

Abstrakte Beschreibung automobiler HMI-Systeme

Björn Reich¹, Michael Weber¹, André Berton², Christoph Schürer²

¹Institut für Medieninformatik
Universität Ulm
James-Franck Ring
D-89069 Ulm
michael.weber@uni-ulm.de,
bjoern.reich@uni-ulm.de

²Daimler AG
Wilhelm-Runge Strasse 11
D-89081 Ulm
andre.berton@daimler.com,
christoph.schuerer@daimler.com

Zusammenfassung: In diesem Artikel wird eine Beschreibungssprache für automobiler HMI-Systeme definiert, die es ermöglicht ein solches System unabhängig von einem zur Modellierung eingesetzten Programm zu beschreiben. Vor der Definition dieser eigenen Beschreibungssprache werden vorhandene Sprachen auf ihre Tauglichkeit hin untersucht. Anschließend wird am Beispiel von Web-Services gezeigt, wie mit einer solchen Beschreibungssprache auch die Anbindung neuer Dienste an ein HMI-Modell beschrieben werden kann.

1 Einleitung

Die Spezifikation eines Head-Unit HMI-Systems wird in Zukunft größtenteils modellbasiert ablaufen. Bei dieser Art der Spezifikation, wird das Verhalten des HMI-Systems in Form von Zustandsautomaten beschrieben. Unterstützung liefern dabei spezielle Modellierungstools, wie etwa *Elektrobit GUIDE* (Elektrobit 2008).

Damit die auf diese Weise erstellten HMI-Modelle auch an externe Zulieferer gegeben werden können, ist es notwendig ein Austauschformat zu besitzen, das es ermöglicht, ein komplettes HMI-Modell auf abstrakte Art und Weise beschreiben zu können. Die Abstraktion ist erforderlich, damit eine Unabhängigkeit der Beschreibungssprache von einem eingesetzten Modellierungstool erreicht wird. Des Weiteren ist es wünschenswert, dass mit der Beschreibungssprache eine Trennung zwischen Struktur (Beschreibung der grafischen Objekte) und Verhalten erreicht werden kann. Außerdem ist eine Funktion gewünscht, die es erlaubt, existierende HMI-Modelle durch die Beschreibungssprache zu erweitern, ohne das komplette Modell neu beschreiben zu müssen. Zusätzlich stellt die Lesbarkeit des Formats durch Menschen eine zentrale Anforderung dar.

2 Analyse existierender Beschreibungssprachen

Es wurden sowohl Beschreibungssprachen aus dem automobilen Bereich, als auch allgemeine Beschreibungssprachen ohne automobilen Hintergrund auf ihre Tauglichkeit zur Beschreibung automobiler HMI-Systeme untersucht.

2.1 Beschreibungssprachen für automobile HMI

Folgende Beschreibungssprachen aus dem automobilen Bereich wurden analysiert:

OEM-XML ist eine Beschreibungssprache, die aus dem Arbeitskreis HMI-Methodik entstanden ist, dem die deutschen Automobilhersteller Audi, BMW, Daimler, Porsche und Volkswagen angehören (Brunhorn 2007). Mit OEM-XML wird eine sehr gute Abstraktion der Beschreibung erreicht, was für ein Austauschformat nützlich ist. Die Verhaltensbeschreibung in Form von Zustandsautomaten weist allerdings einige Schwächen auf und erreicht durch die Verknüpfung von Widget und Zustand nicht die gewünschte Trennung zwischen Verhalten und Struktur.

ICUC-XML ist eine Sprache der Firma Elektrobit zur Beschreibung eines Kombi-Instrumentes (Hübner & Grüll 2007). Nachteilig an diesem Format ist, dass keine Abstraktion erreicht wird, da die Beschreibungssprache zur Verwendung mit dem Elektrobit Programm GUIDE definiert wurde. Die Konstrukte des Programms werden dabei identisch in der Beschreibungssprache verwendet. Positiv hingegen ist der Aufbau der Verhaltensbeschreibung und die damit verbundene Trennung zwischen Struktur und Verhalten eines HMI-Systems.

