

A Visual Formalism for Graphical User Interfaces based on State Transition Diagrams

Carsten Maple, Tim French, Marc Conrad
Departement of Computing and Information Systems
University of Luton, U.K.

carsten.maple@luton.ac.uk, tim.french@luton.ac.uk, marc.conrad@luton.ac.uk

Abstract

In this paper we present a “lightweight” visual formalism that can be used to examine the state space complexity of an interface. The method can form a basis for designing, testing and documenting the ‘front-end’ component of a typical GUI forms based interface. The standard is intended to be free standing and seeks to address issues specific to forms based GUI interfaces. The method is based on a State Transition Diagram (STD) notation that is used to model the GUI component. STDs can be used at an early stage in the design of an interface to quantify and control surface level complexity. Further to allowing a visualisation of the complexity of an interface, techniques used in graph theory are applied to determine a novel measure of complexity that can then be used to better usability.

1. Introduction

It has become increasingly apparent that traditional ‘hard’ system approaches to HCI requirement elicitation, where often the focus is on the rigorous construction of a formal requirements catalogue or other formal models are often failing to address relevant issues. Innovative systems methodologies are required which though constructed from a rigorously applied intellectual paradigm, nevertheless are genuinely exploitable by teams of interdisciplinary stakeholders. E-commerce systems are but the latest example where all those involved are required to deliver credible front-end components that engender the trust and confidence of the users and stakeholders alike. Despite years of research in both academia and industry, existing approaches have

largely failed to deliver systems that are secure and optimally matched to their audience groups. Recent anecdotal public-sector examples of failure include: the failed launch of e-tax returns (Inland Revenue), the acknowledged failure of e-Government sites to meet EU accessibility guidelines, and most recent of all, the failure of the London ‘congestion charge’ web-site (<https://www.cclondon.com>) to cope with peak loading prior to launch. Indeed, few system development methods appear to be either genuinely end-user or web-centric or seek to genuinely address the central issue of competing web-based system ownership and diverse stakeholder group needs often inherent in large-scale B2C e-commerce sites. Typically, the development tools used and server and client side technologies employed drive such projects rather than ‘true’ end user needs, as intended.

Furthermore, the highly distributed and heterogeneous nature of the intended user population of B2C sites makes it virtually impossible to empirically validate the usability of the final site solely using focus groups prior to launch. Thus the HCI component often becomes critical to the success of the site (or otherwise). It would seem that that a potentially worldwide and thus intrinsically cross-cultural or multi-ethnic customer audience cannot be easily replicated with any confidence, (or at least without great expense) without firstly explicit identification of the shared meanings of the application itself amongst all the stakeholder groups from its inception.

Recent failures have been caused in the main by a fundamental lack of shared understanding amongst end-user and other stakeholder groups. Our vision is that a lightweight state modelling notation as derived and

extended from the Shared Meanings Design Framework (www.smdf.org), can do much to mitigate such problems by promoting client and analyst visualisation of the navigation (state-space behaviour as triggered by end users) of e-commerce systems.

The paper outlines some of the existing techniques for web visualisation and sitemap analysis in section 2 before presenting the proposed method in section 3. The paper concludes with a summary and recommendations for further work

2. Existing Techniques

While state-transition diagrams have existed in the realms of computer science for over sixty years, Harel [14] reinvented the state transition diagram in a form that is now most widely used. Harel discusses statecharts as a visual formalism for complex systems. They can be used at an early stage in the design of an interface to quantify and control surface level complexity, and to assess whether the interface embodies any potentially unstable or undefined states. We give a brief summary of some of the uses of STDs, as well as a background of some current methods for visualisation of the semantic web and methods for modelling the internet.

2.1 State Transition Diagrams

Over the last 15 years there has been a great deal of work undertaken that utilises the state-transition diagram concept. Perhaps the most notable use is their employment as part of UML. For details of UML the reader is referred to [10], [18] and the Object Management Group website [22]. UML state diagrams have been used for class testing, see [15] for example where coverage criteria based on control and data flow in UML state diagrams are proposed. Transforming UML state diagrams into extended finite state machines generates the test cases. In [12] France gives consideration to both formal and informal specification techniques. It is discussed that while graphical techniques have great flexibility for program specification, the lack of firm semantics limits their use for validation and verification purposes.

