

Глубокое мета моделирование

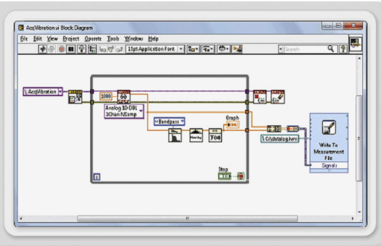
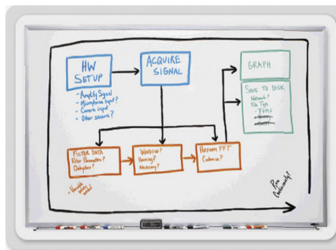
Ю.В. Литвинов

y.litvinov@spbu.ru

???

О визуальных языках

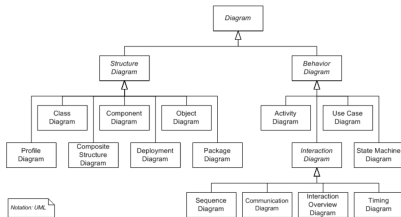
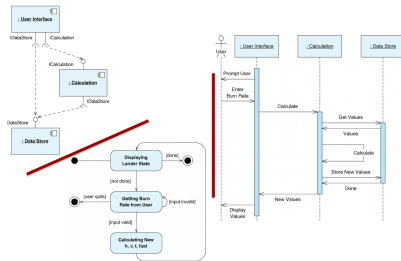
- ▶ Модели – неформальные или формальные
- ▶ Языки моделирования: UML, IDEFx, BPMN, ...
- ▶ Языки визуального программирования: Simulink, LabView, ...
- ▶ Предметно-ориентированные визуальные языки: TRIK Studio, Robolab, Node-RED, ...



© N. Medvidovic

Язык UML

- ▶ Самый известный визуальный язык
- ▶ Появился в середине 90-х
- ▶ Не язык, а набор языков
- ▶ 14 разных видов диаграмм
- ▶ Единое описание, общее “ядро” языка
- ▶ Плохо с семантикой
- ▶ Использует метамоделирование для задания синтаксиса

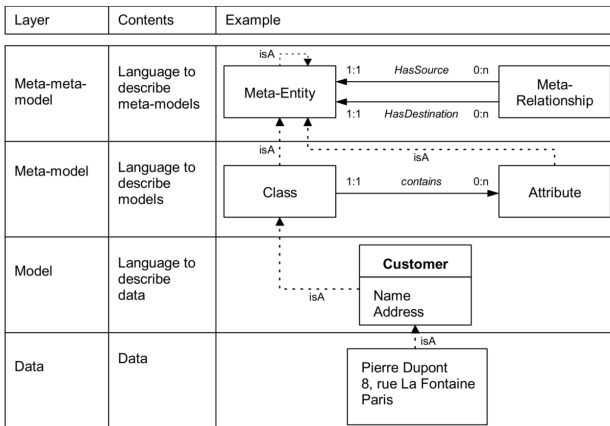


Метамоделирование

Уровни моделирования	Языковые средства	Пример
Предметная область	Нет	Каталог фильмов
Модель	Визуальный язык	<p>Диаграмма классов</p> <pre> classDiagram class Film { -Название : String -Год : int -Жанр : String } class Режиссёр { -Имя : String -Фамилия : String -Год рождения : int } Film -- Режиссёр </pre>
Метамодель	Метаязык	<p>Метамодель диаграммы классов</p> <pre> classDiagram class Класс { -Имя : String } class Атрибут { -Имя : String -Тип : String } class Ассоциация Класс "1" -- "*" Ассоциация Класс "2" -- "1" Ассоциация </pre>
Метаметамодель	Метаязык	<p>Метамодель метаязыка</p> <pre> classDiagram class Узел { -Имя : String } class Атрибут { -Имя : String -Тип : String } class Роль { -Множественность : String } class Связь Узел "1" -- "*" Атрибут Узел "1" -- "1" Роль Роль "2" -- "1" Связь </pre>

Неоднозначность толкования “instanceOf”

