



DSL w środowisku Eclipse

Grzegorz Białek
Architekt techniczny, Sygnity S.A.

Agenda

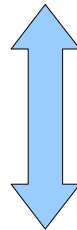
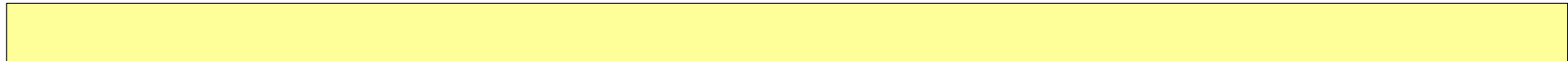
- Wstęp do tematu (10 min)
 - Sens tworzenia języków biznesowych
 - UML jako język biznesu?
 - Zintegrowane środowisko deweloperskie
 - DSL IDE
- Eclipse przegląd technologii (20 min),
- Demo... (5 min)

Co to jest DSL?

- Domain Specific Language
(Język specyficzny dla domeny/biznesu)
- Wyzwania:
 - Podniesienie poziomu abstrakcji,
 - Próba formalnego opisu problemu i rozwiązania,
 - Modele zamiast dokumentacji stanowią kod źródłowy,

Poziom abstrakcji

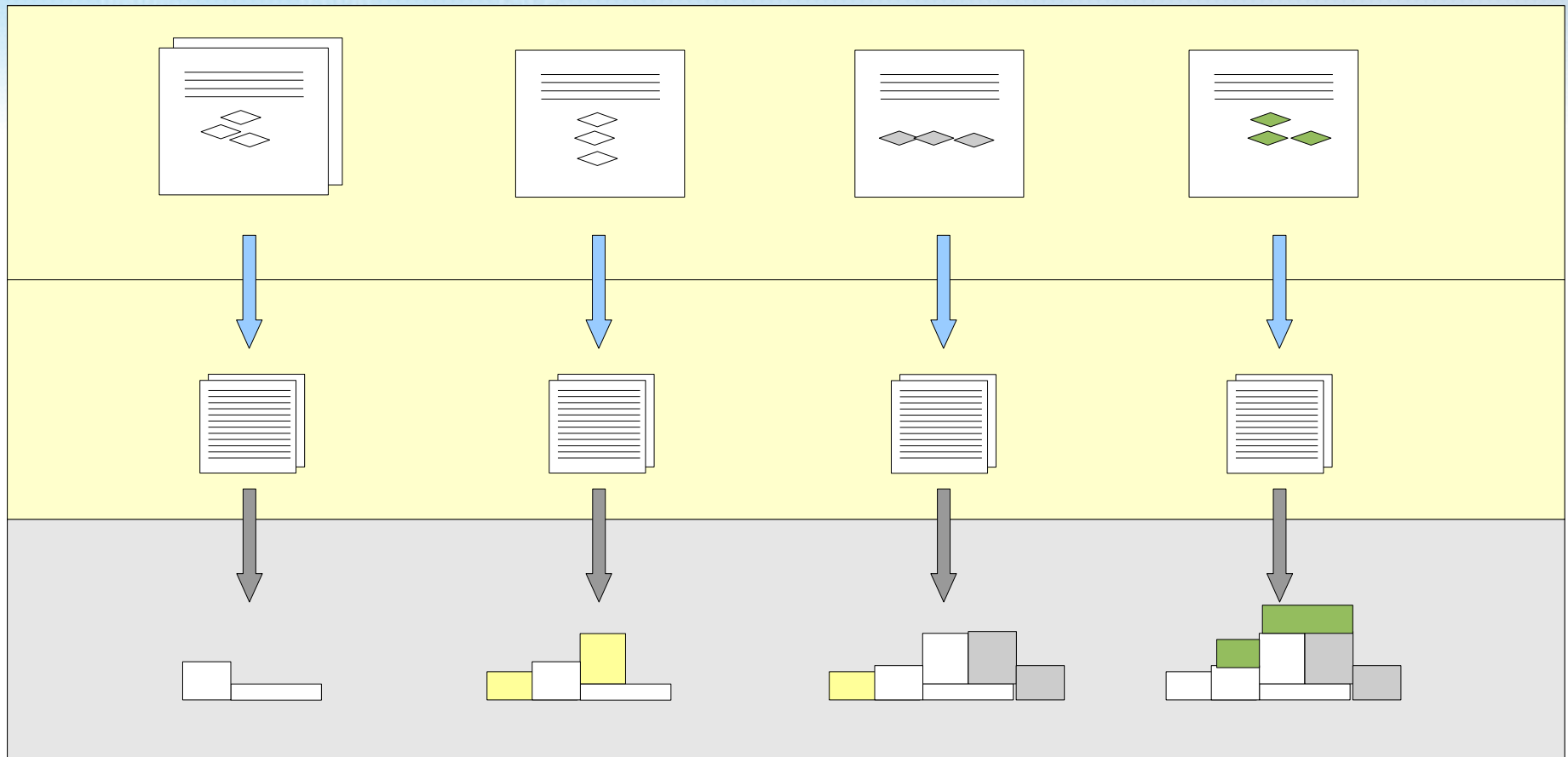
Wymagania biznesowe (przestrzeń problemu)



