

## ***Exploiting HCI Pattern Collections for User Interface Generation***

How well do Existing Pattern Definitions support Automated UI Construction?

Juergen Engel, Christian Herdin, Christian Maertin

Faculty of Computer Science  
Augsburg University of Applied Sciences  
An der Hochschule 1  
86161 Augsburg, Germany

[Juergen.Engel | Christian.Herdin | Christian.Maertin]@hs-augsburg.de

**Abstract**—In the field of human-computer interaction various pattern languages and pattern collections have evolved during the last years. They include patterns for domain-independent and domain-specific graphical user interface design, interaction design, interactive application structure and navigation, design for usability, and design for user experience. However, the description and specification structure and attributes vary between existing pattern languages. In order to unify the various approaches and make them exploitable for automated graphical user interface construction, we have mapped the information structure of the different pattern collections to the pattern language markup language. By adding some additional attributes, the resulting pattern specifications can be used for facilitating and automating the development process for real world interactive applications, e.g., for knowledge sharing systems.

**Keywords** – HCI patterns; pattern language; standardized pattern definition; pattern attributes; pattern relations; user interface generation.

### I. INTRODUCTION

Human-computer interaction (HCI) patterns, pattern languages, and pattern collections recently have gained much influence in the field of model-based user interface development approaches (MUID) [2]. HCI pattern languages and their various types of user interfaces related patterns provide abstract, semi-abstract and/or concrete modeling information for the (semi-)automatic construction of domain-specific, highly-usable, adaptable, and accessible user interfaces, e.g. for web-based and mobile applications. In order to simplify the selection of useful pattern types for a specific development project, ways for hierarchically structuring such languages and grouping their patterns by specific inter-pattern-relationships have been proposed [14]. Also, tools for visually defining patterns and their attributes, for browsing pattern hierarchies and exploiting the selected patterns for automated user interface generation purposes have been introduced [9].

Most available and popular pattern collections, however, as useful for the developer as they may be, still consist of rather unrelated pattern sets. Neither do such collections provide hierarchical or content-based relationships between similar or aggregated patterns, nor do pattern definitions

provide attributes for using the structural, contextual and content information provided by a pattern for exploiting the pattern description for automated user interface generation.

Nevertheless, the pattern collections discussed in the following sections provide valuable information for structuring and designing the user interfaces and the navigation paths between the interaction objects of various types of interactive applications. Patterns from different collections should therefore be easily combinable within the target model of an interactive application. The patterns should be accessible from tool environments, browsers, and generators. In order to reach this rather practical goal, standardization efforts for reaching a common HCI pattern definition approach are necessary.

In the following section, several major standard HCI pattern collections are introduced and compared. Section III discusses the PLML approach for standardizing pattern definitions used in HCI and discusses several extensions to PLML. Section IV examines whether the PLML pattern description approach can be used for specifying real world interactive applications, e.g., for knowledge sharing systems. Finally, Section V presents concepts and specification extensions for HCI pattern definitions that can be used for automated user interface construction and generation.

### II. ANALYSIS OF PREVALENT PATTERN COLLECTIONS

During the last decades, various HCI pattern collections have been developed, introduced and published providing valuable and reusable design know-how. Among others there are Jenifer Tidwell's *Designing Interfaces, Patterns for Effective Interaction Design* [15], Martijn van Welie's *Patterns in Interaction Design* [17], Douglas van Duyne's *The Design of Sites, patterns for Creating Winning Web Sites* [6], Todd Coram's and Jim Lee's *A Pattern Language for User Interface Design* [3], *Yahoo! Design Pattern Library of Yahoo! Inc.* [18], or the community-driven UX and UI pattern library *Quince* operated by Infragistics [11].

Besides our own general and domain-specific pattern languages used for facilitating the design of knowledge sharing systems and portable mobile applications [8, 12], we predominantly have utilized patterns from Tidwell [15], van Welie [17], and van Duyne [6] in our previous work. Due to that fact we have analyzed and evaluated these three pattern

collections in detail to assess their suitability to be used in automated and semi-automated user interface generation process steps respectively.

A. Pattern Collection by Jenifer Tidwell

The pattern collection by Jenifer Tidwell is one of the most extensive pattern libraries and consists of 125 patterns organized in 11 categories from “What users do” to “Make it look good”. Compared to the previous version the over-worked pattern collection described in the second edition of her book [15] is extended by three additional categories. Many patterns have been revised and several new patterns have been introduced. A digest is available on the Internet [16].

