AMSS Lecture 10: The UML Meta-Model & Profiles

Traian-Florin Șerbănuță

2025

Agenda

Goal: Understand the UML meta-model and how **Profiles** extend/customize it.

The UML Meta-Model

1. **UML Meta-Model Basics**
2. **Meta-Model Architecture (MOF layers)**
3. **How UML Constructs Are Defined**

Profile Diagrams

5. **Profiles as Meta-Model Customization**

The UML Meta-Model

1. What Is a Meta-Model? (10 minutes)

- A *model* represents a real-world system.
- A *meta-model* defines the **rules for building models**.
- UML itself is not just a set of diagrams — it is a **modeling language** defined by a meta-model.

Key idea

The UML meta-model defines:

- What a *Class*, *Attribute*, *Operation*, *Association* are
- How they relate
- What diagrams can contain

Model-Driven Architecture (MDA)

MDA is a software development approach defined by the Object Management Group (OMG)

- Focuses on creating and transforming models rather than writing code directly
- Separates business logic from platform-specific implementation
- Supports automation: models $\rightarrow$ transformations $\rightarrow$ generated code

Key MDA Model Types

CIM	Computation-Independent Model (business/domain understanding)
PIM	Platform-Independent Model (logic without tech details)
PSM	Platform-Specific Model (technology-bound design)

Examples of MDA Languages / Modeling Standards

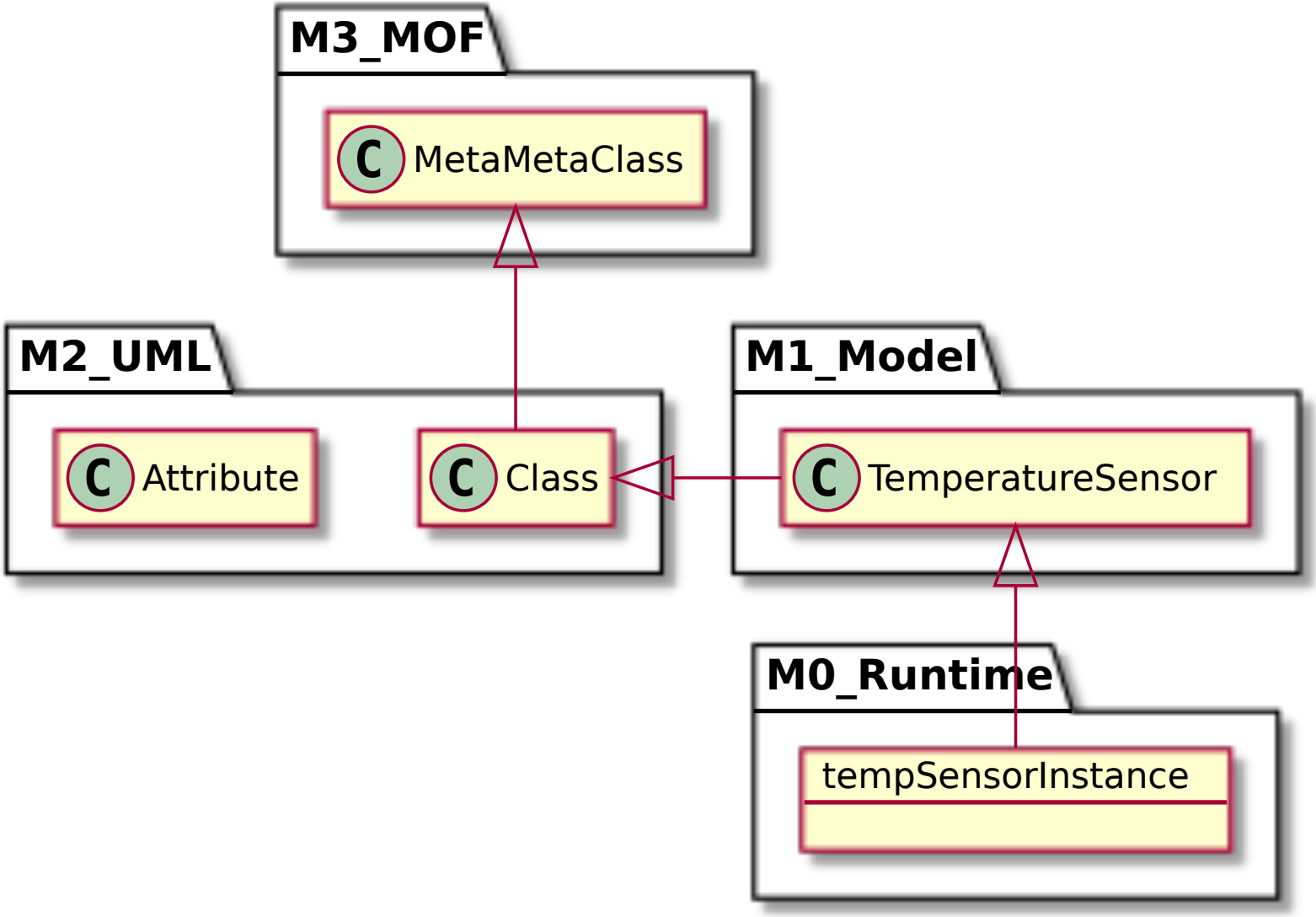
UML	Unified Modeling Language
	(primary MDA modeling language)
MOF	Meta-Object Facility
	(meta-modeling framework)
QVT	Query/View/Transformation
	(model transformation language)
OCL	Object Constraint Language
	(add constraints to models)
XMI	XML Metadata Interchange
	(model serialization/exchange format)