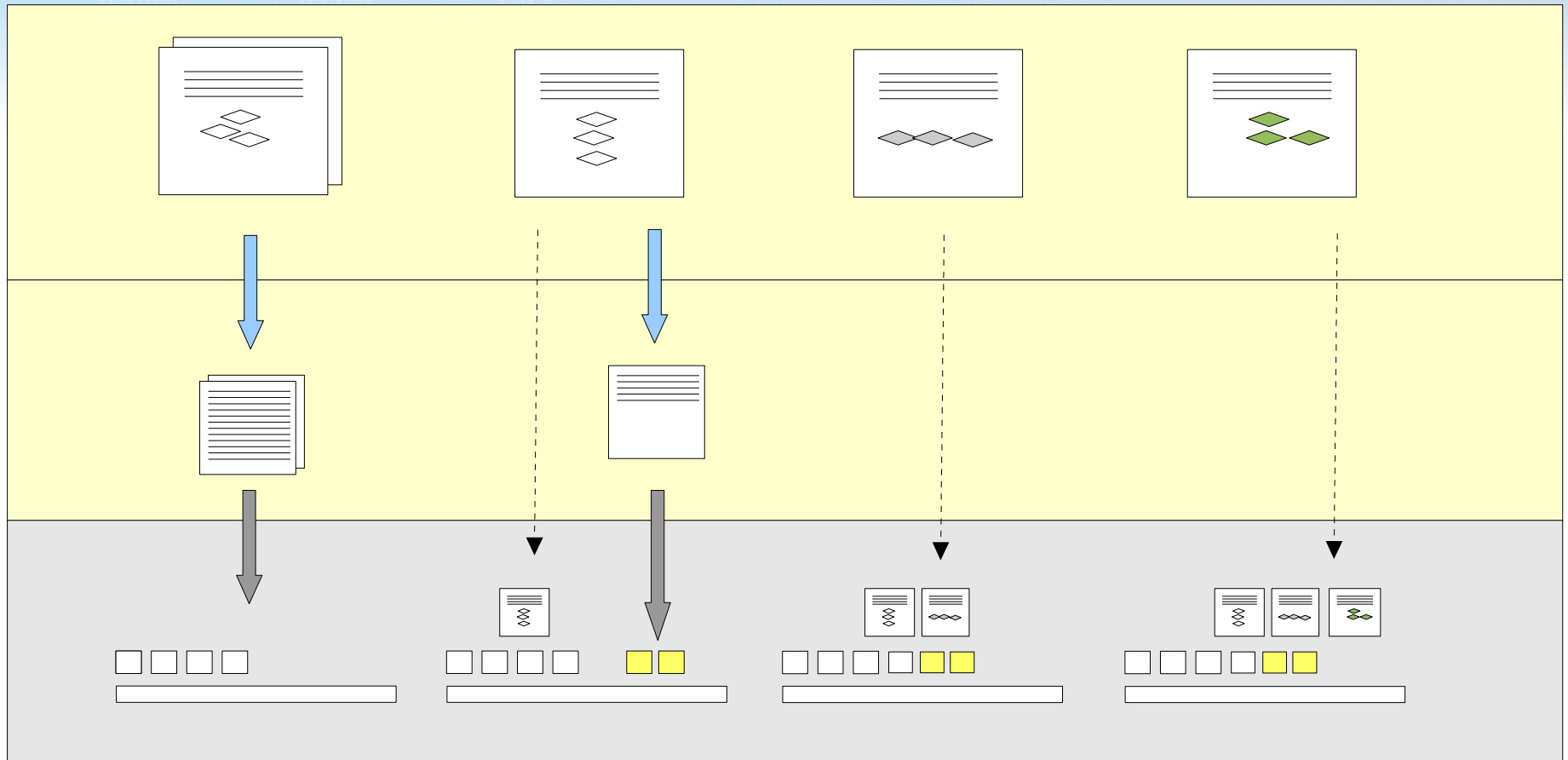
Platforma techniczna (przestrzeń rozwiązania)



