# Document Processing:
# Assignment Part 2a: Attribute-Driven Design (ADD)

### Software Architecture: Project Assignment 2014-2015

The project involves the development of a system for document processing. In this part (2 labs), you are asked to sketch an early draft of the architecture using Attribute-Driven Design (ADD). Later in the course (part 2b), you will produce a complete architectural design.

**Supporting material:** The Attribute-Driven Design (ADD) process is described in Chapter 7 of the book (*"Software architecture in practice (Second edition)"*)[1]. The catalog of patterns is contained in the book "Pattern Oriented Software Architecture, Volume 4 – A Pattern Language for Distributed Computing". On Toledo you will find a set of slides explaining ADD and a thorough example (patient monitoring system). You are strongly advised to refer to and use this material.

**Starting point:** Appendix A presents the requirements which serve as the starting point for Part 2a. Tables 1 and 2 shows the intrinsic priority of the requirements. You will need these in priorities in the context of the ADD process in order to select suitable architectural drivers. Note that these requirements are meant to be a starting point for Part 2a and are not necessarily complete with regard to the assignment of Part 1.

**Assignment:** Perform **three decompositions** (i.e. runs of the ADD process). That is, the initial decompositions plus two zoom-ins.

| Quality | Av1 | Av2 | Av3 | P1 | P2 | P3 | M1 | M2 | M3 |
|---|---|---|---|---|---|---|---|---|---|
| **Priority** | H | H | L | H | M | L | M | L | L |

Table 1: Priority of quality requirements ('H' stands for 'High', 'M' for 'Medium' and 'L' for 'Low').

| Priority | Functionality | Use cases |
|---|---|---|
| High | *Authentication;* | *UC1, UC2;* |
| | *Document processing;* | *UC3, UC4, UC5, UC6, UC8, UC9, UC11, UC14, UC15;* |
| Medium | *Receipt tracking;* | *UC7, UC10;* |
| | *Personal document store;* | *UC12, UC13, UC16, UC17;* |
| | *Customer notifications;* | *UC22;* |
| Low | *Customer registration;* | *UC18, UC19;* |
| | *Customer self-management;* | *UC20;* |
| | *Status overview;* | *UC21;* |

Table 2: Priority of functional requirements.

## The report

In the report for part 2a you will provide (1) a detailed description of your ADD execution, and (2) a standalone description of the resulting architecture after three decompositions (ADD runs).

---

[1]Chapter 17 in the third edition.

**1. ADD execution** We expect you to keep a detailed log during the execution of the Attribute-Driven Design (ADD) process, documenting each decomposition in a different subsection. Apply the following naming convention for the title of these subsections: "`Module_that_is_decomposed` (`Architectural_Drivers`)" (e.g. "eDocs System (Av1, UC4, UC5)"). Each time you complete a decomposition, we expect you to **clearly and concisely** document:

- The selected architectural driver(s) for the module you are decomposing (you can and are in fact encouraged to group architectural drivers in one ADD run if they are sufficiently related) and the reasons for selecting and grouping these (rationale for driver selection).

- A systematic explanation of how you solve these requirements; i.e. the key architectural decisions taken during this iteration. Provide UML diagrams to visualize these solutions: at least one component diagram, deployment diagrams when needed (e.g. to show replication).

  – If you have adopted tactics to address the quality requirements, mention explicitly **which tactic and why you picked it.**
  – If you applied architectural pattern(s), provide an explanation of how they support the selected tactics. Pay attention to instantiate the patterns correctly and document what each module of the decomposition does, in collaboration with the other modules. Use the book *"Pattern Oriented Software Architecture Volume 4 – A Pattern Language for Distributed Computing"* as a catalog of patterns to choose from.

- An overview of the alternatives (solutions, patterns, tactics) you considered, and the reasoning behind not adopting them after all.

- The requirements (quality and functionality) that are left and need to be considered for each module of the current decomposition.

**2. Resulting architecture** This section should present the component diagram of the overall system (after three decompositions). At this point, you are not required to provide the deployment diagram of the overall system.

**Formatting rules** To author the report, you can use the tool of your choice, but you must deliver a single PDF file, possibly with indexes enabled for easy navigation (easy to achieve in LaTeX with the hyperref package). No other digital formats are accepted. The file must be self-contained (e.g., no extra figures in attachment) and readable (e.g., font size in pictures is adequate). Pages must be numbered. The cover page must mention the course name and the team member names (including their student id numbers). Be aware that failing to comply with these instructions will cost you points. If you decide to use LaTeX, you can use the template that has been provided (on Toledo).

For UML modeling, you are required to use *Visual Paradigm for UML* (available in the PC labs, and installable on your own machine).

**Delivery** The deadline to turn in your report for part 2a is **Friday 27th of March (noon)**. You are expected to (i) upload a self-contained PDF document on Toledo, and (ii) deliver a printed copy of this report in project post boxes (in the student room A00.03) on the main floor of the Department of Computer Science.

Good luck!

The Software Architecture team.