

Beschreibung von Usability-Mustern für georäumliche Anwendungen

Benno Schmidt¹

Labor für Geovisualisierung, Hochschule Bochum¹

benno.schmidt@hs-bochum.de

Zusammenfassung

Hinsichtlich ihrer Gebrauchstauglichkeit (Usability) weisen georäumliche Anwendungen oft domänen-spezifische Defizite auf. Zur Dokumentation und Diskussion diesbezüglicher Lösungsansätze bietet sich das Konzept des „Musters“ („pattern“) an. Da innerhalb der Musterbeschreibungen auf wiederkehrende domänentypische Elemente und Konzepte zurückgegriffen wird, ist allerdings die Frage nach den konkret benötigten sprachlichen Konstrukten zu stellen. Weiterhin ergibt sich die Aufgabe, das in den Mustern abgelegte Wissen innerhalb des Software-Entwicklungsprozesses praktisch nutzbar zu machen.

1 Einleitung

Muster lassen sich als strukturierte Beschreibungen bewährter Lösungen für wiederkehrend auftretende Probleme innerhalb spezieller Kontexte definieren. Im Umfeld der objektorientierten Programmierung hat sich beispielsweise die Nutzung von Entwurfsmustern („design patterns“; Gamma et al., 1994) auf Klassenebene etabliert. Bezogen auf Probleme des Benutzerschnittstellenentwurfs oder der Gebrauchstauglichkeit informationstechnischer Produkte lassen sich spezielle Interaktions- oder Usability-Muster identifizieren; vgl. auch Röder (2012).

Der vorliegende Beitrag befasst sich mit der Beschreibung von Usability-Mustern für georäumliche Anwendungen, z. B. bezogen auf die Gestaltung von Benutzeroberflächen für Geoinformationssysteme, Web-Mapping-Anwendungen oder interaktive Anwendungen im Umfeld von Geodateninfrastrukturen (GDIs). Tatsächlich weisen georäumliche Anwendungen nicht selten erhebliche Usability-Defizite auf (Resch & Zimmer, 2013; Komarkova et al., 2017; Henzen, 2017). Ausgehend von der Prämisse, dass eine explizite Bezugnahme auf domänenspezifische Elemente sinnvoll ist, wird im Folgenden ein Beispiel einer (semi-) formalen Notation zur Musterbeschreibung betrachtet und hinsichtlich ihres Anwendungspotenzials

diskutiert. Zielsetzung des Sprachentwurfs ist es zunächst, die Dokumentation und Katalogisierung bewährter Lösungsansätze zu ermöglichen und die Diskussion der praktischen Eignung sowie der Verträglichkeit kombiniert angewendeter Ansätze zu unterstützen.

2 Formale Beschreibung von Usability-Mustern

Musterbeschreibungen lassen sich auf vielfältige Art und Weise zum Zweck der maschinellen Verarbeitung enkodieren. Eine Übersicht über praktisch eingesetzte Ansätze gibt z. B. Gundelsweiler (2012). Eine XML-basierte Umsetzung bietet u. a. die im Zuge der CHI 2003 spezifizierte Pattern Language Markup Language, kurz PLML (Fincher, 2003), welche dem mathematisch formulierten Konzept von Borchers (2000) folgt und Aspekte der Dokumentation und Beschreibung allgemeiner kontextualisierter Muster fokussiert. Der Bezug zum Usability-Aspekt fehlt dort jedoch ebenso wie der zur hier betrachteten fachlichen Anwendungsdomäne. Weiterhin ist eine maschinelle Auswertbarkeit der vorwiegend natürlichsprachlich verfassten Dokumentationen kaum gegeben. Dies führt zu der Frage, auf welche Weise sich strukturiertere und präzisere Beschreibungen bereitstellen lassen.

Ein möglicher Ansatz für Web-Anwendungen im GDI-Umfeld wird z. B. von Henzen (2017) skizziert, indem eine weitere Sprachsyntax zur Beschreibung von grundsätzlichen Interaktionselementen und von in den Anwendungen verankerten Arbeitsmechanismen vorgeschlagen wird. Prinzipiell lassen sich derartige Sprachen in bestehende Implementierungen von Mustersprachen einbetten. So wurde im nachfolgenden Beispiel in der XML-Schema-Definition der PLML ein XML-Element ergänzt, um die Beschreibung des Musters „Kartenlink in Datensatzbeschreibung“ (beschrieben durch Henzen, 2017) aufzunehmen (Code aus <https://github.com/bogeo/upatterns>, der besseren Lesbarkeit halber unwesentlich verändert):

```
<pattern patternID='42'>
  <name>Map Link In Service Description</name>
  <problem>
    The user wants to navigate from ...
  </problem>
  ...
  <formalization>
    precondition:
      EXISTS d1 : DATASET, EXISTS s1 : SERVICE, d1 IS_COUPLED_RESOURCE_OF s1
    pattern: Map_link_in_service_description =
      type navigation
      CLICK((LINK | BUTTON) IN (RESULT(d1) | DETAIL(d1))(s1)) => MAP(s1)
      internal RESULT IN RESULT_LIST
    pattern-relations:
      IS_RELATED_TO(Map_link_in_dataset_description)
  </formalization>
  ...
</pattern>
```

