Liet. mat. rink. LMD darbai, **50**, 2009, 380–385

Problems of portal users behaviour specification*

Audronė LUPEIKIENĖ, Albertas ČAPLINSKAS, Gintautas DZEMYDA
Institute of Mathematics and Informatics
Akademijos 4, LT-08663 Vilnius, Lithuania
e-mail: audronel@ktl.mii.lt; alcapl@ktl.mii.lt; dzemyda@ktl.mii.lt

Abstract. This paper is a position paper, so it presents not well-established scientific truths but remarks and notices on the phenomena and their generalisations. The paper asserts that current portal design practices are not sound, advocates for an explicit specification of the assumptions about the predicted users' behaviour, and requires that a specification language to formulate such the assumptions should be based on the formalism of temporal logic of actions.

Keywords: portal, user behaviour specification, temporal logic of actions.

Introduction

The design of software systems including Internet portals is always based on a number of assumptions about the future behaviour of users. However these assumptions usually are not formulated in an explicit way. Up to date even a specification language to state the preliminary assumptions in a precise form has not been developed. The development of such a language purported to specify the predicted users behaviour for to-be Internet portal is one of the tasks provided by the research project in the frame of the Program of Lithuanian–French bilateral cooperation in scientific research and experimental activity "Gilibert". This task is far not simple. First of all, users' range is not limited. They have different preferences and personal features including background knowledge and reasoning, belief systems, cognitive styles, and experience. Second, their needs and goals are different. The portals visitors need to accomplish tasks (doers), seek to get the information (viewers) or relaxation (readers) [10]. Third, the type and amount of information the users are interested in differs. Forth, their technical and technological backgrounds are different. Therefore, only by knowing *whom*, *why*, *what*, and *how* a portal design can be modelled accordingly.

The presented approach uses a kind of temporal logic as a basic formalism. In general, a portal is a powerful Web site that gives users a single point of access to application and information in a unified interface [1]. The paper discusses only problems, which are related to the specification of corporate portals presenting information about a company, its activities, people, and etc. Such a portal is a set of client-oriented Web

*The research is funded by Program of Lithuanian–French bilateral cooperation in scientific research and experimental activity "Gilibert" and by the Lithuanian State Science and Studies Foundation under the contract No. V-33/2009.

sites that personalise the portal's tools and information to the specific needs and characteristics of the person who is visiting this web site, searching information that is presented directly in the Web sites or in the associated databases. As a rule, it provides a standard set of tools. Corporate portals promote the gathering, sharing, and dissemination of information throughout the intranet, extranet and Internet. However, the traditional approaches to portal design often ignore the information needs, searching practices and personalities of users. As a result, the organisation portals suffer from quality problems that prevent or inhibit their use (for example, impossible to find information, poor navigation or inappropriate display of information).

The reminder of the paper is organised as follows. Section 2 provides some background information about user behaviour model and temporal logic of actions. Section 3 builds on the preceding section by showing how to use temporal logic of actions to specify the preliminary assumptions about the portal users and their behaviour. Finally, conclusions and some future research directions are presented.

1. Temporal logic of actions for behaviour specification

A precise model of user behaviour becomes the first problem of the portal design or redesign if the expected user behaviour mismatches its observed behaviour. A portal user behaviour model can be defined as an infinite sequence of states where a state is specified by a set of variables (features). An action is fundamental unit of user behaviour model and its execution represents some transformation or processing. Actions relate the next state to the current one – they describe the state transition relations. Finding a finite sequence of actions that will transform an initial state to a state that satisfies a given goal is the typical problem in specifying user behaviour. Hence, an action should be explicitly perceived and defined.

The values of state variables are changing in time, so user behaviour represents the temporal relationships between states. This implies the necessity of temporal logic to encode temporal knowledge and to model time-dependent nature of user behaviour.

Temporal logic in its basic form is a particular type of modal tense logic. It provides two basic temporal operators $\Box p$ (*always* p) and $\Diamond p$ (*sometime* p, also read as *eventually* p). Developing of this original logic resulted in logics of increasingly greater expressive power. Systems of temporal logic are classified along a number of axes: propositional versus first order, branching versus linear, points versus intervals, global versus compositional, and past versus future tense [4]. Temporal logic has been shown to be a powerful formalism for expressing dynamic properties of a system [8,3,11,9,5]. To define and verify the arbitrary temporal formulas about states is a standard problem addressed for example by [3,11,9]. However the critical problem of using a temporal logic for behaviour modelling is the description of events and/or actions. Additionally, different temporal logics are based on different conceptualisations. Therefore one logic can be ideal for describing some systems or some properties of systems, but awkward for others. Taking into account all the mentioned above, the temporal logic of actions [6], logic for specification and reasoning about action and change, is to be employed.