Biznes → Programista → System



Biznes → System



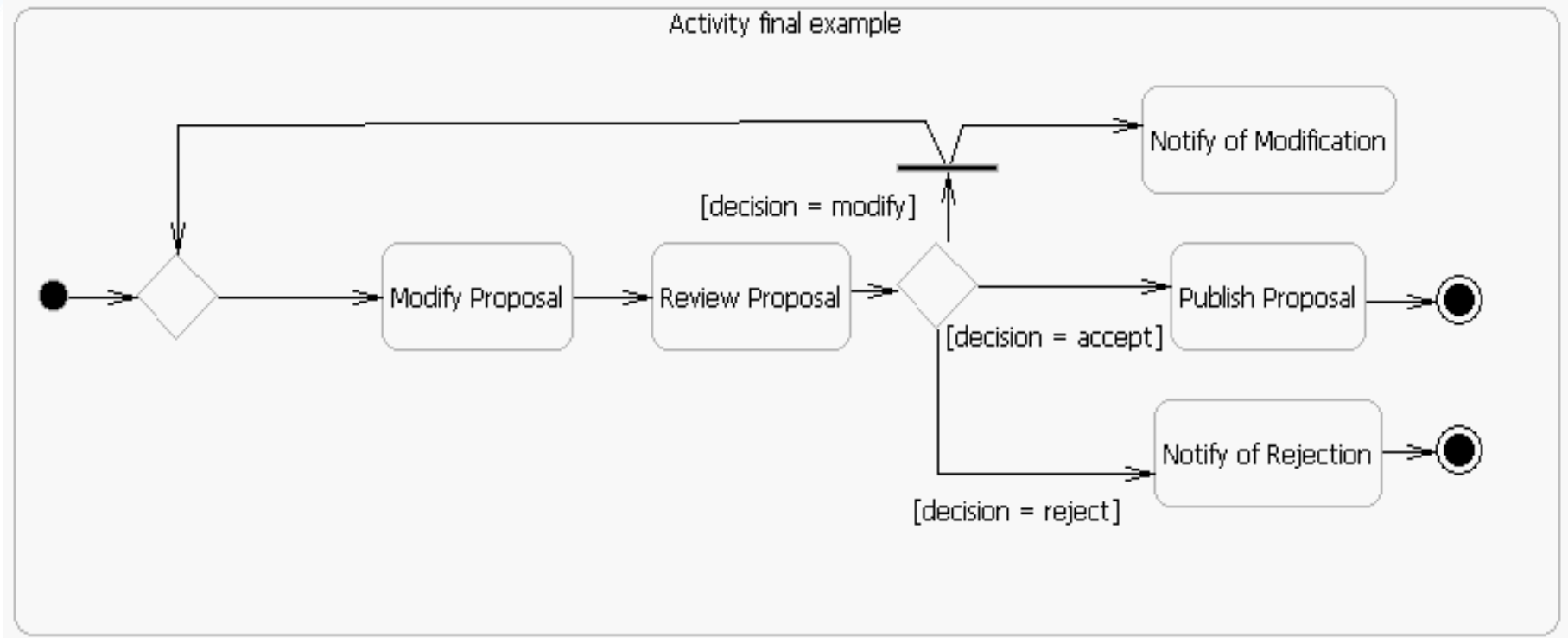
Język biznesowy

- Bardziej **zrozumiały dla biznesu**,
 - Opis w przestrzeni problemu,
 - Wykorzystanie znanych i popularnych notacji,
- Język **formalny**
 - Wyrażenia stanowiąc będą kod źródłowy
 - Przetwarzanie mechaniczne

Platforma techniczna

- **Każdy system** posiada **własny język**,
 - Przypadek użycia jako 'zdanie'
- Tworzenie języka poprzez opis składników
 - Frameworki, biblioteki oraz ich konfiguracja posiadają własne języki formalne

UML jako język biznesu (MDA)



UML jako język biznesu (MDA)

- Zalety:
 - Standard OMG,
 - Gotowe narzędzia,
 - Kilka typów diagramów,
 - Teoretyczna możliwość modyfikacji za pomocą profili
- No dobra ale **to się nie udało...** Dlaczego?

Środowisko zintegrowane

- Środowisko dostosowane do problemu,
- Wsparcie użytkownika
 - Walidacja poprawności
 - Ułatwienie edycji / uzupełnianie treści
 - Praca grupowa
- Integracja i współpraca wielu (różnych) elementów

UML | środowiska zintegrowane

- ~~Środowisko dostosowane do problemu,~~
- ~~Wsparcie użytkownika~~
 - ~~Walidacja poprawności~~
 - ~~Ułatwienie edycji / uzupełnianie treści~~
 - ~~Praca grupowa~~
- ~~Integracja i współpraca wielu (różnych) elementów~~

DSL IDE jako rozwiązanie

- Eclipse JDT kontra MS Word
 - Które środowisko jest bardziej wszechstronne
 - Które środowisko jest wygodniejsze (do jakich zadań)
- **Tylko** narzędzia optymalnie dopasowane do problemu mogą pomóc **wydajnie go rozwiązać**

Agenda

- Wstęp do tematu (10 min)
- **Eclipse przegląd technologii (20 min),**
 - Modelowanie (EMF, GEF/GMF, Antlr...)
 - Użytkowanie (EMF VF|MT, Compare...)
 - Transformacje (M2T, M2M)
 - Rozbudowa (modularyzacja, integracja...)
- Demo... (5 min)

