properties, thereby providing a model that can be used as a structure for implementation.

Two types of process receive information from the dispatcher. The sensor process (Figure 3) combines the behaviour of the public display with the room sensor. Each room or space in the station contains a sensor (how the sensor is implemented is not the concern of this level of modelling). The generic sensor is instantiated for each space: entry hall, stand 1, stand 2, main hall, gate and so on. The other type of process is the passenger process (Figure 4) that describes the passenger and the relevant behaviour of the passenger’s mobile device.
The sensor process (Figure 3) describes the key interaction features of:

- The public display located in the room
- The sensor updating its knowledge of whether passengers are present in the space—this assumes an interaction between the sensor and the passenger device. The sensor communicates by means of three channels.
  - It receives messages that have been distributed to it from the dispatcher by means of the channel mchan.
  - It receives requests from the passenger’s hand-held devices (via arrive) where they arrive in the room that relates to the sensor.
  - It receives requests from the passenger’s hand-held devices (via depart) when they leave the sensor’s room.

When the sensor receives a message from the dispatcher, the function read ( ) checks the tags on the message and if the location tag coincides with the location of the sensor then the display is updated. Of course, a realistic implementation of this system would update a transit information array for display each time a relevant message is received. The array updating mechanism is not of interest to interaction analysis.

When the sensor receives a message from the arrive channel this signals the entry of a passenger. The array present [ ] keeps a count of the number of passengers present for a particular transit and is incremented with the arriving passenger’s coach bus number. When the sensor receives a message from the depart channel then the array is decremented using the departing passenger’s coach bus number. If the result of this is that there are no passengers for a particular coach bus left in the room then the transit information is removed from the display.

As a consequence, when the last passenger moves out of the space this information is cleared from the display. When the passenger is newly arrived in the space then the array present is incremented and so next time a message arrives about this coach bus the information will be displayed for the first time.

The passenger process ( Figure 4) describes the activity of the passenger and the key features of their mobile phone. This activity has a number of characteristics:

- The passenger is given a specific path to follow. This is defined in the array path, one would anticipate more than one passenger defined associated with different paths.
- The process notifies the room sensor that it has arrived. The passenger ticket is updated to
point to the current location.

- The passenger moves to a state where it receives messages from the dispatcher via mchan. If the received message is tagged with the passenger’s current location and the passenger’s coach bus number then the mobile phone display is updated.

4. Discussion

P1 requires that when the passenger moves into a new location in the coach station then move status information is presented to the passenger’s hand-held device within a maximum delay time (this is defined by the variable \textit{maxdelay}). It must be updated within a period of delay after a passenger enters the location (regardless of coach bus number) the relevant message will be received by the passenger within the time delay. Two transitions are of interest in the passenger process (Figure 4). The first occurs as the passenger moves into the new location and the second occurs when the passenger receives a message from the dispatcher that matches the coach bus number and location of the passenger. P1 is checked by introducing an observer process (Figure 5) and adding a communication \textit{(newroom)} in the passenger process (Figure 4) to signal arrival in the new location and similarly a communication \textit{(newmessage)} to signal receiving a relevant message. If the message does not arrive while the passenger is in a location (this time is determined in the model by the variable \textit{dwell}) then the observer will deadlock. The deadlock occurs because the observer is waiting to receive notification that a message has been received but instead receives a message from the passenger that it has entered another room. When appropriate diagnostics are switched on, deadlock generates a trace that can then be further analysed to work out why the system does not satisfy the properties.

![Figure 5. The observer process](image)

In practice the generalized deadlock property is very computer intensive and on a no-frills specification PC the Uppaal system ran out of memory after three hours execution. The specific properties relevant to P1 and P2 relate to checking that fresh messages arrive within the \textit{maxdelay} interval. In the case the properties was checked experimentally for different values of \textit{maxdelay}. When the property failed, and the passenger did not receive the message in time, the diagnostic trace was investigated to check which assumption caused this failure to occur thereby allowing the designer to explore alternative strategies for distributing messages so that passengers would receive timely updates.

5. Conclusions

The advantage of these simple models is that they can be explored using model checking. These approaches explore paths exhaustively to check whether the model satisfies the property. The model therefore is described in one of the languages above. The model checker converts it into an appropriate form for analysis. The analyst provides properties, often expressed in a temporal logic, to analyze the model. The model checker then attempts to find a counter-example to the property, or an instance where the property is true, and provides the analyst with a sequence of states that describes the counter-example or true instance. Model checking has the advantage that it is a decidable algorithmic
approach that is therefore relatively easy to use. The disadvantage of the technique is that there are a number of properties of interactive systems that cannot be adequately analyzed using these approaches, for example representation properties.

The problems with using formal techniques relate to how well the techniques scale, how easy it is to use the methods and whether the methods bias the analysis so that key issues may be ignored. It is not contended that this type of approach is the answer to all of the usability engineering problems. However, if usability engineering is to be a true engineering discipline, it does need rigour and formality. Formal methods of software engineering have a role to play during interactive system design and analysis. They enable a type of reasoning that is more thorough and repetitive, without precluding the use of other more traditional approaches to usability analysis. How to best integrate the two worlds is a challenging and interesting research area. No single current technique has the expressive power to capture, and enable the analysis of, all interesting facets of an interactive system.

Instead, different techniques should be applied as needed.

Acknowledgement

Financial support provided by the Humanities and Social Sciences Fund of the Ministry of Education, China (17JAZH100) & the Special Project on Humanities and art of Xi'an University of Technology (106-451017004).

References

[1] Clarke, E.M., Grumberg, O. and Peled, D.A. Model Checking [M]. Cambridge, MA: MIT Press, 1999
[2] Behrmann, G., David, A. and Larsen, K.G. ‘A tutorial on Uppaal’. Formal Methods for the Design of Real-time Systems[M]. Springer Lecture Notes in Computer Science, Berlin: Springer, 185: 00-36, 2004
[3] Gow, J., Thimbleby, H. and Cairns, P. Automatic critiques of interface modes [M]. Springer Lecture Notes in Computer Science, Berlin: Springer, 3941:201-12, 2006
[4] Holzmann, G.J. The SPIN Model Checker [M]. Primer and Reference Manual. Harlow, UK: Addison Wesley, 2003
[5] Jackson, D. Software Abstractions: Logic, Language and Analysis [M]. Cambridge, MA: MIT Press, 2006
[6] Campos, J.C. and Harrison, M.D. Model checking interactor specifications[M]. Automated Software Engineering, 8:275-310, 2001
[7] Campos, J.C. Automated Deduction and Usability Reasoning [D]. PhD thesis, Department of Computer Science, University of York, UK, 1999
[8] Gilroy, S.W., Olivier, P.L., Cao, H., Jackson, G. D., Kray, C. and Lin, D. Cross-board: Crossmodal access of dense public display[C]. International Workshop on Multimodal and Pervasive Service (MAPS06), Lyon, France, 2006
[9] NijunLi , Tongchi Zhou , Lin Zhou , Zhenyang Wu . Action recognition by Huffman coding and implicit action model [C]. 2015 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2015
[10] Y. P. Huang, G. Z. Wang, T. S. Chang, T. H. Chen. Three-dimensional virtual touch display system for multi-user applications[J]. Display Technol., vol. 9, no. 11, pp. 921-928, 2013
[11] Bonnechère, B.: Serious Games in Physical Rehabilitation: From Theory to Practice [M]. Springer, Brussels, 2017
[12] Garner, T.: Applications of virtual reality[M]. In: Echoes of Other Worlds: Sound in Virtual Reality, pp. 299–361, 2018