Temporal logic of actions (TLA) combines standard temporal logic with a logic of actions: action formulas describe state transition and temporal formulas describe state sequences. TLA syntax assumes given a first order signature – function and predicate symbols, and two disjoint sets of rigid and flexible variables. Rigid variables denote values as in first order logic; flexible variables represent state elements (variables). The semantic of TLA is based on the behaviour, which is an infinite sequence of states $\{s_0, s_1, s_2 \ldots\}$. A state s assigns a value $s(x)$ to a variable x (it will be used $s[[x]]$ to denote $s(x)$). A state function is a non-boolean expression built from variables and constant symbols. The meaning $[[f]]$ of a state function f is a mapping from the collection of states to the collection of values. A state predicate (or predicate) is a boolean expression built from variables and constant symbols. The meaning $[[P]]$ of a predicate P is a mapping from states to booleans.

An action represents a relation between the old and new state, where the unprimed variables (e.g., x, y) refer to the old state and the primed variables (e.g., x', y') refer to the new state. The meaning of the non-primed variables is the variable's value in the current state. The meaning of primed variables is the variable's value in the next state. The meaning $[[A]]$ of an action A is a relation between states – a function that assigns a boolean $s[[A]]t$ to a pair of states s (old) and t (new):

$$s[[A]]t \stackrel{def}{=} A(\forall\text{``}v' \colon s[[v]]/v, t[[v]]/v').$$

The meaning of $[[A]]t$ is that either A is valid now, or the variables appearing in t do not change. This allows for stuttering steps, in which none of the state variables change their values.

The action operators of TLA are as follows:

$[A]_e$ – $A \vee (e' = e)$, where e stands for elements of a set $\{e_1, \ldots, e_n\}$;

$< A >_e$ – $A \wedge (e' \neq e)$;

Enabled A – an A step is possible, i.e., an action A is enabled in a state s iff there exists some state t such that $\{s, t\}$ satisfies A;

Unchanged e – $e' = e$;

$A * B$ – composition of actions.

$s[[P]]$ is a boolean for any state s. Predicate P can be viewed as an action that contains no primed variables. Thus, $s[[P]]t$ is a boolean, which equals $s[[P]]$, for any states s and t.

To summarise, TLA is the linear-time temporal logic whose atomic formulas are predicates and formulas of the form $\Box[[A]]_f$, where A is an action and f is a state function. The temporal aspects are characterised by the following temporal operators:

$\Box F$ – formula F is always true;

$\Diamond F$ – formula F is eventually true;

$WF_e(A)$ – weak fairness[1] for action A, i.e., either an operation must be executed (it means taking an $< A >_f$ for some action A and state function f) or it becomes impossible to execute it;

$SF_e(A)$ – strong fairness for action A, i.e., either an operation is eventually executed or it is not infinitely often possible to execute it;

$F \rightarrow G$ – formula F leads to formula G, i.e., whenever F is true, G will eventually become true;

[1] A fair transition is one that must eventually be taken (assuming that it is possible).

$F \overset{+}{\rightarrow} G$ – formula F guarantees formula G, i.e., is true for a behaviour iff G is true for at least as long as F is;

$\exists x : F$ – this formula is satisfied by a behaviour iff there is some (infinite) sequence of values that can be assigned to x, which would produce a behaviour satisfying F.

Therefore, the users and portals are real-world objects but their behaviours are described by sequences of states, that is, are mathematical objects [6]. It means that to specify the assumptions about the users behaviours we must describe the behaviour of each user class in a precise mathematical way and verify the formal properties of specified behaviours.

2. Users behaviour modelling

Temporal logic of actions is being defined as logic for specifying and reasoning about the concurrent systems [7]. In other words, it was primarily developed for the specifying the behaviour of concurrent and/or active systems, which are characterised by ongoing, typically non-terminating and highly nondeterministic behaviour. To complete the behaviour description this logic foresees the specification and verification of safety, liveness and fairness properties of discrete systems.

We use TLA for other purposes – to specify the behaviours of portal users. It should be noted that users' behaviours can be non-terminating but typically terminate at the goal state, when the required information needs are satisfied.

As a user browses an organisation portal, he visits pages and takes actions on these pages, such as following a link, backing up to a previous page, or bookmarking a page. Therefore, the portal developers should have answers to "what are the user categories?", "what are the goals of any user category?", "what are the information seeking processes for the different categories of users?", "what information should be provided on the portal?", "what is the traffic of portal's users?", etc. questions.

Here are the examples of statements (answers) that can be expressed in TLA:

a user is of the category X;

in the initial state any user of the category X has some information, but the user's goal (i.e., what information he needs) is unknown;

each user of the category X has a goal to find the definitions of key terms;

any reachable (from initial) state in which p is *TRUE* has a path from it on which r is eventually *TRUE*, and until then q is *TRUE*;

any portal's state should always be characterised as follows: no links to already deleted documents.

