A Unified Pattern Specification Formalism to Support User Interface Generation

Jürgen Engel[1,2(✉)], Christian Märtin[1], and Peter Forbrig[2]

[1] Faculty of Computer Science, Augsburg University of Applied Sciences, An der Hochschule 1, 86161 Augsburg, Germany
{Juergen.Engel, Christian.Maertin}@hs-augsburg.de
[2] Institute of Computer Science, University of Rostock, Albert-Einstein-Straße 21, 18059 Rostock, Germany
{Juergen.Engel, Peter.Forbrig}@uni-rostock.de

Abstract. The development of interactive software typically requires the combined skills of software developers, HCI, platform and marketing specialists in order to create applications with good software quality, usability, and user experience. The combination of model-based user interface development practices with pattern-based approaches that specify HCI- and software-patterns in a formalized way and respect emerging standards, has the potential to facilitate and automate the software development process, reduce development costs, and provide solutions that can easily be adapted to varying contexts of use. To satisfy these goals we have developed a framework for Pattern-based Modeling and Generation of Interactive Systems (PaMGIS). The paper at hand describes the topical version of the PaMGIS Pattern Specification Language (PPSL) which is designed to accomplish both, capturing all information required to support model-based user interface generation and offering the highest level of compatibility to existing pattern description languages.

Keywords: Model-based user interface development · Pattern-based development · User interface modeling · User interface generation · HCI patterns · Formal pattern specification

1 Introduction

The model-driven part of the PaMGIS framework is designed in the style of the CAMELEON Reference Framework [1]. Particularly, the ontological domain and context-of-use models are used as proposed by the CRF. However, we decided to split the CRF platform model into a device model and a user interface (UI) toolkit model. While the former comprises all relevant characteristics of the respective end-user device, the latter holds information about the user interface elements that are available on the respective underlying software development platform. This avoids redundancies especially in cases where the same software basis supports significantly different devices, e.g. Android on smartphones and tablet computers. On the basis of the domain model a dialog model is prepared that, in turn, helps to derive an abstract user interface model which is subsequently transformed into a concrete and lastly into the final user

M. Kurosu (Ed.): HCI 2016, Part I, LNCS 9731, pp. 445–456, 2016.
DOI: 10.1007/978-3-319-39510-4_41

interface model. The various UI model transformations are affected by the information which is deposited within the context-of-use model. An overview of the PaMGIS models and their interrelations is illustrated in (Fig. 1).

However, the construction of the required models is a time-consuming and

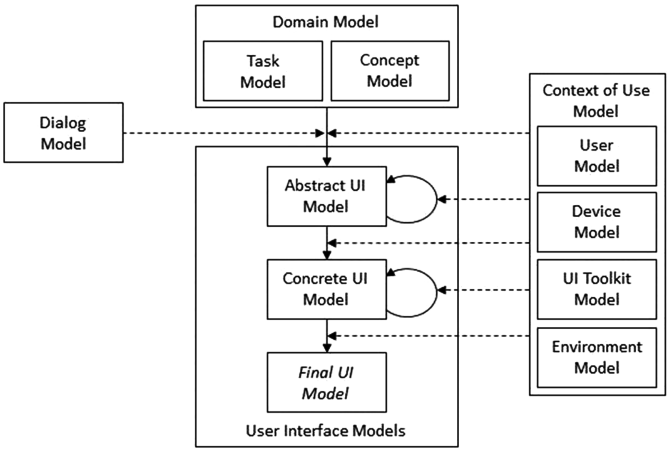

Fig. 1. Overview of the PaMGIS models and their interrelations

error-prone task. Hence, our combined user interface development approach uses patterns as means to alleviate the complexity of the model-driven design and transformation process. The basic idea is to equip patterns with pre-assembled model fragments which can be used as building blocks for the construction of the domain and UI models. In addition, certain patterns could also provide valuable input for the various model transformation steps as highlighted in (Fig. 1) .