Modelowanie języka

- Utworzenie modelu bazowego (meta-modelu)
- Wybór notacji języka,
 - Graficzna (GEF / GMF)
 - Tekstowa (Antlr, xText)
 - Strukturalna (tabele, drzewa etc. JFace)

EMF

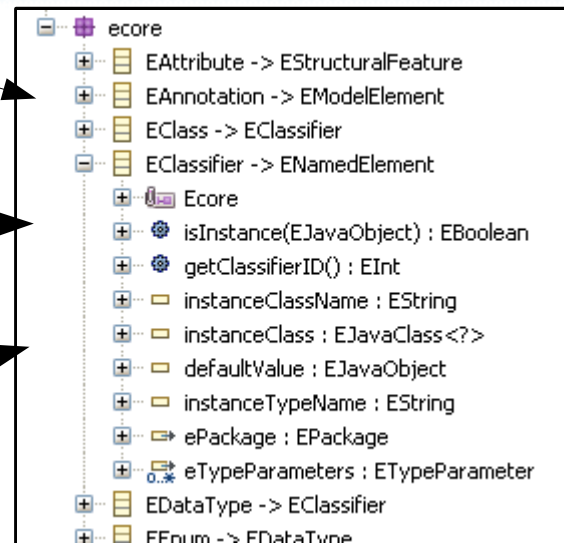
- Model bazowy twojego języka (meta-model),
- Baza dla większości edytorów w Eclipse,
- Śledzenie i modyfikacje (Edit)
- Trwałość (XMI, XML...)
- Model Ecore (uniwersalny meta-meta-model),
- Model XSD

EMF

Bezpośrednia edycja

Odwzorowanie XSD

Import: UML2, MDL, Java...