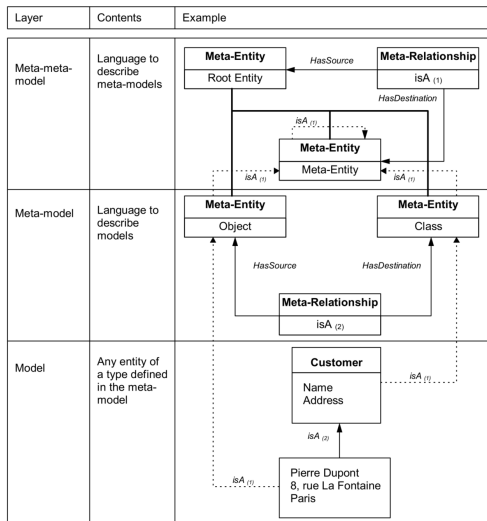
Пример



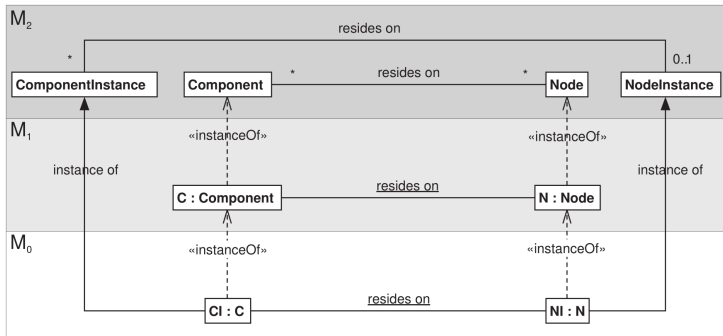
© J. Bezivin et al., Ontology-Based Layered Semantics for Precise OA&D Modeling, 1997

Неоднозначность толкования “instanceOf”

Собственно проблема

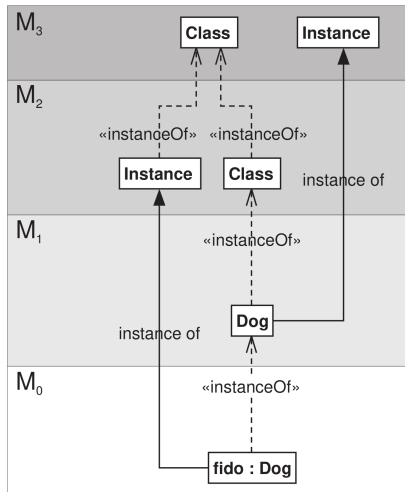


Ещё один пример



© C. Atkinson, Th. Kuhne, The Essence of Multilevel Metamodeling, 2001

Умножение сущностей



© C. Atkinson, Th. Kuhne, The Essence of Multilevel Metamodeling, 2001

“Пользовательский уровень”, подпрограммы

The screenshot shows the 'Robot's Behaviour Diagram' editor. The main workspace contains a sequence of actions: a start node, three function blocks labeled 'f', a 'New Food' block, a 'Go!' block, and an end node. A callout box above the first 'f' block says: 'This is "pacman" game demo. Play it in 2D model using controller buttons in display emulator or on real robot using gamepad or controller buttons.' Another callout box above the 'f' blocks says: 'Global constants.' The right sidebar contains a 'Редактор свойств' (Properties Editor) with a table of properties, a 'Палитра' (Palette) with various blocks categorized under 'Алгоритмы' (Algorithms) and 'Подпрограммы' (Subprograms), and a 'Редактор сенсоров' (Sensor Editor) button. The bottom status bar indicates 'Режим редактирования - нажмите Ctrl+2 или кликните здесь для переключения в режим отладки' (Edit mode - press Ctrl+2 or click here to switch to debug mode).

Файл Правка Вид Инструменты Настройки Справка

Robot's Behaviour Diagram

Редактор

Отладка

This is "pacman" game demo. Play it in 2D model using controller buttons in display emulator or on real robot using gamepad or controller buttons.

Global constants.

New Food

Go!

Редактор свойств

Свойство	Значение

Редактор сво... Настройки сенс...

Палитра

Введите текст поиска...

Алгоритмы

- Начало
- Конец
- Функция
- Условие
- Конец условия

Подпрограммы

- CheckGamepad
- Go!
- Draw Food
- Redraw
- Draw Score

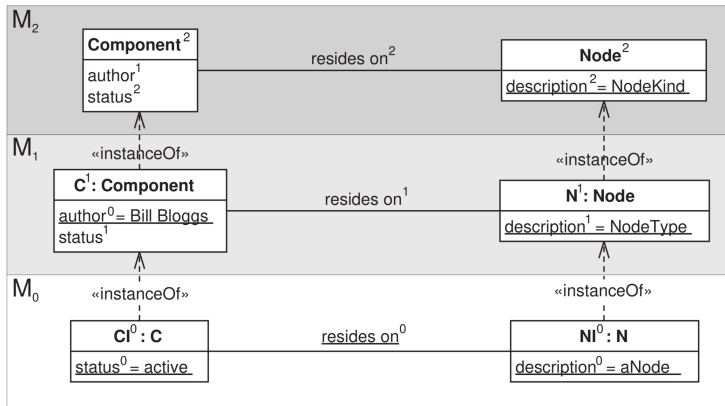
Палитра Переменные

Режим редактирования - нажмите Ctrl+2 или кликните здесь для переключения в режим отладки

