



550, rue Sherbrooke Ouest, bureau 100
Montréal (Québec) H3A 1B9
Tél. : 514 840-1234; Téléc. : 514 840-1244
Place de la Cité – Tour de la Cité
2600, boul. Laurier, bureau 625
Québec (Québec) G1V 4W1
Tél. : 418 648-8080; téléc. : 418 648-8141
<http://www.crim.ca>

CRIM - Documentation/Communications

Rapport technique
WeSPaS: A Specification Pattern System
for Web Verification

Première version

CRIM-07/10-17

May Haydar
Boursière du CRIM
Doctorante, DIRO, Université de Montréal

Houari Sahraoui
Professeur agrégé
DIRO, Université de Montréal

2007-11-23

Collection scientifique et technique

ISBN-13 : 978-2-89522-101-2

ISBN-10 : 2-89522-061-1

(Note facultative sur le verso de la page titre)

Pour tout renseignement, communiquer avec:

CRIM Centre de documentation

CRIM

550, rue Sherbrooke Ouest, bureau 100

Montréal (Québec) H3A 1B9

Téléphone : (514) 840-1234

Télécopieur : (514) 840-1244

Tous droits réservés © 2007 CRIM

ISBN-13 : 978-2-89522-101-2

ISBN-10 : 2-89522-061-1

Dépôt légal - Bibliothèque et Archives nationales du Québec, 2007

Dépôt légal - Bibliothèque et Archives Canada, 2007

TABLE OF CONTENTS

TABLE OF CONTENTS	3
LIST OF TABLES	5
ABSTRACT	6
1. INTRODUCTION	7
2. QUALITY ASSURANCE OF WEB APPLICATIONS	8
2.1 Reliability and Functionality	8
2.2 Usability	9
2.3 Security and Privacy	9
2.4 Custom Attribute	9
3. LINEAR TEMPORAL LOGIC	10
4. PATTERN BASED APPROACH	12
4.1 Categorization	14
4.2 Template	15
5. LIBRARY OF WEB PROPERTY PATTERNS	16
5.1 Non-Functional Patterns	16
5.1.1 Links	16
5.1.2 Contents	17
5.1.3 Navigation	18
5.2 Functional Patterns	19
5.2.1 General WAs	20
5.2.2 E-Commerce WAs	22
6. DISCUSSION	26
7. RELATED WORK	27

8. CONCLUSION AND FUTURE WORK..... 28

REFERENCES 29

APPENDIX 32

LIST OF TABLES

<i>Table 1. Link Pattern number 5.....</i>	<i>17</i>
<i>Table 2. Content Pattern number 22.</i>	<i>18</i>
<i>Table 3. Navigation Pattern for number 7.....</i>	<i>19</i>
<i>Table 4. Reachability Pattern number 1.....</i>	<i>20</i>
<i>Table 5. Security Pattern number 6.....</i>	<i>21</i>
<i>Table 6. Others Pattern number 13.</i>	<i>22</i>
<i>Table 7. Customer Support Pattern number 4.....</i>	<i>23</i>
<i>Table 8. Trust Pattern number 2.....</i>	<i>24</i>
<i>Table 9. Product Info and Navigation Pattern number 4.....</i>	<i>25</i>

ABSTRACT

Quality assurance of Web applications is usually an informal process. Meanwhile, formal methods have been proven to be reliable means for the specification, verification, and testing of systems. However, those methods require learning their mathematical foundations, specifically those behind temporal logics. Specifying properties using temporal logic is often complex even to experts, while it is a daunting task and error prone for non-expert users. To assist web developer and testers in formally specifying web related properties, we present a library of about 120 web specification patterns. This library is a result of a survey of various resources in the field of quality assurance of Web Applications, which characterize successful web application using a set of standardized attributes. The patterns are categorized into two main classes: functional and non-functional.

This work has been completed during the internship of the first author at CRIM, within the Analysis and Distributed Systems Group under the supervision of Dr. Alexandre Petrenko.

Acknowledgements. We would like to thank Dr. Alexandre Petrenko and Dr. Sergiy Boroday for their valuable involvement in this work. Their comments, suggestions, and discussions have greatly influenced the obtained results. We also acknowledge the fruitful discussions with Ghazwa Malak on the subject of web quality measurement that affected this work. Also, the first author acknowledges the financial support offered by CRIM.

1. INTRODUCTION

At the heart of the research efforts dedicated to the web development and analysis, lie implicit postulations about “quality” and “usefulness” of web resources and services. Web applications (WAs) are drastically evolving and becoming more complex. WAs run large-scale software applications for numerous areas of usage including e-commerce, distributed information, collaborative work, distributed services, etc. WAs development has become one of the largest and most important parts of the software industry, and yet little is known about how to fully ensure their quality attributes [21]. Currently quality assurance of WAs is conducted informally. Meanwhile, during the last few years, formal approaches to verification and testing of software applications have been proven to be reliable means to insure quality of software. Likewise, several attempts have been proposed for formal verification, and testing of WAs [2,5,6,11,22,24]. However, these approaches are often inapplicable by the web community for several reasons. First, researchers addressing the problem of quality assurance of WAs by means of formal techniques do not exhaustively address the attributes and norms set by quality assurance specialists, which makes the contributions of those researchers relevant mostly to the formal methods research community. Second, it has been long acknowledged that formal methods are not fully applied in real software development let alone web development. The reason is the difficulty for software developers, testers, and users to understand the mathematical foundation underlying the formal methods.

In this report, we propose a practical solution to the above stated problem which facilitates the acceptance of formal verification technology by the web community. We study the attributes that designate the good quality of such applications and propose a library of temporal properties of WAs. These specifications relate mainly to the behavioral aspects of WAs based on user interactions and the way web resources are rendered. Web quality assurance specialists have long studied the requirement, design, and implementation of various types of WAs and were able to set norms which could serve as vital guidelines to the quality assurance of WAs. Needless to say that those norms are described in English prose and cannot be used in a formal context of quality assurance. In this paper, we discuss the main attributes that determine the quality of WAs. We also present a non-exhaustive inventory of web quality requirements that are indispensable to meet those attributes and could designate formal properties of WAs necessary for quality assurance of WAs. After surveying several resources related to web quality assurance and usability, we were able to collect about 120 web requirements. In the context of our formal framework for run-time verification of WAs [16,17], these formal properties can be verified against WA models. However, to reduce the hurdle of learning the mathematical foundations underlying the temporal logic and model checking theory for web developers and testers, we map those quality requirements into LTL. The formalized properties could serve as library of web specification patterns, called WeSPaS, which can be verified in a given WA. Finally, we

discern existing quality requirements that could not be formulated in LTL and thus cannot be verified using formal verification techniques.

The rest of the report is organized as follows. Section 2 includes a discussion on the major web quality attributes that need to be satisfied to ensure a high quality of WAs. In Section 3, we discuss the pattern based approach to specify web properties, the classification we have used to categorize the web patterns, and the template used for those patterns. In Section 4 we present the properties listed according to their categories (functional and non-functional) and sub-categories, as well as examples of patterns for each sub category. The full pattern system can be found in the Appendix of this report. In Section 5, we present a discussion on the limitations of requirements which could not be formally specified. Section 6 includes the related work, and we conclude in Section 7.

2. QUALITY ASSURANCE OF WEB APPLICATIONS

In [18,21], the authors discuss the main quality attributes of WAs, which are explicitly dubbed as success criteria [18] for WAs. The most important quality attributes of WAs are found to be the frequently called reliability/functionality, usability, and security/privacy. These attributes embody the appeal and trust of users, who if dissatisfied simply move away to another better quality WA. Therefore, these attributes are crucial to the success of WAs. Other necessary attributes to high quality WAs are availability, scalability, maintainability, and performance. In this research, we study the first three attributes, namely reliability and functionality, usability, and security and privacy. The main reason is that since we concentrate on producing temporal web specifications related to the behavior (execution) of the WA itself rather than those specifications that relate to web servers and access load. Therefore, the four latter attributes fall outside the scope of this paper. On the other hand, we identify another attribute which we believe is indispensable to the success of WAs, which we call the *custom* attribute.

2.1 Reliability and Functionality

Reliability has long been perceived as an attribute of safety critical software applications, such as telecommunications, aerospace, and medical devices. Therefore, most of the industry's software applications are non-safety critical and thus do not need to be highly reliable. However, WAs are critical to the commercial success of many businesses and if their WAs do not work reliably, the businesses will not succeed. In [18], the author characterizes web users as customers, unlike software applications users. Therefore, web users are able to make choices based on how reliable the WA is, while software application users tend to tolerate many flaws of the software as long as it can serve their expected purposes. For example, e-commerce applications deal with crucial items, which are money and personal information, such as credit cards, addresses, and buying habits. These applications also offer unrecoverable transactions when making online purchases, and have

deferred results since shipping purchases takes days or weeks. These factors mean that if WAs are unreliable, businesses will lose customers, and may lose large amounts of money. Therefore, WAs reliability is vital to their business's success and companies are willing to spend resources to ensure high reliability.

Functionality of WAs is a quality attribute that is intertwined with reliability since a dysfunctional WA cannot be reliable. WA pages and components should reflect its requirements. Therefore, each component and web page should have specific functions that it must perform in order to fulfill one or more of the WA requirements such as catalogue information presentation or users' payment data gathering.

2.2 Usability

Web users expect WAs to be as simple to learn as shopping at a store. Over the years, many web specialists have studied software usability and web usability. However, many WAs still do not meet the usability requirements anticipated by web users. On the other hand, web users exhibit little loyalty to a given WA since they expect to use it without any training; i.e., WAs that are not usable will not be used [21]. In summary, WAs must behave according to the user's expectations, offer only needed information, and provide navigation controls that are clear and obvious.

Specifically, WAs features need to be logical, accessible, and intuitive. Therefore, web developers should provide web users with a pleasant and efficient experience. Usability engineering represents the effort to make WAs understandable and usable by each web user.

2.3 Security and Privacy

At times when the web consisted only of static documents accessed via hyperlinks, the consequences of security breaches were relatively small. With the evolution and growth of WAs nowadays, business corporations face significant losses in revenue, large repair costs, and legal consequences due to security breaches of their WAs, let alone the loss of their credibility with their customers. Thus, it is essential that WAs safeguard user's data and other electronic information. At the same time, web users should be confident that no unauthorized users can access transactions or personal information stored and used by the WA.

