

CaMa-Frame – Ein Framework für Case Management Prozesse

Jochen Küster, André Zensen (Fachbereich Wirtschaft & Gesundheit, Fachhochschule Bielefeld)

1. Einführung

Case Management (CM) ist ein neues Paradigma in der Wirtschaftsinformatik zur Modellierung und Implementierung von wissensintensiven Prozessen. Anders als beim klassischen Geschäftsprozessmanagement sind Abläufe in ihrer Abarbeitungsreihenfolge nicht von vornherein festgelegt, sondern werden um ein Geschäftsartefakt strukturiert. Dies kann beispielsweise bei Gesundheitsprozessen eine Patientenakte sein.

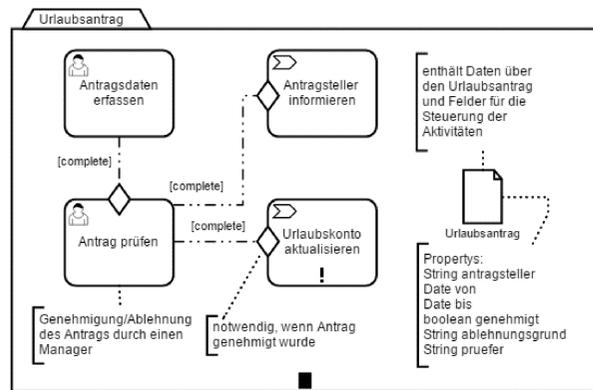


Abb.1: Einfaches CMMN-Modell „Urlaubsantrag“

- „Case Management Model and Notation“ (CMMN) erster Ansatz zur Modellierung von CM
- Grafische Notation (s. Abb. 1) und komplexe, standardisierte Ausführungssemantik
- Ermöglicht sehr flexible Abläufe

- CaMa-Frame baut auf CMMN auf und ist eine leichtgewichtige Alternative zu kommerziellen Angeboten, um CM-Applikationen zu realisieren

2. Architektur einer CM-Applikation

Aus verschiedenen Anforderungen an CM-Anwendungen wurde ein Konzept entwickelt, welches prototypisch in Java EE implementiert wurde. Eine mit dem Framework erstellte CM-Applikation ist in einen Applikationsserver eingebettet und teilt sich in mehrere Schichten auf (s. Abbildung 2).

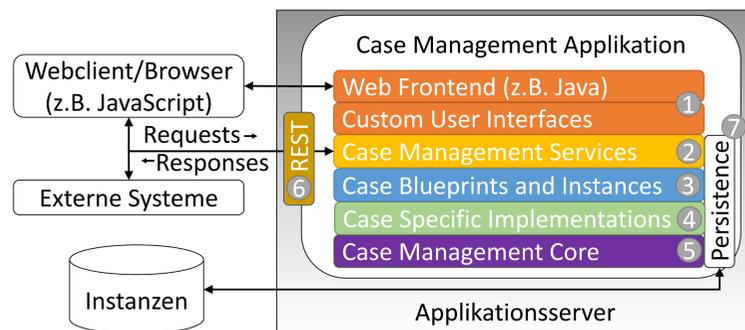


Abb.2: Struktur einer mit dem Framework erstellten CM-Applikation

1. Web Frontend/Custom **User Interfaces**: Grafische Benutzerschnittstellen für Fallbearbeiter zur Interaktion mit dem Framework.
2. Case Management **Services**: Schnittstellen, um mit dem Framework, Fällen und Daten zu arbeiten
3. Case Blueprints and Instances: Vorgefertigte Fallstrukturen, die instanziiert werden können oder bereits instanziiert sind. Zugriff erfolgt durch Services.
4. Case Specific Implementations: Anwendungsspezifische Implementierungen für Geschäftslogik.
5. Case Management Core: Wiederverwendbare Basisbausteine des Frameworks basierend auf vereinfachten CMMN-Strukturen.
6. REST-Interfaces: Schnittstellen für externe Systeme.
7. Persistence: Objektrelationaler Ansatz für die Speicherung und Manipulation von Fall-Instanzen.

3. Basisbausteine des Frameworks

CMMN bietet verschiedene Elemente zur Modellierung von CM-Anwendungen an. Hierzu gehören verschiedene Task-Typen (bspw. HumanTasks, die von menschlichen Nutzern bearbeitet werden, oder ProcessTasks, welche Prozesse anstoßen können) sowie Datenelemente. Durch Milestones können zu erreichende Ziele modelliert werden. Das Framework vereinfacht diese komplexen Strukturen der Spezifikation und bietet repräsentative Klassen zur Implementierung von CM-Anwendungen an. Abbildung 2 zeigt als Ausschnitt die grundlegenden CM-Kernelemente des Frameworks.

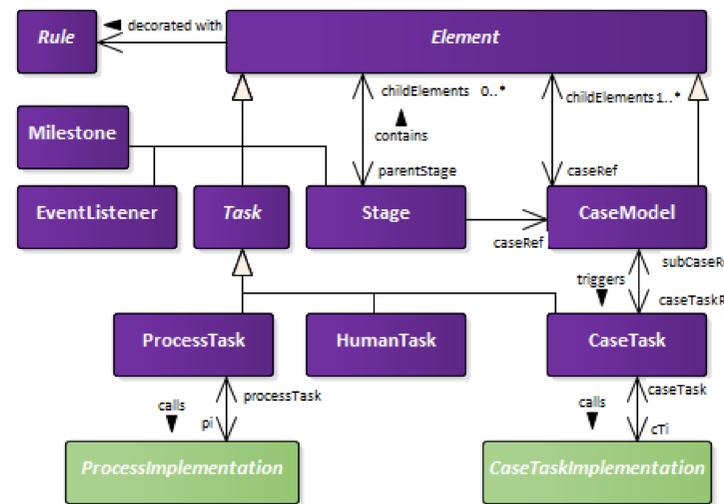


Abb.3: CM-Kernelemente des Frameworks (Ausschnitt)

4. Services des Frameworks

Das Framework bietet eine ausgeprägte Service-Schicht an (s. Abb. 4), um Fall-Instanzen zu starten, abzurufen und enthaltene Elemente entsprechend der Ausführungssemantik von CMMN manipulieren zu können. Für Basisbausteine werden entsprechende Services angeboten, um unter anderem Daten anzuzeigen, oder einen HumanTask abzuschließen.