For this purpose, we have developed the pattern specification language PPSL which allows us both, to particularize patterns in a formal and machine-readable way and to coevally incorporate any kind of information required to support the PaMGIS model-driven mechanisms. Beyond that, the design of PPSL was driven by two further major requirements. On the one hand, we aimed at the possibility to easily adopt already existing patterns from popular and well-accepted sources, such as the pattern collections of van Duyne [4], Tidwell [11], and van Welie [12]. On the other hand, we intended to raise the acceptability of PPSL by offering the highest level of compatibility to existing pattern description languages, i.e. the *Pattern Language Markup Language* (PLML) 1.1 [7], PLML 1.2 [3], and the *Extended Pattern Language Markup Language* (PLMLx) [1]. We also considered the *Pattern and Component Markup Language* (PCML) [9] and the *Task Pattern Markup Language* (TPML) [10]. However, these two languages are not elaborated within the paper at hand due to space restrictions. They can be translated to PPSL in an analogous manner.

We already have demonstrated in [5] that patterns from [4, 11, 12] can be mapped to PLML 1.1 in at least in a semi-automatic manner. Therefore, we now emphasize on the aforementioned pattern description languages and how respective pattern definitions can be translated into PPSL-based specifications.

2 Existing Pattern Description Languages

Within the context of the *CHI 2003 Conference on Human Factors in Computing Systems* a workshop was held named *Perspectives on HCI Patterns: Concepts and Tools*. One of the goals was the definition of a uniform description language for Human-Computer Interaction (HCI) patterns. Result was the XML-compliant *Pattern Language Markup Language* (PLML) version 1.1 which was intended to standardize pattern specifications and hence allow automatic pattern processing by means of standardized tools. PLML comprises several description elements resp. attributes, which are listed in Table 5 [7]. Brief descriptions can be found in Sect. 3.

PLML 1.2 is a further improved refinement of PLML 1.1 and was defined in 2006 within the context of the implementation of a pattern tool called *Management of User Interface Patterns* (MUIP). Major changes have been made in terms of the PLML 1.1 elements <forces>, <examples>, and <management>. Here, <forces> consists of a sub-element <force> which allows to specify particular influence factors separately and hence make them individually searchable and reusable. The same applies to the <example> element. In addition, individual examples are structured by means of the sub-elements <example-name>, <example-diagram>, <description>, and <known-uses>. The <management> element is enriched by the additional <change-log> sub-element which, in turn, is composed of <log-creation-date> and <log-content>. The entirety of amendments in PLML 1.2 compared to PLML 1.1 is illustrated in (Table 6) [3].

The primary goal of PLMLx is to describe patterns in a presentable, human- and machine-readable article-like format. Like PLML 1.2, PLMLx is based on PLML 1.1. PLMLx introduced three anew description elements, i.e. <acknowledgements>, <resulting-context>, and <organization>. Here, <acknowledgements> provides information on persons who were involved in the definition of the pattern and <resulting-context> contains a description of the context which is effective after the application of the pattern. This might include one or more new problems that need to be resolved. The <organization> element is intended to hold pattern-related metadata and comprises of three sub-elements, i.e. <category>, <collection> und <classification>. Additionally, the <management> element was supplemented by three further sub-elements, i.e. <copyright>, <license>, and <change-log>. The complete list of differences between PLMLx and PLML 1.1 can be found in (Table 7).

3 Introduction of the PaMGIS Pattern Specification Language

In order to appropriately support the modeling and UI generation process the patterns are described according to a particular markup language named *PaMGIS Pattern Specification Language* (PPSL). It incorporates the original expressiveness of PLML 1.1 and remedies some of PLML's inherent weaknesses, notably in terms of pattern relation modeling and provision of details required for automated pattern processing [6]. In addition, PPSL is designed to allow straight forward mapping of patterns defined in existing description formats, i.e. PLML 1.1, PLML 1.2, PLMLx, PCML, and TPML. According to our terminology, the current version is called PPSL 3.0.