2.4 Custom Attribute

While the above stated attributes are generic to WAs and relate to the satisfaction and appeal of the user, the custom attribute is related to the satisfaction of the various

stakeholders that are involved and have interest in the WAs. For example, while business owners are mostly interested in features which could result in directly achieving financial gains from the WA, advertisers are interested more to ensure that their commercial advertisements properly appear in key web pages of WAs and strategically published. Also, some WAs require particular specifications that are covered by the existing quality requirements related to the above stated attributes. For instance, in WAs related to governments or political parties, the interest might be in carefully using some statements, phrases or words, which has to appear a certain number of times, in key web pages. Also, in banking WAs, where security is the most regarded feature even above many other features, clients are allowed to mistakenly try to login to their accounts no more than three times, where the system is permanently locked afterwards. We list few examples of these requirements that we were able to formulate:

- *The number of a certain string/object should not appear more than a specific threshold*
- *If yahoo search is available, then google search should also be available*
- *A page X should be reachable without going through page Y*
- *The User visits Authentication page then Secure page then returns to Authentication and does this exactly twice*
- *A combination of specific strings/objects is absent throughout the WA*
- *Incorrect login info is allowed only three times, and then login is forbidden*
- *Number of links is balanced among multiple displays*

Therefore, the requirements or specifications to satisfy the custom attribute are pretty much subjective to the stakeholders themselves. For this reason, it is not possible to set a fix number of specific requirements satisfying this attribute. We collected a number of these requirements from the surveyed research work in Chapter 2, as well as few of them we were able to deduce from browsing through particular WAs.

3. LINEAR TEMPORAL LOGIC

Before we proceed to our proposed pattern based approach, we give an overview of LTL. LTL (sometimes called PTL or PLTL) extends traditional propositional logic with temporal operators.

LTL is used for property specification by major formal verification tools, for instance model checkers Spin and NuSMV.

Thus, LTL allows assertions about the temporal behavior of a system. An LTL formula φ has the following syntax:

$$\varphi ::= p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \cup \varphi) \mid (\text{G } \varphi) \mid (\text{F } \varphi) \mid (\text{X } \varphi)$$

where p is an atomic proposition, \cup is the *until* operator, G (or \square) is the *always* operator, F (or \diamond) is the *eventually* operator, and X (or \circ) is the *next* operator. Informally, $\varphi \cup \psi$ means that φ remains true until ψ becomes true. $\text{G } \varphi$ means that φ is always true. $\text{F } \varphi$ means that φ becomes true in a certain state. $\text{X } \varphi$ means that φ is true in the next state.

LTL semantics is defined by Pnueli [23] over infinite sequences of states that correspond to infinite or non-terminating sequences of computations. LTL deals only with infinite behavior.

Let $M = (S, T, S_0, P, \mathcal{L})$ be a Kripke structure, where S is a set of states, $T \subseteq S \times S$ is a transition relation, $S_0 \subseteq S$ is a set of initial states, P is a set of atomic propositions, and \mathcal{L} is a labeling function from S to the power set of P . An infinite state sequence $\pi = \langle s_0, s_1, \dots \rangle$ is called a *path* (execution) of M if $s_0 \in S_0$, $(s_i, s_{i+1}) \in T$ for all $i, i \geq 0$. $\pi^i = \langle s_i, s_{i+1}, \dots \rangle$ denotes the suffix of a sequence $\pi = \langle s_0, s_1, \dots \rangle$ starting at s_i . Also, note that $\pi^0 = \pi$.

The semantics of LTL formulae on a sequence of states is defined as follows:

- $\pi \models p \Leftrightarrow p \in \mathcal{L}(s_0)$,
- $\pi \models \neg\varphi \Leftrightarrow \pi \not\models \varphi$,
- $\pi \models \varphi \wedge \psi \Leftrightarrow \pi \models \varphi$ and $\pi \models \psi$,
- $\pi \models \text{X } \varphi \Leftrightarrow \pi^1 \models \varphi$,
- $\pi \models \text{G } \varphi \Leftrightarrow$ for all $i, i \geq 0$, $\pi^i \models \varphi$,
- $\pi \models \text{F } \varphi \Leftrightarrow$ for some $i, i \geq 0$, $\pi^i \models \varphi$,
- $\pi \models \varphi \cup \psi \Leftrightarrow$ there exists an $i, i \geq 0$, such that $\pi^i \models \psi$ and for all $j, 0 \leq j < i$, $\pi^j \models \varphi$.

The operator $\bar{\text{W}}$ is the *weak until* operator, such that $\varphi \bar{\text{W}} \psi = \varphi \cup (\psi \vee \text{G } \varphi)$.

Note that in the context of verification of WAs, WA is typically modeled by a Kripke structure, such that states represent web pages, transitions designate the links between the pages, and atomic propositions of each state are the attribute valuations in the

corresponding web page. These page attributes are usually identified by the user/tester of the WA.

While LTL is one of most popular formal notations for properties or requirements, it has certain limitations. For instance some properties, such as reachability of certain page can not be described. However, reachability of page could be described in the negative way, that is as unreachability ($G \neg p$). Clearly, if page is not unreachable, it is reachable. Other difficulties in use LTL along with proposed solutions are discussed the next section.

4. PATTERN BASED APPROACH

Formal methods have been recently recognized as a valuable means to provide dependable and practical solutions for the analysis and verification of applications in various domains. In particular, model checking techniques have been widely supported by diverse tools and methods for automatic verification of systems, overcoming several drawbacks related to the scalability of the involved modeling and analysis techniques, and the state space explosion problem. However, researchers have long acknowledged an important impediment to the transfer of those techniques from research to practice. Despite the automation of verification techniques, users of model checkers still must be able to specify system requirements in the specification language of the model checker. Namely, there exists a costly learning curve associated to the understanding of the underlying mathematical foundations, in particular, mastering the temporal logic theory for property specifications for model checking of systems.

As an example, we present the following requirement for some types of WAs:

Incorrect login info is allowed only three times in a row, and then login is forbidden.

To verify this requirement using LTL model checker, the web developer or tester has to translate it into the following LTL formula:

$$F(\text{login}) \rightarrow F(\text{login} \wedge X(\neg(\text{forbidden}) \vee (\text{login} \wedge X(\neg(\text{forbidden}) \vee (\text{login} \wedge X(\neg(\text{forbidden}) \vee X(\text{forbidden} \wedge G(\neg \text{login}))))))))))$$

such that *login* designates the login page, and *forbidden* designates the page indicating that the user is not allowed to login anymore after 3 trials, and that for instance he has to contact the company. This property apply mostly to WAs for online banking, where the user is given only 3 chances to an incorrect login, and for security reasons, the login is blocked, until the problem is fixed by the bank.

Clearly, writing such formula is a daunting task for web developers and testers. Not only the formula is difficult to read and understand, but it also difficult to correctly formulate without having the expertise in LTL.

To alleviate the above stated problem, we propose a pattern based approach to present the web specifications in an intuitive and easy to use manner. Originally, design patterns [11] were developed to capture descriptions of recurring solutions to design problems, requirements of these solutions, the means to satisfy those requirements, and instances of the solutions. Practitioners, when identifying similar requirements in their systems, use the appropriate design patterns addressing those requirements, and instantiate solutions that represent those patterns. Dwyer et al. have developed the Specification Pattern System (SPS) [26,8,9], where a property specification pattern describes the essential structure of some aspect of a system's behavior and provides expressions of this behavior in a range of common formalisms [26]. However, we are not aware of any attempt to build a library of formal specification patterns that catalogue the requirements necessary to ensure high quality WAs, and represent them in an easy to use manner mainly to the web developer and tester.

One could argue that the SPS could be used to represent web related specifications. Nonetheless, the SPS is a result of a conducted survey of property specification examples collected from various resources that deal with finite state verification of distributed systems, reactive systems, and timed systems; none of the surveyed resources included verification of WAs. For this reason, when we formulate the collected web quality requirements, although we employ patterns from the SPS, many of the web related formulae fall outside of the range of the SPS's patterns. In other cases, we should use of several SPS patterns to represent a single web specification.

On the other hand, the SPS defines a set of scopes in which a property can be verified. Each of the scopes is an interval of states/events with start and end delimiters. Meanwhile, in our work previously published [12,13,14] we identify scopes that constitute arbitrary sets of states. Also, many of the quality requirements surveyed concerned only a subset of the web pages of the WAs. Therefore, a number of our patterns employ the proposed scope operators to specify the related properties. On the other hand, we believe that the scope operators can be easily used in most of the patterns to specify them for multi frame WAs.

In the following, we present a library of web property specifications; we call it the Web Specification Pattern System (WeSPaS). This library is the result of surveying several resources related to web quality assurance [26,21,19,26,25], and web usability namely, IBM usability group [26], and various research work in the area of analysis and verification of WAs. We also relied on results of a Ph.D. research work in progress [19]. We also deduce some requirements by observing particularities related to some types of applications. In these resources, various requirements are developed for quality assurance of WAs. Among them, we identified about 120 common requirements that can be formally specified and used in verification of WAs.

These patterns range from the simple ones such as: *Number of fonts used in each page is less than 4*, mapped into LTL as: $G(\text{numfonts} < 4)$, to the complex ones such as: *Banking information is entered no more than once before submitting form*, mapped into LTL as:

PrecedenceGobally $((\neg (\textit{banking_info}) W (\textit{banking_info} W (G \neg (\textit{banking_info}))))), \textit{submit})$. Note that PrecedenceGobally is a pattern from the SPS [26] .

4.1 Categorization

To classify the web specifications, we do not keep the classical categorization of quality attributes. The reason is that some specifications identified as custom, or stakeholders specific, might overlap with the other quality attributes, while others might not fit in any of them. On the other hand, when analyzing the various quality requirements, it is possible to distinguish between requirements related to the ergonomics and design of web pages, and requirements related to the functionality of WAs which could cover a range of web pages of interest. For this reason, we propose a classification of those requirements (thus patterns) using two main categories:

Non-Functional. These requirements apply mainly to the design and ergonomics of web pages. They deal with standardization requirements (as in page 57) related to links in pages, content management of pages, and navigation between pages. Therefore, we identify sub-categories:

1. Links
2. Content
3. Navigation

Functional. These requirements concern the functionality of WAs. We notice that many of these requirements concern a wide range of WAs. On the other hand, we realize that e-commerce WAs have specific quality requirements when it comes to the functionality, which do not necessarily apply to other types of WAs. Therefore, we identify two sub-categories, one concerns WAs in general, and the other concerns e-commerce applications. Within each sub-category, we identified the following groups:

1. General
 - 1.1. Reachability
 - 1.2. Security and Authentication
 - 1.3. Other
2. E-commerce
 - 2.1. Customer Service

2.2. Trust

2.3. Product Info and Navigation

2.4. Purchase Transaction

4.2 Template

In order to archive the web property patterns in a library, we have to define a standard template that provides useful information about a given property pattern. We propose the template which consists of:

- *ID*: a unique identifier for each pattern.
- *Pattern description*: an English description of the quality requirement.
- *Category*: the category and subcategory to which the pattern belongs.
- *Page Attributes*: the involved page attributes in the LTL formulation.
- *LTL Mapping*: the mapping of the quality requirement into LTL formula.
- *Comments*: additional information concerning the pattern and its formulation.
- *Source*: the source where the quality requirement of the pattern has been found.

These fields of the template help the web developer and tester identify the properties of interest and provide the LTL formulation of the properties to verify. Also, the template's fields help in extending the library of patterns with new patterns. Note that each pattern is assigned a unique ID which encodes the first letters of the main category and subcategories followed by the number of the requirement. For convenience, we identify the sources of patterns as follows:

- *Opquast* for [25],
- *IBM usability group* for [26],
- *Research in progress* for [19],
- *Literature* for [24,5], and
- *Newly introduced* for patterns identified during the research work of this thesis.

