



Towards Integration Aspects and Components

Houssam Fakh^{1,2}
fakh@ensm-douai.fr

Noury Bouraqadi¹
bouraqadi@ensm-douai.fr

Laurence Duchien²
duchien@lifl.fr

¹ École des mines de Douai, GIP
<http://csl.ensm-douai.fr/research>
Douai - France

² LIFL Laboratory, GOAL Team
<http://www.lifl.fr/GOAL>
Lille - France

Facets of Integration

- Three Facets of integration of aspect and component
 - **Facet 1 : Componentizing Aspects**
 - **Facet 2 : Aspectualizing Components-Based Software**
 - **Facet 3 : Unification of aspect and component**

F1 : Componentizing Aspects

Overview

- Representing each **aspect as a single reusable component**
- **Map** the characteristics of a component (attributes, provided and required services, contracts, etc..) on aspect
- Explore **the applicability of concepts like connector, sub-component and composite** on aspect

F1 : Componentizing Aspects

Towards a Solution

- **Provided Part** of aspect : **Generic implementation** of aspect (Context Independent)
- **Required part** of aspect : **Specific implementation** of aspect and mapping with the generic implementation
- Using **contract** to execute **advices** and supporting **introductions**

F2 : Aspectualizing Components

Overview

- **Defining aspects** that acts on **base code** expressed in terms of components and related concepts
- Defining **weaving** and **join points** on execution flow and structure of components and related concepts
- **Definition of AOSD concepts** (aspect, weaving and join point) **varies according to the component model** used

F2 : Aspectualizing Components

Towards a Solution

- Definition of **entry point on components** that allows introspection and/or intercession of its internal state
- **Plugging aspect** on this entry point
- Join points depends on component model
 - Creation, configuration, ...
 - Services call, connexion or disconnection, ...
- **Assembly aspects and components**
 - Binding aspects to components
 - Weaving aspects

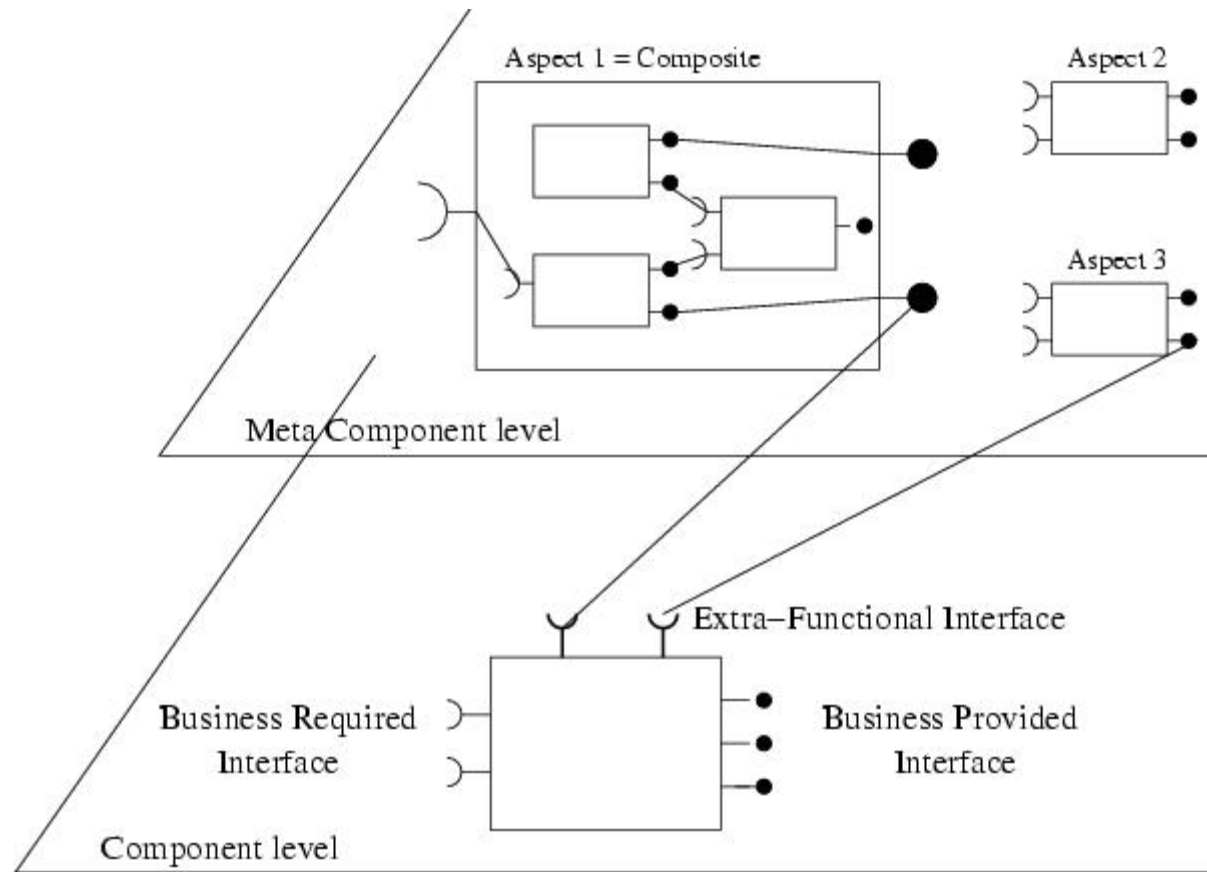
F3 : Unification

Overview

- Merge of the two previous facets
 - **Facet 3 = Facet 1 + Facet 2 + ...**
- Unification by defining a **global Component Model** that encompasses not only ‘traditional’ CBSD concepts but also AOSD ones
- Weaving vs Assembly
 - Components in CBSD are aware of their assembly points. In the AOSD, only aspects are aware of assembly points (join points)
 - The assembly mechanism is not intrusive like weaving

F3 : Unification

Towards a Solution : Meta-Component



- **Interface Meta** on component to plug Meta-control
- **Aspect = Meta-component or Meta-composite**

March 23, 2004

Houssam Fakhri :: fakhri@ensm-douai.fr

<http://csl.ensm-douai.fr/fakhri/phd>

Perspectives

- Exploring and specifying solutions to the three facets
- Use of Fractal Component Model
 - Fractal is an INRIA and France Telecom project supported by the objectweb consortium
- Application of aspect and component integration on **Ubiquitous computing**