TABLE I. TIDWELL PATTERN COLLECTION ORGANIZATION

Tidwell pattern collection organization		
Category	#	Patterns
What users do	14	Safe exploration; Instant gratification; Satisficing; Changes in midstream; Deferred choices; Incremental construction; Habituation; Microbreaks; Spatial memory; Prospective memory; Streamlined repetition; Keyboard only; Other people’s advice; Personal recommendations
Organizing the content	10	Feature, search and browse; News stream; Picture manager; Dashboard; Canvas plus palette; Wizard; Settings editor; Alternative views; Many workspaces; Multi-level help
Getting around	13	Clear entry points; Menu page; Pyramid; Modal panel; Deep-linked state; Escape hatch; Fat menus; Sitemap footer; Sign-in tools; Sequence map; Breadcrumbs; Annotated scrollbar; Animated transition
Organizing the page	13	Visual framework; Center stage; Grid of equals; Titled sections; Module tabs; Accordion; Collapsible panels; Movable panels; Right/left alignment; Diagonal balance; Responsive disclosure; Responsive enabling; Liquid layout
Lists of things	12	Two-panel selector; One-window drilldown; List inlay; Thumbnail grid, Carousel; Row striping; Pagination; Jump to item; Alphabet scroller; Cascading lists; Tree table; New-item row
Doing things	11	Button groups; Hover tools; Action panel; Prominent “done” button; Smart menu items; Preview; Progress indicator; Cancelability; Multi-level undo; Command history; Macros
Showing complex data	11	Overview plus detail; Datatips; Data spotlight; Dynamic queries; Data brushing; Local zooming; Sortable table; Radial table; Multi-Y graph; Small multiples; Treemap
Getting input from users	11	Forgiving format; Structured format; Fill-in-the-blanks; Input hints; Input prompt; Password strength meter; Autocompletion; Dropdown chooser; List builder; Good defaults; Same-page error messages

Tidwell pattern collection organization		
Category	#	Patterns
Using social media	12	Editorial mix; Personal voices; Repost and comment; Conversation starters; Inverted nano-pyramid; Timing strategy; Specialized streams; Social links; Sharing widget; News box; Content leaderboard; Recent chatter
Going mobile	11	Vertical stack; Filmstrip; Touch tools; Bottom navigation; Thumbnail-and-text list; Infinite list; Generous borders; Text clear button; Loading indicators; Richly connected apps; Streamlined branding
Make it look good	7	Deep background; Few hues, many values; Corner treatments; Borders that echo fonts; Hairlines; Contrasting font weights; Skins and themes

In the following, we consider the fully-fledged suite of patterns [15]. In Table I, an overview of the diverse pattern categories is given. In addition, it is shown how many and which patterns are assigned to the particular groups.

The actual pattern definitions consist of the eight attributes “Name”, “Figure”, “What”, “Use when”, “Why”, “How”, “Examples”, and “In other libraries”. Brief descriptions of these attributes are provided in Table II.

TABLE II. TIDWELL PATTERN DEFINITIONS

Tidwell pattern definitions	
Attribute	Brief description
Name	Name of the pattern
Figure	Meaningful example of pattern application, i.e. a screen shot
What	Brief description of the pattern
Use when	Information about the problem to be solved and the context of application
Why	Information about why one should use this particular pattern
How	How to use the particular pattern
Examples	Instances of application of the particular pattern
In other libraries	Links to other pattern collections where the selfsame pattern can be found

The Tidwell pattern collection draws a bow over a huge variety of possible fields of applications. The patterns are well-illustrated and described in detail. For all patterns, several meaningful examples of pattern instantiations are included. Descriptive category names support the user when browsing the catalog and help to locate individual patterns. Information about relations between the patterns are occasionally included in the “How” and “Why” attributes. References to the same or very similar patterns in other collections are provided via the “In other libraries” attribute.

B. Pattern Collection by Martijn van Welie

The pattern library of Martijn van Welie is available in the Internet [17]. It comes with 131 patterns grouped in the

three main categories which are again divided into a total of 15 sub categories.

Table III shows the organization of the main category “User needs”, while Table IV illustrates “Application needs” and Table V “Context of design”.

TABLE III. MAIN CATEGORY “USER NEEDS”

Organization of pattern category “User needs”		
Category	#	Patterns
Navigating around	25	Accordion; Headerless menu; Breadcrumbs; Directory navigation; Doormat navigation; Double tab navigation; Faceted navigation; Fly-out menu; Home link; Icon menu; Main navigation; Map navigator; Meta navigation; Minesweeping; Panning navigator; Overlay menu; Repeated menu; Retractable menu; Scrolling menu; Shortcut box; Split navigation; Teaser menu; To-the-top link; Trail menu; Navigation tree
Basic interactions	7	Action button; Guided tour, Paging; Pulldown button; Slideshow; Stepping; Wizard
Searching	13	Advanced search; Autocomplete; Frequently asked questions (FAQ); Help wizard; Search box; Search area; Search results; Search tips; Site index; Site map; Footer sitemap; Tag cloud; Topic pages
Dealing with data	14	Carrousel; Table filter; Collapsible panels; Detail on demand; Collector; Inplace replacement; List builder; List entry view; Overview by detail; Parts selector; Tabs; Table sorter; Thumbnail; View
Personalizing	3	Customizable window; Login; Registration
Shopping	9	Booking; Product comparison; Product advisor; Product configurator; Purchase process; Shopping cart; Store locator; Testimonials; Virtual product display
Making choices	5	Country selector; Date selector; Language selector; Poll; Rating
Giving input	3	Comment box; Constraint input; Form
Miscellaneous	5	Footer bar; Hotlist; News box; News ticker; Send-a-friend link