Das Muster beschreibt das Prinzip, zu recherchierten Geodatenätzen in der Benutzeroberfläche einen anklickbaren Link anzubieten, über den Inhalt und räumliche Lage des Datensatzes in einer Karte grafisch dargestellt werden können. Die Grundmengen der konkret instanzierbaren Datensätze d_i bzw. Webdienste s_j lauten hierbei DATASET bzw. SERVICE, die Relation IS_COUPLED_RESOURCE_OF beschreibt die Bindung eines Datensatzes an einen Webdienst. CLICK() beschreibt eine parametrisierte Nutzerinteraktion, die weiteren parametrisierten Elemente stehen für Dialogelemente der grafischen Benutzerschnittstelle. Die Relation $\rightarrow$ dient zur Beschreibung automatisch ablaufender Mechanismen, hier die unmittelbare Generierung einer Kartendarstellung für den Dienst s_1 , welche durch das Anklicken des Links oder Buttons zu einem angezeigten Datensatz d_1 angestoßen wird.

3 Nutzung von Musterwissen im Software-Entwurf

Der vorgestellte Sprachentwurf lässt die konzeptuellen Elemente der Anwendungsdomäne sichtbar werden. An dieser Stelle lässt sich fragen, welche weiteren sprachlichen Elemente in der betrachteten Domäne bedeutsam sind, wobei sich für verschiedene Nutzergruppen (z. B. GDI-Experten vs. -Laien) und Entwicklerrollen (z. B. Web-Designer vs. Programmierer) relevante Unterschiede ergeben können. Kandidaten für derartige Elemente sind beispielsweise die der ISO/OGC-Welt entstammenden Konzepte „Geodienst“ (Web-Ressource mit definierter Ein-/Ausgabeschnittstelle), „Kartenebene“ (Layer), „Darstellungsvorschrift“ (Style) oder „Metadatenatz“, auf welchen im GDI-Umfeld häufig operiert wird (vgl. OGC Technical Committee, 2017), oder Nutzerschnittstellen-Elemente wie interaktive Kartenansicht (MapView), punktuelle Sachdatenabfragefunktion (Feature-Info, Identify) oder die Anzeige einer Trefferliste, welche in vielen Benutzeroberflächen zu finden sind.

Syntaktisch korrekte Beschreibungen lassen sich nutzen, um weitere Artefakte zu generieren. Exemplarisch sei die automatisierte Ableitung von UML-Code genannt. So lässt sich z. B. ein UML-Klassenmodell erzeugen, insofern die im Eingabecode referenzierten relevanten domänenspezifischen Fachobjekte und Nutzerschnittstellen-Elemente mit geeigneten Templates zur Instanzierung von UML-Objekten bzw. -Klassen verknüpft sind; siehe Schmidt (2018) für ein konkretes Beispiel. So wird u. a. sichtbar, welche Elemente innerhalb einer zu entwerfenden Software kollaborieren. Sind mehrere Muster gleichzeitig aktiv (wovon im Regelfall auszugehen ist), so lassen sich u. U. Hinweise auf mehrfach referenzierte Elemente gewinnen und möglicherweise drohende Inkonsistenzen aufdecken.

Die Aktivierung eines Usability-Musters kann zusätzliche funktionale Merkmale einer Software-Anwendung implizieren, wobei derartige „usability features“ vom Nutzer der Software nicht notwendigerweise als eigenständige Funktionen wahrgenommen werden. Diese Usability-Features lassen sich durch eine einheitliche und integrierte Spezifikation in die Anforderungsspezifikation und somit in den Software-Entwicklungsprozess einbetten (vgl. Röder, 2012).

4 Aufbau von Mustersammlungen

Mit dem Ziel des kollaborativen Aufbaus einer Mustersammlung für die betrachtete Anwendungsdomäne folgt das hier beschriebene Vorgehen folgenden Schritten:

1. *Allgemeine Beschreibung* der Muster (weitgehend natürlichsprachliche Angabe von Problem-Lösungs-Paaren und weiterer Aspekte, z. B. gemäß der Strukturvorgabe der PLML oder einer vergleichbaren Sprache);
2. *Kontextualisierung* der beschriebenen Muster durch formale Angabe der Musterrelationen;
3. *Präzisierung* der Lösungen durch Ergänzung von Beschreibungen mit höherem Formalisierungsgrad;
4. *Ableitung weiterer Artefakte*, z. B. explorierbare Kontextvisualisierungen, prüfbare Musternetze, nachnutzbare UML-Modelle etc.

Der Prozess wird mehrfach iterativ durchlaufen werden, indem z. B. im Laufe der Zeit textliche Formulierungen verbessert, Musterbeschreibungen verfeinert oder Kontextualisierungen aktualisiert werden, um nur einige Anwendungsfälle zu nennen.