BMW-FLUID ist ein komplettes Framework zur Erweiterung eines Head-Unit HMI-Systems mit neuen Diensten. Ziel dabei ist es, dass Dienste-Entwickler möglichst wenig über den Aufbau eines konkreten HMI-Systems wissen müssen (Hildisch et al. 2007). Durch diesen Ansatz wird eine sehr gute Abstraktion bei der Beschreibung erreicht. Spezielle Regeln, sogenannte „Generation Rules“, werden von einem Compiler zur Umsetzung der abstrakten Beschreibung in konkrete Konstrukte für das jeweilige HMI-System verwendet. Als Austauschformat für ein komplettes HMI-System ist es allerdings auf Grund der Spezialisierung auf neue Dienste nicht geeignet.

2.2 Allgemeine Beschreibungssprachen

Bei der Analyse der allgemeinen Beschreibungssprachen UIML 3.1, SCXML und Adobe Flex 3, konnten interessante Kenntnisse erlangt werden, die auch zur Beschreibung eines automobilen HMI-Systems verwendet werden können. Im Folgenden werden die Sprachen näher dargestellt.

UIML ist eine plattform-unabhängige XML-Beschreibungssprache für grafische Oberflächen (UIML.org 2004). Erkenntnis der Analyse dieser Beschreibungssprache ist, dass die grafische Oberfläche eines HMI-Systems sehr gut mit UIML beschreibbar ist und auf Grund der Plattformunabhängigkeit von UIML die gewünschte Abstraktion erreicht wird. Die Verhaltensbeschreibung durch Zustandsautomaten ist mit UIML allerdings nicht möglich.

SCXML ist ein als Draft-Version vorliegender Standard des W3C zur Beschreibung von Zustandsautomaten mit XML (W3C 2007). Die Ansätze, die *SCXML* zur Beschreibung von Zustandsautomaten wählt, sind ebenfalls anwendbar für die Verhaltensbeschreibung eines HMI-Systems. Allerdings werden zum jetzigen Zeitpunkt noch nicht alle Zustandsarten unterstützt, die die UML2 für Zustandsautomaten vorsieht, wie zum Beispiel „Fork-“ und „Join-States“. Die Möglichkeit, eine grafische Oberfläche zu beschreiben, fehlt in *SCXML* völlig und muss daher von anderen Beschreibungssprachen übernommen werden.

Adobe Flex 3 ist ein Framework zur Erstellung kompletter Anwendung für den Web- und Desktop-Bereich (Adobe 2008). *Flex 3* besteht aus zwei Sprachen, mit denen die grafische Oberfläche und das Verhalten einer Anwendung definiert werden kann. *MXML* ist eine auf XML basierende Sprache, mit der die Oberfläche der Anwendung beschrieben wird. Für die Definition des Verhaltens wird *ActionScript* verwendet. *ActionScript* ist aber eher eine Programmier- als eine Beschreibungssprache, so dass für eine modellbasierte Spezifikation die Zustandsautomaten in *ActionScript* programmiert werden müssen. Dies ist allerdings innerhalb einer Spezifikation und mit Blick auf eine mögliche Austauschfähigkeit der Beschreibung nicht gewünscht.

3 Definition der abstrakten Beschreibungssprache AbstractHMI

Die Analyse vorhandener Beschreibungssprachen verdeutlicht, dass keine der untersuchten Beschreibungssprachen die gestellten Anforderungen vollständig erfüllt. Aus diesem Grund wird mit *AbstractHMI* eine eigene Beschreibungssprache definiert, die auch einige interessante Konstrukte der untersuchten Sprachen aufnimmt.

3.1 Aufbau von AbstractHMI

AbstractHMI besteht aus drei Arten von Dokumenten, den Strukturbeschreibungen, der Verhaltensbeschreibung und der Umsetzungstabelle (Mapping Table). In Abbildung 1 auf Seite 4 ist dieser Aufbau grafisch dargestellt.

Die strukturbeschreibenden Dokumente enthalten die Beschreibung der grafischen Oberfläche, zum Beispiel die Anordnung der Widgets. Es existieren mehrere dieser Dokumente, die jeweils einen HMI-Screen beschreiben. Das verhaltensbeschreibende Dokument enthält die Zustandsautomaten und deren Elemente.