Generally, PPSL pattern specifications are structured by means of four top level description elements, i.e. <Head>, <Body>, <Relationship>, and <Deployment>.

Here, the <Head> element incorporates pattern metadata, including information on pattern identification, names, work status, synopsis, originators, change history, further sources and literature, and legal aspects. Amongst others, pattern metadata is exploited

Table 1. PPSL description elements within the top level element <Head>

Element	Brief Description
<UniquePatternID>	Unique pattern identifier
<UPID_PatternCompilationID>	Identifier of the respective pattern collection
<UPID_PatternID>	Identifier of the pattern
<UPID_VersionNumber>	Version number of the pattern (major changes)
<UPID_RevisionNumber>	Revision number of the pattern (minor changes)
<Classification>	Information on classification of the pattern
<CLSS_PatternCompilationName>	Name of the respective pattern collection
<CLSS_Domain>	Name of the domain the pattern belongs to
<CLSS_PatternType>	Type of the pattern, e.g., "HCI pattern"
<CLSS_AbstractionLevel>	Abstraction level of the pattern
<CLSS_Category>	Category within the pattern collection
<CLSS_Subcategory>	Sub-category within the pattern collection
<CLSS_SubSubCategory>	Sub-sub-category within the pattern collection
<CLSS_Keywords>	List of keywords characterizing the pattern
<CLSS_Keyword>	Individual keyword
<Names>	Names of the pattern
<Name>	Preferred name of the pattern
<Aliases>	List of alternative names
<Alias>	Individual alternative name
<Status>	Work status, e.g., "draft" or "released"
<Synopsis>	Concise summary of the pattern's relevance
<Contribution>	Information on contributions to the pattern
<Authors>	List of pattern authors
<Author>	Information on an individual author
<ATHR_AuthorID>	Identifier of the author
<ATHR_FirstName>	Author's first name

<ATHR_LastName>	Author's last name
<ATHR_Organization>	Organization the author works for
<ATHR_PersonalDate>	Further information, e.g., a CV
<ATHR_ContactInformation>	Contact details
<ATHR_Link>	Link to a personal website
<ATHR_LinkDisplayName>	Name of the website
<ATHR_URL>	Actual URL
<Credits>	Information on contributing persons not mentioned within the list of authors
<History>	Development of the pattern specification over time
<Ancestry>	Information on the origin of the pattern
<ANCS_PatternCompilationName>	Name of the original pattern collection
<ANCS_PatternName>	Original name of the pattern within this collection
<ANCS_PatternID>	Identifier of the pattern in this collection
<ANCS_Category>	Name of the respective category in this collection
<ANCS_VersionNumber>	Version number within this collcection
<ANCS_RevisionNumber>	Revision number within this collection
<ANCS_Link>	Link to a website providing more information
<ANCS_LinkDisplayName>	Name of the Website
<ANCS_URL>	Actual URL
<CreationDate>	Date of pattern creation in PPSL format
<LastModified>	Date of latest change to the pattern specification
<Changes>	List of changes which were applied to the pattern
<Change>	Information on an individual change
<CHNG_ChangeID>	Identifier of the change
<CHNG_Date>	Date of change
<CHNG_VersionNumber>	Version number before the change
<CHNG_RevisionNumber>	Revision number before the change
<CHNG_Originator>	Name(s) or ID(s) of author(s)
<CHNG_Description>	Description of the change
<CHNG_Reason>	Reason for the change
<CHNG_Link>	Link to a website with further information
<CHNG_Link DisplayName>	Name of the Website
<CHNG_URL>	Actual URL
<References>	List of related sources and literature
<Reference>	Information on an individual source
<RFNC_Type>	Type of source, e.g., "book" or "website"
<RFNC_Title>	Titel or name of the source
<RFNC_Author>	Name(s) or ID(s) of author(s)
<RFNC_Description>	Summary in terms of purpose and contents
<RFNC_Reference>	Unique identifier, e.g., ISBN or URL
<LegalFoundation>	Legel foundation regarding the usage of the pattern
<Copyright>	Information on copyright restrictions
<License>	License-related information
<LCNS_Description>	Description of the licensing
<LCNS_Type>	Type of license, e.g., "GNU"
<LCNS_Agreement>	License agreement or link to it
<FurtherInformation>	Further information not yet mentioned in <Head>

to retrieve particular patterns within a pattern repository. An overview of the description elements of <Head> are listed in (Table 1).