5. LIBRARY OF WEB PROPERTY PATTERNS

In this section, we present the web property patterns we have archived in our library, categorized in their corresponding groups. Quality requirements are formalized as LTL properties that could be checked by a model checker given that the atomic propositions that constitute those properties are attribute valuations that exist in individual pages. We only present examples of full patterns for each category and sub-category; the complete library can be found in the Appendix of this report. Finally, we discuss examples of quality requirements that are not checkable using a model checker.

5.1 Non-Functional Patterns

In this section we present the non-functional patterns. They are classified into Links, Content, and Navigation. They merely concern standard requirements for good design and ergonomics of WAs.

5.1.1 Links

1. Broken links are absent.
2. Deadlocks are absent.
3. URLs have “/” at the end.
4. Only links are underlined.
5. Number of links in each display (page) should not be more than a given number (depends on size of application).
6. Internal URLs have no spaces.
7. URL of current page is not a link in page itself.

We present in Table 1 an example template of one the Link pattern number 5.

ID	NFL5
Pattern description	Number of links in each display (page) should not be more than n (depends on size of application)
Category	Non-functional – Links
Page Attributes	num_links : number of links in individual pages n : threshold of number of links
LTL Mapping	UniversalityGlobally ($num_links \leq n$)
Comments	
Source	Research in progress

Table 1. Link Pattern number 5.

5.1.2 Contents

1. Number of fonts used is less than 4.
2. Size of text is greater than 10.
3. Always name of the site or application is visible.
4. Always logo of the site or application is visible.
5. Site map exists.
6. Index exists.
7. Glossary exists.
8. Always there is a link to home page.
9. Always there is a link to important information.
10. Titles are short (less than 25 characters).
11. Date of last changes exists.
12. Number of frames less than 4.
13. If images exist, their format is gif or jpeg.
14. Address is present (email, phone, fax, postal).
15. Visitors counter is present.
16. Mandatory and optional form fields are distinct.

17. Every text field has a label in forms.
18. Frames are absent in application.
19. In all addresses country is mentioned.
20. Area code is present with telephone number.
21. Menus are present on all pages.
22. 404 page is personalized and main menu appears on it.
23. If plug in is suggested then size should be indicated.
24. Pop-ups have a close button.
25. The number of a certain string/object should not appear more than a specific threshold.
26. If yahoo search is available then google search should also be available.
27. A combination of specific strings/objects is absent.

Table 2 is an illustration of Content pattern number 22.

ID	NFC22
Pattern description	On 404 page, main menu is always present
Category	Non-functional – Content
Page Attributes	<i>status</i> : HTTP status code <i>num_links</i> : counts the number of links <i>menu_exists</i> : Boolean indicating the presence of menu
LTL Mapping	UniversalityGlobally ($status = 404 \rightarrow menu_exists \wedge num_links > 0$)
Comments	
Source	Opquast

Table 2. Content Pattern number 22.

5.1.3 Navigation

1. Link to a printable version exists.
2. Link to FAQ exists.

3. Link to “what’s new” page exist.
4. Link to copyright exists.
5. Link to Author name, qualifications, editor, webmaster, department, etc. exists.
6. Link to date of creation, publication, expiration, last modified exists.
7. There exist 1 or 2 links to Author or webmaster.
8. There exist 1 or 2 links to moderator of public forums.
9. If forum exists, then conditions of moderation exist.
10. If plug in is suggested then a link for it is present.
11. If plug in is suggested then a link for it is present in all pages.
12. After an order is submitted, the time for delivery is indicated.
13. If newsletter exists, frequency of sending newsletters is present in application.
14. Schedule and charge of services are present.

As an illustration, we present in Table 3, the Navigation pattern number 6.

ID	NFN7
Pattern description	There exist 1 or 2 links to Author or webmaster
Category	Non-functional – Navigation
Page Attributes	<i>webmaster</i> : integer identification of webmaster page <i>webmaster_link</i> : Boolean indicating the presence for link to webmaster page
LTL Mapping	ExistenceGlobally (<i>webmaster</i>) → BoundedExistenceGlobally (<i>webmaster_link</i>)
Comments	
Source	Opquast

Table 3. Navigation Pattern for number 7.

5.2 Functional Patterns

Here we present the functional patterns. They are classified into two main sub-categories: General, and E-commerce. These patterns are related to the functionality and expected behavior of WAs.

5.2.1 General WAs

The patterns of this class concern WAs in general. They include specifications about the reachability of certain pages. They also include specifications concerning the security and authentication in WAs, as well as other general specifications.

5.2.1.1 Reachability

1. Home page is reachable from every page.
2. Always link to a text version of page exists.
3. Always help page is accessible.
4. Always reference page is accessible.
5. Site map is reachable from every page.
6. Copyright information is reachable from every page.
7. If the WA is for a specific audience, it should be mentioned at least in home page.
8. If different language of application is available, then it is reachable from every page.
9. A page X is reachable without going through a page Y.

As an illustration, we present in Table 4, the Reachability pattern number 1.

ID	FGR1
Pattern description	Home page is reachable from every page
Category	Functional – General – Reachability
Page Attributes	<i>home</i> : integer identifying home page
LTL Mapping	Negation: $\text{AbsenceGlobally}(\textit{home})$
Comments	To check this pattern, it is negated. The property formulated is “on all paths home page is absent”. If the result of verification gives a counter example, it means the model checker found at least a path containing the home page. Then the original property is valid.
Source	Literature

Table 4. Reachability Pattern number 1.

5.2.1.2 Security/Authentication

1. Always the logo for security license is present.

2. Always the link for security information is present.
3. Incorrect login is allowed only 3 times, and then login is forbidden.
4. Secure pages are opened in a different window.
5. Secure pages are not reachable without going through authorization pages.
6. Banking information is entered no more than once before submitting form.
7. User visits Authentication page then Secure page then returns to Authentication and does this exactly twice.

As an illustration, we present in Table 5, the Security pattern number 6.

ID	FGS6
Pattern description	Banking information is entered no more than once before submitting form
Category	Functional – General – Security and Authentication
Page Attributes	<i>Banking_info</i> : Boolean identifying the presence of fields for banking information <i>Submit</i> : identification of page where form submit action exists
LTL Mapping	PrecedenceGlobally ($(\neg (banking_info) \ W (banking_info \ W (G \neg (banking_info))))$), <i>submit</i>)
Comments	
Source	Newly introduced

Table 5. Security Pattern number 6.

5.2.1.3 Others

1. Search function is always present on all pages.
2. If WA is larger than 15 pages, then advanced search is always available.
3. If page is a questionnaire or comments, then feedback page is next.
4. In forms, a link to go back and correct fields exists.
5. In forms, a validation / feedback page exists before submit action.
6. In forms, a confirmation of submission exists after submit action.
7. In forms, a link to info on how to fill out forms exists.
8. In forms, a link to security and privacy info is always present.

9. In forms, a preview page allowing changing info exists before submit action.
10. User can proceed navigation with links after form submission.
11. Pop-ups appear only once.
12. Pop-ups are absent from internal pages and home page.
13. Certain content (adult, violence, etc.) are absent.
14. Whenever there is subscribe to newsletter option, unsubscribe is possible.

In Table 6, we present the pattern number 13 of this category.

ID	FGO13
Pattern description	Certain content (adult, violence, etc.) are absent
Category	Functional – General – Other
Page Attributes	<i>adult_str</i> : string count of a specific prohibited adult related phrase/word <i>violence_str</i> : string count of a specific prohibited violence related phrase/word
LTL Mapping	AbsenceGlobally $((adult_str > 0) \wedge (violence_str > 0))$
Comments	
Source	Newly introduced

Table 6. Others Pattern number 13.

5.2.2 E-Commerce WAs

The class of E-commerce patterns comprises specification patterns that concern mainly e-commerce WAs. We archive those patterns into the following sub-categories: Customer support, Trust, Product Info and Navigation, and Purchase Transaction. Note that this classification is mainly adopted from the Usability Group at IBM [26].

5.2.2.1 Customer Support

1. Contact info exists and is accessible from every page.
2. Assistance for passwords (change, remember, ...) exist.
3. There are three modes of product search: search function, list of products, product suggestion.
4. FAQ / glossary is always within 3 clicks from every page.

5. Link to FAQ exists on all shopping pages and “How to order” pages.
6. Shipping and delivery date info is accessible within 3 clicks from any page.
7. After submitting an order, there is a link for tracking order or the shipper’s site.
8. It is possible to change or cancel an order after submission.
9. After order submission, info on warranties and service agreements is accessible.
10. Info on terms and conditions is accessible before order submission or check out.
11. Link to info on reimbursement / return conditions and procedure is present on order pages.
12. Info on security / level / mode is always present.
13. Info on conditions of buying / usage is reachable from all pages.

In Table 7, we present the pattern number 4 of this category.

ID	FEC4
Pattern description	FAQ / glossary is always within 3 clicks of every page.
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>faq</i> : integer identification of the FAQ page
LTL Mapping	Negation: AbsenceGlobally($\neg faq \rightarrow X (faq \vee X (faq \vee X (faq)))$)
Comments	To check this pattern, it is negated. The property formulated is “on all paths FAQ/glossary cannot be found within 3 clicks of every page”. If the result of verification gives a counter example, it means the model checker found at least a path containing the home page. Then the original property is valid.
Source	IBM usability group

Table 7. Customer Support Pattern number 4.

5.2.2.2 Trust

1. Privacy policy is accessed from every page.
2. Security policy is accessed from store front, order list, shopping cart, and order form.
3. There exists third party validation seal for trust.

4. Mission statement, history, etc. exist.

In Table 8, we present the pattern number 2 of this category.

ID	FET2
Pattern description	Security policy is accessed from store front, order list, shopping cart, and order form
Category	Functional – E-commerce – Trust
Page Attributes	<i>front, order_list, shopping_cart, security_policy</i> : integers identifying the corresponding web pages
LTL Mapping	$G (front \vee order_list \vee shopping_cart \rightarrow X (security_policy))$
Comments	
Source	IBM usability group

Table 8. Trust Pattern number 2.

5.2.2.3 Product Info and Navigation

1. Product info is reachable within 3 clicks.
2. From home page the following is provided: search products, browse categories, and recommend products.
3. Shopping pages are accessed from every other page.
4. “Best selling” pages are accessed directly either from home page or product category page.
5. Link to a product description and adding product is available from shopping cart page.
6. Within the same product category, product description pages are linked with forward and backward buttons/links.
7. A link to catalogue exists on every page.
8. Promotions exist only on home page and category navigation pages.
9. 1 or 2 promotions exist simultaneously with a link to more promotions.
10. Promotions of certain products are only present either on the Home page or on Shopping pages and their number does not exceed 2.
11. Promotions are absent on product comparison pages and ordering pages.
12. Cross-selling and up-selling pages are reachable only from product detail pages.

13. Types of accepted payments are indicated and present on all product pages.

In Table 9, we present the pattern number 4 of this category.

