

16.1.1 Motivation und Zielstellung

Prozessmodelle für TIM lassen sich einfach erstellen, sind transparent und können ohne großen Aufwand veröffentlicht werden.

Prozessbegleitformulare können genau so einfach, schnell und preiswert entwickelt werden wie Prozessmodelle:

- Configuration Bases Application (CBA) bietet genau dafür eine Lösung.
- Mit dem gleichen Arbeitsprinzip – Modellierung und Konfiguration – werden Bedienoberflächen und Anwendungsteile erstellt, sind flexibel anpassbar und mit Prozessmodellen kombinierbar.

Damit sind Human Workflows mit TIM eine runde Sache.

16.1.2 Begrifflichkeiten Modellierung, Konfiguration, Parametrierung

Modellierung:

Ist eine freie Gestaltung sachlicher Zusammenhänge auf der Basis einer einfach verständlichen, häufig grafischen und standardisierten Notationsform.

Beispiele:

- Prozessmodelle nach BPMN 2.0
- Klassenmodelle nach UML
- Datenbankmodelle nach Entity-Relationship-Modellierung
- Bedienoberflächenstrukturen in CBA

Konfiguration:

Stellt die Definition von Eigenschaften eines Entwurfsobjektes nach einem vorgegeben Schema während des Design dar.

Beispiele:

- Service-Aktivität in BPMN 2.0
- Eigenschaften einer Klasse in UML
- Eigenschaften eines Attributes in einem Entity-Relationship-Modell
- Eigenschaften eines Datentyps in CBA

Parametrierung:

Entspricht der Definition von Eigenschaften eines Entwurfsobjektes nach einem vorgegeben Schema zu Laufzeit.

Beispiele:

- Übergabe von Parametern an einen Subprozess
- Initialisierungsparameter in einer Klasse
- Übergabe von Parametern per Script an eine Methode in CBA

16.1.3 Allgemeine Konventionen

CBA operiert mit **Namenskonventionen**:

- CBA kennt logische Bezeichnungen (z.B. Attributbezeichnungen), die in den Bedienoberflächen wie das entsprechende Entwurfsobjekt benannt sind, und Klartexte (z.B. Überschriften, Beschreibungen) für die Bedienoberflächen.
 - Während Klartexte/Überschriften beliebige Zeichen (inkl. Umlaute und Sonderzeichen) enthalten können, haben alle logischen Bezeichnungen der Namenskonvention von XML-Tags zu entsprechen, können also nur die Zeichen a-zA-Z0-9-_ enthalten. Leerzeichen und andere Trennzeichen sind verboten.
 - **Eine Nichteinhaltung dieser Konventionen führt zu nicht definierten Zuständen.** Das Verhalten des Systems hängt vom Kontext ab, in dem die logische Bezeichnung verwendet wird. Wird sie z.B. beim Speichern/Laden von Konfigurationsdateien verwendet, wird der XML-Parser eine Fehlermeldung ausgeben. Bei einem autocomplete werden Leer- und andere Trennzeichen als Ende der zu erkennenden Zeichenkette erkannt, was zu unerwünschtem Verhalten führen kann.
 - Besonders zu beachten ist, dass diese **Namenskonventionen schon bei der Benennung des Prozesses sowie der Aktivitäten in der Prozessdefinition** anzuwenden sind. Diese Informationen werden ohne Prüfung an CBA weitergegeben und können dort zu oben genannten Effekten führen. iGrafx kennt die Unterscheidung in logische Bezeichnungen und Klartexte nicht. Deshalb sind die strengeren Konventionen einzuhalten.
-

16.1.4 Entwurfsobjekte

CBA kennt folgende Entwurfsobjekte:

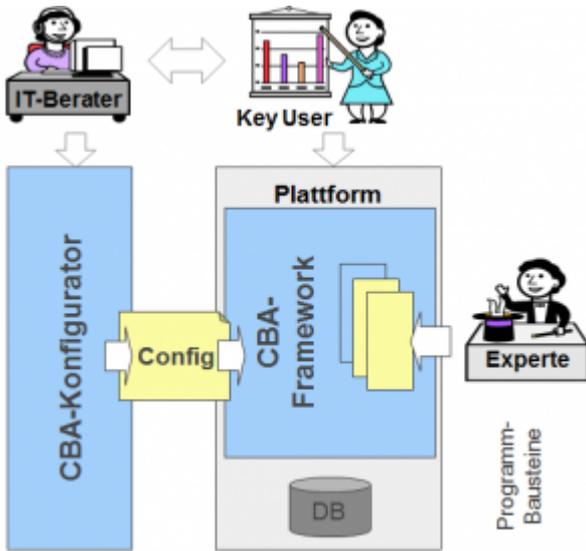
- **Bedienoberflächen:** strukturierte Einheiten mit Funktionen zur Datenbereitstellung, Navigation und der Realisierung von Bedienlogik
- **Templates:** Definition des grundsätzlichen Layouts einer Bedienoberfläche
- **Tabellen:** Definition logischer Datenstrukturen und ihres Mapping auf die Datenquelle, Umsetzung von Geschäftslogik
- **Datentypen:** Definition der Datenablage, der erlaubten Werte, der Bedienerführung, der Validierung und Angebot von datentyp-spezifischen Darstellungs-Varianten
- **Einheiten:** spezielle Plugins zur Implementierung algorithmischer Bestandteile
- **Meldungen:** Nachrichten für Fehlermeldungen, Nutzerhinweise, verwendbar als Textbausteine z.B. für emails oder in Bedienoberflächen
- **Dokumententypen:** vordefinierte Typen von Dokumenten zur Einbindung in eine Anwendung

Durch die Modellierung der Entwurfsobjekte wird die Struktur der Anwendung und der Informationsfluss zwischen den Entwurfsobjekten definiert. Für die Entwurfsobjekte entsteht eine funktionale Struktur. Die Verwendung der Entwurfsobjekte kann abgefragt werden.

16.1.5 Arbeitsweise

CBA favorisiert eine besondere Arbeitsweise. Sie ist charakterisiert durch:

- businessorientiertes Framework als Zielsystem
- Anwendung in CBA-Konfigurator erstellt
- Konfigurations-Datei in CBA-Framework eingespielt
- Programmteile als definierte Bausteine hinzugefügt



16.1.6 Komplexitätsstufen

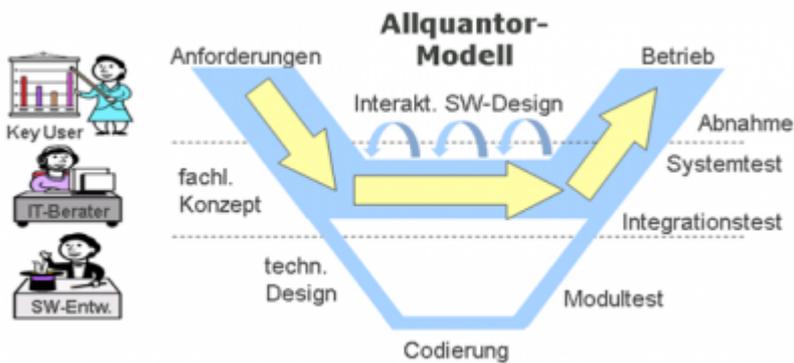
CBA beinhaltet folgenden Komplexitätsstufen:

pure	simple	medium	complex
<ul style="list-style-type: none"> ■ Modellierung des Prozesses in BPMN 			
<ul style="list-style-type: none"> ■ kein Prozess-Begleitformular 	<ul style="list-style-type: none"> ■ einfaches Prozess-Begleitformular 	<ul style="list-style-type: none"> ■ strukturiertes Prozess-Begleitformular 	
<ul style="list-style-type: none"> ■ Abarbeitung ausschließlich durch Starten und Beenden von Tasks 	<ul style="list-style-type: none"> ■ fachliche Daten in Prozessvariablen ■ Validierung entsprechend Prozessfortschritt 	<ul style="list-style-type: none"> ■ fachliche Daten in prozessspez. Datenbanken/XML ■ umfangreiche Nutzerführung und Validierung ■ Stammdatenanbindung 	<ul style="list-style-type: none"> ■ kompl. Geschäftslogiken ■ Integration in IT-Anwendungen/Landschaften
Modellierung und Konfiguration			Programmierung
fachlicher Nutzen			
<ul style="list-style-type: none"> ■ einfache Prozess-Modellierung, Dokumentation, Handlungsanweisungen, hilfreiche Links 			
<ul style="list-style-type: none"> ■ zielgerichtete Steuerung der Abarbeitung, insbesondere Ressourcensteuerung, Exception/Escalation Handling 			
<ul style="list-style-type: none"> ■ Transparenz und Prozeßqualität, Prozeßmonitoring und -steuerung, BAM 			
<ul style="list-style-type: none"> ■ Steuerung der Prozesslogik durch fachliche Daten, Suchfunktionen 			
<ul style="list-style-type: none"> ■ einfache fachliche Daten 	<ul style="list-style-type: none"> ■ strukturierte fachliche Daten 	<ul style="list-style-type: none"> ■ Komfort und Robustheit 	<ul style="list-style-type: none"> ■ integrierte Informationsflüsse

16.1.7 Methodik

Mit CBA wird eine spezielle Methodik angewandt:

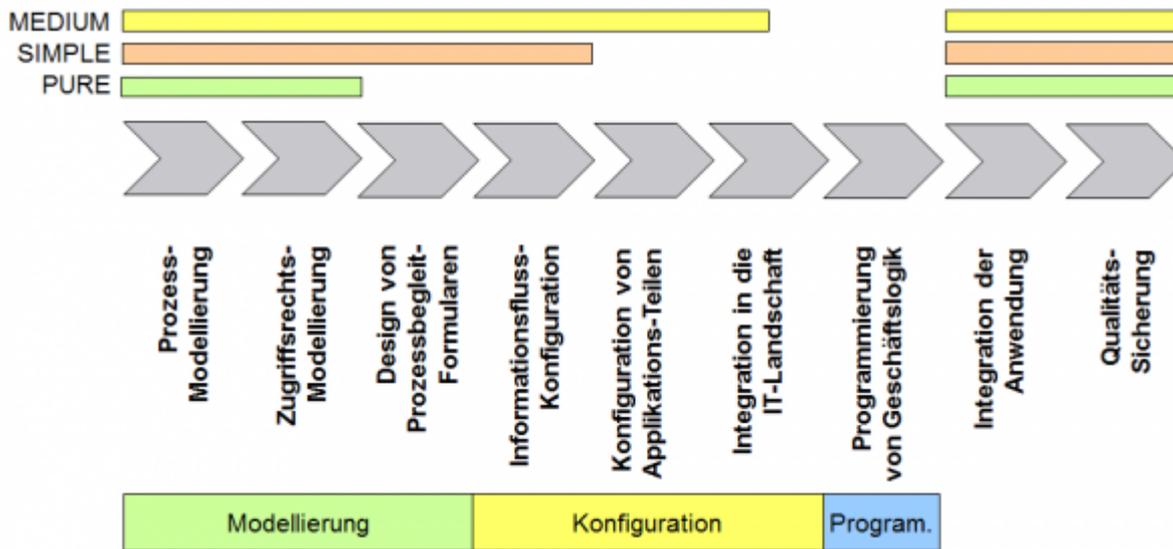
- CBA ermöglicht einen **interaktiven Software Design-Prozess** für prozessorientierte Anwendungen.
- Das Software-Design wird unmittelbar im Zielsystem schrittweise entwickelt und verfeinert.
- Die Lösung wird für den Key User erlebbar.
- Die Struktur und Bedienbarkeit der Bedienoberflächen können schnell an die Anforderungen der Key User angepasst werden.



