



Jens Nerche

Leiter Anwendungsentwicklung

Kontext E GmbH

www.kontext-e.de

Email j.nerche@kontext-e.de

Twitter [@jensnerche](https://twitter.com/jensnerche)

Blog <http://wp.me/p3leb2-Z>

1. Teil

Was ist eine Language Workbench?

2. Teil

Beispiele aus der Praxis

Language Workbench

Sprachelemente

Generator

Editor

Typesystem

Intentions

Constraints

Actions („Editor-Magie“)



Language Workbench

Scope

VCS-Integration

Debugger

Testunterstützung

Scripts, Plugins,
Refactoring,
Data Flow



Einige Workbenches



Xtext



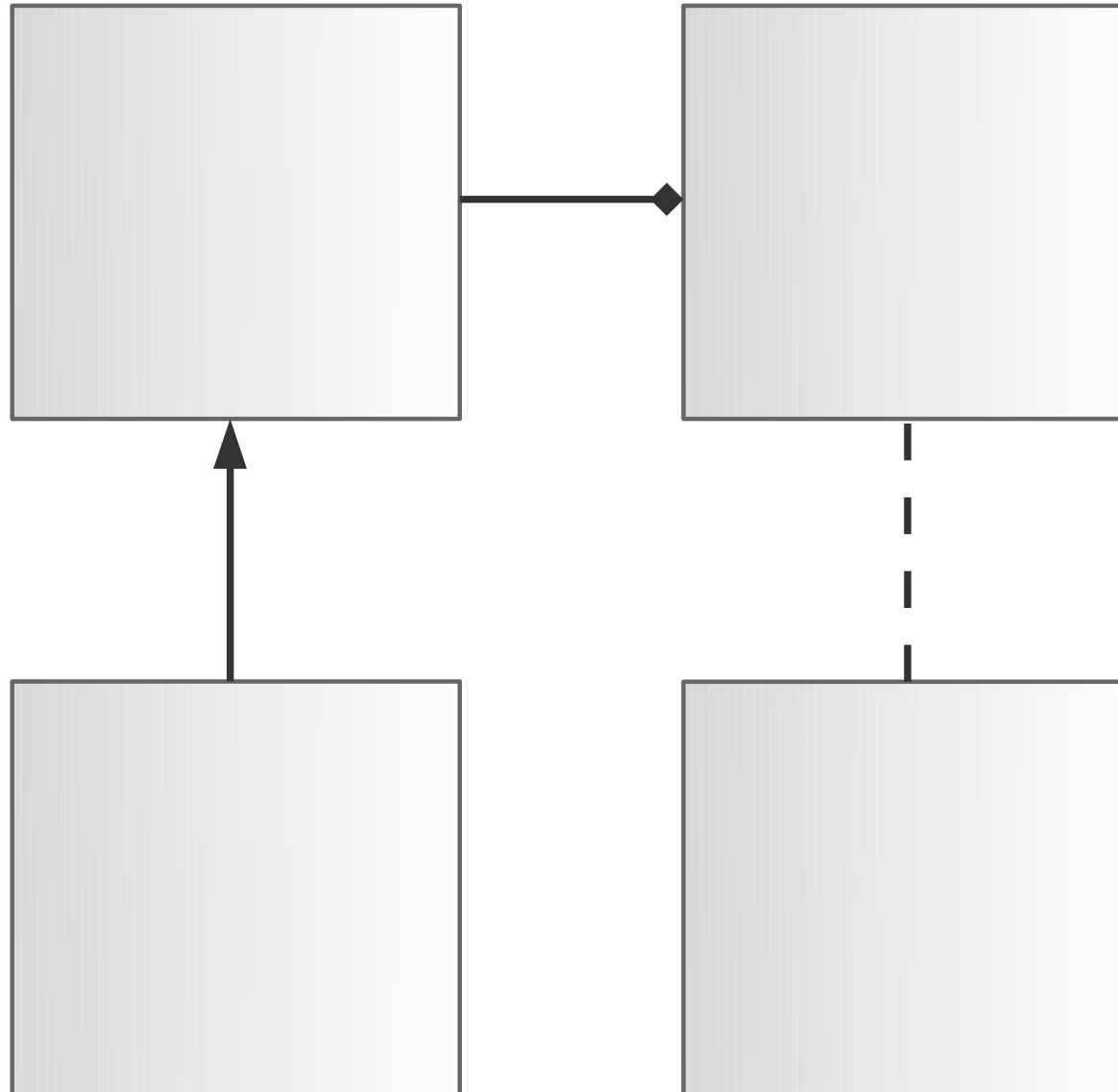
SpooFax

MPS



Hersteller:	JetBrains
Lizenz:	Apache 2.0
Released:	Q3/2009
Aktuelle Version:	2.5, 3.0 EAP
Hosting:	GitHub
Referenzen:	YouTrack, mbeddr