2.2 Visualising the semantic web

There has been a great deal of work in recent years into visualising the web. They have largely been based on semantic considerations and can prove invaluable for clustering and content-based information retrieval. A significant area of research is that of Pathfinder networks.

Pathfinder networks were originally developed to analyse and model patterns in proximity data, see [20]. Chen [7], [8], [9] and Fowler [11], among others, have since adapted these for use with the World Wide Web.

2.3 Modelling the WWW

The World Wide Web WWW can be represented by a (very large) graph where HTML pages are considered to be nodes (or vertices) and the hyperlinks between these pages are (directed) edges (or arcs). Much work has been undertaken in this area, see for example [3], [5], and [16]. This representation of the web has proved invaluable, especially in the area of data mining. Many papers consider the indegree and outdegree distribution of the nodes. The indegree of a vertex is defined to be equal to the number of directed edges with that vertex as the endpoint. The outdegree is similarly defined but considers those directed edges emanating from the vertex. It has been shown, see [2] or [16] for example, that the indegree and outdegree of nodes follow a power-law distribution.

The work described above attracts a great deal of funding and is an interesting area. However this work addresses the internet on a macroscopic scale and as yet does not consider more localised aspects that are of interest to website development and analysis.

3. The Proposed Method

It is essential that a web-site (being a public access system) contains a simple state-space model which supports user goals (such as on line shopping) but which does not overwhelm users. To control and model user navigation therefore is seen to be critical. We propose a solution that is:

- Lightweight and detached from any particular existing method (such as UML)
- Supports cognitive visualisation by client and analyst
- Defines user navigation objectively using a semi-formal notation (cf formal methods)

The benefits are seen as supporting “real” client-analyst dialogue and active client and stakeholder involvement, rather than the detached and disconnected manner in which many e-commerce systems typically evolve. In this paper we define a standard as a basis for designing, testing and documenting the ‘front-end’ component of a typical GUI forms based interface. The standard is intended to be free standing and seeks to address issues specific to forms based GUI interfaces.

The methods that are proposed are based upon both state-transition diagrams and elements of graph theory.

3.1 An STD-like representation of websites

We adapt the standard state-transition diagram method to suit the needs of the internet. We reiterate that it is not the aim to propose a “heavy” formalism such UML, but rather a lightweight representation. Forms behaviour is indicated by overlaying the STD with a dashed ellipse that will include as many underlying states as necessary. Each form has a label such as p-1, p-2, etc.

We propose that the arcs in the STD are not weighted in the visualisation - this can lead to a lack of clarity and can complicate the diagram unnecessarily. The visualisation is intended to be cognitive and relay information regarding the structure of the site or portion of site.

To aid understanding of the diagram the area of each ellipse representing a state is of the same area - π square unit (where a circle of radius 1 would satisfy the condition) say. The reason for these states to be of the same size is that all are considered equally important.

Each state lies within some form, frame or webpage. The area of an object O_k (such as a page or frame) that contains one or more smaller objects, s_1, s_2, \dots, s_n is given by the formula: $\text{Area}(O_k) = 5 \times \sum \text{area}(s_i) \times (1 + 2 \times (\text{indegree}(O_k) - 1) / (\text{max_indegree} - 1))$. Here $\text{indegree}(O_i)$ is a function equal to the number of edges ending at state O_k , and max_indegree is the maximum of the function indegree over all states. We notice that since the area of an ellipse of width $2a$ and height $2b$ is πab , if the states are of unit radius and thus area of π then calculating the dimensions of the ellipse is very easy.

Using such a formula ensures that objects with greater importance are given bigger representations in the diagram. The minimum area assigned to O_k is given as $5 \times \sum \text{area}(s_i)$ while the maximum is 3 times this value. The formula for the area is somewhat arbitrary. All that is intended is that the area allocated to an object is proportional to its significance. We define the significance as a measure of how many direct links there are to that object and how many embedded states or objects it has. There are other metrics for importance that may be equally valid.