ID	FEPN4
Pattern description	“Best selling” pages are accessed directly either from home page or product category page
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>home, category, best_selling</i> : integers identifying the corresponding web pages
LTL Mapping	UniversalityGlobally ($home \vee category \rightarrow X (best_selling)$)
Comments	
Source	IBM usability group

Table 9. Product Info and Navigation Pattern number 4.

5.2.2.4 Purchase Transaction

1. It is possible to view a product description, total cost, change a quantity, add/remove a product, save an order, and proceed to order form.
2. After an order is submitted, a confirmation page is shown to the user.
3. From confirmation page, it is possible to reach registration page (acquire ID, and password).
4. If the user registers, then he can use shortcut order (by entering only ID, password, and credit card number).
5. There is a link from the order form to the order list.
6. Credit card info is submitted no more than once before submitting an order.
7. After ordering, conditions of guarantee are available for the user.
8. After ordering, address of return is indicated.
9. After ordering, return service (charge, hours of work, contact, etc.) are indicated.
10. Availability of products is indicated before validation of order.

In Table 10, we present the pattern number 6 of this category.

ID	FEPT6
Pattern description	Credit card info is entered no more than before submitting an order
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>Credit_card</i> : Boolean indicating the presence of fields requesting credit card info in pages <i>submit</i> : Boolean indicating the submit order page
LTL Mapping	PrecedenceGlobally ($(\neg (credit_card) W (credit_card W (G \neg (credit_card))))$), <i>submit</i>)
Comments	
Source	Newly introduced

Table 10. Purchase Transaction Pattern number 6.

6. DISCUSSION

In this paper, the library of web specification patterns covers a number of important quality requirements of WAs. The list of those patterns is non exhaustive and can be further extended with new patterns. Also, this library does include all types of quality requirements. The reason is that our approach is based on formally specified patterns. Therefore, it is important to identify the elements of the quality requirements as attributes that can be evaluated within the web pages of the WA. Those attributes when evaluated, constitute the atomic propositions of the LTL formulae of the corresponding patterns.

In fact, there exists a range of quality requirements which we have identified but are not able to specify as LTL properties. Those requirements are either:

1. too generic to the extent that it is impossible to identify corresponding attributes in web pages of the WA, or
2. describe necessary requirements for the quality which do not directly involve the WA itself, but rather involve the business or company that owns the WA.

We list here few examples of such requirements:

- 404 error page is personalized.
- At least 2 modes of payments can be performed by the user.
- An answer to a request for information is sent within 5 working days of reception.
- An acknowledgment of reception of a request is sent by email within 60 minutes.

- The nature and quantifiable characteristics of products and services (dimensions, duration, capacity, etc.) are stated.
- Subscription to newsletters is submitted to a verification procedure.

On the other hand, many of the patterns of our library employ specification patterns from the SPS. Also, the patterns of the SPS can be mapped not only to LTL, but also to a range of specification languages such as CTL, and QRE. Therefore, we believe that this adds to the flexibility and richness of our library of patterns. In particular, the web patterns which are composed of SPS patterns could be easily formulated using the available SPS's mappings to the various specification languages. Such flexibility makes our system of web specification patterns more generic, making verification of WAs possible using various model checkers and analysis tools.

7. RELATED WORK

Several efforts [3,8,9,7] have been made to reconcile complex formal approaches based solutions with their practical usage in industrial software applications, and make formal verification techniques understandable and easy to use by software developers. However, we still see that the contributions resulting from those efforts are difficult to use by web developers since they are too abstract and cannot easily be projected in the web domain.

Dwyer et al [8,9] present and analyze over 500 temporal properties, classified in the proposed specification pattern system [26]. The patterns constitute abstractions of specifications formulated for different formalisms in which such abstractions are not supported. The patterns are categorized into *occurrence* and *order*, and are defined on five scopes: *Globally*, *Before R*, *After Q*, *Between Q and R*, and *After Q until R*, which represent intervals/regions in which properties should be validated. The temporal properties surveyed in Dwyer's work refer to finite state verification of distributed systems, reactive systems, and timed systems, but not verification of WAs. While in our work, we not only survey existing temporal web properties, but we study the essential requirements that satisfy quality attributes of WAs, which in turn we translate into temporal properties. Although we employ many of Dwyer's patterns at the specification level, we introduce the web patterns in templates using a terminology familiar to the web community. We also use several patterns of the SPS to compose a single web pattern. Moreover, our WeSPaS include specifications using scopes of arbitrary subsets of states [12,13], which is not possible using SPS alone.

Chechik et al [3] extend the SPS with edge based LTL property formulations. Dillon et al [7] introduce in the Graphical Interval logic (GIL) which is similar to the pattern system. Although the GIL employ a graphical based mechanism to specify properties, it is still considered at a high level of abstraction, and web users would need to put efforts in understanding and correctly using it.

The only work we have found that uses patterns in Web related properties is that of Pereira et al [22]. The authors use patterns of the SPS to formulate properties related e-commerce applications. However, this approach was used as a proof of concept and for experimentation purposes. Unlike our work, they do not provide a library of specifications that helps in correctly choosing and instantiating it for a particular e-commerce specification.

In conclusion, we claim that our pattern based approach to provide a library of formal specifications is novel and provides a practical solution to the complex problem of temporal logic formulation of web properties. To our knowledge, there has been no other attempt in the literature or industry to provide such a solution.

8. CONCLUSION AND FUTURE WORK

In this paper, we developed a Web Specification Pattern System to reduce the hurdle for web users and testers of formally specifying web properties. Based on various quality attributes that are necessary for the quality assurance of WAs, we surveyed a range of resources in industry and research that set numerous requirements to meet those quality attributes. We collected about 120 requirements and built the WeSPaS, which is a repository of web patterns, categorized into Functional and Non-functional patterns. The Non-functional category includes the sub-categories Links, Contents, and Navigation. The Functional category includes the subcategories General, and E-commerce, which in turn have their own sub categories. Each web pattern includes the description of a quality requirement, a unique ID reflecting its category, the involved page attributes, the corresponding LTL formula, and its source.

In the future, a tool support can be developed and integrated with our web analysis prototype tool [16]. This tool support would allow the user to easily browse through the patterns, choose a particular pattern, and use its LTL formula for model checking. This opens the door to another direction of making the WeSPaS a public repository where other researchers and experts can add their contributions.

REFERENCES

1. Boroday S, Petrenko A, Sing J, Hallal H. Dynamic Analysis of Java Applications for Multi Threaded Anti patterns. *3rd International Workshops on Dynamics Analysis*, St-Louis, MI, USA, May 17th, 2005.
2. Benedikt M, Freire J, Godefroid P. VeriWeb: Automatically Testing Dynamic Web Sites. *11th International World Wide Web Conference*, Hawaii, U.S.A.
3. Chechik M, Paun D. Events in Property Patterns. *6th Int. SPIN Workshop on Theoretical and Practical Aspects of SPIN Model Checking*, LNCS, September 1999, 1680: 154-167.
4. Clarke EM, Grumberg O, Peled DA. *Model Checking*. MIT Press, 2000.
5. De Alfaro L. Model Checking the World Wide Web. *13th International Conference on Computer Aided Verification*, Paris, France, July 2001.
6. Di Sciascio E, Donini FM, Mongiello M, Totaro R, Castelluccia D. Design Verification of Web Applications Using Symbolic Model Checking. *5th Int. Conference on Web Engineering*, LNCS 3579, Sydney, Australia, 2005.
7. Dillon LK, Kutty G, Moser LE. A Graphical Interval Logic for Specifying Concurrent Systems. *ACM Transactions on Software Engineering and Methodology*, April 1994, 3(2): 131-165.
8. Dwyer M, Avrunin GS, Corbett JC. Patterns in Property Specifications for Finite-state Verification. *21st Int. Conference on Software Engineering*, May, 1999.
9. Dwyer M, Avrunin GS, Corbett JC. Property Specification Patterns for Finite-state Verification. *2nd Workshop on Formal Methods in Software Practice*, March, 1998.
10. Dwyer MB and Clarke LA. Data flow analysis for verifying properties of concurrent programs. *Proceedings of the 2nd ACM SIGSOFT symposium on Foundations of software engineering*: 62-75, 1994.
11. Gamma E, Helm R, Johnson R, and Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
12. Haydar M, Boroday S, Petrenko A, Sahraoui H. *Adding Propositional Scopes to Linear Temporal Logic*. Technical Report [CRIM 05/05-06], Centre de Recherche Informatique de Montreal, May 2005.

13. Haydar M, Boroday S, Petrenko P, Sahraoui H. Propositional Scopes in Linear Temporal Logic. *5th Int. Conference on Novelles Technologies de la Repartition*, Gatineau, Canada, August 30-September 1, 2005.
14. Haydar, M., Boroday, S., Petrenko, A. and H. Sahraoui. "Properties and Scopes in Web Model Checking" In *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering (ASE 2005)*. Long Beach, California, USA, November 7-11, 2005.
15. Haydar M, Petrenko A, Boroday S, and Sahraoui H. A Formal Approach for Run-Time Verification of Web Applications using Scope-extended LTL. *Journal of Software Testing, Verification and Reliability*, Wiley, 2007 (under review).
16. Haydar M, Petrenko A, Sahraoui H. Formal Verification of Web Applications Modeled by Communicating Automata. *24th IFIP WG 6.1 IFIP Int. Conference on Formal Techniques for Networked and Distributed Systems*, LNCS, Spain, September 2004, 3235:115-132.
17. Haydar M. Formal Framework for Automated Analysis and Verification of Web-based Applications. *19th IEEE Int. Conference on Automated Software Engineering*, Linz, Austria, September 20-24, 2004.
18. Kan S. *Metrics and Models in Software Quality Engineering*, 2nd ed. Pearson, 2002.
19. Malak, G. Évaluation de la Qualité des Applications Web: Approche Probabiliste. *Département d'informatique et de recherches opérationnelles*. Montreal, Université de Montreal. Ph.D. Thesis (in progress), 2007.
20. Millerand F, Martial O. *Guide pratique de conception et d'évaluation ergonomique de sites Web*. Montréal, Centre de recherche informatique de Montréal, 2001. [CRIM-01/08-21].
21. Offutt J. Quality attributes of Web software applications. *IEEE Software*, 19(2): 25-32, 2002.
22. Pereira A, Song M, Gorgulho G, Meira Jr. W, Campos S. A Formal Methodology to Specify E-commerce Systems. *4th Int. Conf. on Formal Engineering Methods*, Shanghai, China, October 21-25, 2002.
23. Pnueli, A. The Temporal Logic of Programs. *18th IEEE Symposium on Foundations of Computer Science*, 1977, 46-57.
24. Stotts PD, Cabarrus CR. Hyperdocuments as Automata: Verification of Trace-Based Browsing Properties by Model Checking. *ACM Transactions on Information Systems*, January 1998, 16(1): 1-30.