Language Workbenches



Language Workbenches

Sprachentwicklung

für eigene Projekte

technischer Art

domänenspezifischer Art

IDE

MPS: baseLanguage = Java

Java-IDE

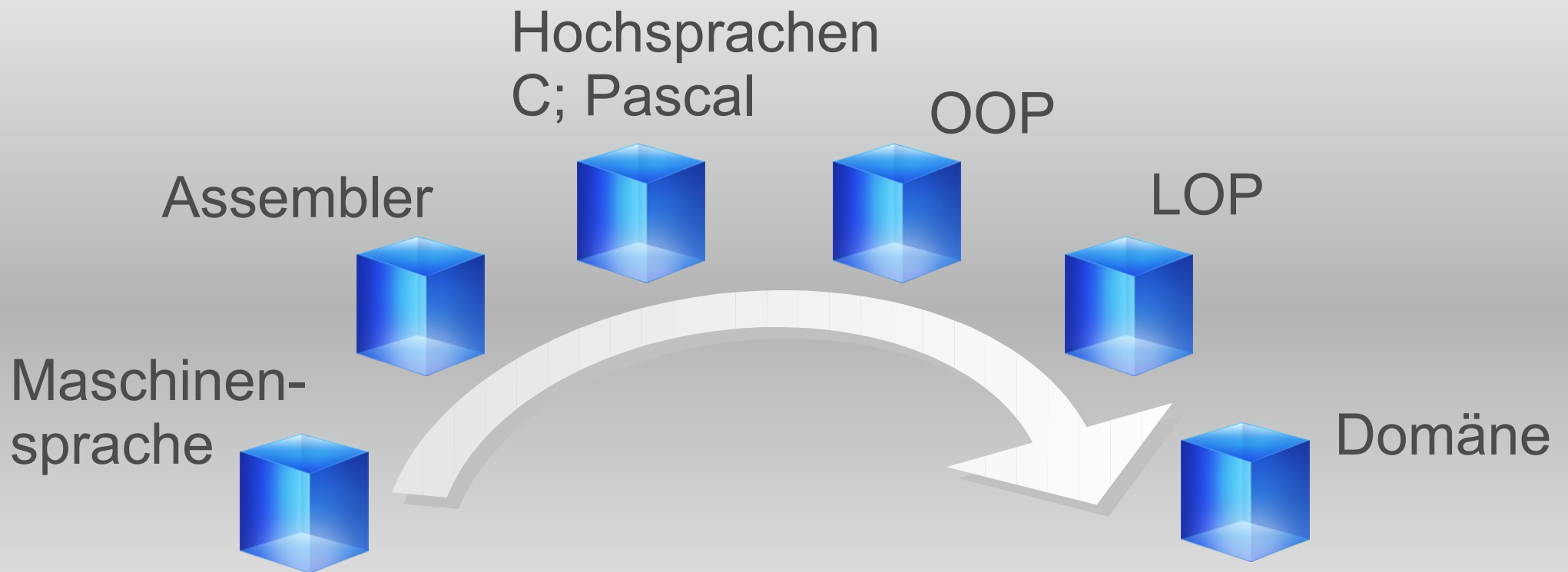
C: mbeddr (IDE++)

Applikation: Workbench als Grundlage

MDA/MDSD vs. Language Workbenches

Traditionell:

- teilweise modellieren
- konventionell vervollständigen
- Toolwechsel
- Medienbruch
- mangelhafte Integration
- One-Way
- Debugger? VCS? Refactorings?



1. Teil

Was ist eine Language Workbench?

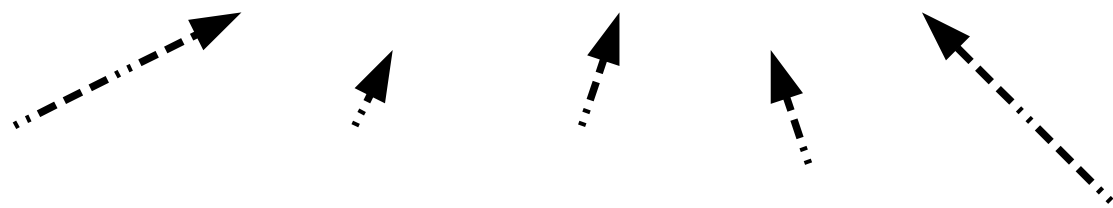
2. Teil

Beispiele aus der Praxis

Single Level of Abstraction

```
int/m/ x = Main.x();  
int/m/ a = x;  
int/m/ length = 1 m;  
int/s/ time = 2 s + 4 s;  
int/kg/ mass = 3 kg + 2 kg;  
int/m/ bla = 1 m - 2 m + 3 s;
```