2. The MOF Architecture

UML is defined using a 4-layer meta-model architecture (OMG MOF):

Layer	Meaning	Example
M3	Meta-meta-model	MOF defining UML's structure
M2	Meta-model	UML specification (classes, states, components...)
M1	Model	Your diagrams (class diagrams, state diagrams...)
M0	Runtime	Real objects in the running system

MOF Visualization (as a Package Diagram)



MOF in More Detail

- Meta-Object Facility (MOF) is an Object Management Group (OMG) standard
- Defines how meta-models are built
- UML, SysML, BPMN meta-models are all built using MOF
- Enables interoperability between modeling tools

Key MOF Concepts

Classes	Meta-classes used to define modeling concepts (e.g., UML Class)
Properties	Define attributes and relationships in the meta-model
Packages	Group meta-model elements
Associations	Link meta-classes together

MOF Variants

Essential MOF (EMOF)

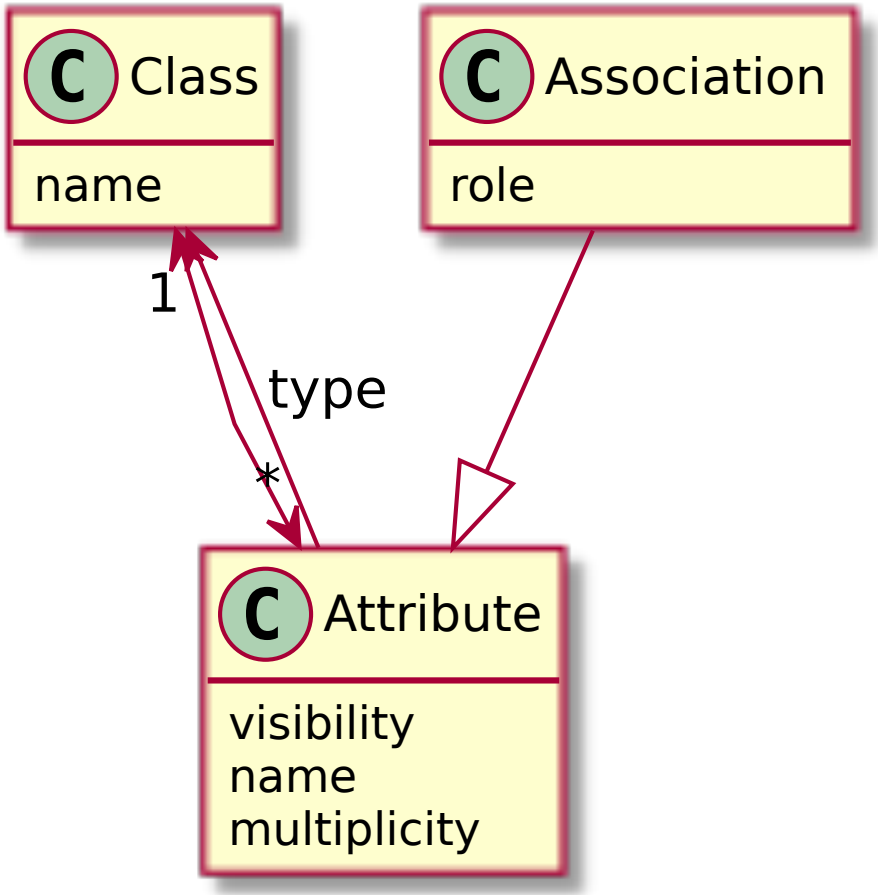
- A simplified subset of MOF
- Used for dimple DSLs, transformation systems
  - Many DSLs (Domain-Specific Languages) use EMOF for simplicity