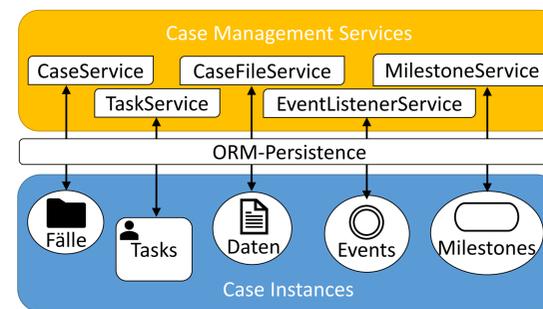


Abb.4: Services zur Bearbeitung von Fall-Instanzen und Elementen

Auf den Services setzen REST-Schnittstellen auf, die Kommunikation mit externen Systemen ermöglicht. Die Services nutzen in der Persistenzschicht objektrelationale Abbildungen, um direkt mit Objekten arbeiten zu können.

5. Entwicklung einer CM-Applikation mit dem Framework

Zunächst werden CMMN-Modelle in wiederverwendbare Kernelemente des Frameworks übertragen und Fallstrukturen erstellt (s. Abb. 5). Diese Fall-Blaupausen können anschließend mehrfach instanziiert werden.

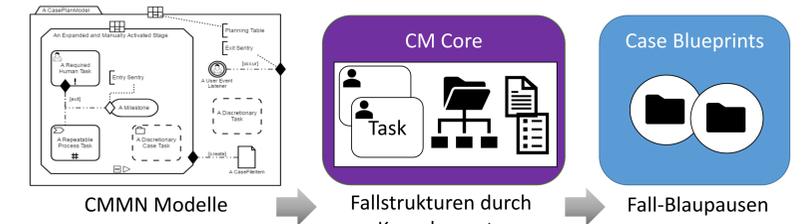


Abb.5: Transformation von CMMN-Modellen in Framework-Strukturen

Der konkrete Arbeitsablauf kann durch Aufgaben, Daten und Fallbearbeiter beeinflusst werden. CMMN ermöglicht es, Elemente wie Aufgaben und Daten dynamisch miteinander zu verknüpfen. Die korrekte Ausführung instanziiertener Fall-Blaupausen übernimmt das Framework und folgt dabei der komplexen Ausführungssemantik der CMMN-Spezifikation. Entwickler können sich so auf die Implementierung der Geschäftslogik konzentrieren.

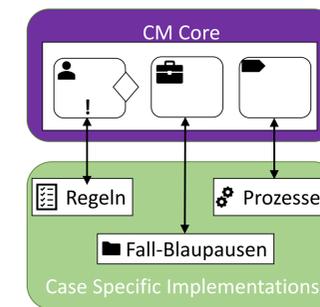


Abb.6: Individuelle Logik

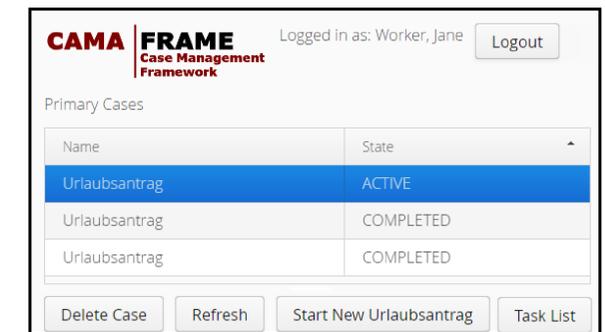


Abb.7: Im Framework enthaltene Benutzerschnittstellen

Entwickler ergänzen erstellte Fall-Blaupausen durch Geschäftsregeln, untergeordnete Fallstrukturen und Prozesse (s. Abb. 6). Das Framework bietet hierzu entsprechende Strukturen an. Daneben werden grafische Schnittstellen implementiert, um Fallbearbeiter bei der Ausführung laufender Fälle zu unterstützen. Das Framework bietet hierzu bereits einfache grafische Benutzermasken an, die mit Hilfe bestehender Services entwickelt worden sind (s. Abb. 7).

6. Fazit

- Systematische Entwicklung leichtgewichtiger CM-Applikationen ausgehend von CMMN-Modellen als gemeinsame Grundlage von Analysten und Entwicklern
- Modelle werden in die Strukturen des Frameworks übertragen und um anwendungsspezifische Geschäftslogik ergänzt
- Entwickler konzentrieren sich auf die Implementierung von Geschäftslogik und grafischer Benutzerschnittstellen, unterstützt durch ausgeprägte Services

Referenzen

- Zensen, A., Küster, J. (2018): A Comparison of Flexible BPMN and CMMN in Practice. A Case Study on Component Release Processes. In: *IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*.
- Zensen, A., Küster, J. (2019): A Java-based Framework for Case Management Applications. In: *17th Int. Conference on Business Process Management (BPM) BPM Forum (LNBIP)*
- Abschlussbericht des Forschungsprojekts CaMa-Frame – Ein Framework für Case-Management Prozesse und Anwendungen

Besuchen Sie unsere Projektwebseite unter <https://www.fh-bielefeld.de/wug/forschung/ag-pm/cama-frame>.