TABLE IV. MAIN CATEGORY “APPLICATION NEEDS”

Organization of pattern category “Application needs”		
Category	#	Patterns
Drawing attention	8	Captcha; Center stage; Color coded section; Premium content lock; Grid-based layout; Liquid layout; Outgoing links; Alternating row colors
Feedback	2	Input error message; Processing page
Simplifying interaction	2	Enlarged clickarea; Font enlarger

TABLE V. MAIN CATEGORY “CONTEXT OF DESIGN”

Organization of pattern category “Context of design”		
Category	#	Patterns
Site types	14	Web-based application; Artist site; Automotive site; Branded promotion site; Campaign site; E-commerce site; Community site; Corporate site; Multinational site; Museum site; Personalized ‘my’ site; News site; Portal site; Travel site
Experiences	8	Community building; Information management; Fun; Information seeking; Learning; Assistance; Shopping; Story telling
Page types	13	Article page; Blog page; Case study; Contact page; Event calendar; Forum; Guest book; Help page; Homepage; Newsletter; Printer-friendly page; Product page; Tutorial

The pattern definitions contain the nine attributes “Name”, “Problem”, “Solution”, “Use when”, “How”, “Why”, “More examples”, “Literature”, and “Comments”. Brief descriptions of these attributes are provided in Table VI.

TABLE VI. VAN WELIE PATTERN DEFINITIONS

van Welie pattern definitions	
Attribute	Brief description
Name	Name of the pattern
Problem	Description of the problem to be solved
Solution	Description of the intended solution of the problem
Use when	Information of the context of application
How	Information about how to apply the pattern
Why	Information about why one should use this particular pattern
More examples	Instances of application of the particular pattern
Literature	Links to background literature
Comments	Comments from users of the pattern

In contrast to the renewed Tidwell collection the patterns, in the van Welie pattern library have not been revised during the last four years. According to an entry on the Internet page [17], the last change was performed in September 2007.

The two-tiered categorization supports the user in terms of searching and finding appropriate patterns. Collection-internal relationships between patterns are occasionally included in the attributes “Use when”, “How”, and “Why”. References to other pattern catalogues do not exist. A potentially valuable resource of information is the attribute “Comments”, which contains annotations that pattern users have left on van Welie’s website.

C. Pattern Collection of Douglas K. van Duyne

The pattern collection of Douglas van Duyne consists of 107 web-related patterns separated in 13 categories from “Site genres” to “The mobile web” [6]. All patterns are also published in the internet [7] but in a shortened and therefore less exhaustive manner.

Again, we consider the fully-fledged patterns library as published in [6]. Table VII provides an overview of how many and which patterns are available and how they are classified.

The pattern definitions consist of the seven attributes “Pattern ID”, “Name”, “Figure”, “Background”, “Problem”, “Solution”, and “Other patterns to consider”. Brief explanations are provided in Table VIII.

The target audience of the pattern collection is the web designer community. This is already indicated by the category names. The actual pattern descriptions are extensive and detailed. Very positive is the treatment of collection-internal dependencies and relation between patterns which are explicitly listed and accentuated in terms of font color. References to external pattern collection are not included.

TABLE VII. VAN DUYNÉ PATTERN COLLECTION ORGANIZATION

van Duyne pattern collection organization		
Category	#	Patterns
Site genres	12	Personal e-commerce; News mosaics; Community conference; Self-service government; Nonprofits as networks of help; Grassroots information sites; Valuable company sites; Educational forums; Stimulating arts and entertainment; Web apps that work; Enabling intranets; Blogs
Creating a navigation framework	9	Multiple ways to navigate; Browsable content; Hierarchical organization; Task-based organization; Alphabetical organization; Chronological organization; Popularity-based organization; Category pages; Site accessibility
Creating a powerful homepage	2	Homepage portal; Up-front value proposition
Writing and managing content	11	Page templates; Content modules; Headlines and blurbs; Personalized content; Message boards; Writing for search engines; Inverted-pyramid writing style; Printable pages; Distinctive HTML titles; Internationalized and localized content; Style sheets
Building trust and credibility	9	Site branding; E-mail subscriptions; Fair information practices; Privacy policy; About us; Secure connections; E-mail notifications; Privacy preferences; Preventing phishing scams
Basic e-commerce	9	Quick-flow checkout; Clean product details; Shopping cart; Quick address selection; Quick shipping method selection; Payment method; Order summary; Order confirmation and thank-you; Easy returns
Advanced e-commerce	7	Featured products; Cross-selling and up-selling; Personalized recommendations; Recommendation community; Multiple destinations; Gift giving; Order tracking

