

HyperAdapt: Aspect-oriented Adaptation for the Web

Motivation

- users access web applications from different places, with different devices and individual goals
- complexity of developing systems that adapt to these differences increases exponentially with the number of adaptation concerns that are considered (e.g., location adaptation, internationalization)

Obtaining new knowledge on aspect interactions

- interactions between aspects can be undesired, but they can also be the result of a conscious decision of the author
- apply automatic conflict resolution strategies by using the semantics of aspects
- offer semi-automatic resolution of conflicts by involving the author

Concept

Extending invasive software composition (ISC) towards XML documents

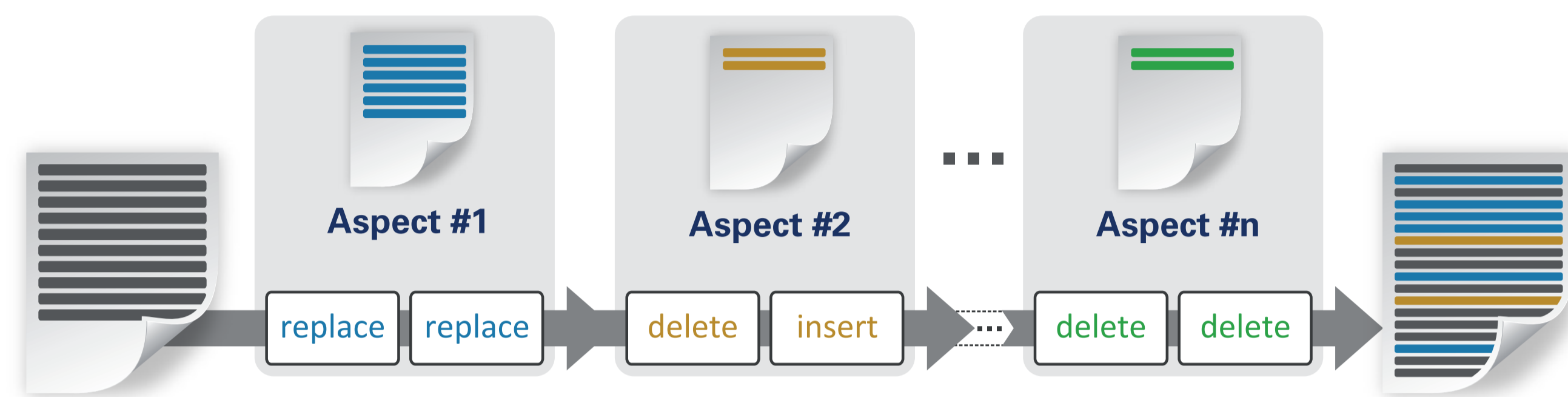
- ISC as a basis for both type-safe weaving and definition of interfaces between XML components
- development of a mechanism to declare extension and variation points within XML documents
- design and implementation of a pipeline for type-safe weaving of XML fragments
- break down aspects into atomic operations