The PPSL top level element <Body> represents the description of the actual pattern and is divided into the two sub-elements <Theory> and <Practice>. The former holds theoretical background information, including descriptions of the problem to be solved, the application context, and the solution for the given problem. The latter provides practical examples and known uses as well as results of usability activities, e.g. user testing. Details of the <Body> element are provided in (Table 2).

Relations between patterns can be defined by the <Relationship> top level element of PPSL. Compared to PLML 1.1 the identification of related patterns is more precise. Hence, relations between particular pattern versions resp. revisions of patterns can be expressed. (Table 3) shows the structure of the <Relations> element.

Finally, the <Deployment> top level element is a further development of the PLML 1.1 element <Implementation> and incorporates, amongst others, the information required for automatic processing and model construction within the PaMGIS framework.

It is designed to hold fragments of PaMGIS's fundamental models, including task model, concept model, dialog model, abstract UI (AUI) model, concrete UI (CUI) model, and final UI (FUI) model fragments. These model fragments are intended to be automatically integrated into the overall PaMGIS models as soon as the framework user selects and applies them. They can be regarded as building blocks for the PaMGIS models which have the potential to speed up the model design process, feature reuse of design work, and positively contribute to high usability and acceptable user experience of the final user interface. The structure of the PPSL top level element <Deployment> is depicted in (Table 4).

4 Mapping Existing Pattern Description Languages to PPSL

The mapping of PLML 1.1 description elements to PPSL is shown in Table 5. The left column contains the PLML 1.1 source elements while the right column specifies the respective PPSL target elements.

In a similar way, the correlation between PLML 1.2 and PPSL is illustrated in (Table 6) Due to space limits solely the differences between PLML 1.1 and PLML 1.2 are listed.

Finally, (Table 7) outlines the relation between PLMLx and PPSL. Like before, only the differences to PLML 1.1 are listed.

Due to lack of space we do not elaborate on the relationship between PCML resp. TPML and PPSL. However, the mapping is feasible analogously in both cases.

Table 2. PPSL description elements within the top level element < Body>

Element	Brief Description
<Theory>	Theoretical aspects regarding the pattern
<Problem>	Description of the problem to be solved
<PRBL_Digest>	Summary
<PRBL_Elaboration>	Detailed description
<Context>	Situations in which the pattern can be applied
<CNTX_Digest>	Summary
<CNTX_Elaboration>	Detailed description
<Solution>	Description of how to resolve the problem
<SLTN_Digest>	Summary
<SLTN_Elaboration>	Detailed description
<Forces>	List of forces in the environment
<Force>	Force that the pattern will resolve
<FRCE_Digest>	Summary
<FRCE_Elaboration>	Detailed description
<Rationale>	Discussion and any principled reasons
<Confidence>	Rating of how likely the pattern provides an invariant solution for the given problem
<ResultingContext>	Context after the application of the pattern
<Diagrams>	List of schematic visualization of the pattern
<Diagram>	Individual schematic visualization
<Practice>	Pratical aspects regarding the pattern
<Examples>	List of application examples (within PaMGIS)
<Example>	Information on one individual example
<XMPL_Type>	Example type, i.e., (counter)example
<XMPL_ExampleID>	Identifier of the example
<XMPL_Label>	Name of the example
<XMPL_Description>	Textutal description of the example
<XMPL_Diagram>	Schematic visualization of the example
<XMPL_Images>	List of relevant images
<XMPL_Image>	Individual image, e.g., a screenshot
<XMPL_Realizations>	List of links to implementations
<XMPL_Realization>	Individual link to an implementation
<XMPL_Modelings>	List of modelings
<XMPL_Modeling>	Individual modeling (list of links)
<XMPL_ModelFragmentID>	Link to model fragments within the PPSL description element <Deployment>
<Known Uses>	List of known uses (outside of PaMGIS)
<KnownUse>	Information on one individual known use
<KNWN_CreationDate>	Date of creation
<KNWN_KnownUseID>	Identifier of the known use
<KNWN_Label>	Name of the known use
<KNWN_Description>	Textual description
<KNWN_Originator>	Name of the originator
<KNWN_Link>	Link to a website using the pattern
<KNWN_LinkDisplayName>	Name of the website
<KNWN_URL>	Actual URL
<KNWN_Images>	List of relevant images
<KNWN_Image>	Individual image, e.g., a screenshot
<UsabilityFeedback>	Results of usability activities, e.g., user testing