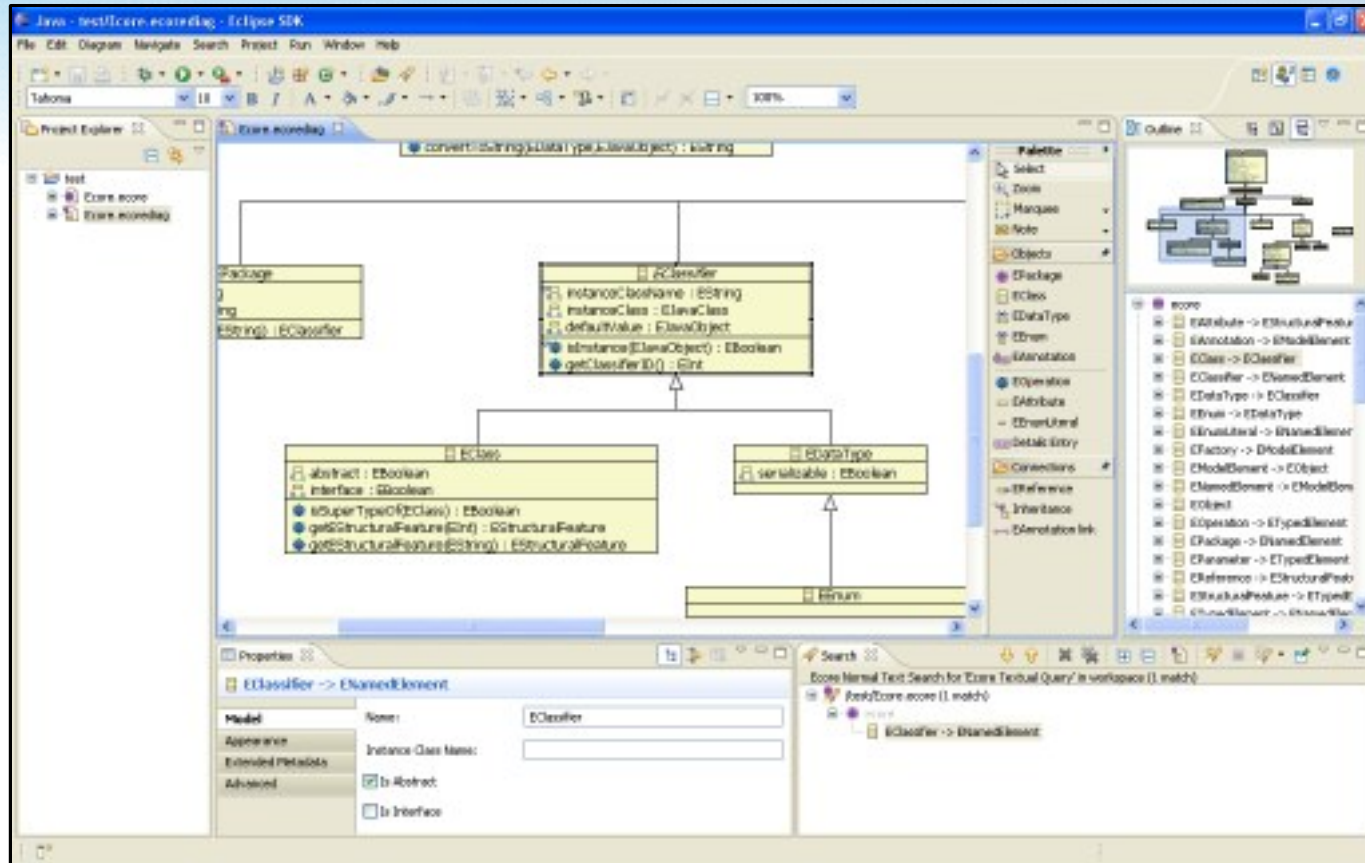


Model

Edit

Edytor (czy warto?)

EMFT Ecore Tools



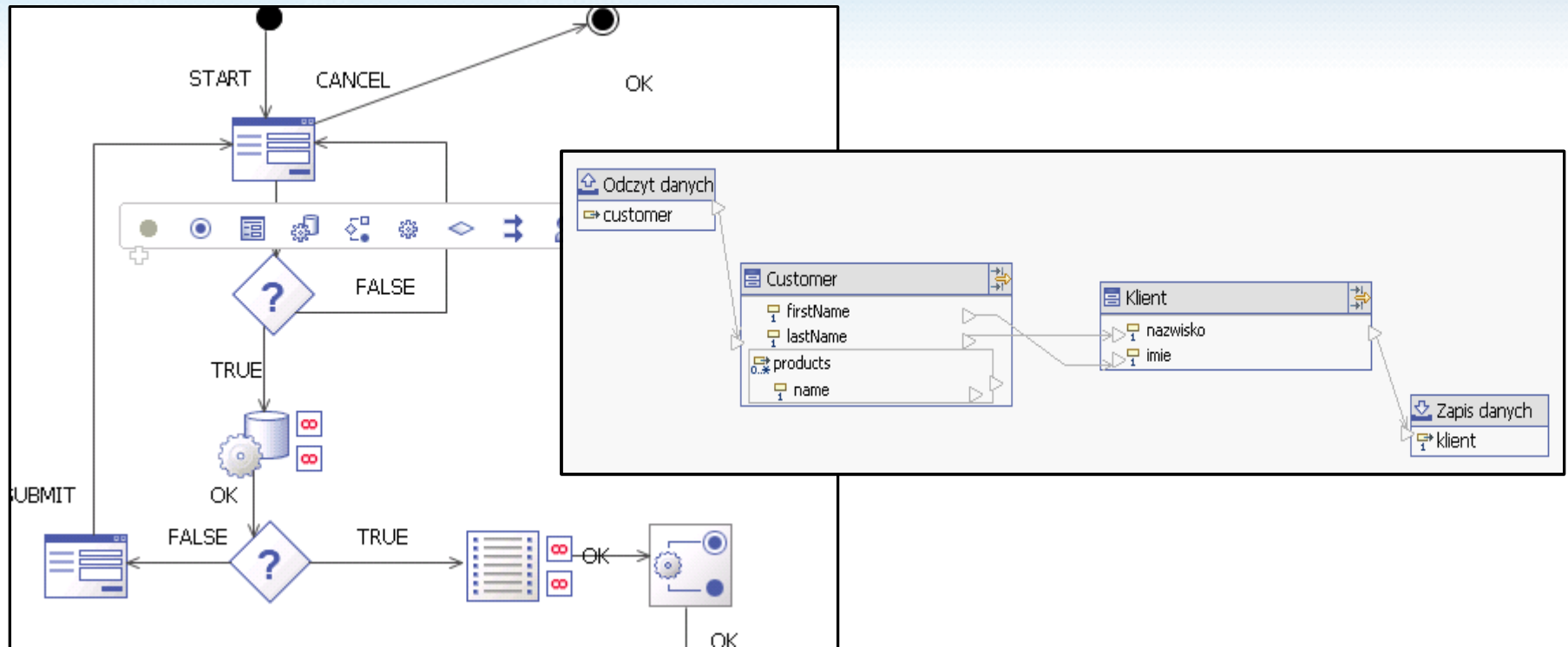
EMFT Ecore Tools

- Doskonałe narzędzie od dokumentacji modelu
 - Wiele diagramów dla jednego modelu

Notacje języka

- Cechy poszczególnych notacji
- Notacja **graficzna**
 - Grafy, kompozycja etc.
- Notacja **tekstowa**
 - Wyrażenia, proste struktury etc.
- Notacja **strukturalna**
 - Tabele (np. tabele decyzyjne), drzewa etc.