3.2 The characteristic matrix

A characteristic matrix can be used to represent the STD. However, the standard characteristic matrix needs modification to be applied to websites and GUIs. Usually there are no temporal considerations in STDs and

characteristic matrices. However, this is not reflected in the real situation. Users may become disinterested in the site, the system may crash, or they may simply run out of time. All these possibilities can result in any state becoming a terminal state. To model this in the characteristic matrix, we place values in the diagonal of the matrix. We use self-loops that reflect the probability that the current state is a terminal state.

The initialisation of the characteristic matrix is such that if the outdegree of S_i is k then there are $k + 1$ non-zero values in row i . Each non-zero value is equal to $1 / k + 1$, reflecting the initial assumption that any subsequent state (including the possibility that the current state is the terminal state) has an equal probability of being the next state. This initial value can then be changed once a suitable history of usage has been obtained.

		Successor state				
		S_1	S_2	S_3	S_4	S_5
Predecessor state	S_1	1/6	1/6	0	0	0
	S_2	0	1/3	1/3	0	0
	S_3	0	0	1/2	1/2	0
	S_4	0	0	0	1/2	1/2
	S_5	0	0	0	0	1
	S_6	0	0	0	0	1/2

Table 1. Excerpt of the characteristic matrix

Note that in the characteristic matrix, we have $\sum p = 1$ in each row for standard STDs.

3.3 A metric for GUI / website complexity

There exist a number of possible metrics to use to determine the complexity of a GUI or website. The number of pages and links is one of the most obvious measures. The larger a site the more complex it can appear and structuring is needed to prevent disorientation [4]. The complexity of a site can be related to the amount of navigational freedom the user is given. A simple metric that can be applied is the ratio between the number of links and the number of pages. Distance, the shortest path between pages, can be viewed as an indicator of navigation effort. It has been shown that user satisfaction is inversely proportional to the distance see [21]. Some researchers have proposed studying the average length of all shortest paths instead, the average connected distance [5]. Others have suggested using the diameter [1]. Our algorithm considers the shortest paths from state S_1 to S_i , a_i say, the outdegree of the state S_1 and the outdegree of the states S_i .

$$\text{Complexity} = \text{outdegree}(S_1) + \sum a_i \times \text{outdegree}(S_i) / n$$

We consider that great complexity arises by having a large number of possible states reachable immediately from state S_1 . The complexity metric also considers the number of intermediate states required to reach any state S_i . Finally the outdegree of each state as a proportion of the number of states is a further factor of the metric.

We localise the domain of interest as a set of “goal” driven paths from an initial state S_1 (usually the homepage of the site), triggered via user intervention (actions) which terminates in one or more goal states. The proposed visualisation does not attempt to consider all possible states and paths within the Universe of discourse defined within the entire Finite State Model (as others have attempted to do). Neither are we attempting to model (as in UML) all object states and actions of our system. Only “just enough” system actions are notated so as to facilitate user/designer visualisation of user “goals”. In this way we choose not to regard the system as “goal-less” or as capable of being initiated and visualised starting from any given state within the Universe of discourse of the Finite State Model.

Example 1: The Nationwide Building Society

Consider the website of the Nationwide Building Society (www.nationwide.co.uk) where the user goal is to obtain a “quick mortgage quote”. The state-transition diagram is shown in Figure 1 and the dialogue shown in Table 2. Since a taxonomy of user and system actions adds to the understanding of the semantic of user actions each transition has a label of the form user-action/system-action. Similar to UML statechart diagrams we can have transitions where only a user action or only a system action is specified. System actions are shown in *italics*. In the diagram we use the following acronyms for system actions:

- *PCO*: The system closes the current page and opens a new page in the same window.
- *BO*: The system opens a dialogue box.
- *BC*: The system closes a dialogue box.

Regarding the user actions we distinguish between actions triggered by key strokes denoted by square brackets and actions triggered by mouse events prefixed by M. In particular we have:

- [...]: A series of characters has been entered into a form in the web page.
- [RET]: The return key has been hit.
- MH: The left mouse button had been clicked on hypertext.