25. Temesis - Qualité, conformité, et accessibilité des sites Internet. Opquast: Quality Best Practices for On-line Services. 2005, from <http://en.opquast.com/>.
26. The Specification Patterns System. 2002, from <http://patterns.projects.cis.ksu.edu/>.
27. IBM Usability. "Web Design Guidelines." 2005, from http://www-306.ibm.com/ibm/easy/eou_ext.nsf/publish/611.

APPENDIX

Non-Functional

Links

ID	NFL1
Pattern description	Broken links are absent
Category	Non-functional – Links
Page Attributes	<i>status</i> : HTTP status code
LTL Mapping	AbsenceGlobally (<i>status</i> = 404)
Comments	
Source	Research in progress

ID	NFL2
Pattern description	Deadlocks are absent
Category	Non-functional – Links
Page Attributes	<i>numlinks</i> : integer counting number of links in each page
LTL Mapping	AbsenceGlobally (<i>numlinks</i> = 0)
Comments	
Source	Research in progress

ID	NFL3
Pattern description	URLs have “/” at the end
Category	Non-functional – Links
Page Attributes	<i>URLslash</i> : boolean indicating if page URL is ended with a “/”
LTL Mapping	UniversalityGlobally (<i>URLslash</i>)
Comments	
Source	Research in progress - Opquast

ID	NFL4
Pattern description	Only links are underlined
Category	Non-functional – Links
Page Attributes	<i>textunderlined</i> : boolean indicating if any textual content is underlined in pages
LTL Mapping	AbsenceGlobally (<i>textunderlined</i>)
Comments	
Source	Research in progress - Opquast

ID	NFL5
Pattern description	Number of links in each display (page) should not be more than n (depends on size of application)
Category	Non-functional – Links
Page Attributes	<i>numlinks</i> : integer counting number of links in pages
LTL Mapping	UniversalityGlobally ($numlinks \leq n$)
Comments	
Source	Research in progress

ID	NFL6
Pattern description	Internal URLs have no spaces
Category	Non-functional – Links
Page Attributes	<i>internalURLnospaces</i> : boolean indicating if internal URLs have no spaces
LTL Mapping	UniversalityGlobally (<i>internalURLnospaces</i>)
Comments	
Source	Research in progress - Opquast

ID	NFL7
Pattern description	URL of current page is not a link in page itself
Category	Non-functional – Links
Page Attributes	<i>URLlink</i> : boolean indicating if page URL is a link inside the page
LTL Mapping	AbsenceGlobally (<i>URLlink</i>)
Comments	
Source	Research in progress

Contents

ID	NFC1
Pattern description	Number of fonts used in each page is less than 4
Category	Non-functional – Content
Page Attributes	<i>numfonts</i> : integer to count the number of fonts in the page
LTL Mapping	UniversalityGlobally ($numfonts \leq 4$)
Comments	
Source	Research in progress

ID	NFC2
Pattern description	Size of text is greater than 10
Category	Non-functional – Content
Page Attributes	<i>textsize</i> : integer indicating the size of text used in the page

LTL Mapping	UniversalityGlobally (<i>textsize</i> ≥ 10)
Comments	
Source	Research in progress

ID	NFC3
Pattern description	Always name of the site or application is visible
Category	Non-functional – Content
Page Attributes	<i>nameofsiteexists</i> : Boolean indicating the presence of the name of the site or application
LTL Mapping	UniversalityGlobally (<i>nameofsiteexists</i>)
Comments	
Source	Research in progress

ID	NFC4
Pattern description	Always logo of the site or application is visible
Category	Non-functional – Content
Page Attributes	<i>logoexists</i> : Boolean indicating of the presence of the logo in the page
LTL Mapping	UniversalityGlobally (<i>logoexists</i>)
Comments	
Source	Research in progress

ID	NFC5
Pattern description	In each page, link to Site map exists
Category	Non-functional – Content
Page Attributes	<i>Sitemapexists</i> : Boolean indicating the presence of a link to the site map
LTL Mapping	UniversalityGlobally (<i>sitemapexists</i>)
Comments	
Source	Research in progress

ID	NFC6
Pattern description	In each page, link to Index exists
Category	Non-functional – Content
Page Attributes	<i>indexexists</i> : Boolean indicating the presence of a link to the index page
LTL Mapping	UniversalityGlobally (<i>indexexists</i>)
Comments	
Source	Research in progress

ID	NFC7
Pattern description	In each page, link to Glossary exists

Category	Non-functional – Content
Page Attributes	<i>glossaryexists</i> : Boolean indicating the presence of a link to the Glossary page
LTL Mapping	UniversalityGlobally (<i>glossaryexists</i>)
Comments	
Source	Research in progress

ID	NFC8
Pattern description	Always there is a link to home page
Category	Non-functional – Content
Page Attributes	<i>homelinkexists</i> : Boolean indicating the presence of a link to Home page
LTL Mapping	UniversalityGlobally (<i>homelinkexists</i>)
Comments	
Source	Research in progress

ID	NFC9
Pattern description	Always there is a link to important information
Category	Non-functional – Content
Page Attributes	<i>infolinkexists</i> : Boolean indicating the presence of a link to the page containing important information
LTL Mapping	UniversalityGlobally (<i>infolinkexists</i>)
Comments	
Source	Research in progress

ID	NFC10
Pattern description	Titles are short (less than 25 characters)
Category	Non-functional – Content
Page Attributes	<i>length_title</i> : integer indicating the length of the title of the page
LTL Mapping	UniversalityGlobally ($length_title \leq 25$)
Comments	
Source	Research in progress

ID	NFC11
Pattern description	Date of last changes exists
Category	Non-functional – Content
Page Attributes	<i>last_change_date</i> : Boolean indicating of the date of last changes exist in a page
LTL Mapping	Negation: AbsenceGlobally (<i>last_change_date</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the date of last changes. then the original

	property is valid
Source	Research in progress

ID	NFC12
Pattern description	Number of frames less than 4
Category	Non-functional – Content
Page Attributes	<i>num_frames</i> : integer counting the number of frames of each page.
LTL Mapping	UniversalityGlobally ($num_frames \leq 4$)
Comments	
Source	Research in progress

ID	NFC13
Pattern description	If images exist, their format is gif or jpeg
Category	Non-functional – Content
Page Attributes	<i>num_images</i> : integer that counts the number of images in a page <i>not_gif_jpeg</i> : Boolean indicating the presence image format other than gif or jpeg
LTL Mapping	UniversalityGlobally ($((num_images > 0) \rightarrow \neg (not_gif_jpeg))$)
Comments	
Source	Research in progress

ID	NFC14
Pattern description	Address is present (email, phone, fax, postal)
Category	Non-functional – Content
Page Attributes	<i>addressexists</i> : Boolean indicating the presence of address in each page
LTL Mapping	Negation: AbsenceGlobally (<i>addressexists</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Research in progress

ID	NFC15
Pattern description	Visitors counter is present
Category	Non-functional – Content
Page Attributes	<i>visitcounter</i> : Boolean indicating the presence of visitor counter in each page
LTL Mapping	Negation: AbsenceGlobally (<i>visitcounter</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at

	least a page containing the address. then the original property is valid
Source	Research in progress

ID	NFC16
Pattern description	Mandatory and optional form fields are distinct
Category	Non-functional – Content
Page Attributes	<i>num_forms</i> : integer counting the number of forms in each page <i>fields_distinct</i> : Boolean indicating if fields of forms are distinct in each page
LTL Mapping	UniversalityGlobally ($num_forms > 0 \rightarrow fields_distinct$)
Comments	
Source	Research in progress

ID	NFC17
Pattern description	Every text field has a label in forms
Category	Non-functional – Content
Page Attributes	<i>num_forms</i> : integer counting the number of forms in each page <i>text_unlabeled</i> : Boolean indicating if at least one text field does not have a label
LTL Mapping	UniversalityGlobally ($num_forms > 0 \rightarrow \neg (text_unlabeled)$)
Comments	
Source	Research in progress

ID	NFC18
Pattern description	Frames are absent in application
Category	Non-functional – Content
Page Attributes	<i>num_frames</i> : integer counting the number of frames in each page
LTL Mapping	UniversalityGlobally ($num_frames = 0$)
Comments	
Source	Research in progress – Opquast

ID	NFC19
Pattern description	In all addresses country is mentioned
Category	Non-functional – Content
Page Attributes	<i>address_exists</i> : Boolean indicating the presence of an address in each page <i>country_exists</i> : Boolean indicating the presence of a field for country with each address
LTL Mapping	UniversalityGlobally ($address_exists \rightarrow country_exists$)
Comments	
Source	Opquast

ID	NFC20
Pattern description	Area code is present with tel. number
Category	Non-functional – Content
Page Attributes	<i>phonenum_exists</i> : Boolean indicating the presence of a field of phone number in each page <i>num_digits</i> : integer counting the number of digits for telephone number
LTL Mapping	UniversalityGlobally (<i>phonenum_exists</i> → <i>num_digits</i> = 10)
Comments	
Source	Opquast

ID	NFC21
Pattern description	Menus are present on all pages
Category	Non-functional – Content
Page Attributes	<i>menu_exists</i> : Boolean indicating the presence of menu
LTL Mapping	UniversalityGlobally (<i>menu_exists</i>)
Comments	
Source	Opquast

ID	NFC22
Pattern description	On, 404 page main menu is present
Category	Non-functional – Content
Page Attributes	<i>status</i> : HTTP status code <i>num_links</i> : counts the number of links <i>menu_exists</i> : Boolean indicating the presence of menu
LTL Mapping	UniversalityGlobally (<i>status</i> = 404 → <i>menu_exists</i> and <i>num_links</i> > 0)
Comments	
Source	Opquast

ID	NFC23
Pattern description	If plug in is suggested then size should be indicated on the same page
Category	Non-functional – Content
Page Attributes	<i>plugin_exists</i> : Boolean indicating the presence of a plugin <i>pluginsizepresent</i> : Boolean indicating the presence of size of plugin
LTL Mapping	UniversalityGlobally (<i>plugin_exists</i> → <i>pluginsizepresent</i>)
Comments	
Source	Opquast

ID	NFC24
Pattern description	Pop-ups have a close button
Category	Non-functional – Content
Page Attributes	<i>popup_exists</i> : Boolean indicating the presence of a popup <i>closebuttonexists</i> : Boolean indicating the presence of close button within the popup
LTL Mapping	UniversalityGlobally ($popup_exists \rightarrow closebuttonexists$)
Comments	
Source	Opquast

ID	NFC25
Pattern description	The number of a certain string/object should not appear more than a specific threshold
Category	Non-functional – Content
Page Attributes	<i>num_string</i> : integer counting the number of a certain string in each page
LTL Mapping	UniversalityGlobally ($num_string > n$)
Comments	
Source	Newly introduced

ID	NFC26
Pattern description	If yahoo search is available then google search should also be available
Category	Non-functional – Content
Page Attributes	<i>yahoosearch</i> : Boolean indicating of search function of yahoo exists in each page <i>googlesearch</i> : Boolean indicating of search function of google exists in each page
LTL Mapping	UniversalityGlobally ($yahoosearch \rightarrow googlesearch$)
Comments	
Source	Newly introduced