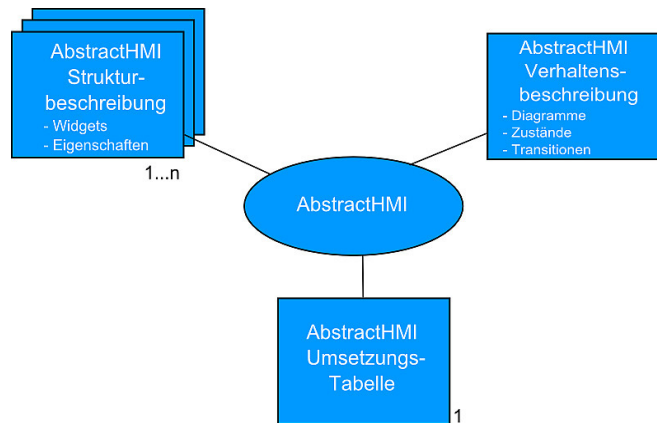


Abbildung 1: Aufbau von AbstractHMI

Die dritte Dokumentenart ist eine Mapping-Tabelle. Diese enthält die Anweisungen, wie die in der Strukturbeschreibung verwendeten abstrakten Widget-Typen auf konkrete Widget-Typen des jeweiligen Modellierungstools umgesetzt werden sollen.

Die strukturbeschreibenden Dokumente sind intern in vier Abschnitte gegliedert. Der erste Abschnitt dient der Definition der Widgets und ihrer Hierarchie. Abbildung 2 zeigt das Beispiel einer Widgetdefinition.

```
<hmi:widget id="System" class="View">
  <hmi:widget id="StatusLine" class="Container">[... ]
  <hmi:widget id="ApplicationLine" class="Container">[... ]
  <hmi:widget id="SystemContainer" class="Container">[... ]
  <hmi:widget id="SubFunctionLine" class="Container">
    <hmi:widget id="ButtonSettings" class="Button"/>
    <hmi:widget id="ButtonTime" class="Button"/>
    <hmi:widget id="ButtonDisplay" class="Button"/>
  </hmi:widget>
</hmi:widget>
```

Abbildung 2: Definition von Widgets in AbstractHMI

Der zweite Abschnitt ist für die Zuweisung von Eigenschaften an die Widgets vorgesehen. Diese Zuweisung kann mehrfach für ein Widget auftreten, damit zum Beispiel mehrere Baureihen beschrieben werden können. Im dritten Abschnitt können den einzelnen Widgets Inhalte hinzugefügt werden, etwa die Beschriftung eines Buttons. Auch an dieser Stelle ist es möglich, den Widgets mehrere verschiedene Inhalte zuzuweisen, um damit eine multi-linguale Oberfläche beschreiben zu können. Im letzten Abschnitt können für die Widgets Ereignisse definiert werden, auf die diese reagieren, zum Beispiel die Ereignisse „pressed“ und „released“ für einen Button.

Das verhaltensbeschreibende Dokument enthält alle für das HMI-System definierten Zustandsautomaten. Die Zustandsautomaten enthalten Diagramme, die wiederum die Zustände, Transitionen und Bedingungen enthalten. Die Anordnung der Zustände innerhalb eines Diagramms entspricht ihrer Hierarchie.

Die Mapping-Tabelle enthält die Anweisungen zur Abbildung der abstrakten Widget-Typen, die in der Strukturbeschreibung verwendet werden (siehe Abbildung 2), auf die tatsächlich zu verwendenden Widget-Typen. Der Vorteil einer solchen Mapping-Tabelle ist, dass eine zentrale Stelle existiert, die bei einer Änderung der zu verwendenden Widgets einfach angepasst werden kann. Durch die Kombination dieser verschiedenen Dokumente ist es nun mit *AbstractHMI* möglich, ein komplettes HMI-System auf abstrakte Weise zu beschreiben.

