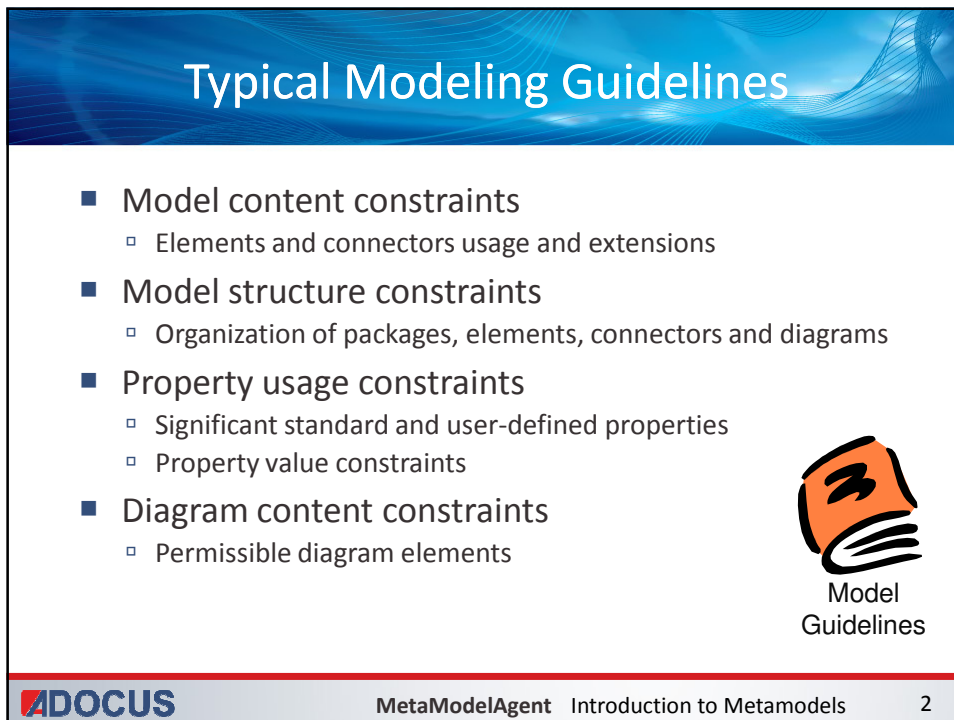


MetaModelAgent

Introduction to Metamodels


**ADOCUS**

Ready for  
**IBM Rational**  
software



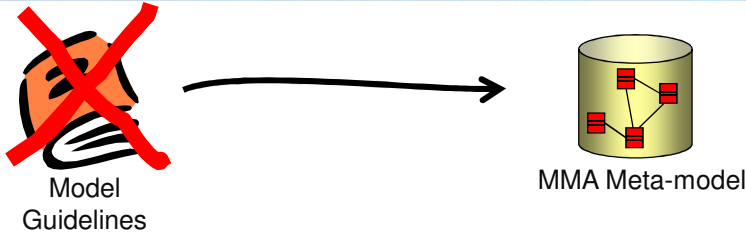
Typical Modeling Guidelines

- Model content constraints
  - Elements and connectors usage and extensions
- Model structure constraints
  - Organization of packages, elements, connectors and diagrams
- Property usage constraints
  - Significant standard and user-defined properties
  - Property value constraints
- Diagram content constraints
  - Permissible diagram elements

  
Model Guidelines

**ADOCUS** MetaModelAgent Introduction to Metamodels 2

## Formalization of Guidelines



Model Guidelines

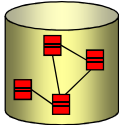
MMA Meta-model

- Natural Language
- Informal
- Vague semantics
- UML
- Formal
- Exact semantics

MetaModelAgent Introduction to Metamodels 3

## Metamodel Notation

- A UML-based Definition Language for defining UML-based Domain-Specific Languages (DSL)
  - The complete DSL-definition is visible in UML-diagrams
- Expressed as positive rules
  - Specification of valid and significant construction in the DSL
  - Anything not explicitly specified is not significant nor allowed
- Focusing on the DSL syntax and static semantics
  - Model structure & content
  - Diagram presence and contents
  - Element properties