Grundsätzlich können in den Schritten 1 und 2 alle Interessenbeteiligten – der einfache Anwender ebenso wie der GDI-Experte, Web-Designer oder Programmierer – gleichermaßen aktiv werden. Die nachfolgenden Schritte 3 und 4 werden hingegen die Kenntnis des konzeptuellen Mustermodells und der eingesetzten formalen Fachsprache und Werkzeuge voraussetzen. Prinzipiell erscheint die in diesem Beitrag exemplarisch betrachtete Sprache gut skalierbar, indem die grundlegenden Bestandteile der Musterbeschreibungen z. B. über einen geeigneten Editor eingegeben werden können und die präzisierten, formalen Beschreibungen von Experten ausgearbeitet werden.

5 Schlussfolgerungen und Ausblick

Zu konstatieren ist, dass für den betrachteten Anwendungsbereich der georäumlichen Anwendungen die in einfachen Mustersprachen (wie z. B. der PLML) vorhandenen Sprachkonstrukte nicht ausreichen. Vielmehr ist es wünschenswert, ein erweitertes Vokabular unter Berücksichtigung einheitlicher domänenspezifischer Elemente bereitzustellen, um die Usability-relevanten Beschreibungen präzise fassen zu können und eine maschinengestützte Nachnutzung zu ermöglichen.

Mustersammlungen lassen sich während der Anforderungsspezifikation und des Nutzerschnittstellen-Entwurfs beratend nutzen, indem z. B. geeignete Recherchefunktionen bereitgestellt werden, die in den Suchkriterien Merkmale des zu entwerfenden Zielsystems berücksichtigen und zugleich das formalisierte Musterwissen nutzen. Mit Blick auf die Prozessierbarkeit der Beschreibungen und die Ableitbarkeit im Software-Entwicklungsprozess nutzbarer Artefakte liefern die Erfahrungen aus der Anwendung der exemplarisch betrachteten Beschreibungssprache Ansatzpunkte. Wünschenswert für die Zukunft wäre der Aufbau einer größeren

Sammlung von Usability-Mustern für den betrachteten Anwendungsbereich, die nutzbringende Hinweise auf die tatsächlich benötigten Sprachelemente liefern kann.

Weiterhin lässt sich die Forschungsfrage formulieren, ob und gegebenenfalls wie sich der Brückenschlag zur praktischen Software-Entwicklung herstellen lässt. Zwar sind diesbezüglich erste erfolgreiche Arbeiten verfügbar (z.B. Röder, 2012; Carvajal et al., 2013; Nguyen et al., 2016). Für die Anwendungsdomäne georäumlicher Software-Anwendungen besteht jedoch weiterhin Forschungsbedarf.

Literaturverzeichnis

- Borchers, J. O. (2000). *A Pattern Approach to Interaction Design*. Proceedings DIS 2000 International Conference on Designing Interactive systems, Aug. 16-19, 2000, New York. New York: ACM Press.
- Carvajal, L., Moreno, A. M., Segura, M. I. & Seffah, A. (2013). *Usability through Software Design*. IEEE Transactions on Software Engineering, Vol. 39, No. 11, pp. 1582-1596.
- Fincher, S. (2003). *Perspectives on HCI Patterns: Concepts and Tools (introducing PLML)*. Interfaces, Vol. 56, Sept. 2003, pp. 26-28.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.
- Gundelsweiler, F. (2012). *Design-Patterns zur Unterstützung der Gestaltung von interaktiven, skalierbaren Benutzungsschnittstellen*. München: Oldenbourg Verlag.
- Henzen, C. (2017). *Usability-Patterns für Webanwendungen in Geodateninfrastrukturen*. Dissertation, Fakultät Umweltwissenschaften der Technischen Universität Dresden.
- Komarkova, J., Sedlak, P., Habrman, J. & Cermakova, I. (2017). *Usability Evaluation of Web-Based GIS By Means of a Model*. Proceedings of the International Conference on Information and Digital Technologies (IDT), Zilina, Slovakia, 2017, July 5-7, pp. 191-197.
- Nguyen, T.-D., Vanderdonckt, J. & Seffah, A. (2016). *UIPLML: Pattern-based Engineering of User Interfaces of Multi-Platform Systems*. Proceedings of IEEE 10th International Conference on Research Challenges in Information Science (RCIS'2016), 2016, June 01-03, IEEE Press: Piscataway, pp. 273-284.
- OGC Technical Committee, eds. (2017). *Abstract Specifications*, <http://www.opengeospatial.org/docs/as>, Wayland, MA: Open Geospatial Consortium (OGC).
- Resch, B. & Zimmer, B. (2013). *User Experience Design in Professional Map-Based Geo-Portals*. ISPRS International Journal of Geo-Information, 2013(2), pp. 1015-1037.
- Röder, H. (2012). *Usability Patterns : Eine Technik zur Spezifikation funktionaler Usability-Merkmale*. Dissertation, Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart.
- Schmidt, B. (2018). *Konzeptuelle Elemente einer Beschreibungssprache für Usability-Muster in GDI-Anwendungen*. Labor für Geovisualisierung der Hochschule Bochum, Bericht Nr. 18-01, https://www.hs-bochum.de/fileadmin/media/fb_v/prof_schmidt/Bericht_18-01.pdf