ID	NFC27
Pattern description	A combination of specific strings/objects is absent
Category	Non-functional – Content
Page Attributes	<i>num_string1</i> : integer counting the number of occurrences <i>string1</i> <i>num_string2</i> : integer counting the number of occurrences <i>string2</i> <i>num_string3</i> : integer counting the number of occurrences <i>string3</i>
LTL Mapping	UniversalityGlobally ($num_string1 = 0 \wedge num_string2 = 0 \wedge num_string3 = 0$)
Comments	This pattern could be extended to check for the combination of any number of strings/objects

Source	Newly introduced
---------------	------------------

Navigation

ID	NFN1
Pattern description	Link to a printable version exists in every page
Category	Non-functional – Navigation
Page Attributes	<i>printable_link</i> : Boolean indicating the presence for link to printable version of page
LTL Mapping	UniversalityGlobally (<i>printable_link</i>)
Comments	
Source	Research in progress - Opquast

ID	NFN7
Pattern description	Link to FAQ exists
Category	Non-functional – Navigation
Page Attributes	<i>faqlink</i> : Boolean indicating the presence for link to FAQ page
LTL Mapping	Negation: AbsenceGlobally (<i>faqlink</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Research in progress

ID	NFN3
Pattern description	Link to “what’s new” page exist
Category	Non-functional – Navigation
Page Attributes	<i>whatnewlink</i> : Boolean indicating the presence for link to what’s new page
LTL Mapping	Negation: AbsenceGlobally (<i>whatnewlink</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Research in progress - Opquast

ID	NFN4
Pattern description	Link to copyright exists
Category	Non-functional – Navigation
Page Attributes	<i>copyrightlink</i> : Boolean indicating the presence for link to what’s new page
LTL Mapping	Negation: AbsenceGlobally (<i>copyrightlink</i>)

Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Research in progress - Opquast

ID	NFN5
Pattern description	Link to Author name, qualifications, editor, webmaster, department, etc. exists
Category	Non-functional – Navigation
Page Attributes	<i>authorinfo</i> link: Boolean indicating the presence for link to author's and webmaster information page
LTL Mapping	Negation: AbsenceGlobally (<i>authorinfo</i> link)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Research in progress - Opquast

ID	NFN6
Pattern description	Link to date of creation, publication, expiration, last modified exists
Category	Non-functional – Navigation
Page Attributes	<i>webinfo</i> link: Boolean indicating the presence for link to page of information on dates of creation and updates, publication, expiration, etc.
LTL Mapping	Negation: AbsenceGlobally (<i>webinfo</i> link)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Research in progress

ID	NFN7
Pattern description	There exist 1 or 2 links to Author or webmaster
Category	Non-functional – Navigation
Page Attributes	<i>webmaster</i> : integer identification of webmaster page <i>webmaster_link</i> : Boolean indicating the presence for link to webmaster page
LTL Mapping	ExistenceGlobally (<i>webmaster</i>) → BoundedExistenceGlobally (<i>webmaster_link</i>)
Comments	This property could not be specified in LTL since there is no quantification of paths in LTL, so the LTL mapping necessitated

	the validity of the property on all paths.
Source	Opquast

ID	NFN7
Pattern description	There exist 1 or 2 links to Author or webmaster
Category	Non-functional – Navigation
Page Attributes	<i>webmaster</i> : integer identification of webmaster page <i>webmaster_link</i> : Boolean indicating the presence for link to webmaster page
LTL Mapping	ExistenceGlobally (<i>webmaster</i>) \rightarrow BoundedExistenceGlobally (<i>webmaster_link</i>)
Comments	This property could not be specified in LTL since there is no quantification of paths in LTL, so the LTL mapping necessitated the validity of the property on all paths.
Source	Opquast

ID	NFN9
Pattern description	Conditions of moderation of forums exist
Category	Non-functional – Navigation
Page Attributes	<i>forum</i> : Boolean designating the existence of forum page <i>moderationcond</i> : Boolean designating the existence of forum moderation condition
LTL Mapping	F (<i>forum</i>) \rightarrow F (<i>moderationcond</i>)
Comments	
Source	Opquast

ID	NFN10
Pattern description	If plug in is suggested then a link for it is present
Category	Non-functional – Navigation
Page Attributes	<i>plugin_exists</i> : Boolean indicating of request for plugin exists in page <i>plugin_link</i> : Boolean indicating the presence for link of plugin
LTL Mapping	UniversalityGlobally (<i>plugin_exists</i> \rightarrow <i>plugin_link</i>)
Comments	
Source	Opquast

ID	NFN11
Pattern description	If plug in is suggested then a link for it is present in all pages
Category	Non-functional – Navigation
Page Attributes	<i>plugin_exists</i> : Boolean indicating of request for plugin exists in page <i>plugin_link</i> : Boolean indicating the presence for link of plugin

LTL Mapping	UniversalityGlobally ($plugin_exists \rightarrow \wedge$ UniversalityGlobally ($plugin_link$))
Comments	
Source	Opquast

ID	NFN12
Pattern description	After an order is submitted, the time for delivery is indicated
Category	Non-functional – Navigation
Page Attributes	<i>order</i> : Boolean indicating that order is submitted <i>deliverytime</i> : Boolean indicating the presence for time of delivery
LTL Mapping	ResponseGlobally ($order, deliverytime$)
Comments	
Source	Opquast

ID	NFN13
Pattern description	If newsletter exists, frequency of sending newsletters is present in application
Category	Non-functional – Navigation
Page Attributes	<i>newsletter</i> : Boolean indicating if page is a newsletter <i>newsfrequency</i> : Boolean indicating the presence frequency of newsletter
LTL Mapping	Negation: $F(newsletter) \wedge$ AbsenceGlobally ($newsfrequency$)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Opquast

ID	NFN14
Pattern description	Schedule and charge of services are present
Category	Non-functional – Navigation
Page Attributes	<i>webmaster</i> : integer identification of webmaster page <i>webmaster_link</i> : Boolean indicating the presence for link to webmaster page
LTL Mapping	Negation: AbsenceGlobally ($schedulechargeinfo$)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Opquast

Functional

General

Reachability

ID	FGR1
Pattern description	Home page is reachable from every page
Category	Functional – General – Reachability
Page Attributes	<i>home</i> : integer identifying home page
LTL Mapping	Negation: AbsenceGlobally (<i>home</i>)
Comments	To check this pattern, it is negated. The property formulated is “on all paths home page is absent”. If the result of verification gives a counter example, it means the model checker found at least a path containing the home page. Then the original property is valid.
Source	Literature

ID	FGR2
Pattern description	Always link to a text version of page exists
Category	Functional – General – Reachability
Page Attributes	<i>linkto_text_version</i> : Boolean indicating the presence of link to text version of page
LTL Mapping	UniversalityGlobally (<i>linkto_text_version</i>)
Comments	
Source	Research in progress - Opquast

ID	FGR3
Pattern description	Always help page is accessible
Category	Functional – General – Reachability
Page Attributes	<i>help</i> : integer identifying help page
LTL Mapping	UniversalityGlobally (ExistenceGlobally (<i>help</i>))
Comments	
Source	Research in progress - Opquast

ID	FGR4
Pattern description	Always reference page is accessible
Category	Functional – General – Reachability
Page Attributes	<i>reference</i> : integer identifying reference page
LTL Mapping	UniversalityGlobally (ExistenceGlobally (<i>reference</i>))
Comments	

Source	Research in progress - Opquast
---------------	--------------------------------

ID	FGR5
Pattern description	Home page is reachable from every page
Category	Functional – General – Reachability
Page Attributes	<i>sitemap</i> : Boolean expression identifying sitemap page
LTL Mapping	Negation: $\text{AbsenceAfter}(\textit{sitemap}, \neg(\textit{sitemap}))$
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Littrature

ID	FGR6
Pattern description	Copyright information is reachable from every page
Category	Functional – General – Reachability
Page Attributes	<i>copyright</i> : Boolean expression identifying copyright page
LTL Mapping	Negation: $\text{AbsenceAfter}(\textit{copyright}, \neg(\textit{copyright}))$
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Opquast

ID	FGR7
Pattern description	If WA for a specific audience, it should be mentioned at least in home page
Category	Functional – General – Reachability
Page Attributes	<i>home</i> : integer identifying home page <i>audience</i> : Boolean indicating the designation of audience in page
LTL Mapping	$G(\textit{home} \rightarrow (\textit{audience} \wedge X(G(\neg(\textit{audience})) \vee F(\textit{audience}))))$
Comments	
Source	Opquast

ID	FGR8
Pattern description	If different language of application is available, then it is reachable from every page
Category	Functional – General – Reachability
Page Attributes	<i>Diff_lang_home</i> : integer identifying home page of application in a different language
LTL Mapping	Negation: $\text{AbsenceAfter}(\textit{diff_lang_home}, \neg(\textit{diff_lang_home}))$
Comments	To check this pattern, it is negated. If the result of verification

	gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Opquast

Security/Authentication

ID	FGS1
Pattern description	Always the logo for security license is present
Category	Functional – General – Security and Authentication
Page Attributes	<i>security_logo</i> : Boolean identifying the presence of security logo
LTL Mapping	UniversalityGlobally (<i>security_logo</i>)
Comments	
Source	Research in progress - Opquast

ID	FGS2
Pattern description	Always the link for security information is present
Category	Functional – General – Security and Authentication
Page Attributes	<i>link_securityinfo</i> : Boolean identifying the presence of the link to security information
LTL Mapping	UniversalityGlobally (<i>link_securityinfo</i>)
Comments	
Source	Opquast

ID	FGS3
Pattern description	Incorrect login info is allowed only 3 times, and then login is forbidden
Category	Functional – General – Security and Authentication
Page Attributes	<i>login</i> : Boolean identifying the login page <i>forbidden</i> : Boolean identifying the page which indicate that access is permanently forbidden
LTL Mapping	$F(login) \rightarrow F(login \wedge X(\neg(forbidden) \vee (login \wedge X(\neg(forbidden) \vee (login \wedge X(\neg(forbidden) \vee X(forbidden \wedge G(\neg(login))))))))))$
Comments	
Source	Newly introduced

ID	FGS4
Pattern description	Secure pages are opened in a different window
Category	Functional – General – Security and Authentication
Page Attributes	<i>secure</i> : Boolean indicating of page is a secure one

	<i>main_window</i> : Boolean indicating if page is opened in main window
LTL Mapping	UniversalityGlobally ($secure \rightarrow \neg (main_window)$)
Comments	
Source	Newly introduced

ID	FGS5
Pattern description	Secure pages are not reachable without going through authorization pages
Category	Functional – General – Security and Authentication
Page Attributes	<i>authorization</i> : Boolean identifying the page for authorization to access secure pages <i>secure</i> : Boolean identifying if the page is a secure page
LTL Mapping	ExistenceBetween (<i>authorization</i> , <i>secure</i> , $\neg (secure)$)
Comments	
Source	Literature