van Duyne pattern collection organization		
Category	#	Patterns
		and history
Helping customers to complete tasks	13	Process funnel; Sign-in/new account; Guest account; Account management; Persistent customer sessions; Floating windows; Frequently asked questions; Context-sensitive help; Direct manipulation; Clear forms; Predictive input; Drill-down options; Progress bar
Designing effective page layouts	6	Grid layout; Above the fold; Clear first reads; Expanding screen width; Fixed screen width; Consistent sidebars of related content
Making site search fast and relevant	3	Search action module; Straightforward search forms; Organized search results
Making navigation easy	17	Unified browsing hierarchy; Navigation bar; Tab rows; Action buttons; High-visibility action buttons; Location bread crumbs; Embedded links; External links; Descriptive, longer link names; Obvious links; Familiar language; Preventing errors; Meaningful error messages; Page not found; Permalinks; Jump menus; Site map
Speeding up your site	6	Low number of files; Fast-loading images; Separate tables; HTML power; Reusable images; Fast-loading content
The mobile web	3	Mobile screen sizing; Mobile input controls; Location-based services

TABLE VIII. VAN DUYNÉ PATTERN DEFINITIONS

van Duyne pattern definitions	
Attribute	Brief description
Pattern ID	Unique identifier of the pattern
Name	Name of the pattern
Figure	Meaningful example of pattern application, i.e. a screen shot
Background	Information of the context of application
Problem	Description of the problem to be solved
Solution	Description of the intended solution of the problem
Other patterns to consider	List of patterns which correlate with the particular pattern.

III. PLML: A STANDARDIZATION APPROACH FOR HCI PATTERN-DEFINITION

Like most of the available pattern collections, the three libraries described above lack a common organizational structure in order to facilitate appropriate and convenient pattern retrieval and selection. Manageability aspects of various existing UI pattern catalogues are discussed and compared in [3].

Another significant obstacle is that the authors describe their patterns in different and inconsistent styles. This makes it hard or even impossible to search, select and reference

patterns across pattern collections. Hence, in a workshop [10] held within the scope of the CHI 2003 conference the attendees aimed for unification of pattern descriptions and guidance for the authors. As a result, the Pattern Language Markup Language (PLML) has been constituted and documented [10].

Version PLML v1.1 specifies that the documentation of a certain pattern should consist of the following elements: a pattern identifier, name, alias, illustration, descriptions of the respective problem, context and solution, forces, synopsis, diagram, evidence, confidence, literature, implementation, related patterns, pattern links, and management information [10]. Brief descriptions of these elements are provided in Table IX.

A. Extensions and changes to PLML

Over the years, PLML continued to develop and new extensions were added in order to adapt PLML to developing trends and needs of authors and users.

TABLE IX. PLML ATTRIBUTES

PLML pattern description elements	
Element	Brief description
Pattern ID	Unique pattern identifier
Name	Name of the pattern
Alias	Alternative names; also known as
Illustration	Good example of instantiation of the pattern
Problem	Description of the problem to be solved
Context	Situations in which the pattern can be applied
Forces	Description of forces in the environment that the use of the pattern will resolve
Solution	Description of how to resolve the problem
Synopsis	Summary of the pattern description
Diagram	Schematical visualization of the pattern
Evidence Example Rationale	Verification that it is in fact a pattern by at least three known uses of the pattern discussion and any principled reasons
Confidence	Rating of how likely the pattern provides an invariant solution for the given problem
Literature	References to related documents or papers
Implementation	Code fragments or details of technical realization
Related-patterns	Relationship to other patterns
Pattern-Link Type Pattern ID Collection ID Label	Catenation of patterns kind of relationship identifier of related pattern identifier of pattern collection name of the pattern link
Management Author Credits Creation-date Last-modified Revision-number	Authorship and change management name of the pattern author Merits date of pattern compilation date of last change version of the pattern definition

B. PLML version 1.2

An extension to PLML was suggested by Deng et al. [4]. New in this version 1.2 is an improved pattern- and attributes-description. This change allows for the attributes <chance-log> and <force> to be used by the pattern management tool MUIP.

C. PLMLx

The first published extension to PLML is PLMLx, the Extended Pattern Language Markup Language by Bienhaus [1]. The extension allows for better support of search functionality and classification of the patterns. The additional attributes that were added or modified are shown in Table X.