Notacje graficzne



Notacje graficzne

- Biblioteki **GEF / GMF**
- GEF jako nakładka na Draw2D
- GMF jako nakładka łącząca GEF
- GMF
 - Uniwersalny diagram + Model specyficzny
 - Biblioteka usług podstawowych
 - Rozbudowa poprzez rozszerzenia

Notacje tekstowe

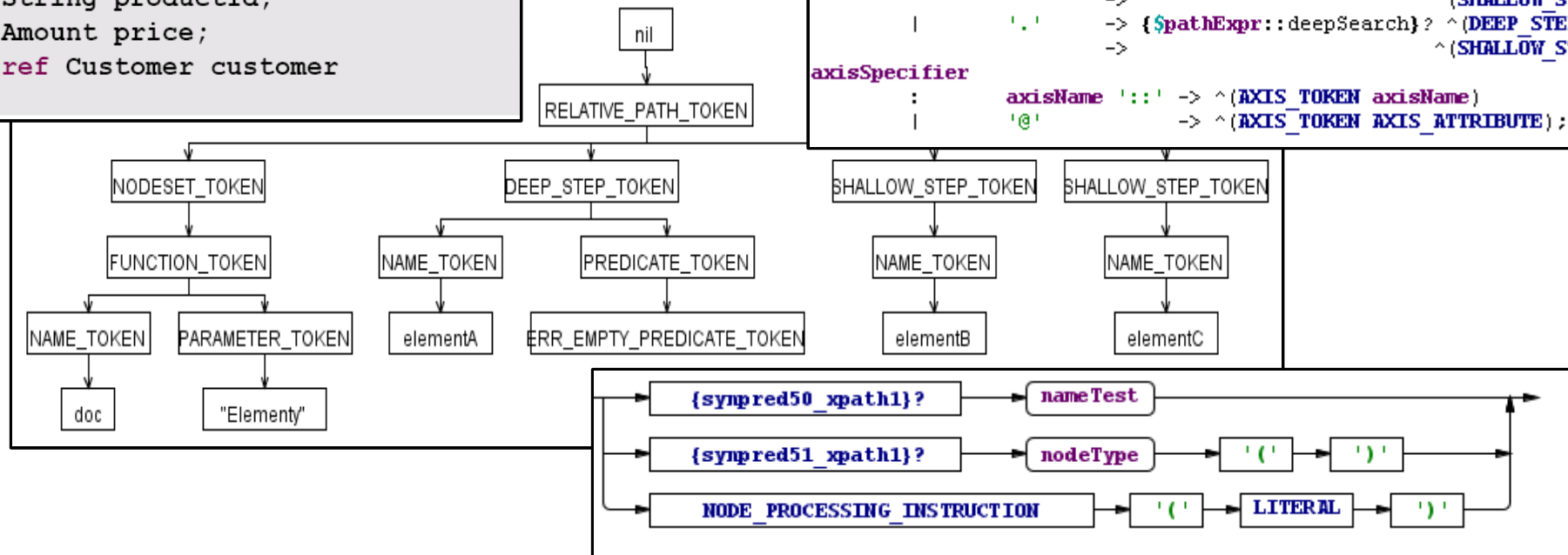
```
entity Customer {
  String name;
  ref +Order orders <-> customer
}

entity Order {
  String productId;
  Amount price;
  ref Customer customer
}
```

```
relativeLocationPath
  :      step (('///'! {$pathExpr::deepSearch=true;} |
             '//'! {$pathExpr::deepSearch=false;}) ste

step
  :      axisSpecifier? nodeTest predicate*
      -> {$pathExpr::deepSearch}? ^{(DEEP_STEP
      ->                               ^{(SHALLOW_STEP
      |      '...' -> {$pathExpr::deepSearch}? ^{(DEEP_STEP
      ->                               ^{(SHALLOW_STEP
      |      '..' -> {$pathExpr::deepSearch}? ^{(DEEP_STEP
      ->                               ^{(SHALLOW_STEP

axisSpecifier
  :      axisName '::' -> ^{(AXIS_TOKEN axisName)
  |      '@' -> ^{(AXIS_TOKEN AXIS_ATTRIBUTE);
```



Notacje tekstowe

- Proces zdecydowanie **bardziej złożony**
 - Edycja (Eclipse Text Framework)
 - Parsowanie tekstu
 - Budowa drzewa **AST**
 - Wizytacja i budowa modelu

Notacje tekstowe

- **ANTLR**
 - Wydajny parser LL(*)
 - Dobre narzędzia z debuggerem etc.
 - Idealny przy zaawansowanych problemach
- **xText** (nakładka na ANTLR)
 - Automatyczna generacja kodu Eclipse
 - Raczej proste gramatyki (model generowany)
- TEF, IMP...