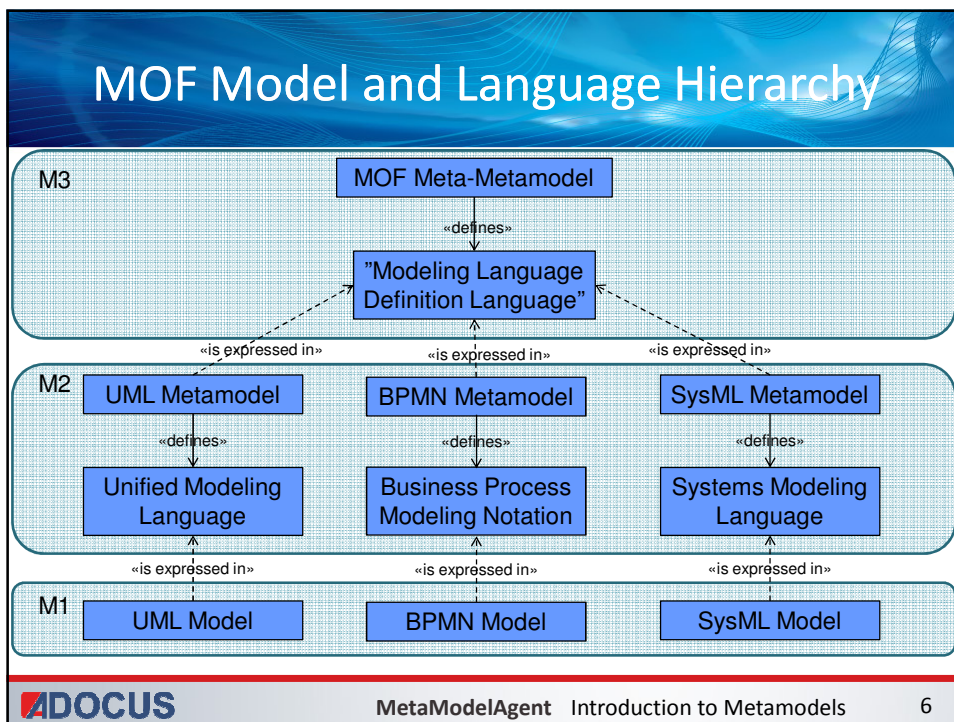
MMA Meta-model

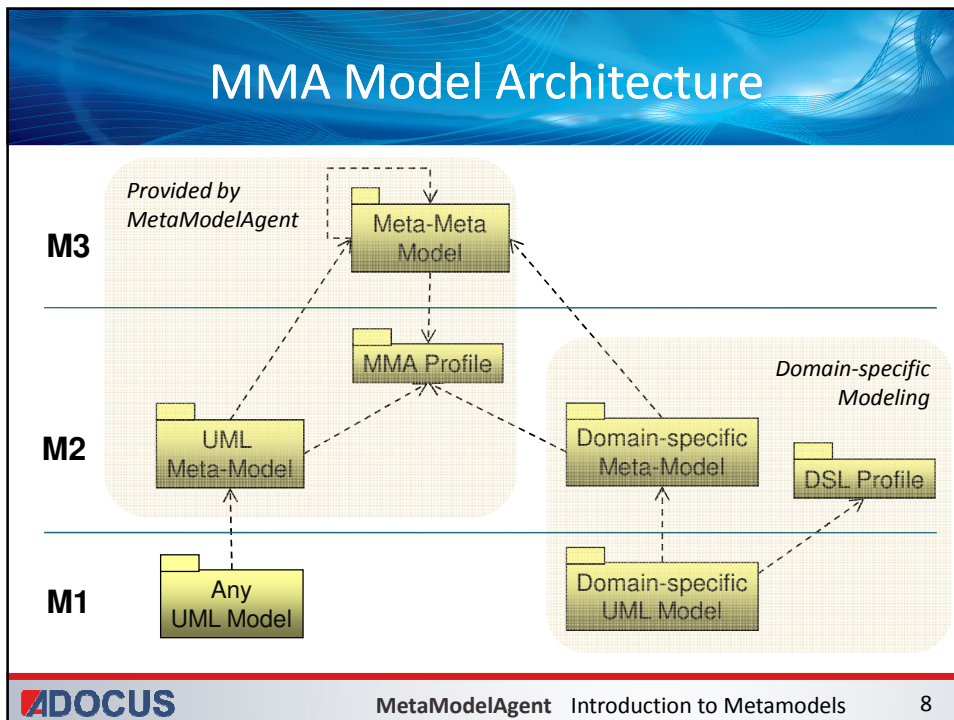
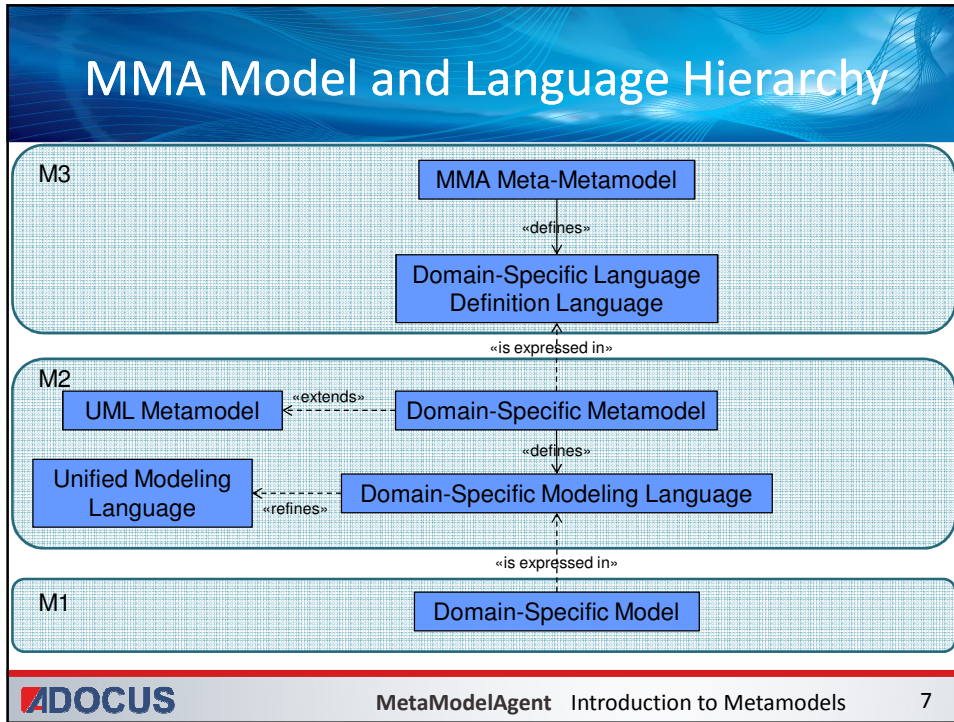
**ADOCUS** MetaModelAgent Introduction to Metamodels 4