Because of the space limitations only the last two examples are presented in TLA-based specification language TLA+:

$Choose\ (u, link) \overset{def}{=} link \in Data \wedge link \in available[u]$

$\wedge\ Addr\ [N] = \bot$

$\wedge\ Addr'\ [N] = link$

$\wedge \forall i \in 1..(N-1)$ UNCHANGED $Addr\ [i]$ – to follow the link= $link$ is an action user u chooses in the state;

$\forall u \in Users,\ D \in$ SUBSET $Documents,\ A \in$ SUBSET $Documents$:

$\Diamond(obtain\ (u, D) \Rightarrow \Box\ look\text{-}for\ (u, A))$ – whenever user obtains text-based resources, he eventually looks for audio resources and it holds in the current state (web page) and all the following states.

The users of a corporate portal typically seek for information by navigating from state to state. They want to achieve a certain goal state or maybe several goal states. The user's actions are guided, for example, by so called "information scent" [2] or, in other words, user's behaviour is guided by some patterns. Although these patterns are individual ones and built on user experiences, mentality and intuition, it is possible to speak about some typical patterns. In any case, the developers should assume and specify some information search patterns for each user class. Patterns should be expressed in terms of the state sequences and implemented as navigational paths. Later they can be validated using the log data and data mining techniques. Of course, the validation is possible only in case if the assumptions about patterns are formulated in a formal way. Thus the TLA formalism should be used to specify also the user behaviours patterns. For this aim, TLA should be augmented with a goal modality. So, a proper expressiveness of the formalism is one more problem to be dealt with before the users' behaviour modelling.

3. Conclusions and future research

As the corporate portals have grown into a channel of communication and a vehicle for information dissemination and retrieval, the problem of understanding and modelling users' behaviour becomes more and more important. The proper formalism is necessary for this aim. It seems that TLA is one of most promising candidates for this role – it enables the specification and reasoning about action and change. However, to apply this formalism in the portal engineering practice, several serious theoretical problems must be solved. Firstly, the TLA formalism should be augmented with the goal modality. Next, the developer-oriented specification language should be developed on the basis of this formalism. Finally, the validation techniques, which allow matching the data mining results and specifications of user behaviour patterns, should be developed.

References

1. *BEAWebLogic Portal*™, Version 8.1. Portal Overview, BEA Systems, Inc., 2003. Available: `http://edocs.bea.com/wlp/docs81/pdf/overview.pdf`.
2. E.H. Chi, P. Pirolli, K. Chen, J. Pitkow. Using information scent to model user information needs and actions on the web. In: *Proceedings of the SIG-CHI on Human Factors in Computing Systems*, Seattle, WA, USA, 2001. ACM, 490–497.
3. C. Dixon, M. Fisher, B. Konev, A. Lisitsa. Efficient first-order temporal logic for infinite-state systems. Computing Research Repository, article 0702036, Cornell University, 2007.
4. E.A. Emerson. Temporal and modal logic. In: J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science*, *Vol. B*: *Formal Models and Semantics*. North-Holland Pub. Co./MIT Press, 995–1072, 1990.
5. E.A. Emerson. Automated temporal reasoning about reactive systems. In: F. Moller, G. Birtwistle (Eds.), *Logics for Concurrency*: *Structure Versus Automata*, *Lecture Notes on Computer Science*, Vol. 1043. Springer, 111–120, 1996.
6. L. Lamport. Temporal logic of actions. *Technical Report*. Digital Equipment Corporation, 1991.
7. L. Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, 1994.

8. A. Pnueli. System specification and refinement in temporal logic. In: R. Shyamasundar (Ed.), *Proceedings of Foundations of Software Technology and Theoretical Computer Science*, *Lecture Notes in Computer Science*, Vol. 652. Springer, 1–38, 1992.
9. C. Rossi, M. Enciso, A. Mora. A first order temporal logic for behavior representation. In: Ch. Lemaitre, C.A. Reyes, J.A. Gonzalez (Eds.), *Advances in Artificial Intelligence*, *Lecture Notes in Computer Science*, Vol. 3315. Springer, 408–418, 2004.
10. J. Zeldman. Designing your audience. *A List Apart*: *For People Who Make Websites*, 1999–2001. Available: `http://www.alistapart.com/stories/ who/index.html`.
11. Ch. Zhou1. Specifying communicating systems with temporal logic. In: B. Banieqbal, H. Barringer, A. Pnueli (Eds.), *Temporal Logic in Specification*, *Lecture Notes in Computer Science*, Vol. 398. Springer, 304–323, 1989.

REZIUMĖ

A. Lupeikienė, A. Čaplinskas, G. Dzemyda. Portalo naudotojų elgsenos specifikavimo problemos

Straipsnyje yra išsakoma autorių pozicija korporacijų portalų kūrimo klausimais. Jame teigiama, kad kuriant portalus šiuo metu dažniausiai naudojami nepakankamai formalūs metodai. Autoriai gina teiginį, kad prielaidos apie prognozuojamą naudotojų elgseną turėtų būti formuluojamos išreikštiniu būdu ir siūlo tokias prielaidas užrašyti specifikavimo kalba, kuri būtų grindžiama atitinkamai praplėstu laiko ir veiksmų logikos formalizmu.

Raktiniai žodžiai: portalas, naudotojo elgsenos specifikavimas, laiko ir veiksmų logika.