Глубокое метамоделирование

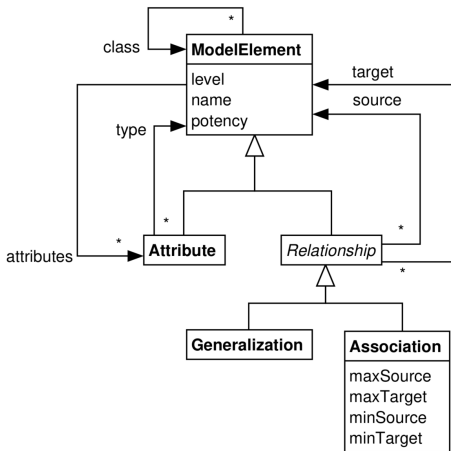
- ▶ Разрешить метатипам определять структуру элементов на несколько метауровней ниже
- ▶ *Clabject* — класс-объект, с атрибутами и полями со значениями
- ▶ *Potency* — число, показывающее, на сколько метауровней вниз элемент может быть инстанцирован
- ▶ *Level* — метауровень, на котором определён элемент
- ▶ Инстанцирование применимо только к элементам с *potency* > 0 и уменьшает *potency* и *level* на 1
- ▶ *Dual fields* — поля, имеющие значение, и могущие быть инстанцированы
- ▶ *Single fields* — “обычные” поля, принимающие значение, только если их *potency* $= 0$

Пример



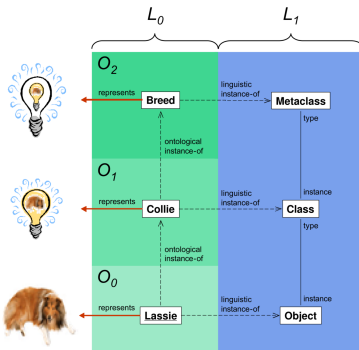
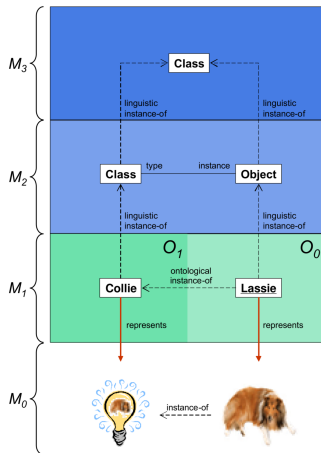
© C. Atkinson, Th. Kuhne, The Essence of Multilevel Metamodeling, 2001

Метамета модель для глубокого метамоделирования



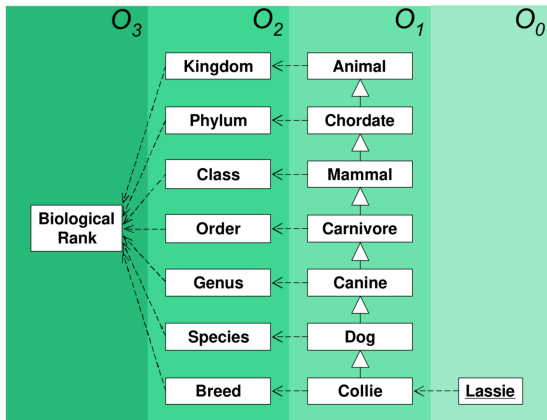
© C. Atkinson, Th. Kuhne, The Essence of Multilevel Metamodeling, 2001