TABLE X. ADDITIONAL OR MODIFIED PATTERN DESCRIPTION ELEMENTS FROM THE PLMLx

Additional or modified PLMLx pattern description elements	
Element	Brief description
<organisation>	Meta information about the category, classification and the collection. A pattern can be part of different collections
<resulting context>	Describes the context in which we find ourselves when we applied the pattern. By applying a pattern it can create new problems that must be solved again.
<acknowledgments>	Acknowledgment for each who contributed significantly to the development of the pattern or the techniques description. If the pattern has been through a "shepherding process" or "writer's workshop", significant contributors are candidates for the acknowledgment.
<management>	Added two new Subtasks copyright and license. The <last modified>-Tag from the original PLML<management>-Tag was deleted. The data from the last chance and the pertinent pattern name were integrated in the <revisions-number>-Tag
<example> and <rationale>	<example> and <rationale> are now separate tags. In the original PLML 1.1.2 they are Subtasks of the tag <evidence>.

The extensions and modifications in PLMLx make sense, because the <organization>-Tag helps to give a better overview of the different patterns from separate collections. However, an attribute that would be helpful for describing relationships and hierarchy between patterns or pattern catalogs is not included in PLMLx.

D. XPLML

Another extension to PLML is XPLML that is especially appropriate for human-computer interaction (HCI) design and for networking. XPLML describes the patterns and their relationship to each other with the help of Open Source standards such as XML, RDF and OWL [13].

The XPLML-Framework is a summary of specifications and tools to formalize HCI Patterns. The framework is intended to close the gap between the textual specification of patterns and their application in software. The conversion of

an existing design pattern to XPLML does not work automatically.

The implementation of the XPLML framework is based on the following seven modules [13]:

- Unified HCI Pattern Form
- Semantic Metadata
- Semantic Relations among Patterns
- Atomic Particles of HCI Design Patterns
- Requirements engineering in HCI community
- Survey of HCI Design Pattern Management Tools
- Specification Documentation

XPLML is based on PLML, but adds components (e.g., Metadata) not in the HCI pattern form. The new components were added outside in a separate form. This is a good approach to define the relationships between patterns and collections.

The problem is that XPLML uses the older and not enhanced PLML backbone for pattern description. In addition, many pattern collections are not available in an appropriate specification and need to be translated separately for the use in XPLML. An automatic translation is not possible.

Furthermore, there neither exists a detailed description of the individual components of the framework, nor a release version until today.

#### IV. COMPARISON OF PATTERN DEFINITIONS

For all three discussed pattern collections the particular pattern definition attributes can be mapped almost entirely to corresponding PLML 1.1 elements. In some cases pattern attributes show 1:1 relationships to particular PLML elements, while not rarely exist equivocal correlations, i.e. 1:n, n:1, or n:m. We identified just one single pattern attribute within van Welie’s pattern library that has no appropriate counterpart in PLML, i.e., the “Comments” attribute. Contrariwise four PLML elements are totally unconsidered, i.e., <alias>, <forces>, <confidence>, and <management>. Content that maps to the <implementation> PLML element is solely provided by van Duyne patterns.

Within the Tidwell pattern collection the “Use when” attribute provides information that partly maps to both, the <problem> and the <context> PLML elements. On the other hand the “What” and “How” attributes provide content that fits into <solution>. Data from the “In other libraries” attribute relates to the PLML element <literature>. However, not all Tidwell patterns feature this type of information. The <pattern-link> PLML element is completely disregarded.

Van Welie’s patterns keep information within the “Solution” and “How” attributes that maps to the <solution> PLML element. Moreover the “How” attribute occasionally delivers figures or sketches that refer to <diagram> in PLML. Likewise the “Literature” attribute is not specified within every pattern definition. Information that maps to the <related-patterns> PLML element is provided by the “Use when”, “How”, and “Why” attributes albeit <pattern-link> data is missing.

Patterns within the van Duyne collection possess a <pattern ID> type property which consists of a capital letter indicating the particular pattern category and a consecutive number. The PLML <context> is split over the “Background” and “Problem” attributes. Notably, “Problem” is a very rich attribute whose content maps to various PLML elements, i.e., <problem>, <context>, <evidence>, <implementation>, and <related-patterns>. PLML <pattern-link> data is merely rudimentary available. Notably, the <type> specification of the pattern links are provided not at all.

The complete comparison of the pattern attributes of the discussed pattern collections and their mappings to PLML elements is consolidated in Table XI.