Table 3. PPSL description elements within the top level element < Relationship>

Element	Brief Description
<RLTN_Information>	Textual description of relations to other patterns
<Relations>	List of relations to other patterns
<Relation>	Information on an individual relation
<RLTN_RelationID>	Identifier of the relation
<RLTN_Label>	Name of the relation
<RLTN_Nature>	Nature of relation (within PaMGIS or external)
<RLTN_Type>	Type of relation, e.g., “aggregation”
<RLTN_Reference>	Identity of the related pattern
<RLTN_PatternCompilationID	Identifier of the respective pattern collection
<RLTN_PatternCompilationName>	Name of the respective pattern collection
<RLTN_PatternID	Identifier of the related pattern
<RLTN_PatternName>	Name of the related pattern
<RLTN_VersionNumber>	Version number of the related pattern
<RLTN_RevisionNumber>	Revision number of the related pattern
<RLTN_Description>	Textual description, comments, and hints

Table 4. PPSL description elements within the top level element < Deployment>

Element	Brief Description
<Implementations>	List of possible implementations
<Implementation>	Information on an individual implementation
<IMPL_ImplementationID>	Identifier of the implementation
<IMPL_Label>	Name of the implementation
<IMPL_Description>	Textual description of the implementation
<IMPL_Code>	Code fragments
<EmbeddingLinks>	List of links to OOA and/or OOD models (see [8])
<EmbeddingLink>	Information on an individual link
<EMBL_EmbeddedLinkID>	Identifier of the link
<EMBL_Label>	Name of the link
<EMBL_ReferenceClassID>	Class within the OO model being referenced
<EMBL_UMLRelationshipType>	UML type of relationship
<PaMGIS>	Information for processing with PaMGIS
<ModelFragments>	List of model fragments
<ModelFragment>	Information on an individual model fragment
<MDFR_Type>	Type of model fragment, e.g., task model
<MDFR_FragmentID>	Identifier of model fragment
<MDFR_Label>	Name of model fragment
<MDFR_Purpose>	Description of the purpose of the fragment
<MDFR_Fragment>	Actual model fragment
<MDFR_Base>	List of references to model fragments from which the current fragment has been derived
<MDFR_BaseReference>	ID of an individual model fragment
<MDFR_Annotation>	Comments and annotations in textual format