}



```
private static int/m/ x() {  
    return 1 m;  
}
```

Currencies

USD

EUR

GBP

```
public static void main(string[] args) {  
    Money Savings = Main.getSavings();  
    Money wrong = 5 EUR + 6 USD;  
}
```

```
private static Money getSavings() {  
    return 3 EUR + 4 EUR + 5 EUR;  
}
```

`Coordinate[] neighbors = (cell +`

	left	middle	right
upper	(-1, 1)	(0, 1)	(1, 1)
middle	(-1, 0)	(0, 0)	(1, 0)
lower	(-1, -1)	(0, -1)	(1, -1)

`) - cell;`

`if (boolean Default: dead`

	<code>generation.contains(c)</code>	<code>!(generation.contains(c))</code>
<code>countAliveNeighbors(generation, c) < 2</code>	dead	dead
<code>countAliveNeighbors(generation, c) == 2</code>	alive	dead
<code>countAliveNeighbors(generation, c) == 3</code>	alive	alive
<code>countAliveNeighbors(generation, c) > 3</code>	dead	dead

`) {`
`nextGeneration.add(c);`
`}`

Principles of Verifiable Decision Table

Anforderungen verlinken: Reviews

```
public void run() {  
    int i = 3; -> int must be 3  
}
```


Formular

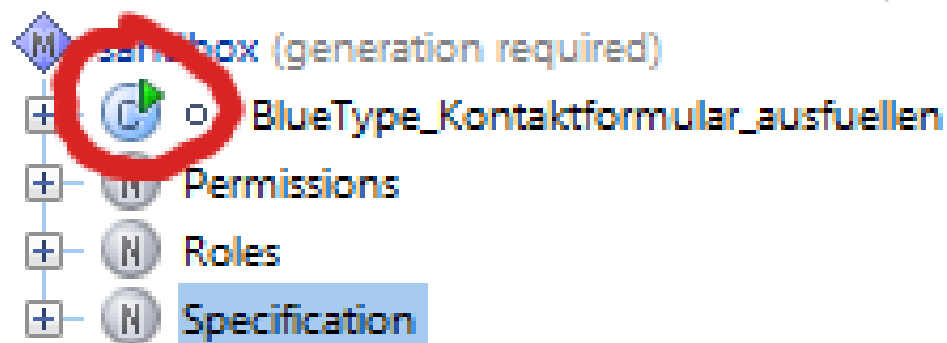
Requirements

Name	Neues Passwort festlegen
Art	Functional Requirement
Beschreibung	Das System muss dem Admin die Möglichkeit
Stabilität	absolut stabil
Verbindlichkeit	muss
Priorität	hoch
Details	bla bla bla
Andere Modellelemente:	BlueType_Kontaktformular_ausfuellen

Automatisierte Akzeptanztests
Fisurte Akzeptanztests
Testfälle und Integrationstests

Use Case

ID:		
Name:	BlueType_Kontaktformular_ausfuellen	
Goal:	Ein ausgefülltes Kontaktformular wird abgesendet.	
Actor:	Anonymer Webnutzer	
Steps:	Aktion	Erwartetes Ergebnis
	Open URL http://www.kontext-e.de Click link mehr... Click link Kontakt Fill form drop down box (name contains ContactList) = E-Mail drop down box (name contains TitleList) = Frau input (name contains SurnameBox) = Form input (name contains FirstnameBox) = Bot input (name contains FirmnameBox) = Kontext E GmbH input (name contains EmailBox) = info@kontext-e.de drop down box (name contains ConnectionList) = Berater click on radio button (name contains radioContact) = Ja click on radio button (name contains RadioNews) = Ja click on button name = SendButton	title= Unternehmen title= blueType - WebTop Publishing title= blueType - WebTop Publishing title= blueType - WebTop Publishing
Date:	09.12.2012	
Author:	Jens Nerche	
Version:	1	



Feature Set Mahnmails versenden

@gutfall

Feature Standardfall, Level 1: Der Kunde bekommt die Mahnungen per Emails.
Die Angabe <email/> direkt unter <debitor/> erst mal ignorieren.
HTML-Format ausprobieren, um Invoices als Tabelle darstellen zu können