TABLE XI. PATTERN DEFINITION COMPARISON

Comparison of pattern attributes			
PLML 1.1	Tidwell	van Welie	van Duyne
<pattern ID>	-	-	pattern category + consecutive number
<name>	“Name”	“Name”	“Name”
<alias>	-	-	-
<illustration>	“Figure”	included in “Solution”	“Figure”
<problem>	included in “Use when”	“Problem”	“Problem”
<context>	included in “Use when”	“Use when”	partly in “Background” and partly in “Problem”
<forces>	-	-	-
<solution>	“What” and “How”	“Solution” and “How”	“Solution”
<synopsis>	-	-	-
<diagram>	-	occasionally in “How”	included in “Solution”
<evidence>	yes	yes	to some extent
<example>	“Examples”	“More examples”	included in “Problem”
<rationale>	“Why”	“Why”	-
<confidence>	-	-	-
<literature>	occasionally included “In other libraries”	occasionally available in “Literature”	-
<implementation>	-	-	occasionally included in “Problem”
<related-patterns>	occasionally included in “Why” and “How”	occasionally included in “Use when”, “How”, and “Why”	“Other patterns to consider”, and contained in “Problem” and partly in “Background”
<pattern-link> <type> <pattern ID> <collection ID>	-	-	rudimentary “Pattern ID”

Comparison of pattern attributes			
PLML 1.1	Tidwell	van Welie	van Duyne
<label>			
<management>	-	-	-
-	-	“Comments”	-

V. DOES PLML MEET OUR DEMANDS?

In this section, we would like to approach this question on the basis of an application in the area of knowledge sharing systems, which has been developed in the context of the project p.i.t.c.h. (pattern-based interactive tools for improved communication habits in knowledge transfers) [12]. It was conducted by the Automation in Usability Engineering group (AUE) at Augsburg University of Applied Sciences in cooperation with two medium-sized enterprises with engineering and production background and several partners from communication sciences and the knowledge management domain.

As an introductory example, Figure 1 shows a screenshot of the search function of the p.i.t.c.h. application. The red rectangles indicate the essential patterns which have been used to design the search capabilities.

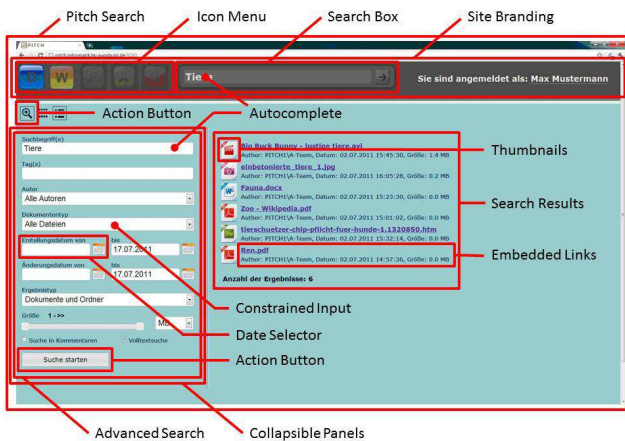


Figure 1. Screenshot of the p.i.t.c.h. Search Functionality.

Figure 2 gives an overview of the intrinsic pattern hierarchy of the p.i.t.c.h. search dialogue. Root element is the “p.i.t.c.h. Search” pattern which contains three further patterns, i.e., “Site Branding” [6], “Action Button” [17], and “Search Results” [17]. “Site Branding” in turn contains the two patterns “Icon Menu” [17] and “Search Box” [17], which finally relates to the “Autocomplete” pattern [17]. The other branches of the tree are constructed in analogous manner. The capital letters in the lower right corners of the boxes indicate from which pattern collection the particular pattern has been retrieved. The arrows represent the respective relationships between the patterns.

Such relationships are domain-independent and can be exploited for guiding structural and layout-related decisions

during the automated pattern-based construction of the interactive application.

As yet, we have identified different relation types including “contains”, “invokes”, “employs”, and “is-similar-to” relationships. Certainly there are respective inverse relations, i.e., “is-contained-by”, “is-invoked-by”, and “is-employed-by”.

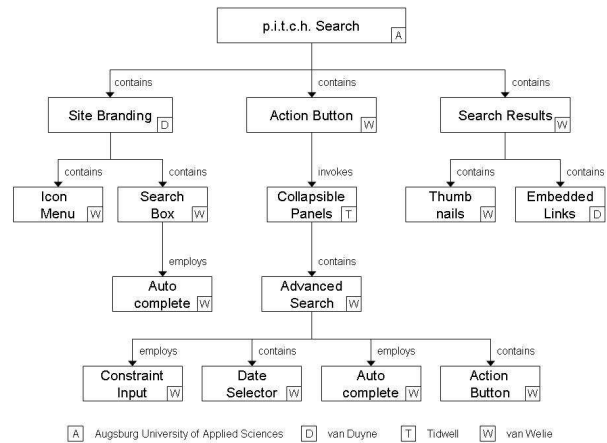


Figure 2. Excerpt of the Pattern Hierarchy of the p.i.t.c.h. Search Functionality.