Treatment of aspect interactions

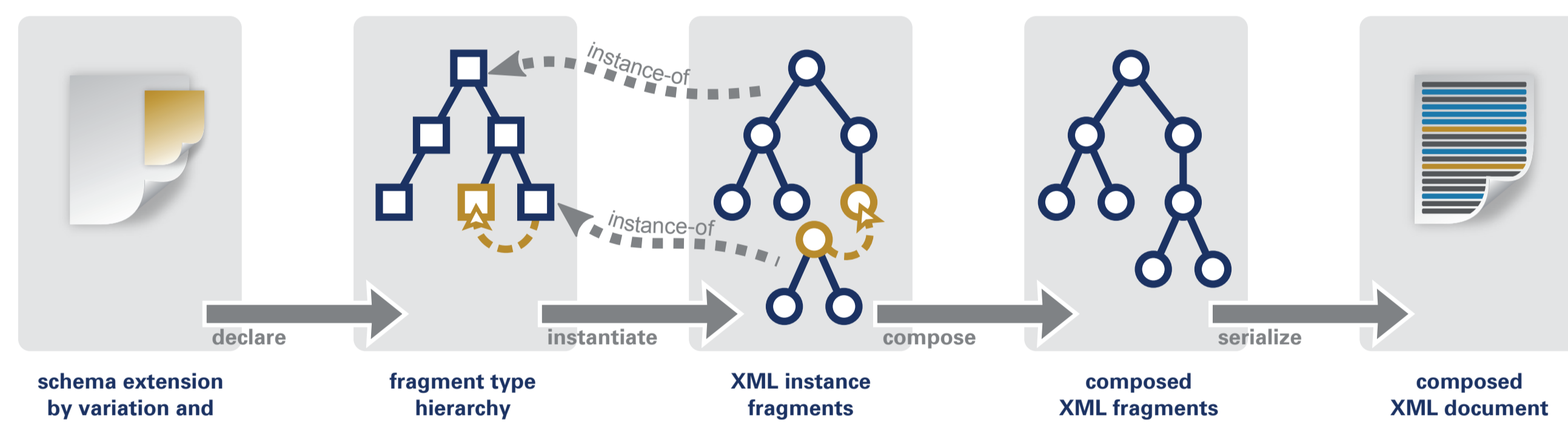
- classification of known adaptation techniques according to their conflict potential
- annotation of these adaptation techniques by semantic descriptions in order to ease conflict analysis
- resolution strategies for combinations of certain adaptation techniques, afterwards extension of these strategies towards aspects in general

Introduction of limitations on aspects and base applications

- locking parts of a web application from being extended by aspects
- allow specification of conditions on the ordering of aspects (e.g., pre- and postconditions)



▲ Aspects are broken down into atomic operations



▲ Through schema extensions, documents can be extended with variation elements and later be composed

- conflicts between adaptation concerns can arise; resolving these manually requires lots of work and insight by the author
- idea for solution: aspect-oriented programming (AOP) applies the principle of dividing a program into its concerns and can thus offer different, specialized views on the application as a whole

