

# Hera-S: Design of Semantic Web Information Systems

## From Adaptation Engineering to Aspect-Oriented Context-Dependency

### HERA METHODOLOGY

The purpose of Hera is to support the design of applications that provide navigation-based Web structures (hypermedia presentations) over Semantic Web data in a personalized and adapted way. The design approach centers on *models* that represent the core aspects of the application design:

- **Domain Model:** in Hera we use as a starting point a *Domain Model* (DM) that describes the semantical structure of the content data.
- **Application Model:** based on this DM, the designer creates an *Application Model* (AM) that describes a hypermedia-based navigation structure over the content. This navigation structure is devised for the sake of delivering and presenting the content to the user in a way that allows for a (semantically) effective access to the content.
- **Context Model:** effective access can imply the personalization or adaptation that is deemed relevant. Hera allows dynamic personalization and adaptation of the content. For this purpose, context data is maintained in a so-called *Context Model* (CM). This context data is typically updated based on the (inter)actions of the users with the system as well as on external information (e.g. device information, browser information, etc).
- **Presentation Model:** a *Presentation Model* (PM) specifies the concrete presentation design in terms of layout and other (possibly browser-specific) presentation details.

An implementation framework supports the execution of these models to generate actual Web pages. See Figure 1.

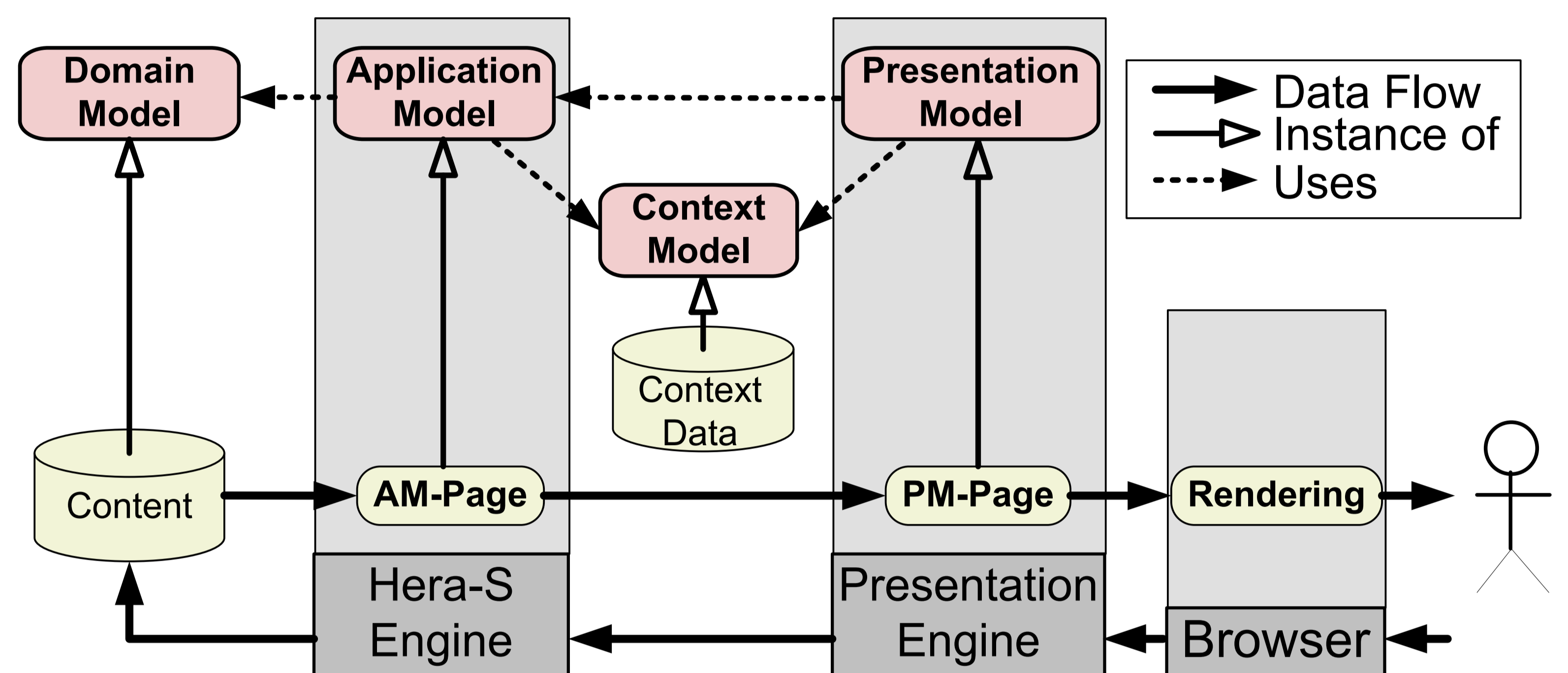


Figure 1: Hera Models and Implementation Pipeline

```
:MovieUnit a am:NavigationUnit ;
  am:hasInput [
    am:variable [
      am:varName "M" ;
      am:varType imdb:Movie
    ] ] ;
  am:hasAttribute [
    rdfs:label "Title" ;
    am:hasQuery
      "SELECT T
      FROM {$M} rdf:type {imdb:Movie};
      imdb:movieTitle {T} "
  ] .
```

Figure 2: Example AM showing the title of a movie

### ADAPTATION IN HERA

**-Adaptation and SeRQL:** flexible adaptation support is provided by manipulating the AM SeRQL queries, which determine the data that appears in the instantiated models

**-Adaptation and AOP:** most adaptation concerns are *cross-cutting* and require adaptation support (i.e. SeRQL query modification) at distributed places in the AM. Therefore, Hera supports the distributed addition of adaptation conditions, in the form of a SeRQL queries, by means of pointcut/advice pairs. Each such pair gives rise to SeRQL query manipulation or addition at multiple places in the design. Figure 3 displays an example AOP that adapts a set of movies given the current user's age and the movie's MPAA-rating.

### INTERNALS: RDF(S), SESAME and SeRQL

**-RDF(S):** Hera uses RDF and RDFS for representation of DM, AM and CM. Using Web standards such as RDF and RDFS facilitates easy deployment on very heterogeneous data sources, is less costly to develop than any alternative, enables reuse of existing knowledge and flexible integration of several separate data sources in a single hypermedia presentation. As access to the data and models is query-based also semantically richer languages (based on RDF) like OWL can be implicitly used.

**-Sesame:** Sesame is used as an RDF repository framework, with its expressive query language SeRQL catering for extra flexibility and interoperability. Furthermore, Sesame allows uniform access and querying of all model-data, it facilitates easy integration of multiple data sources and allows reasoning over domain content and models.

**-SeRQL:** Hera's Application Model (AM) connects to the content (DM) and context (CM) using Sesame. In this setting, Hera associates with each element (unit or relationship) of the AM a SeRQL query, which expresses how the element will be instantiated. Figure 2 displays an example AM that displays a title of a particular movie by means of SeRQL query over the domain.

```
POINTCUT SET WITH PARENT cm:movie
ADVICE
  SELECT M
  FROM {M} am:MPAA-rating {R};
          rdf:type {imdb:Movie}
  WHERE R != 'NC-17'
  OR EXISTS
    (SELECT * FROM {$U} cm:age {G}
     WHERE G > 17)
```

Figure 3: Example AOP specification for parental control