- Key User und IT-Berater diskutieren und schärfen die Anforderungen.
- Prototyping mittels Modellierung und Konfiguration
- schrittweise Verfeinerung und Ableitung von Programmieraufgaben
- Implementierung von Programmteilen und Integration
- gemeinsamer Systemtest

16.1.8 Modellierungsmethodik

CBA kennt folgende Modellierungsmethodik:



16.1.9 Aufruf CBA-Konfigurator

Der CBA-Konfigurator wird aus dem TIM-Prozess-Repository aufgerufen.

The screenshot shows the CBA-Konfigurator interface with the 'Sections' tab selected. The interface is divided into several sections:

- Basisinformationen:** ID, Kurzbez., Name, Beschreibung, Email, Abteilung.
- Prozessinformationen:** Hardware 1, Hardware 2, Software 1, Software 2, Kosten Hardware, Kosten Software.
- Zusat:** Freige, Datum, Freige.

Buttons for 'Bearbeiten' (Edit) and 'Starten' (Start) are visible at the top right.

- Prozessmodell im TIM-Repository vorhanden, aber nicht veröffentlicht
- Vorschau eines vorhandenen Prozessbegleitformulars
- Aufruf des CBA-Konfigurators
- Wenn kein Prozessbegleitformular existiert, wird bei Aufruf des CBA-Konfigurators automatisch eine leere Vorlage geladen.
- Übergang von CBA-Konfigurator SIMPLE zu PROFESSIONAL

16.1.10 Wichtige Bedienelemente

CBA implementiert in den UIU's eine Bedienphilosophie, die mit Hilfe von Icons an allen Stellen in den

Bedienoberflächen einheitlich verfügbar ist:

- Neue Datensätze können eingefügt werden (INSERT).
- Existierende Datensätze können editiert werden (Standard-Einstellung - EDIT).
- Existierende Datensätze können gelöscht werden (DELETE).
- Neue und/oder geänderte Datensätze können gespeichert werden (SAVE).
- Der Ursprungszustand eines Datensatzes kann wieder hergestellt werden (UNDO).
- Es kann der nächste Datensatz aktiviert werden (NEXT).
- Es kann der vorhergehende Datensatz aktiviert werden (PREVIOUS).



16.1.11 Konfiguration validieren, testen und veröffentlichen

Nach dem Design der Konfiguration kann diese validiert, getestet und veröffentlicht werden.

- Das Speichern mit Hilfe des Speichern-Icons speichert die Daten aus der Bedienoberfläche in einen internen Cache – nicht in die Konfigurations-Datei.
- Konfiguration wird mit dem **Veröffentlichen-Button** in der Konfigurationsdatei gespeichert.
- Öfter mal Speichern ist empfohlen.
- Klick auf den **Test-Button** ermöglicht das Testen der vorhandenen Konfiguration (d.h. die, die im Cache ist ohne diese vorher in der Konfigurationsdatei zu speichern).
- Beim Test wird der eingestellte Status berücksichtigt.
- Testen bedeutet, dass die meisten der konfigurierten Bedienoberflächen und Funktionen ausprobiert werden können.
- Einschränkungen gibt es, wenn in der Design-Umgebung nicht alle Ressourcen wie Datenbanken, Schnittstellen, Webservice etc. verfügbar sind.
- Klick auf den **Validieren-Button** prüft die vorhandene Konfiguration (d.h. die, die im Cache ist ohne diese vorher in der Konfigurationsdatenbank zu speichern).
- Ergebnis ist ggf. eine mehrzeilige Fehlermeldung mit allen aufgetretenen Problemen.

From:

<https://wiki.tim-solutions.de/> - **TIM Wiki** / [NEW TIM 6 Documentation](#)

Permanent link:

<https://wiki.tim-solutions.de/doku.php?id=software:cba:overview>

Last update: **2021/07/01 09:52**