Ортогональное метамоделирование



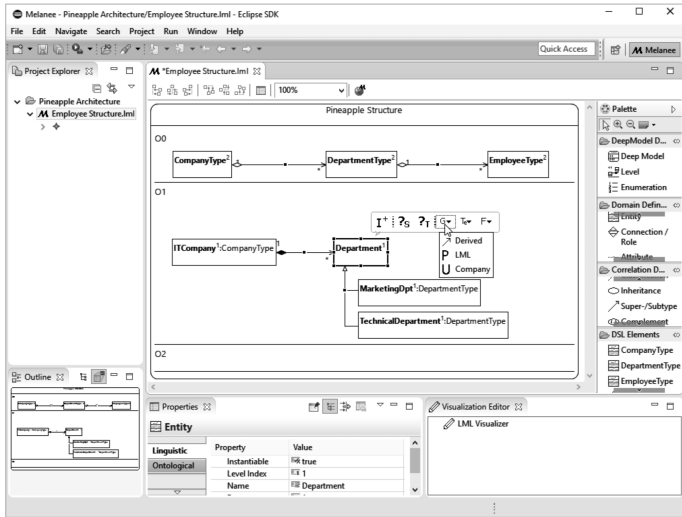
© C. Atkinson, Th. Kuhne, Model-driven development: a metamodeling foundation, 2003

Пример из реальной жизни



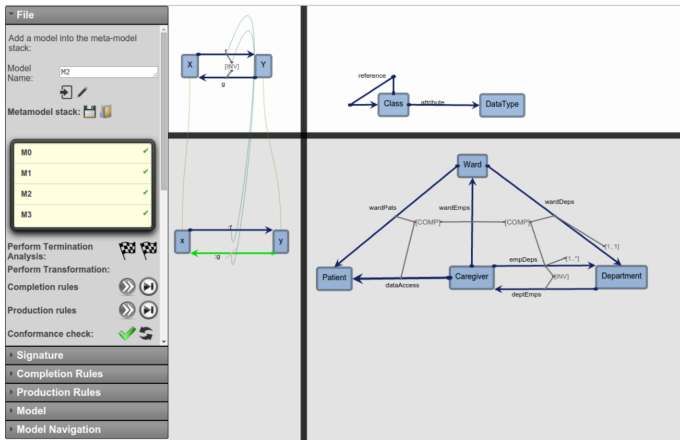
© C. Atkinson, Th. Kuhne, Model-driven development: a metamodeling foundation, 2003

Melanee



© C. Atkinson, R. Gerbig, Flexible Deep Modeling with Melanee, 2016

WebDPF



© F. Rabbi et al. WebDPF: A web-based metamodeling and model transformation environment, 2016

REAL.NET

REAL.NET

RobotsTestModel

```

graph LR
    aInitialNode --> aMotorsForward
    aMotorsForward --> aTimer
    aTimer --> aFinalNode
  
```

Link

- Timer
- MotorsStop
- MotorsBackward
- MotorsForward
- FinalNode
- InitialNode

Elements:

- aTimer
- aMotorsForward
- aFinalNode
- aInitialNode

→ aTimer - aFinalNode

→ aMotorsForward - aTimer

→ aInitialNode - aMotorsForward

Selected entity:

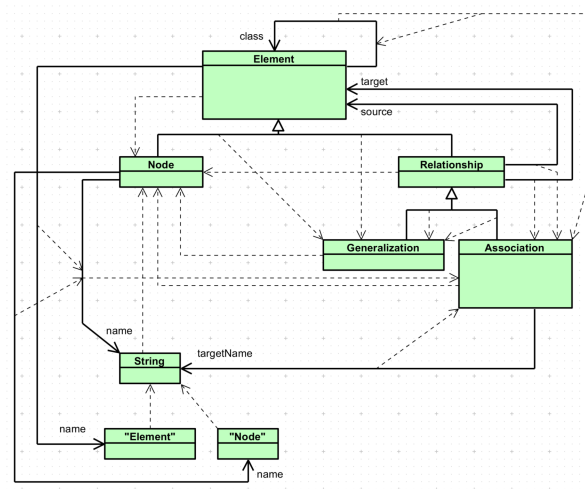
Name	Type	Value
delay	Int	3000
instanceMetatype	String	Metatype.Node
isAbstract	Boolean	false
shape	String	View/Pictures/timerBlock

Sample add-on successfully launched

Sample add-on successfully launched

Messages Errors

Корневая метамодель REAL.NET



Заклучение

- ▶ UML 2.0, вышедший в 2005 году, придерживается четырёхуровневой схемы, как и UML 2.5.1
- ▶ Реализации глубокого метамоделирования начали появляться только в 2010-х, далеки от внедрения
 - ▶ MetaDepth, текстовая среда моделирования от авторов *AToM³*
- ▶ Есть и другие интересные подходы, например, Powertypes