Table 5. Mapping of PLML 1.1 to PPSL

PLML 1.1 Element	Corresponding PPSL Element
<patternID>	<Head> <UniquePatternID> <UPID_PatternID>
<name>	<Head> <Names> <Name>
<alias>	<Head> <Names> <Aliases>
<illustration>	<Body> <Practice> <Examples> <Example>
<problem>	<Body> <Theory> <Problem>
<context>	<Body> <Theory> <Context>
<forces>	<Body> <Theory> <Forces> <Force>
<solution>	<Body> <Theory> <Solution>
<synopsis>	<Head> <Synopsis>
<diagram>	<Body> <Theory> <Diagrams> <Diagram>
<evidence>	Merely used for structuring purposes; not required in PPSL
<example>	<Body> <Practice> <Examples> <Example>
<rationale>	<Body> <Theory> <Rationale>
<confidence>	<Body> <Theory> <Confidence>
<literature>	<Head> <References> <Reference>
<implementation>	<Deployment> <Implementations> <Implementation>
<related-patterns>	<Relationship> <RLTN_Information>
<pattern-link>	<Relationship> <Relations> <Relation>
<type>	<RLTN_Type>
<patternID>	<RLTN_Reference> <RLTN_PatternID>
<collection>	<RLTN_Reference> <RLTN_PatternCompilationName>
<label>	<RLTN_Label>
<management>	Merely used for structuring purposes; not required in PPSL
<author>	<Head> <Contribution> <Authors> <Author>
<credits>	<Head> <Contribution> <Credits>
<creation-date>	<Head> <History> <CreationDate>
<last-modified>	<Head> <History> <LastModified>
<revision-number>	<Head> <UniquePatternID> <UPID_RevisionNumber>

5 Conclusion and Outlook

Within the present paper we introduced the topical version of the PaMGIS Pattern Specification Language (PPSL). On the one hand, it is designed to equip patterns with all information required to use them as building blocks for model construction in the context of the PaMGIS framework. On the other hand, PPSL strives for best possible compatibility to existing pattern description languages in order to enable the reutilization of already available patterns without major efforts. To this effect, we demonstrated how PLML 1.1, PLML 1.2, and PLMLx can be mapped to PPSL. Mapping of PCML and TPML is also possible, but not elaborated in the paper due to space limits.

Table 6. Mapping of PLML 1.2 PPSL (differences of PLML 1.2 compared to PLML 1.1)

PLML 1.2 Element	Corresponding PPSL Element
<collection>	<Head> <Classification> <CLSS_PatternCompilationName>
<forces>	<Body> <Theory> <Forces>
<force>	<Force>
<example>	<Body> <Practice> <Examples> <Example>
<example-name>	<XMPL_Label>
<example-diagram>	<XMPL_Diagram>
<description>	<XMPL_Description>
<known-uses>	<Body> <Practice> <KnownUses>
<literature>	Merely used for structuring purposes; not required in PPSL
<work-name>	<Head> <References> <Reference> <RFNC_Title>
<reference>	<Head> <References> <Reference> <RFNC_Reference>
<implementation>	<Deployment> <Implementations> <Implementation>
<implementation-name>	<IMPL_Label>
<code>	<IMPL_Code>
<other-details>	<IMPL_Description>
<pattern-link>	<Relationship> <Relations> <Relation>
<revision-number>	<RLTN_Reference> <RLTN_RevisionNumber>
<management>	Merely used for structuring purposes; not required in PPSL
<change-log>	<Head> <History> <Changes>
<log-creation-date>	<Change> <CHNG_Date>
<log-content>	<Change> <CHNG_Description>

To fulfill the aforementioned requirements PPSL consists of a multitude of description elements. In order to preserve manageability of the pattern specifications we semantically grouped them into the four top level elements <Head>, <Body>, <Relationship>, and <Deployment>. In addition, the patterns are created, managed, retrieved, and applied with the aid of a set of tools included in the PaMGIS framework.

Our upcoming activities will target on the augmentation of our pattern repository. For this, we will focus on both, the translation and enrichment of existing patterns and the development of new domain-specific patterns, e.g. for electronic vending and commerce. Further, we intend to intensify our research on how results of usability activities, e.g. user testing, can be exploited to improve the quality of our patterns and hence also of the user interfaces being generated by PaMGIS. As a consequence, this will entail more substance and structure of the PPSL description element <Body> <Practice> <UsabilityFeedback>.