A “contains” relationship implies that one pattern is integral part of another while “invoke”, respectively “is-invoked-by” means that a pattern is activated owing to an event that happened within the correlated pattern. The “employs” relation means that the parent pattern avails itself of a functionality provided by the child pattern. Finally, “is-similar-to” predicates that the related patterns are either identical or can be utilized alternatively.

As outlined in Section I, we intend to use PLML-style pattern definitions that allow for (semi-)automated processing and UI construction purposes. Besides the fact that the “implementation” element is not yet adequately structured and specified, PLML shows various shortcomings concerning the given options to model the previously described pattern relationships.

Automated pattern processing requires for distinguishing two fundamental types of pattern links. On one hand, there are permanent links to other patterns, which can be regarded as “hard-coded” and generally, will not change for a long period of time, e.g., a “is-similar-to” relationship to the identical pattern within a different pattern collection. If a permanent link is to be changed it would lead to a new revision of the pattern. As soon as a respective parent pattern is applied, all child patterns referenced by permanent links are also applied automatically.

On the other hand, there is a need for defining temporary pattern relationships when the pattern is applied respectively the pattern hierarchy is constructed. For instance, the “Autocomplete” pattern occurs twice within Figure 2. In both cases, it is a matter of the selfsame pattern, but one time it has an “is-employed-by” relationship with the “Search

Box” pattern while the other time it indeed correlates in the same manner, but to a different pattern, i.e., “Advanced Search”. In order to meet these requirements, we defined two child elements belonging to the PLML <related-patterns>, i.e., <permanent-links> and <temporary-links>.

In addition, PLML incorporates information about the pattern revision number within the <management> element. It is a flaw that the data is not used in the <pattern-link> element which allows for referencing other patterns, but not certain revisions of that particular pattern. This can be resolved by simply adding a child element to <pattern-link>, i.e., <revision>. Since an individual pattern may, of course, possess multiple relationships to other patterns, we propose to specify another child element <linkID>.

The described changes to PLML are illustrated in Figure 3. Please, note that the given <permanent-link> is not included within the graph in Figure 2.

```

...
<name>Search Box</name>
<problem>
The users need to find an item or specific
information.
</problem>
<solution>
Offer search functionality consisting of a search
label, a keyword field, a filter if applicable and
a "go" button. Pressing the return key has the
same function as selecting the "go" button.
</solution>
<related patterns>
  <permanent-links>
    <pattern-link>
      <linkID>W00037PL001</linkID>
      <revision>1.0</revision>
      <type>is similar to</type>
      <patternID>D00079</patternID>
      <collectionID>van Duyn</collectionID>
      <label>search action module</label>
    </pattern-link>
  </permanent-links>
  <temporary-links>
    <pattern-link>
      <linkID>W037TL001</linkID>
      <revision>1.3</revision>
      <type>employs</type>
      <patternID>W00034</patternID>
      <collectionID>van Welie</collectionID>
      <label>Autocomplete</label>
    </pattern-link>
  </temporary-links>
</related patterns>
...

```

Figure 3. Fragment of PLML-like Definition of Search Box Pattern.

As already described above, a pattern might occur more than one time in a pattern hierarchy. This means that this particular pattern is applied several times or, in other words, multiple instances of the pattern do exist. In this sense, PLML merely specifies the <patternID> element which is not suitable for distinguishing two or more instances of the same pattern. Therefore, we suggest to introduce a new element named <UID> with two child elements, i.e., <patternID> and <instanceID>. In addition, the <pattern-link> element has to be extended, respectively.

Figure 4 provides a pattern definition fragment that illustrates all proposed PLML changes required for better support of automatic pattern processing. This example specifies the “Advanced Search” pattern in the context of the pattern hierarchy displayed in Figure 2.

```

...
<UID>
  <patternID>W00033</patternID>
  <instanceID>0001</instanceID>
</UID>
<name>Advanced Search</name>
<illustration>
Please refer to Figure 1
</illustration>
<problem>
Users need to find a specific item in a large
collection of items.
</problem>
<solution>
Offer a special advanced search function with
extended term matching, scoping and output
options.
</solution>
<related-patterns>
  <permanent-links>
  </permanent-links>
  <temporary-links>
    <pattern-link>
      <linkID>W00033TL001</linkID>
      <type>employs</type>
      <patternID>W00078</patternID>
      <revision>1.0</revision>
      <instanceID>0001</instanceID>
      <collectionID>van Welie</collectionID>
      <label>Constraint Input</label>
    </pattern-link>
    <pattern-link>
      <linkID>W00033TL002</linkID>
      <type>contains</type>
      <patternID>W00073</patternID>
      <revision>1.1</revision>
      >instanceID>0001</instanceID>
      <collectionID>van Welie</collectionID>
      <label>Date Selector</label>
    </pattern-link>
    <pattern-link>
      <linkID>W00033TL003</linkID>
      <type>employs</type>
      <patternID>W00034</patternID>
      <revision>1.0</revision>
      <instanceID>0002</instanceID>
      <collectionID>van Welie</collectionID>
      <label>Autocomplete</label>
    </pattern-link>
    <pattern-link>
      <linkID>W00026TL004</linkID>
      <type>contains</type>
      <patternID>W00026</patternID>
      <revision>1.0</revision>
      <instanceID>0002</instanceID>
      <collectionID>van Welie</collectionID>
      <label>Action Button</label>
    </pattern-link>
  </temporary-links>
</related patterns>
...