Notacje strukturalne

JobListeners

- JobListeners()
- IListenerDoit
 - aboutToRun : IListenerDoit
 - awake : IListenerDoit
 - done : IListenerDoit
 - running : IListenerDoit
 - scheduled : IListenerDoit
 - sleeping : IListenerDoit
 - global : ListenerList
- newEvent(Job)
- newEvent(Job, IStatus)
- newEvent(Job, long)
- doNotify(IListenerDoit, IJobChangeEvent)
- handleException(Object, Throwable)
- add(IJobChangeListener)
- remove(IJobChangeListener)
- aboutToRun(Job)
- awake(Job)
- done(Job, IStatus, boolean)
- running(Job)
- scheduled(Job, long, boolean)
- sleeping(Job)
- scheduled(Job, long, boolean)

ecore

- EAttribute -> EStructuralFeature
- EAnnotation -> EModelElement
- EClass -> EClassifier
- EClassifier -> ENamedElement
 - Ecore
 - isInstance(EJavaObject) : EBoolean
 - getClassifierID() : EInt
 - instanceClassName : EString
 - instanceClass : EJavaClass<?>
 - defaultValue : EJavaObject
 - instanceTypeName : EString
 - ePackage : EPackage
 - eTypeParameters : ETypeParameter
- EDataType -> EClassifier
- EEnum -> EDataType

Typ	Parametr				
workflow::IN	<input type="checkbox"/>				
workflow::OUT	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customer	<input checked="" type="checkbox"/>				<input type="checkbox"/>

Notacje strukturalne

- Idealny łącznik / uzupełnienie pozostałych notacji
- Proste API dostarczane z platformą (JFace)
- Wsparcie ze strony EMF'a w prostych przypadkach

Agenda

- Wstęp do tematu (10 min)
- Eclipse przegląd technologii (20 min),
 - Modelowanie (EMF, GEF/GMF, Antlr...)
 - **Użytkowanie (EMF VF|MT, Compare...)**
 - Transformacje (M2T, M2M)
 - Rozbudowa (modularyzacja, integracja...)
- Demo... (5 min)

Wsparcie dla użytkownika

- Odwzorowanie problemu
- Ułatwienia edycji / podpowiedzi etc.
 - Mechanizmy modelu, kreatory, podpowiadacze
- Walidacja poprawności
 - EValidator, EMF Validation Framework
- Praca grupowa
 - Notacje, EMF Compare

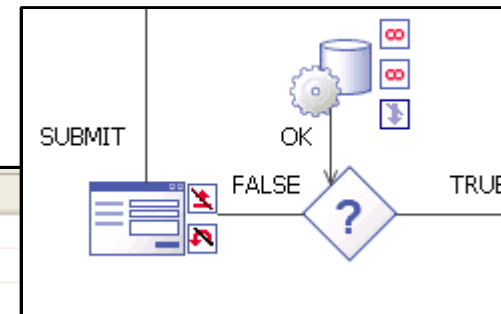
EMF Model Transaction

- Ułatwienie tworzenia poleceń (gestów) atomowych pomiędzy API platformy, a GMF/EMF (3 rodzaje komend)
- Umożliwienie synchronizacji równoległego dostępu do modelu

EMF Validation Framework

- Rozproszona walidacja
 - Walidatory dodawane za pomocą rozszerzeń
- Tryby: Batch, Live (blokada komendy)

Description	Resource
Errors (12 items)	
✘ [3001] Należy zdefiniować przepływ wyjściowy [ERROR] dla elementu	Flow1.webflow
✘ [3001] Należy zdefiniować przepływ wyjściowy [OK] dla elementu	Flow1.webflow
✘ [3001] Należy zdefiniować przepływ wyjściowy [TRUE] dla elementu	Flow1.webflow
✘ [3002] Parametr [customer] sygnału [IN] nie został związany z danymi	Flow1.webflow
✘ [3002] Parametr [customer] sygnału [IN] nie został związany z danymi	Flow1.webflow
✘ [3002] Parametr [customer] sygnału [IN] nie został związany z danymi	Flow1.webflow



Praca grupowa

- Notacja tekstowa
 - Uniknięcie problemów kosztem łatwości obsługi (drogi kompromis)
- Porównywanie modeli
 - Porównanie graficzne i strukturalne
 - EMF Compare

EMF Compare

- Uniwersalny mechanizm porównywania modeli z możliwością dostosowywania

The screenshot displays the Eclipse EMF Compare interface. On the left, two XML files are compared side-by-side. The right pane shows a graphical comparison of the models, with elements like 'pole A', 'pole B', and 'Pole C' represented as input fields. A tree view in the center shows the class hierarchy, including 'ToolGroup', 'ToolGroupItem', and 'Separator'. A detailed view of the 'ToolGroup' class is shown on the right, listing attributes like 'isPhantomElement()', 'metaClass', and 'containmentMetaFeature'.