Designing a generic approach for aspects in XML

- adaption as „add-on“ to existing web applications and other XML domains

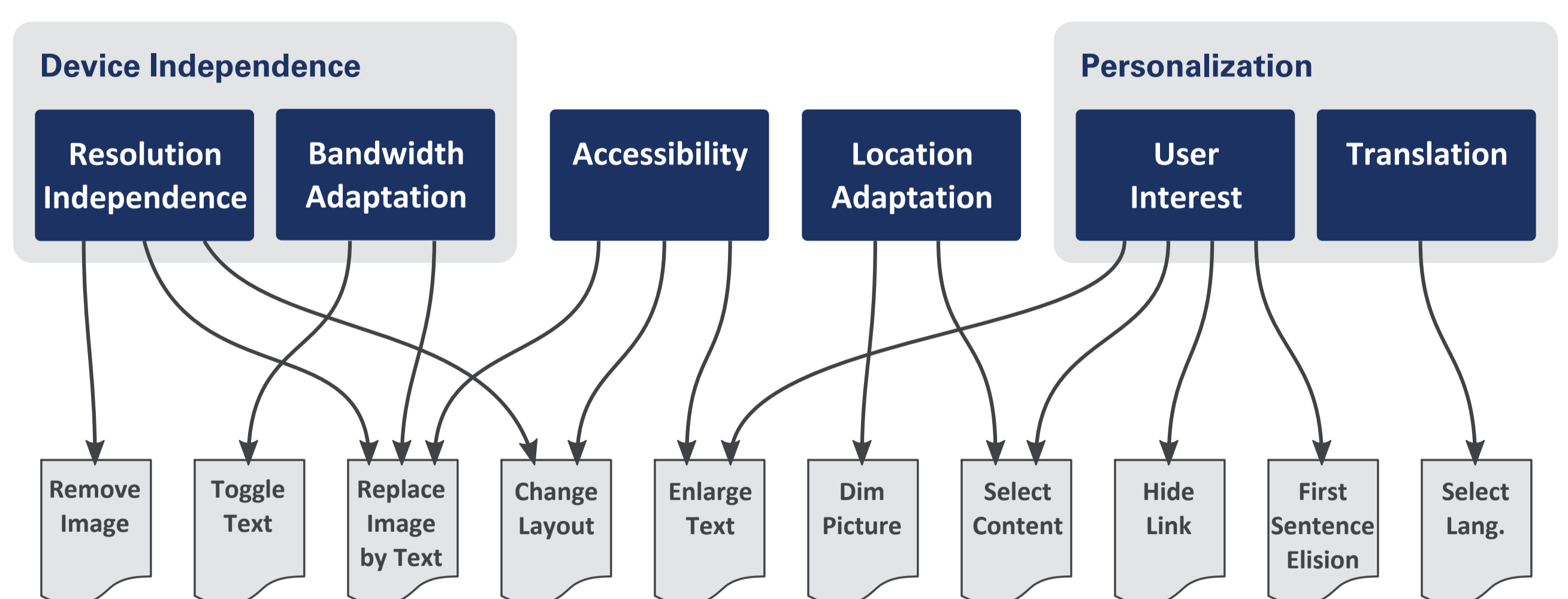
Vision und Research Goals

Encasing of adaptivity in aspects

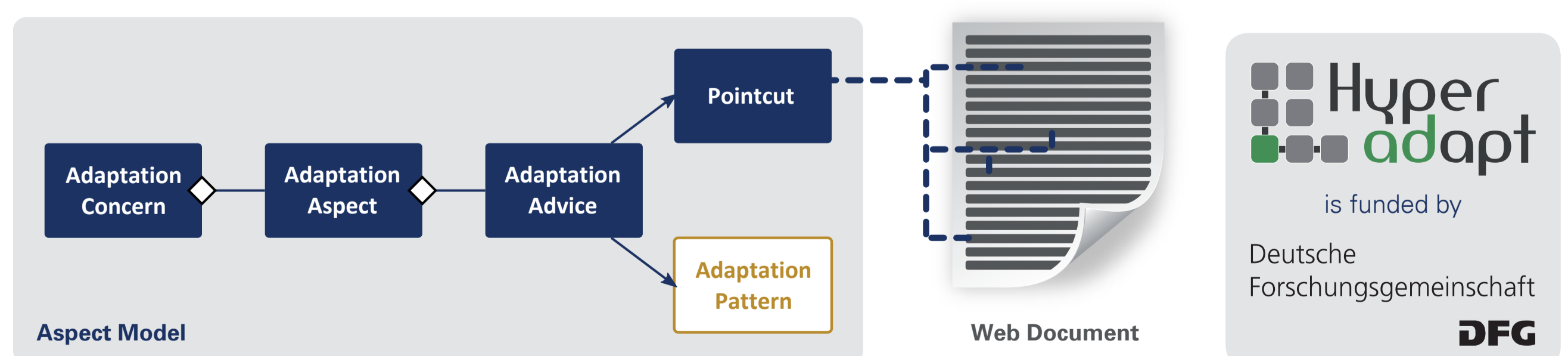
- apply AOP to separately model adaptation within web applications
- organize aspects by their adaptation goal (internationalization, device independence, location adaptation, etc.)

Safe weaving of multiple aspects

- guarantee type safety of aspects (i.e., that they fit accurately into the document)
- enable definition of software contracts between aspects and the base web application
- application of various conflict resolution strategies



▲ Adaptation concerns can be implemented with the help of typical, reusable Adaptation Patterns



▲ Aspects can leverage Adaptation Patterns and are mapped onto an XML document via Pointcuts