@duplikat

Scenario: kein Debitor-Contact
see scenario Gemahnter hat keine Mailadresse

XML File for Scenario: standard

<<add a tag>>

Scenario: genau ein Debitor-Contact
given eine MFG-XML standard
and sendto = DEBTOR
when die XML zum inubit gesendet wird
then 1 Mail wurde gesendet
and der Empfänger ist email = j.nerche@kontext-e.de
and der Betreff ist emailsubject = ZAHLUNGSERINNERUNG
and der Inhalt enthält emailbodyline1 = SEHR GEEHRTER KUNDE,
and der Inhalt enthält emailbodyline4 = TABELLE GEKENNZEICHNETEN BETRAEGE ZU UEBERWEISEN.
and es existiert ein PDF-Anhang
and die Mahnung wird ins Archiv geschrieben

```

<contactlist>
  <contact>
    <gender>m</gender>
    <title></title>
    <name>Jens Nerche</name>
    <phone>88899922</phone>
    <fax>99</fax>
    <email>j.nerche@kontext-e.de</email>
    <functionkey_internaluse_only></functionkey_internaluse_only>
    <functionlookuplevel_internaluse_only></functionlookuplevel_internaluse_only>
  </contact>
  <contact>
    <gender>m</gender>
    <title></title>
    <name>Martin Halatchev</name>
    <phone>88899922</phone>
    <fax>99</fax>
    <email>m.halatchev@kontext-e.de</email>
    <functionkey_internaluse_only></functionkey_internaluse_only>
    <functionlookuplevel_internaluse_only></functionlookuplevel_internaluse_only>
  </contact>
  (--> zwei Debitor-Contacts )
</contactlist>

```

Loginseite JSF Style

```
<?xml version = "1.0" encoding = "default" standalone = "default" ?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html" xmlns:f="http://java.sun.com/jsf/core" xmlns:tim="http://java.sun.com/jsf/composite/tim">
  <h:head>
    <title>Anmeldung</title>
  </h:head>
  <h:body>
    <f:view>
      <div id="loginForm">
        <h:form>
          <div class="formTitle">
            <h:outputText value="Anmelden bei TiM (Build #{ sessionContext.buildVersion.version })" />
          </div>
          <div class="contentDetail">
            <tim:formInputText label="Benutzername" value="#{loginController.username}" required="true" focus="true">
              <f:validateLength minimum="2" for="input" />
            </tim:formInputText>
            <tim:formInputSecret label="#{loginController.passwordLabel}" value="#{loginController.password}" required="true" />
          </div>
          <div class="actions">
            <tim:imageTextButton label="Anmelden" image="lightning.png" action="#{loginController.authenticate}" />
          </div>
          <h:messages globalOnly="true" styleClass="formMessage" />
        </h:form>
      </div>
    </f:view>
  </h:body>
</html>
```

Loginseite DSL Style

Page Anmeldung

Form loginForm **with title** Anmelden bei TiM

Input Fields

| **FormInutText** Benutzername

FormInputSecret Passwort

Actions

ImageTextButton Anmelden

Sign Up | Login | Forgot Password | Home

Zustandsmaschine

State Machine: Projektstatemachine

nicht existent (start) :
anlegen -> in Vorbereitung

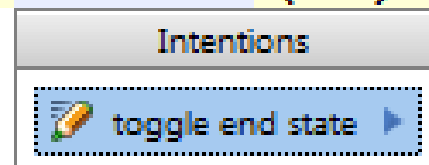
in Vorbereitung : Statische Codeanalyse
starten -> aktiv
starten -> in Gewährleistung

aktiv :
auf Gewährleistung setzen -> in Gewährleistung
abschließen -> abgeschlossen

in Gewährleistung :
abschließen -> abgeschlossen

abgeschlossen :
archivieren -> archiviert

archiviert (end)



Datenmodell der Domäne

Aggregat Projekt

Domain Object Projekt

Interner Name

0..1 alphanumeric default: 1234 length >= 3
length <= 200
unique

Informal: Weitere Informationen usw.