- MB: The left mouse button had been clicked on a button.
- MT: The left mouse button had been clicked on a “tab”.
- MD: A choice has been selected from a drop down menu.

Transitions are shown as dotted lines if they do not lead towards the goal but are quite likely to be followed because of the overall design of the page. Goal states are shown with a dark grey background. States which might be considered as subgoals are shown in light grey. In the diagram we can identify three main “routes” for the user to follow to achieve the “goal” of obtaining a quick mortgage quotation.

Route-1: S_1 to goal state S_5 assuming valid entries are made for house price, loan, and term. (There are a large number of error conditions omitted here for clarity);

Route-2: S_1 to S_{12} initiated via two different user action alternatives to goal state S_{13} . Subgoal state S_{12} has the same calculation functionality as state S_4 differing only slightly in presentation. State S_{13} contains the result;

Route-3: S_1 to S_{12} via an intervening page containing states S_{11} and S_{12} .

Example 2: Using the complexity metric for GUI design

If we consider the two possible configurations of Figure 2, we can assign a complexity of 5.5 to the configuration on the left and 4.5 to the configuration on the right. This certainly seems reasonable since the extra two links or arcs on the left actually do very little in terms of aiding navigation. The longest path in the right-hand configuration is only 4 which is not an unreasonable length. We can define the complexity for any layout such as the one above, in which there are $n + 1$ states, with every i^{th} state directly accessible from S_1 , and with each of the states linked as in the example. In such cases the complexity is given by $\text{Complexity} = n / i + (i + 1)/2$. We see that obviously it is never optimal to have all states (the case where $i = 1$) directly accessible from S_1 as the complexity of the interface is high. The metric can be used to identify if adding in new links increases or decreases the complexity of the interface.

4. Summary and Further Work

We have presented a novel lightweight method for the visual representation of GUIs, aimed mainly at web applications. The visualisation considers the relative