ID	FGS6
Pattern description	Banking information is entered only once before submitting form
Category	Functional – General – Security and Authentication
Page Attributes	<i>Banking_info</i> : Boolean identifying the presence of fields for banking information <i>Submit</i> : identification of page where form submit action exists
LTL Mapping	PrecedenceGlobally ($(\neg (banking_info) \ W (banking_info \ W (G \neg (banking_info))))), submit$)
Comments	
Source	Newly introduced

ID	FGS7
Pattern description	User visits Authentication page then Secure page then returns to Authentication and does this exactly twice
Category	Functional – General – Security and Authentication
Page Attributes	<i>authentication</i> : Boolean indicating if the page is the authentication page <i>secure</i> : Boolean indicating if the page is a secure page
LTL Mapping	$authentication \wedge X(secure \wedge X (authentication \wedge X (secure \wedge X (authentication \wedge G (\neg secure)))))) \mathbf{In} (authentication + secure)$
Comments	$(authentication + secure)$ is the scope of this pattern
Source	Newly introduced

Others

ID	FGO1
Pattern description	Search function is always present on all pages
Category	Functional – General – Other
Page Attributes	<i>search_function</i> : Boolean indicating if search function is present in page
LTL Mapping	UniversalityGlobally (<i>search_function</i>)
Comments	
Source	Research in progress -Opquast

ID	FGO2
Pattern description	If WA is larger than 15 pages, then advanced search is always available
Category	Functional – General – Other
Page Attributes	<i>advanced_search</i> : Boolean indicating if advanced search is present in page <i>num_pages</i> : string count of the number of pages of the application
LTL Mapping	$num_pages > 15 \rightarrow$ UniversalityGlobally (<i>advanced_search</i>)
Comments	
Source	Opquast

ID	FGO3
Pattern description	If page is a questionnaire or comments, then feedback page is next
Category	Functional – General – Other
Page Attributes	<i>questionnaire</i> : Boolean identifying questionnaire page <i>comments</i> : Boolean identifying comments page <i>feedback</i> : Boolean identifying feedback page
LTL Mapping	UniversalityGlobally ($((questionnaire \vee comments) \rightarrow X(feedback))$)
Comments	
Source	Opquast

ID	FGO4
Pattern description	In forms, a link to go back and correct fields exists
Category	Functional – General – Other
Page Attributes	<i>form</i> : Boolean indicating the presence of form <i>link_back</i> : Boolean indicating the presence of a link back to the form
LTL Mapping	UniversalityGlobally ($form \rightarrow X(link_back)$)
Comments	

Source	Opquast
ID	FGO5
Pattern description	In forms, a validation / feedback page exists before submit action
Category	Functional – General – Other
Page Attributes	<i>form</i> : Boolean indicating the presence of form <i>feedback</i> : Boolean indicating the feedback page <i>submit</i> : Boolean indicating the submission of form
LTL Mapping	UniversalityGlobally (<i>form</i> → PrecedenceGlobally (<i>feedback</i> , <i>submit</i>))
Comments	
Source	Opquast

ID	FGO6
Pattern description	In forms, a confirmation of submission exists after submit action
Category	Functional – General – Other
Page Attributes	<i>form</i> : Boolean indicating the presence of form <i>confirmation</i> : Boolean indicating the confirmation page <i>submit</i> : Boolean indicating the submission of form
LTL Mapping	UniversalityGlobally (<i>form</i> → ResponseGlobally (<i>confirmation</i> , <i>submit</i>))
Comments	
Source	Opquast

ID	FGO7
Pattern description	In forms, a link to info on how to fill out forms exists
Category	Functional – General – Other
Page Attributes	<i>form</i> : Boolean indicating the presence of form <i>forminfo</i> : Boolean indicating the page of form filling information
LTL Mapping	UniversalityGlobally (<i>form</i> → <i>link_forminfo</i>)
Comments	
Source	Opquast

ID	FGO8
Pattern description	In forms, a link to security and privacy info is always present
Category	Functional – General – Other
Page Attributes	<i>form</i> : Boolean indicating the presence of form <i>link_security</i> : Boolean indicating the link to security information <i>link_privacy</i> : Boolean indicating the link to privacy information
LTL Mapping	UniversalityGlobally (<i>form</i> → <i>link_security</i> ∧ <i>link_privacy</i>)
Comments	
Source	Opquast

ID	FGO9
Pattern description	In forms, a preview page allowing changing info exist before submit action
Category	Functional – General – Other
Page Attributes	<i>form</i> : Boolean indicating the presence of form <i>preview</i> : Boolean indicating the preview page <i>submit</i> : Boolean indicating the submission of form
LTL Mapping	UniversalityGlobally (<i>form</i> → PrecedenceGlobally (<i>preview</i> , <i>submit</i>))
Comments	
Source	Opquast

ID	FGO10
Pattern description	User can proceed navigation after form submission
Category	Functional – General – Other
Page Attributes	<i>Form_submitted</i> : Boolean indicating that form is submitted <i>Num_links</i> : integer counting number of links
LTL Mapping	UniversalityGlobally (<i>form_submitted</i> → X (<i>num_links</i> > 0))
Comments	
Source	Opquast

ID	FGO11
Pattern description	Pop-ups appear only once
Category	Functional – General – Other
Page Attributes	<i>popup</i> : Boolean indicating the presence of a certain popup
LTL Mapping	$\neg (popup) W (popup W (G \neg (popup)))$
Comments	
Source	Opquast

ID	FGO12
Pattern description	Pop-ups are absent from internal pages and home page
Category	Functional – General – Other
Page Attributes	<i>popups</i> : Boolean indicating the presence of popups <i>home</i> : Boolean designating home page <i>internal</i> : Boolean designating internal pages
LTL Mapping	AbsenceGlobally (<i>popups</i>) In (<i>home</i> ∧ <i>internal</i>)
Comments	Scope of pattern is (<i>home</i> ∧ <i>internal</i>)
Source	Opquast

ID	FGO13
Pattern description	Certain contexts (adult, violence, etc.) are absent

Category	Functional – General – Other
Page Attributes	<i>adult_str</i> : string count of a specific prohibited adult related phrase/word <i>violence_str</i> : string count of a specific prohibited violence related phrase/word
LTL Mapping	AbsenceGlobally ((<i>adult_str</i> > 0) and (<i>violence_str</i> > 0))
Comments	
Source	Newly introduced

ID	FGO13
Pattern description	Subscribe to / unsubscribe from newsletters exists
Category	Functional – General – Other
Page Attributes	<i>newsletter_subscription</i> : Boolean designating page for subscription in newsletter
LTL Mapping	Negation: AbsenceGlobally (<i>newsletter_subscription</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	Opquast

E-Commerce

Customer Support

ID	FEC1
Pattern description	Contact info exists and accessible from every page
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>contact</i> : Boolean indicating the contact information page
LTL Mapping	ExistenceAfter (<i>contact</i> , \neg (<i>contact</i>))
Comments	
Source	IBM usability group

ID	FEC2
Pattern description	Assistance for passwords (change, remember, etc) exist
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>pwd_assist</i> : integer identification of the FAQ page
LTL Mapping	Negation: AbsenceGlobally(<i>pwd_assist</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid

Source	IBM usability group
ID	FEC3
Pattern description	There are three modes of product search: search function, list of products, product suggestion.
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>product</i> : Boolean designating the product page <i>search</i> : Boolean indicating the presence of search function <i>productlist</i> : Boolean indicating the presence of product list <i>productsuggestion</i> : Boolean indicating the presence of product suggestion
LTL Mapping	UniversallyGlobally ($product \rightarrow search \wedge productlist \wedge productsuggestion$)
Comments	
Source	IBM usability group

ID	FEC4
Pattern description	FAQ / glossary is always within 3 clicks of every page.
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>faq</i> : integer identification of the FAQ page
LTL Mapping	Negation: AbsenceGlobally($\neg faq \rightarrow X (faq \vee X (faq \vee X (faq)))$)
Comments	To check this pattern, it is negated. The property formulated is “on all paths FAQ/glossary cannot be found within 3 clicks of every page”. If the result of verification gives a counter example, it means the model checker found at least a path containing the home page. Then the original property is valid.
Source	IBM usability group

ID	FEC5
Pattern description	Link to FAQ exists on all shopping pages and “How to order” pages
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>link_faq</i> : Boolean identification of the presence of link to FAQ page <i>shopping</i> : Boolean identification of the shopping pages <i>how_to_order</i> : Boolean identification of the how to order page
LTL Mapping	UniversalityGlobally ($link_faq$) In ($shopping \wedge how_to_order$)
Comments	Scope of pattern is ($shopping \wedge how_to_order$)
Source	IBM usability group

ID	FEC6
Pattern description	Shipping and delivery date info is accessible within 3 clicks from any page
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>shippinginfo</i> : Boolean indicating the page for shipping information
LTL Mapping	UniversalityGlobally(\neg (<i>shippinginfo</i>) \rightarrow X (<i>shippinginfo</i> \vee X (<i>shippinginfo</i> \vee X (<i>shippinginfo</i>))))
Comments	
Source	IBM usability group

ID	FEC7
Pattern description	After submitting an order, there is a link for tracking order or the shipper's site
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>link_track</i> : Boolean indicating the presence of link to track order <i>shipper_site</i> : Boolean indicating the presence of link to the shipper's site <i>submit_order</i> : Boolean indicating submission of order
LTL Mapping	ResponseGlobally ((<i>link_track</i> \vee <i>shipper_site</i>), <i>submit_order</i>)
Comments	
Source	IBM usability group

ID	FEC8
Pattern description	It is possible to change or cancel an order after submission
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>submitorder</i> : Boolean indicating submission of order <i>change_cancel</i> : Boolean indicating page for changing or cancelling order
LTL Mapping	G (<i>submitorder</i>) \rightarrow ExistenceAfter (<i>change_cancel</i> , <i>submitorder</i>)
Comments	
Source	IBM usability group

ID	FEC9
Pattern description	After order submission, info on warranties and service agreements is accessible
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>submitorder</i> : Boolean indicating submission of order <i>warranties</i> : Boolean indicating page of warranties
LTL Mapping	ExistenceAfter (<i>warranties</i> , <i>submitorder</i>)
Comments	

Source	IBM usability group
ID	FEC10
Pattern description	Info on terms and conditions is accessible before order submission or check out
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>submitorder</i> : Boolean indicating submission of order <i>terms_conditions</i> : Boolean indicating page of terms and conditions
LTL Mapping	ExistenceBefore (<i>terms_conditions</i> , <i>submitorder</i>)
Comments	
Source	IBM usability group

ID	FEC11
Pattern description	Link to info on reimbursement / return conditions and procedure is present on order pages
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>Order_pages</i> : Boolean indicating order pages <i>Link_reimbursement</i> : Boolean indicating the presence of link to reimbursement page
LTL Mapping	UniversalityGlobally (<i>link_reimbursement</i>) In (<i>order_pages</i>)
Comments	
Source	IBM usability group

ID	FEC12
Pattern description	Info on security / level / mode is always present
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>security</i> : Boolean indicating presence of security info <i>level</i> : Boolean indicating presence of security level info <i>mode</i> : Boolean indicating presence of security mode info
LTL Mapping	UniversalityGlobally (<i>security</i> \wedge <i>level</i> \wedge <i>mode</i>)
Comments	
Source	IBM usability group