Projektname	1 alphanumeric
Projektbeschreibung	0..1 alphanumeric
Projektnummer	0..1 alphanumeric
Kunde	1 Kunde
Kundenprojektnummer	0..1 alphanumeric
Kundenansprechpartner	0..1 Kundenansprechpartner
Projektleiter	1 Mitarbeiter
Fakturierungsart	1 Fakturierungsarten
Sollstunden	0..1 numeric default: 0 if Fakturierungsart = Festpreis then length >= 1
Termine	0..1 Termine
Stundennachweise	0..1 yes/no default: false
Welche Stundennachweise	0..1 alphanumeric default: abc
Geschätzter Restaufwand	0..1 alphanumeric
Bei Gewährleistungsende erinnern	0..1 yes/no default: true
Aktivitäten	0..n Aktivitäten
Leistungsarten	0..n Leistungsarten
Projektstatus	1 Projektstatemachine

Datenmodell der Domäne



Anforderungsermittlung

= Modellierung

= Code

= DB-Schemadefinition

= GUI



Kein Scaffolding



Ein- und Ausblenden
von Details

Datenmodell der Domäne



Enumeration



Projektstatus: Verweis
auf Zustandsmaschine



Erklärende
informale Texte



Validierung



Navigation zu anderen Entitäten, Enumerationen,
Zustandsmaschine, Find Usages

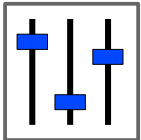
Weiter Möglichkeiten



Buttons, Bilder, Swing



Tools (z.B. FOP-Preview)



L10N (auch Keywords!)



Assessments: Komplexe Prüfungen über
die gesamte Codebasis

Einsatzgebiete

Anforderungserhebung und -verwaltung

Akzeptanztests

Featuremodellierung - Produktlinien

Schnittstellendefinition

Protokolldefinition

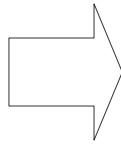
Modellierung - „Entwurf“

Design, Implementierung,

Test

Tool-/Vendorabhängigkeit

Ja

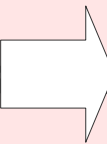


Aber

Open Source

Offen für Erweiterungen
und Anpassungen

Nachteile – Wirklich?

Einarbeitungszeit  und Bleib am Ball

Design betrifft nicht nur UML, sondern auch Sprachmöglichkeiten

Kann in Brownfieldprojekten knifflig sein, wenn kein Importer zur Verfügung steht  Oder man nutzt die Gelegenheit, etwas neues, saubers zu erstellen

Mehr Know How nötig, Mehr Möglichkeiten für schwieriger zu rekrutieren Personal

Zusammenfassung



- Keep it simple, stupid
- Don't Repeat Yourself
- Separation of Concerns
- Single Level of Abstraction
- Single Responsibility Principle
- Information Hiding Principle

Zusammenfassung



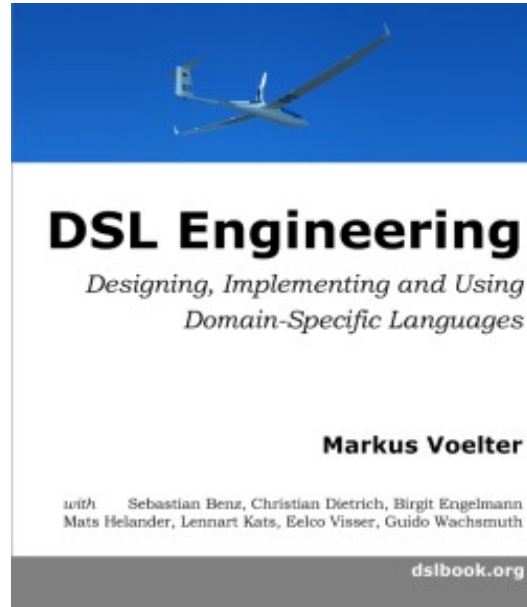
- Principle of Least Astonishment
- Automatisierte Integrations- und Akzeptanztests
- Test first
- Reviews
- Statische Codeanalyse
- Source Code Konventionen

Zusammenfassung



- Ausdruckskraft
- Abstraktionsstufe
- Domänenspezifisch
- Korrektheit
- Evolvierbarkeit
- Produktionseffizienz
- Kontinuierliche Verbesserung

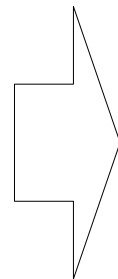
Wie weiter?



Workshop:

Xtext

<http://dslbook.org/>



Tutorials, Demos, Blog,
Online-Doku,
jetbrains.tv, YouTube

Jens Nerche

Leiter Anwendungsentwicklung

Kontext E GmbH

www.kontext-e.de



Email j.nerche@kontext-e.de

Twitter [@jensnerche](https://twitter.com/jensnerche)

Blog <http://wp.me/p3leb2-Z>

Slides <http://slideshare...>

Samples <https://github.com/jensnerche>

Feedback <http://speakerrate.com/jensnerche>



Antworten

&



Fragen

Fragen und Antworten

Language Workbenches

- Smart Client zum Testen
 - Schnittstellen und Protokolle
 - Code Completion, Checks, Intentions
- Applikation
- Simulation und Stubbing für nicht verfügbare Systeme, Hardware,...
- Interpreter