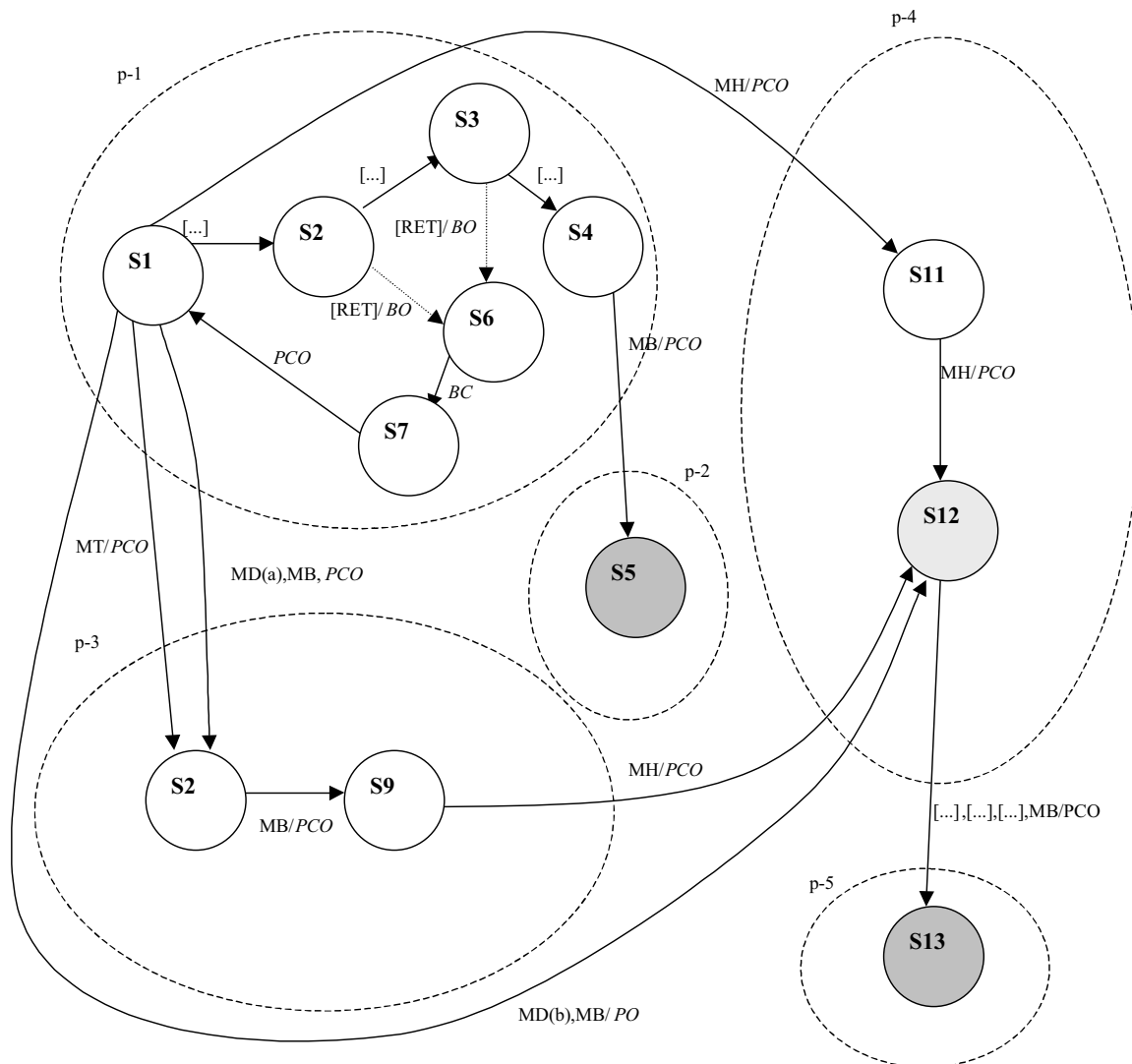


Figure 1: Visualisation of Nationwide Bank Website

"importance" of a form, frame or page and this is represented in a simple cognitive manner.

Whereas standard STDs are non-temporal, we have suggested that using the diagonal of the characteristic matrix can be used for temporal considerations. This allows each state to have outcomes with a combined probability of 1. If we consider the case of many webpages, wherein an alternative state is reached after a certain time has elapsed (such as a redirection) it is difficult not to consider time. It is possible that the user closes the browser / interface before the direction. If this can be the case, then a probability needs assigning to accurately model the state. Our method allows such states.

As others have done [6], the characteristic matrix can be used as a starting point for analysis with Markov models.

We have also discussed a new complexity metric. The metric is based upon attempting to minimise to average path length but without compromising a limit on

the number of links. The metric uses the assumption that the user will have a goal of any of the other states *with equal probability*. The benefit of the metric is its simplicity.

The methods appear to have potential and we hope to apply them to both established platforms (PC, WAP) as well as to newer forms of e-mediation (so called 2.5G, 3G phones) and possibly iTV (Interactive TV) in due course so as to ground the technique firmly in practice.

References

- [1] Albert, R., Jeong, H., et al, Diameter of the World Wide Web, *Nature*, 401, pp. 130-131 (1999).
- [2] Barabási, A.-L., Albert R., Emergence of scaling in random networks. *Science*, pp. 509-512 (1999).
- [3] B. Bollobas, *Random Graphs*, Acad. Press, London (1985).
- [4] Botafogo, R.A., et al., Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics, *ACM Transactions on Information Systems*, Vol. 10, No. 2. pp. 142-180 (1992).

[5] Broder, A., Kumar, R., et al., Graph structure in the web, *Proc. Of the 9th Intl. WWW Conference*, pp. 309-320 (2000).

[6] Chairns, P, Jones, M, Thimberly, H.W, Usability Analysis with Markov Models, *ACM Transactions on Computer Human Interaction*, Vol. 8, No. 2, pp. 99-132 (2001)

[7] Chen, C., Bridging the gap: The use of Pathfinder networks in visual navigation, *Journal of Visual Languages and Computing*, 9(3), pp. 267-286 (1998).

[8] Chen, C., Generalised Similarity Analysis and Pathfinder Network Scaling. *Interacting with Computers*, 10(2), pp. 107-128 (1998).