```

Figure 4. Fragment of PLML-like Definition of Advanced Search Pattern.



## VI. CONCLUSION

In this paper, we have analyzed three prevalent pattern collections in detail. Subsequently, we have discussed whether and how pattern definitions of these collections could be mapped into a PLML-compliant description format. We have identified some shortcomings of PLML in terms of suitability for automated pattern processing and we have proposed some changes and extensions in order to overcome these obstacles.

In our current research, we put our emphasis on the <implementation> element of PLML and target our efforts to enriching pattern definitions with code and model fragments and other valuable design resources.

## REFERENCES

- [1] Bienhaus, D.: PLMLx Doc., available at <http://www.cs.kent.ac.uk/people/staff/saf/patterns/plml.html> Retrieved on February 3, 2012.
- [2] Breiner, K. et al. (eds.): Proceedings of the 1<sup>st</sup> International Workshop on Pattern-Driven Engineering of Interactive Computing Systems (PEICS '10), June 20, 2010, Berlin, Germany, ACM Int. Conf. Proc. Series, 2010
- [3] Coram, T. and Lee J.: "A Pattern Language for User Interface Design", available at <http://www.maplefish.com/todd/papers/Experiences.html> Retrieved on March 13, 2012
- [4] Deng, J., Kemp E., and Todd E.G. (Hg.): Focusing on a standard pattern form: the development and evaluation of MUIP: Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: design centered HCI, 2006
- [5] Deng, J., Kemp E., and Todd E.G.: Managing UI Pattern Collections. Proc. CHINZ '05, Auckland, New Zealand, ACM Press, pp. 31-38, 2005.
- [6] van Duyne, D., Landay J., and Hong J.: "The Design of Sites, Patterns for Creating Winning Websites", 2nd Edition, Prentice Hall International, ISBN 0-13-134555-9, 2006
- [7] van Duyne, D., Landay J., and Hong J.: "The Design of Sites, Patterns for Creating Winning Websites", available at <http://www.designofsites.com/home/> Retrieved on March 13, 2012
- [8] Engel, J., Martin C., and Forbrig P.: HCI Patterns as a Means to Transform Interactive User Interfaces to Diverse Contexts of Use, in: J.A. Jacko (Ed.): Human-Computer Interaction, Part I, HCII 2011, LNCS 6761, Springer-Verlag Berlin Heidelberg pp. 204-213, 2011
- [9] Engel, J., Martin C., and Forbrig P.: Tool-support for Pattern-based Generation of User Interfaces, in [1], pp. 24-27, 2010
- [10] Fincher, S. et al.: Perspectives on HCI Patterns: Concepts and Tools (Introducing PLML), CHI '03 Workshop Report, 2003
- [11] Infragistics, Inc., "Quince", available at <http://quince.infragistics.com> Retrieved on March 13, 2012
- [12] Kaelber, C. and Martin C.: From Structural Analysis to Scenarios and Patterns for Knowledge Sharing Applications, in: J.A. Jacko (Ed.): Human-Computer Interaction, Part I, HCII 2011, LNCS 6761, Springer-Verlag Berlin Heidelberg, pp. 258-267, 2011
- [13] Kruschitz, C.: XPLML - A HCI Pattern Formalizing and Unifying Approach, Proceedings of CHI EA, 2009
- [14] Martin, C. and Roski A.: Structurally Supported Design of HCI Pattern Languages, Jacko, J. (Ed.) Human-Computer Interaction, Part I, HCII '07, Springer LNCS 4550 (2007), pp. 1159-1167, 2007
- [15] Tidwell, J.: "Designing Interfaces, Patterns for Effective Interaction Design", 2nd Edition, O'Reilly Media Inc., ISBN 978-1-449-37970-4, 2011
- [16] Tidwell, J.: "Designing Interfaces, Patterns for Effective Interaction Design", available at <http://www.designinginterfaces.com> Retrieved on March 13, 2012
- [17] van Welie, M.: "Patterns in Interaction Design", available at <http://www.welie.com> Retrieved on March 13, 2012
- [18] Yahoo! Inc.: "Yahoo! Design Pattern Library", available at <http://developer.yahoo.com/ypatterns/> Retrieved on March 13, 2012