Agenda

- Wstęp do tematu (10 min)
- Eclipse przegląd technologii (20 min),
 - Modelowanie (EMF, GEF/GMF, Antlr...)
 - Użytkowanie (EMF VF|MT, Compare...)
 - **Transformacje (M2T, M2M)**
 - Rozbudowa (modularyzacja, integracja...)
- Demo... (5 min)

Model → Text

- Standardowa transformacja za pomocą szablonów
- Zmiana modeli w pliki wymagane przez platformę docelową
- Kilkanaście generatorów:
 - Jet, xPand, Acceleo...

```
«DEFINE Root FOR data::DataModel»
  «EXPAND Entity FOREACH entity»
«ENDDDEFINE»

«DEFINE Entity FOR data::Entity»
  «FILE name + ".java"»
  public class <name> {
    «FOREACH attribute AS a»
      // bad practice
      private <a.type> <a.name>;
    «ENDFOREACH»
  }
«ENDFILE»
«ENDDDEFINE»
```


Model → Model

- Zdecydowanie mniej popularne, raczej przyszłość niż terażniejszość
- **Ganymede** dostarcza dwa silniki:
 - Implementację **QVT Operational**
 - **ATL 2.0**

```
query UML::Package::package2schemas() : OrderedSet(RDB::Schema) {
    self.map package2schema()->asSequence()->
        union(self.getSubpackages()->collect(package2schemas()))->asOrderedSet()
}

mapping UML::Package::package2schema() : RDB::Schema
when { self.hasPersistentClasses() }
{
    name := self.name;
    elements := self.ownedElements->select(oclIsKindOf(UML::Class))->
        collect(oclAsType(UML::Class).map persistentClass2table())->asOrderedSet()
}
```

Agenda

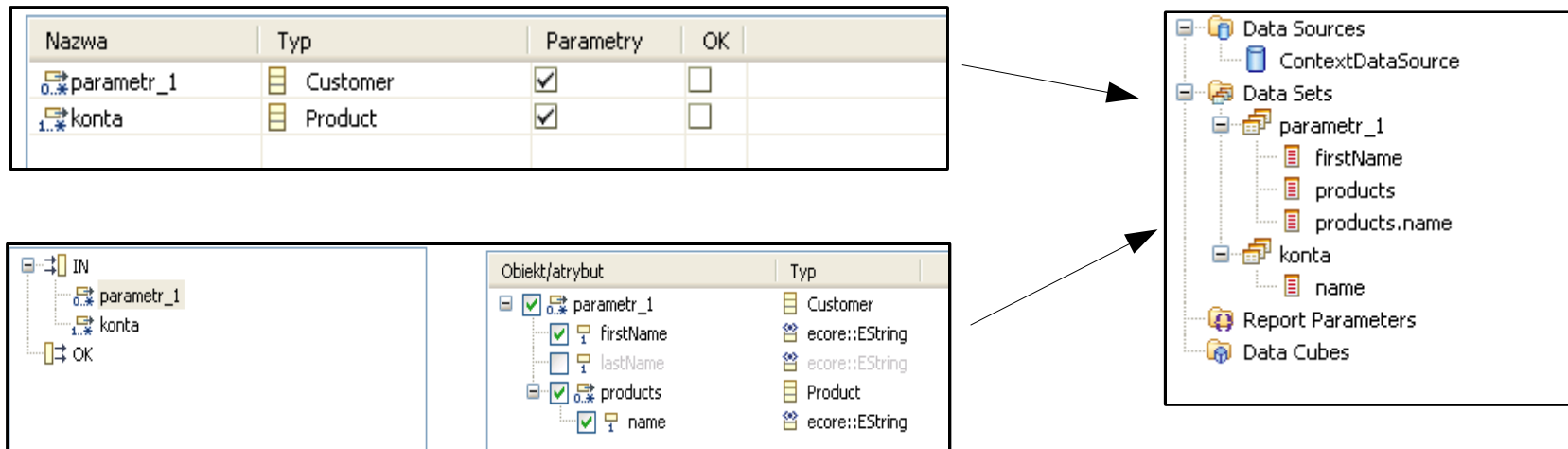
- Wstęp do tematu (10 min)
- Eclipse przegląd technologii (20 min),
 - Modelowanie (EMF, GEF/GMF, Antlr...)
 - Użytkowanie (EMF VF|MT, Compare...)
 - Transformacje (M2T, M2M)
 - **Rozbudowa (modularyzacja, integracja...)**
- Demo... (5 min)

Modularyzacja / Integracja

- **Integracja** z istniejącymi edytorami / notacjami etc.
- **Modularność**
- Microsoft **Software Factories**

Przykład integracji

- Przykład zintegrowanego edytora → BIRT
 - Silna integracja (synchronizacja modeli wewn.)
 - Zastąpienie kodu językiem modelowania



Demo...

Pytania?

Grzegorz Białek
gbialek@sygnity.pl

Zainteresowanych tematyką zapraszam na:
<http://deepdiveinto.blogspot.com>

Prowadzisz bloga lub stronę o Eclipse?
http://wiki.eclipse.org/Regional_Communities/Poland