Complete MOF (CMOF)

- Offers the full expressive power of MOF
  - UML is defined in CMOF

3. Anatomy of the UML Meta-Model

How the UML meta-model defines *Class*, *Attribute*, and *Association*.



- A UML *Class* has *Attribute*s
- *Association*s are *Attribute*s with additional details

Interactive Exercise (10 minutes)

Task: With a partner, reverse-engineer the meta-model elements behind a **sequence diagram**.

Identify: - What meta-model class represents a *lifeline*? - What meta-model class represents a *message*? - What meta-model class represents an *execution specification*?

Write your answers as a small meta-model sketch.

Session 2 (50 minutes)

4. Profiles and Stereotypes (10 minutes)

Profiles are **lightweight extensions** to the UML meta-model.

They allow you to:

- Add domain-specific concepts
- Add constraints
- Specialize existing UML meta-model elements without modifying UML itself

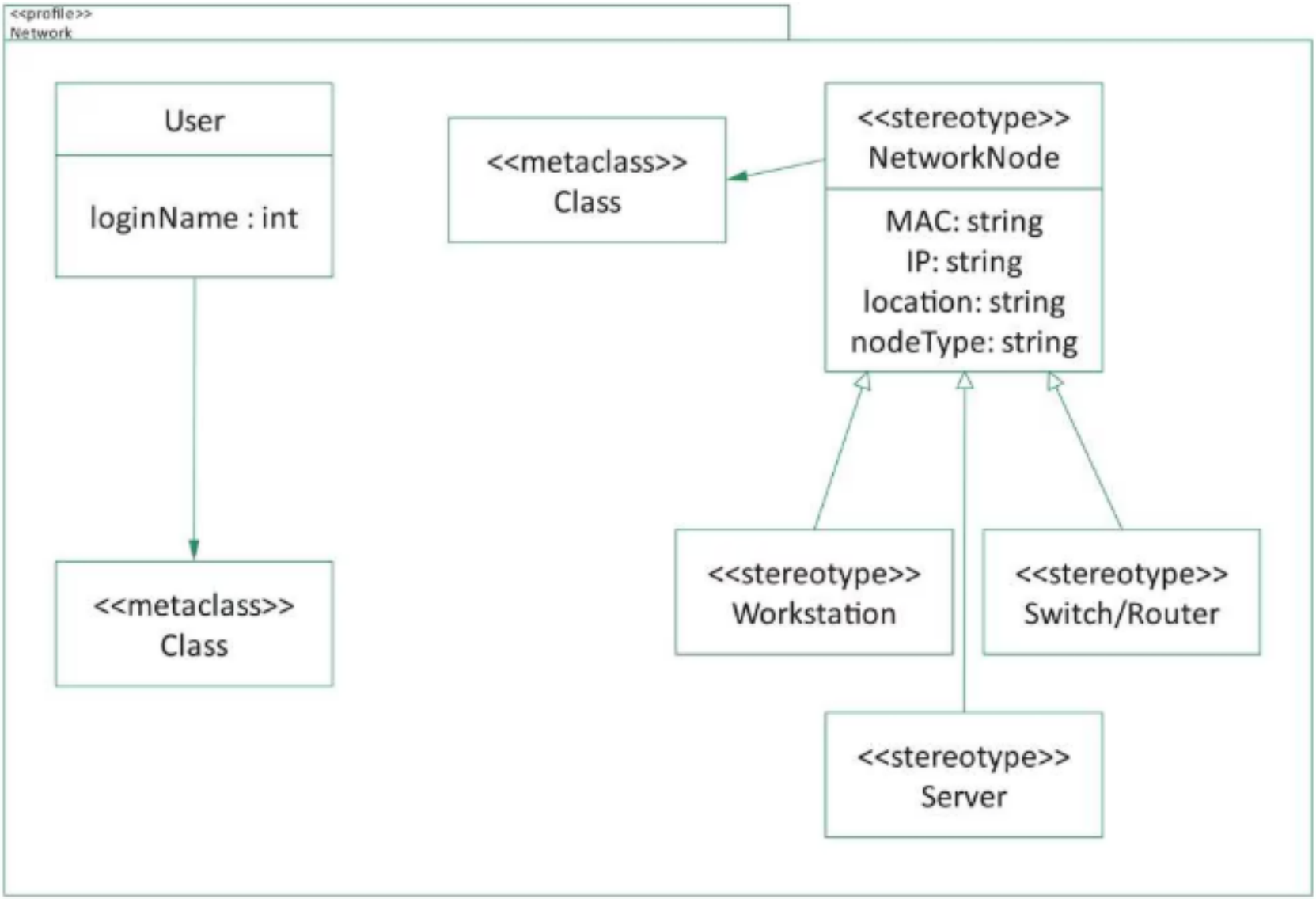
Stereotypes extend UML elements:

- Add tagged values
- Add constraints
- Add semantics

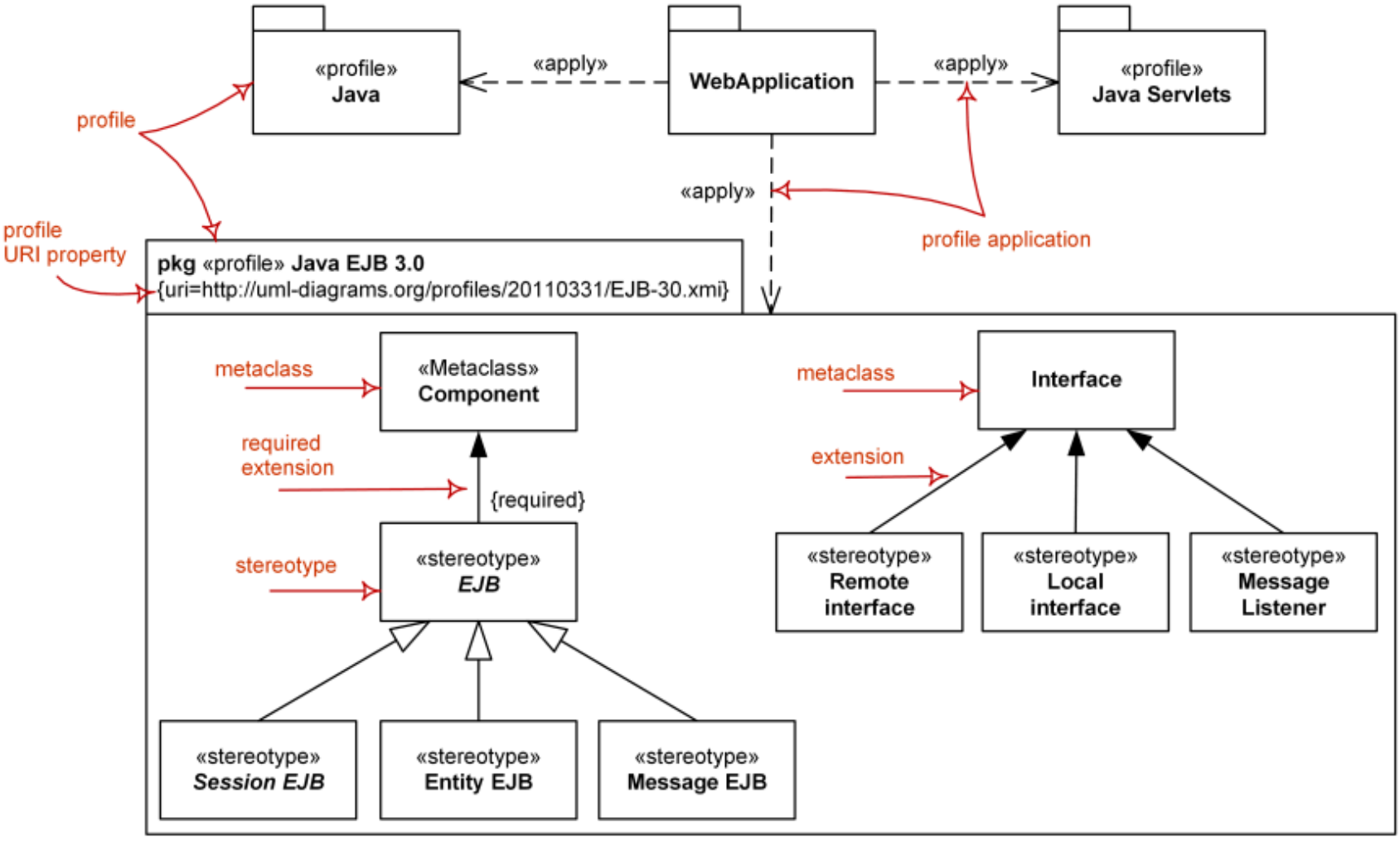
5. Profile Diagrams

Define UML *extensions* for domain-specific modeling.

- custom stereotypes, tagged values, and constraints.



5. Profile Diagram example



5. Profile Diagrams exercise (Secure Web Services profile)

Create a UML Profile Diagram that extends UML to better describe security characteristics of web-service components.

Tasks

1. Create a WebSecurity profile
2. Add stereotypes
    a. SecureCompoent extends Component with encryption and CA tags
    b. SensitiveData extends Class with a dataCategory tag
    c. AuthRequired extends Operation with authLevel tag
3. Add at least one constraint
    - e.g., SensitiveData must have at least one private attribute

How this ties to the meta-model?

- A *stereotype* extends a UML meta-model class, e.g.:
  `stereotype Sensor extends Class`
- `<<Sensor>>` marks all classes that play the role of sensors

Profiles vs. Meta-Model Subclassing (10 minutes)

Why Profiles instead of modifying the UML meta-model?

- Profiles keep UML standard-compliant
- Tool-friendly
- Tailored for specific domains (IoT, automotive, medical, cloud, finance)

Examples:
- SysML = UML Profile
- MARTE (real-time systems) = UML Profile

Interactive Exercise (10 minutes)

Task: Design a profile for the Smart Home system:

Create the following stereotypes: - `SensorDevice` (tag: unit) - `ControllerDevice` (tag: cpuLoad) - `Alerting` (tag: severity)

Apply your stereotypes to: - TemperatureSensor
- SecurityController
- AlarmModule

Bonus: Show how your stereotypes extend meta-model classes (Class, Component, etc.).

Summary

- UML is defined by a **meta-model** (M2 layer) using MOF (M3 layer)
- Your diagrams are **models** (M1), representing real objects (M0)
- Profiles customize UML **without altering the meta-model**
- Stereotypes add domain semantics and constraints
- Profiles are essential for domain-specific modeling (e.g., SysML)

Final Reflection Exercise

Write a short paragraph:

How would you extend UML to better model smart home security concerns?

Consider whether you would add:

- Stereotypes
- Tagged values
- Constraints
- A full Domain-Specific Profile