Table 7. Mapping of PLMLx to PPSL (differences of PLMLx compared to PLML 1.1)

PLMLx Element	Corresponding PPSL Element
<acknowledgements>	<Head> <Contribution> <Credits>
<evidence>	Merely used for structuring purposes; not required in PPSL
<example>	<Body> <Practice> <Examples> <Example>
<rationale>	<Body> <Theory> <Rationale>
<organization>	<Head> <Classification>
<category>	<CLSS_Category>
<collection>	<CLSS_PatternCompilationName>
<classification>	<CLSS_PatternType>
<resulting-context>	<Body> <Theory> <ResultingContext>
<management>	Merely used for structuring purposes; not required in PPSL
<author>	<Head> <Contribution> <Authors> <Author>
<change-log>	<Head> <History> <Changes>
<change>	<Change>
<author>	<CHNG_Originator>
<description>	<CHNG_Description>
<version>	Merely for structuring purposes; not required in PPSL
<date>	<CHNG_Date>
<majorNo>	<CHNG_VersionNumber>
<minorNo>	<CHNG_RevisionNumber>
<copyright>	<Head> <LegalFoundation> <Copyright>
<creation-date>	<Head> <History> <CreationDate>
<credits>	<Head> <Contribution> <Credits>
<last-modified>	<Head> <History> <LastModified>
<license>	<Head> <LegalFoundation> <License>
<license-type>	<LCNS_Type>
<ulink>	<LCNS_Agreement>
<revision-number>	<Head> <UniquePatternID> <UPID_RevisionNumber>

References

1. Bienhaus, D.: PLMLx Doc (2004). http://www.cs.kent.ac.uk/people/staff/saf/patterns/plml.html. Accessed 6 Jan 2016
2. Calvary, G., et al.: The CAMELEON reference framework. Document D1.1 of the CAMELEON R&D Project IST-2000-30104 (2002)
3. Deng, J., et al.: Focusing on a standard pattern form: the development and evaluation of MUIP. In: Proceedings of the 6th ACM SIGCHI New Zealand Chapter's International Conference on Computer-Human Interaction: Design Centered HCI (2006)
4. van Duyne, D., et al.: The Design of Sites: Patterns for Creating Winning Websites, 2nd edn. Prentice Hall International, Upper Saddle River (2006). ISBN 0-13-134555-9
5. Engel, J., et al.: Exploiting HCI pattern collections for user interface generation. In: Proceedings of Patterns 2012 (Nice, France), IARIA 2012, pp. 36–44 (2012)

6. Engel, J., et al.: Pattern-based modeling and development of interactive information systems. In: Frotschnig, A., Raffaseder, H. (eds.) Forum Medientechnik – Next Generation, New Ideas, pp. 155–167. vwh Hülsbusch, Glückstadt (2012)
7. Fincher, S., et al.: Perspectives on HCI patterns: concepts and tools (introducing PLML). In: CHI 2003 Workshop Report (2003)
8. Märtin, C., Herdin, C., Engel, J.: Patterns and models for automated user interface construction – in search of the missing links. In: Kurosu, M. (ed.) HCII/HCI 2013, Part I. LNCS, vol. 8004, pp. 401–410. Springer, Heidelberg (2013)
9. ObjectVenture Inc.: Pattern and Component Markup Language (PCML), Draft 3 (2002). http://www.cryer.co.uk/glossary/p/pcml/PCMLSpecification.pdf. Accessed 6 Jan 2016
10. Sinnig, D.: The complicity of patterns and model-based UI development. Master thesis, Concordia University, Montreal, Québec, Canada (2004)
11. Tidwell, J.: Designing Interfaces: Patterns for Effective Interaction Design, 2nd edn. O'Reilly Media Inc., Sebastopol (2011). ISBN 978-1-449-37970-4
12. van Welie, M.: Patterns in Interaction Design. http://www.welie.com. Accessed 6 Jan 2016