[9] Chen, C., Information Visualisation and Virtual Environments. Springer-Verlag (1999).

[10] Fowler, F., Scott, K., UML Distilled: a brief guide to the standard object modeling language, AddisonWesley (2000).

[11] Fowler, R.H., Fowler, W.A.L, et al, Document Explorer Visualisations of Www Document and Term Spaces, University of Texas - Pan American, CS Dept No NAG9-551 96-6 (1996).

[12] France, R. B., Wu, J., et al., A Tale of Two Case Studies: Using Integrated Methods to Support Rigorous Requirements Specification, *Proceedings of the BCS FACS Methods Integration Workshop* (1996).

[13] Gorshkova, E., Novikov, B., Exploiting UML Extensibility in the Design of Web Applications (2002), www2002.org/CDROM/poster/69.pdf

[14] Harel, D., Statecharts: a visual formalism for complex systems, *Science of Computer Programming*, 8, pp. 231-274, (1987).

[15] Kim, Y. G., et al, Test cases generation from UML state diagrams. *IEEE Software*, 146(4), 187-192 (1999).

[16] Kumar, S. R., Raghavan, P., et al, Trawling the Web for Emerging Cyber Communities, *Proc. of the 8th WWW Conference*, pp. 403-416 (1999).

[17] Mitchell, T., Freitag, D., et al, Webwatcher: A tour guide for the world wide web, *Proc. Int. Joint Conf. AIJCA197*, pp770-777 (1997).

[18] OMG: UML Notation Guide, v1.3. Object Management Group, Inc. (2000).

[19] Rauterberg, M.: A Method of Quantitative Measurement of Cognitive Complexity. In: Van der Veer, G., Tauber, M, et al., (eds), *Human-Computer Interaction: Tasks and Organisation*. CUD-Publ., Roma, pp. 295-307 (1992).

[20] Schvaneveldt, R., Durso, F., Dearholt, D., Network structures in proximity data, *The Psychology of Learning and Motivation*, 24, 249-284 (1989).

[21] Smyth, B., Cotter, P., The Plight of the Navigator: Solving the Navigation Problem for Wireless Portals. In: De Bra, P., et al., (eds), *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer Verlag, 2002. pp. 328-337

[22] UML Ressource Page. Available online at <http://www.omg.org/uml>

S4~S5: MB/PCO	UA/SA: {user clicks "OK" icon / system opens a new page and displays the mortgage quotation = goal state!}
S2~S6: [RET]/BO	UA/SA: {user hits return button / system opens dialogue "error" box}
S3~S6: [RET]/BO	UA/SA: {user hits return button / system opens dialogue "error" box}
S6~S7: MB	UA: {user clicks on "ok" button }
S7~S1: BC	SA: system closes box and returns user to default starting state}
S1~S11: MH/PCO	UA/SA: {user clicks on hypertext link / system opens new page [site map]}
S11~S12: MH/PCO	UA/SA: {user clicks on hypertext / system activates link [form]}
S12~S13: [...],[...],[...], MB/PCO	UA/SA: {user inputs data as in S1, hits "Calculate" button / system closes p-6 and opens a new page and displays the mortgage quotation = goal state!}
S1~S8: MD(a), MB/PCO	UA/SA: {user selects an item from dropdown list (a) and clicks on OK icon / system closes p-1 and opens p-3}
S1~S8: MT/PCO	UA/SA: {user selects an item by clicking on a "tab" / system closes p-1 and opens p-3}
S8~S9: MB/PCO	UA/SA: {user clicks on "Tools & Calculators" button / system closes page and opens p-3}
S9~S12: MH/PCO	UA/SA: {user clicks on hypertext text link /system closes page-3 and proceeds to s12 = sub-goal state}
S1~S12: MB/PO	UA/SA: {user selects an item from dropdown list (b) and clicks on OK icon/system closes p-1 and goes to S12}

Table 2: Actions in the Nationwide diagram

ACTIONS	DESCRIPTION
S1~S2: [...]	UA: {user inputs a number}
S2~S3: [...]	UA: {user inputs a number}
S3~S4: [...]	UA: {user inputs a number}