3.2 Implementierung für AbstractHMI

Damit die definierte Beschreibungssprache *AbstractHMI* produktiv eingesetzt werden kann und eine Aussage darüber getroffen werden kann, in wie weit existierende HMI-Modelle beschreibbar sind, ist es notwendig, eine Implementierung für *AbstractHMI* zu erstellen. Dies wurde als Plug-In für das Modellierungstool *EB GUIDE* realisiert. Die Implementierung bietet eine Import-/Exportfunktion für *AbstractHMI*. Dies ermöglicht es, HMI-Modelle aus der importierten abstrakten Beschreibung in *EB GUIDE* zu erzeugen und auch die modellierten HMI-Systeme in die abstrakte Beschreibung zu exportieren. Anhand dieser Implementierung konnte auch an existierenden HMI-Modellen erfolgreich getestet werden, wie gut eine Beschreibung mit *AbstractHMI* möglich ist.

4 Anbindung von Web-Services an die modellbasierte Spezifikation

Nachdem eine abstrakte Beschreibungssprache definiert und eine entsprechende Implementierung dafür entwickelt wurde, stellt sich nun die Frage, ob auch die Anbindung neuer Dienste, speziell Web-Services, abstrakt beschreibbar sind. Am Beispiel der beiden Web-Services *Google Maps* und *Google LocalSearch* wurde eine Anbindung der Dienste an ein HMI-Modell realisiert. Das Hauptproblem bei der Anbindung und abstrakten Beschreibung dieser Anbindung sind die vielen unterschiedlichen Schnittstellen von Web-Services, wie WSDL, REST und Web-APIs. Mit WSDL existiert zwar ein Standard des W3C zur Beschreibung von Web-Services. Allerdings wird dieser von populären Web-Services wie *Google*, *Yahoo* und *Microsoft* nicht verwendet. Diese Dienste stellen überwiegend eine eigene Web-API zur Verfügung, auf die mit verschiedenen Programmiersprachen zugegriffen werden kann. Die Anbindung der beiden genannten *Google* Dienste an ein HMI-Modell ist trotzdem möglich, erfordert allerdings eine aufwendige Programmlogik im Hintergrund.

Für die Art der Anbindung sind mehrere Ansätze denkbar, die auch davon abhängen, wie tief die Integration der Dienste sein soll. So muss zum Beispiel eine Entscheidung getroffen werden, ob die Dienste im Design des OEM auf der Head-Unit dargestellt werden sollen, oder in ihrer eigenen Darstellung. Die komplette abstrakte Beschreibung einer solchen, auf heterogenen Schnittstellen basierenden Anbindung, ist kaum möglich, da die Web-APIs ständig geändert werden. Beschrieben werden kann das komplette Verhalten bis zum Aufruf einer konkreten Funktion, die der Web-Service anbietet. Dies kann zum Beispiel als Aktion an einer Transition abstrakt definiert sein. Danach existiert keine Möglichkeit mehr, die nun notwendige Verarbeitungslogik abstrakt zu beschreiben.

5 Zusammenfassung

Mit *AbstractHMI* wurde eine XML-basierte Beschreibungssprache definiert, die es ermöglicht, sowohl das Verhalten als auch die Struktur eines HMI-Systems komplett und unabhängig von einem Modellierungstool beschreiben zu können. Die abstrakte Beschreibung der Anbindung von Web-Services an das HMI-Modell ist auf Grund der unterschiedlichen Schnittstellen schwierig. Eine Anbindung lässt sich trotzdem erreichen und auch, mit einigen Abstrichen bezüglich der notwendigen Logik im Hintergrund, mit *AbstractHMI* beschreiben.

6 Literaturverzeichnis

- Adobe (2008). *Adobe Flex 3*, , <http://www.adobe.com/products/flex/>
- Brunhorn J. (2007). *XML-Sprache zur Beschreibung von HMIs für Infotainmentsysteme und Kombiinstrumenten*, Carmeq GmbH / OEM Arbeitskreis HMI Methodik.
- Elektrobit (2007). *Elektrobit GUIDE, Graphical User Interface Designer*.
- Hildisch A., Steurer J., Stolle R. (2007). *HMI generation for plug-in services from semantic descriptions*.
- Hübner M., Grüll I. (2007). *ICUC-XML Format Specification Revision 14*, Elektrobit.
- UIML.org (2004). *Interface Markup Language (UIML) Specification*.
- W3C (2007). *W3C: State Chart XML (SCXML), Working Draft*