ID	FEC13
Pattern description	Info on conditions of buying / usage is reachable from all pages
Category	Functional – E-commerce – Customer Support
Page Attributes	<i>conditions_buying_usage</i> : Boolean indicating the page for conditions of buying and usage
LTL Mapping	ExistenceAfter (<i>conditions_buying_usage</i> , (<i>conditions_buying_usage</i>))
Comments	

Source	IBM usability group
---------------	---------------------

Trust

ID	FET1
Pattern description	Privacy policy is accessed from every page
Category	Functional – E-commerce – Trust
Page Attributes	<i>privacy</i> : Boolean indicating page of privacy policy
LTL Mapping	ExistenceAfter (<i>privacy</i> , \neg (<i>privacy</i>))
Comments	
Source	IBM usability group

ID	FET2
Pattern description	Security policy is accessed from store front, order list, shopping cart, and order form
Category	Functional – E-commerce – Trust
Page Attributes	<i>front, order_list, shopping_cart, security_policy</i> : integers identifying the corresponding web pages
LTL Mapping	G ($front \vee order_list \vee shopping_cart \rightarrow X (security_policy)$)
Comments	
Source	IBM usability group

ID	FET3
Pattern description	There exist third party validation seal for trust
Category	Functional – E-commerce – Trust
Page Attributes	<i>third_party_seal</i> : Boolean indicating presence of third party seal
LTL Mapping	Negation: AbsenceGlobally (<i>third_party_seal</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a page containing the address. then the original property is valid
Source	IBM usability group

ID	FET4
Pattern description	Mission statement, history, etc. exist
Category	Functional – E-commerce – Trust
Page Attributes	<i>mission_hist</i> : Boolean indicating page of mission statement and history information
LTL Mapping	Negation: AbsenceGlobally (<i>mission_hist</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at

	least a page containing the address. then the original property is valid
Source	IBM usability group

Product Info and Navigation

ID	FEPN1
Pattern description	Product info is reachable within 3 clicks
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>product_info</i> : Boolean indicating page of product information
LTL Mapping	UniversalityGlobally ($\neg (product_info) \rightarrow X (product_info \vee X (product_info \vee X (product_info)))$)
Comments	
Source	IBM usability group

ID	FEPN2
Pattern description	From home page the following is provided: search products, browse categories, and recommend products
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>home</i> : Boolean indicating home page <i>search_product</i> : Boolean indicating presence of search function <i>browse_categories</i> : Boolean indicating presence of browsing function <i>recommend_products</i> : Boolean indicating presence of product recommendation
LTL Mapping	UniversalityGlobally ($home \rightarrow search_product \wedge browse_categories \wedge recommend_products$)
Comments	
Source	IBM usability group

ID	FEPN3
Pattern description	Shopping pages are accessed from every other page
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>shopping</i> : Boolean indicating shopping pages
LTL Mapping	ExistenceAfter (<i>shopping</i> , $\neg (shopping)$)
Comments	
Source	IBM usability group

ID	FEPN4
Pattern description	“Best selling” pages are accessed directly from home page or product category page

Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>home, category, best_selling</i> : integers identifying the corresponding web pages
LTL Mapping	UniversalityGlobally ($home \vee category \rightarrow X (best_selling)$)
Comments	
Source	IBM usability group

ID	FEPN5
Pattern description	Link to a product description and adding product is accessed from shopping cart page
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>Shopping_cart</i> : Boolean designating the shopping cart page <i>Link_productdesc</i> : Boolean indicating presence for link to product description <i>Link_add_product</i> : Boolean indicating presence for link to product addition
LTL Mapping	UniversalityGlobally ($shopping_cart \rightarrow link_productdesc \wedge link_add_product$)
Comments	
Source	IBM usability group

ID	FEPN6
Pattern description	Within the same product category, product description pages are linked with forward and backward buttons/links
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>Product_category</i> : Boolean designating the product category pages <i>forward</i> : Boolean indicating presence of forward button/link <i>backward</i> : Boolean indicating presence of forward button/link
LTL Mapping	$((forward \wedge backward) \vee (forward \wedge \neg backward) \vee (\neg forward \wedge backward)) \text{ In } (product_category)$
Comments	Scope of pattern is <i>product_category</i>
Source	IBM usability group

ID	FEPN7
Pattern description	A link to catalogue exists on every page
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>link_catalogue</i> : Boolean indicating presence of link to catalogue
LTL Mapping	UniversalityGlobally ($link_catalogue$)
Comments	
Source	IBM usability group

ID	FEPN8
Pattern description	Promotions exist only on home page and category navigation pages
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>home</i> : Boolean designating home page <i>category</i> : Boolean indicating category page <i>promotions</i> : Boolean indicating presence of promotions
LTL Mapping	$\neg (\textit{promotions}) \textbf{In} \neg (\textit{home} \vee \textit{category})$
Comments	Scope is $(\textit{home} \vee \textit{category})$
Source	IBM usability group

ID	FEPN9
Pattern description	1 or 2 promotions exist simultaneously with a link to more promotions
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>num_promotions</i> : integer counting number of promotions <i>link_morepromotions</i> : Boolean indicating presence of link to more promotions page
LTL Mapping	ExistenceGlobally $((\textit{num_promotions} \leq 2) \wedge \textit{num_promotions} > 0) \wedge \textit{link_morepromotions}$
Comments	
Source	IBM usability group

ID	FEPN10
Pattern description	At most 2 promotions exist only on home page and category navigation pages
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>home</i> : Boolean designating home page <i>category</i> : Boolean indicating category page <i>num_promotions</i> : integer counting number of promotions in page
LTL Mapping	$((\textit{num_promotions} \leq 2) \textbf{In} (\textit{home} \vee \textit{category})) \wedge ((\textit{num_promotions} = 0) \textbf{In} \neg (\textit{home} \vee \textit{category}))$
Comments	Scope is $(\textit{home} \vee \textit{category})$
Source	IBM usability group

ID	FEPN11
Pattern description	Promotions are absent on product comparison pages and ordering pages
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>product_comparison</i> : Boolean designating product comparison pages <i>ordering</i> : Boolean indicating ordering pages

	<i>num_promotions</i> : integer counting number of promotions in page
LTL Mapping	$G (num_promotions = 0) \mathbf{In} (product_comparison \wedge ordering)$
Comments	Scope is $(product_comparison \wedge ordering)$
Source	IBM usability group

ID	FEPN12
Pattern description	Cross-selling and up-selling pages are reachable only from product detail pages
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>home, category, best_selling</i> : integers identifying the corresponding web pages
LTL Mapping	$ExistenceAfter (cross_selling, product_detail) \wedge ExistenceAfter (up_selling, product_detail) \wedge AbsenceAfter (cross_selling, \neg product_detail) \wedge AbsenceAfter (up_selling, \neg product_detail)$
Comments	
Source	IBM usability group

ID	FEPN13
Pattern description	Types of accepted payments are indicated and present on all product pages
Category	Functional – E-commerce – Product Info and Navigation
Page Attributes	<i>Type_payment</i> : Boolean indicating presence of types of payments <i>product</i> : Boolean designating product pages
LTL Mapping	$(type_payment) \mathbf{In} (product)$
Comments	Scope is <i>product</i>
Source	IBM usability group

Purchase Transaction

ID	FEPT1
Pattern description	It is possible to view a product description, total cost, change a quantity, add/remove a product, save an order, and proceed to order form
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>description, cost, change_quantity, addrem_product, save</i> : Booleans indicating presence of corresponding functions <i>order_form</i> : Boolean indicating order form page
LTL Mapping	$PrecedenceGlobally ((description \wedge cost \wedge change_quantity \wedge addrem_product \wedge save), order_form)$
Comments	

Source	IBM usability group
ID	FEPT2
Pattern description	After an order is submitted, a confirmation page is shown to the user
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>confirmation</i> : Boolean indicating the confirmation page <i>order</i> : Boolean indicating the order page
LTL Mapping	ResponseGlobally (<i>confirmation</i> , <i>order</i>)
Comments	
Source	IBM usability group

ID	FEPT3
Pattern description	From confirmation page, it is possible to reach registration page (acquire ID, and password)
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>registration</i> : Boolean indicating the registration page <i>confirmation</i> : Boolean indicating the confirmation page
LTL Mapping	Negation: AbsenceAfter (<i>registration</i> , <i>confirmation</i>)
Comments	To check this pattern, it is negated. If the result of verification gives a counter example, it means the model checker found at least a path where registration is reached from confirmation page.. then the original property is valid
Source	IBM usability group

ID	FEPT4
Pattern description	If the user registers, then he can use shortcut order (by entering only ID, password, and credit card#)
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>user_login</i> : Boolean indicating the login page <i>shortcut_order</i> : Boolean indicating the shortcut order page
LTL Mapping	G (<i>user_login</i> → XF (<i>shortcut_order</i>))
Comments	
Source	IBM usability group

ID	FEPT5
Pattern description	There is a link from the order form to the order list
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>link_orderlist</i> : Boolean indicating the presence of link to order list <i>order_form</i> : Boolean indicating the order form page
LTL Mapping	(<i>link_orderlist</i>) In (<i>order_form</i>)

Comments	Scope of pattern is <i>order_form</i>
Source	IBM usability group

ID	FEPT6
Pattern description	Credit card info is submitted once only before submitting an order
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>Credit_card</i> : Boolean indicating the presence of fields requesting credit card info in pages <i>submit</i> : Boolean indicating the submit order page
LTL Mapping	PrecedenceGlobally ($(\neg (credit_card) W (credit_card W (G \neg (credit_card))))$), <i>submit</i>)
Comments	
Source	Newly introduced

ID	FEPT7
Pattern description	After ordering, conditions of guarantee are available for the user
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>guarantee</i> : Boolean indicating the page of conditions of guarantee <i>order</i> : Boolean indicating the order page
LTL Mapping	ResponseGlobally (<i>guarantee, order</i>)
Comments	
Source	IBM usability group

ID	FEPT8
Pattern description	After ordering, address of return is indicated
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>return_address</i> : Boolean indicating the presence of return address <i>order</i> : Boolean indicating the order page
LTL Mapping	ResponseGlobally (<i>return_address, order</i>)
Comments	
Source	IBM usability group

ID	FEPT9
Pattern description	After ordering, return service (charge, hours of work, contact, etc.) are indicated
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>return_service</i> : Boolean indicating the presence of return service <i>order</i> : Boolean indicating the order page
LTL Mapping	ResponseGlobally (<i>return_service, order</i>)

Comments	
Source	IBM usability group

ID	FEPT10
Pattern description	Availability of products is indicated before validation of order
Category	Functional – E-commerce – Purchase Transaction
Page Attributes	<i>product_availability</i> : Boolean indicating page of product availability <i>validation_order</i> : Boolean indicating page of order validation
LTL Mapping	PrecedenceGlobally (<i>product_availability</i> , <i>validation_order</i>)
Comments	
Source	IBM usability group