## OMG Model Hierarchy

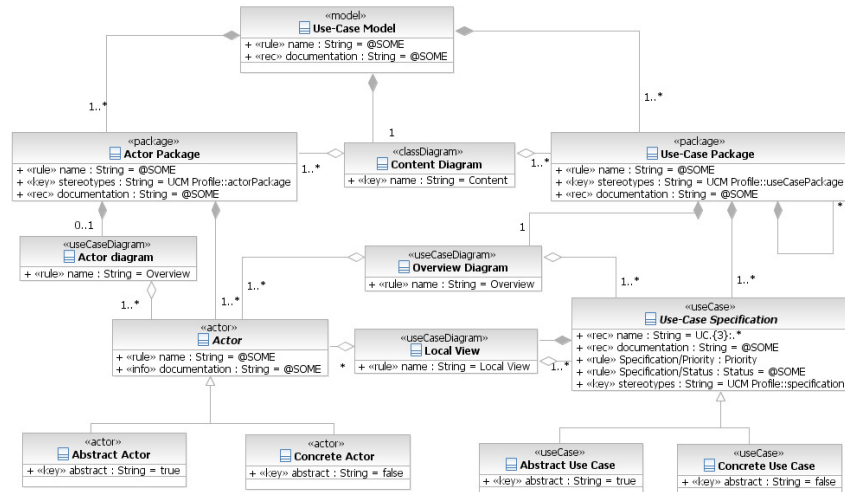
Level	Content	Explanation	Typical items
<b>M3</b>	Meta-meta-model	A meta-model defining how meta-models on level M2 should be designed.	Meta-classes
<b>M2</b>	Meta-model	A meta-model defining how models on level M1 should be designed.	Meta-classes
<b>M1</b>	Model	A model defining executable model instances on level M0.	Classes Use Cases
<b>M0</b>	Instantiation of a model	An executing instance of a model on level M1.	Objects


MetaModelAgent Introduction to Metamodels
5





## Example - Use Case Model Guidelines



## UML Coverage in the MMA Metamodel Notation

- All Elements
  - except some minor elements as Constraints, Comments, etc.
- All Relationships
  - except metaclass extensions
- All Diagrams
  - except Timing Diagram & Interaction Overview Diagram
- All built-in element properties
  - except some action variants specific properties
- All user-defined element properties
  - Except multi-value properties (multiplicity greater than 1)

## UML Elements Used in the MMA Metamodel Notation

- **Packages**
  - For organizing the rules
- **Classes**
  - Represent modeling concepts, e.g. elements and diagrams
- **Attributes**
  - Represent significant element properties
- **Operations**
  - Represent additional element characteristics
- **Aggregations**
  - Represent model composition and diagram content
- **Interfaces**
  - Represent general modeling concept abilities
- **Enumerations & Literals**
  - Represent valid value sets
- **Generalizations**
  - Represent specialization between modeling concepts and value sets
- **Package Imports**
  - Represent reuse between metamodels
- **Constraints**
  - Represent additional constraints

**ADOCUS** MetaModelAgent Introduction to Metamodels 11

## Significant Element Properties in the MMA Metamodel Notation

- **Packages**
  - Stereotypes
  - Name
  - Documentation
- **Interfaces**
  - Stereotypes
  - Name
  - Documentation
- **Classes**
  - Stereotypes
  - Name
  - Abstract
  - Visibility
  - Documentation
- **Associations**
  - Stereotypes
  - Name
- **Association Ends**
  - Name
  - Documentation
  - Visibility
  - Multiplicity
  - Type
  - Aggregation Kind
- **Constraints**
  - Name
  - Body
- **Attributes**
  - Stereotypes
  - Name
  - Visibility
  - Documentation
  - Type
  - Default Value
- **Operations**
  - Name
- **Literals**
  - Name
- **Generalizations**
- **Package Import**

**ADOCUS** MetaModelAgent Introduction to Metamodels 12



## Benefits of working with Metamodels

- **Common understanding of guidelines**
  - Well defined syntax and semantics
  - Easy to ensure complete, consistent and unambiguous guidelines
- **Integrated tool support**
  - Model Validation by use of MetaModelAgent
  - Guidelines adapted wizards and views by use of MetaModelAgent