Implementation of an exploratory workbench for identifying similar design decisions

Prateek Bagrecha, Garching, 09.10.2017 Advisor: Manoj Mahabaleshwar

Software Engineering betrieblicher Informationssysteme (sebis)
Fakultät für Informatik
Technische Universität München

wwwmatthes.in.tum.de

Agenda

- ❑ Introduction: Comparing Two Decisions
- ❑ Introduction: Why Compare ?
- ❑ Motivation
- ❑ Research Questions
- ❑ Approach: K-Means
- ❑ Observations
- ❑ Further Research
- ❑ End User System
- ❑ Configurable Backend System (Pipelines)
- ❑ Evaluation Strategy
- ❑ Timeline

In software engineering and software architecture design, **architectural decisions (ADs)** are **design decisions that address architecturally significant requirements**; they are perceived as **hard to make** and/or **costly to change.**

- Grady Booch, Architecting the unknown, Saturn 2016

Introduction: Comparing Two Decisions

Issues	SPARK-8321	SPARK-19625
Description	Authorization Support(on all operations not only DDL) in Spark Sql	Authorization Support(on all operations not only DDL) in Spark Sql version 2.1.0
Concepts	Apache, SQL, authentication	Apache, SQL, authentication
Keywords	Spark, operations, Support, Authorization	Spark, operations, Support, Authorization
Components	SQL	Spark Core, SQL
Issue Type	Improvement	Improvement
Created	12/Jun/15 03:34	16/Feb/17 09:36
Resolved	16/Jun/16 08:22	24/Mar/17 01:21

Introduction: Why Compare ? (End User Perspective)

Helpful if the second reporter could have been informed about the similar design decision made in past

- ❑ Reduced time for analysis
- ❑ Reduced time to resolution
- ❑ Reduced time to turn around for expert feedback

Given an new open design decision, search the knowledge base for similar earlier made design decisions.

Identifying similarities in design decisions (for an organization)

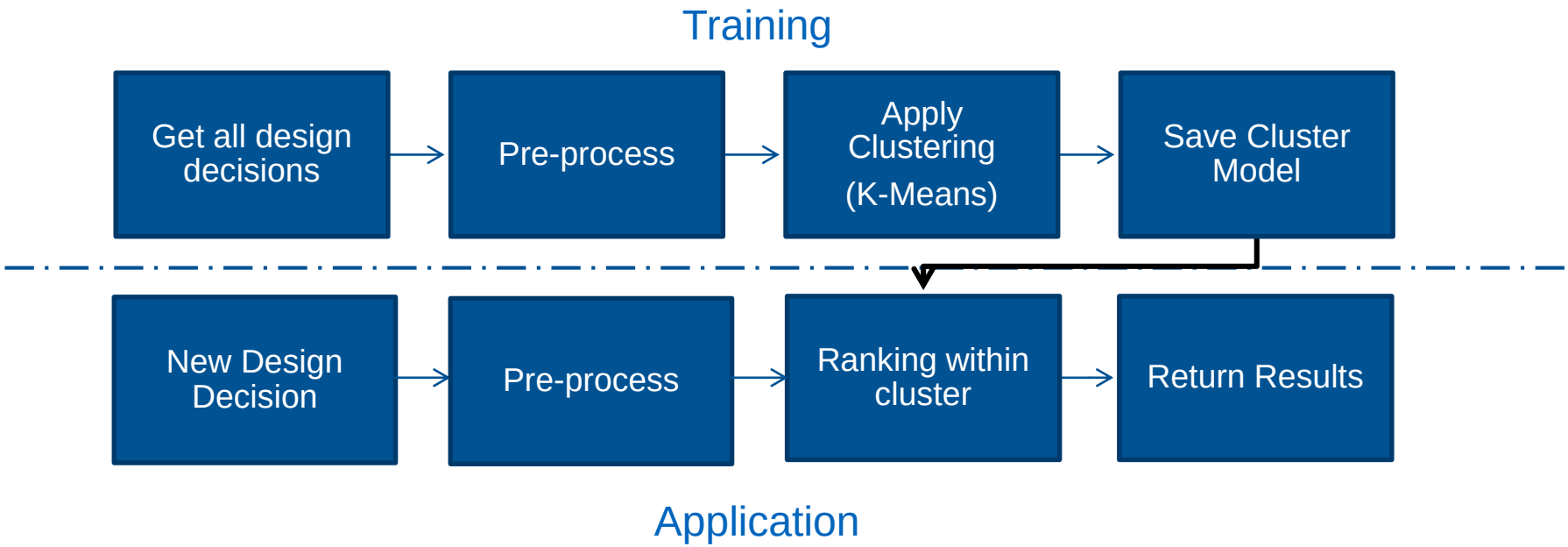
Motivation

- Documentation - specifying constraints on similar design decisions
- Communication - visual representation of related design decisions
- Complexity - Inferring the complexity for addressing similar design decisions

Research Questions

- How to identify similar design decisions?
- What are the context parameters that needs to be considered?
- Which similarity measures are most efficient for comparing context parameters?

Approach



Goal: Analyse alternatives for performing text similarity

- **Machine learning model for unsupervised clustering of design decisions**
- Predicting cluster label for a new design decision
- Ranking within cluster to find most similar design decisions using context

Observations
K-Means (Spark and Hadoop Datasets)

With lower K value (k <= 4 clusters) and no pre-processing

- ❑ Inconsistent cluster
- ❑ Large first cluster
- ❑ Clustering based on missing values

Lessons learnt ➔ Need pre-processing

With higher K value (k = 8 & k = 20) and with pre-processing

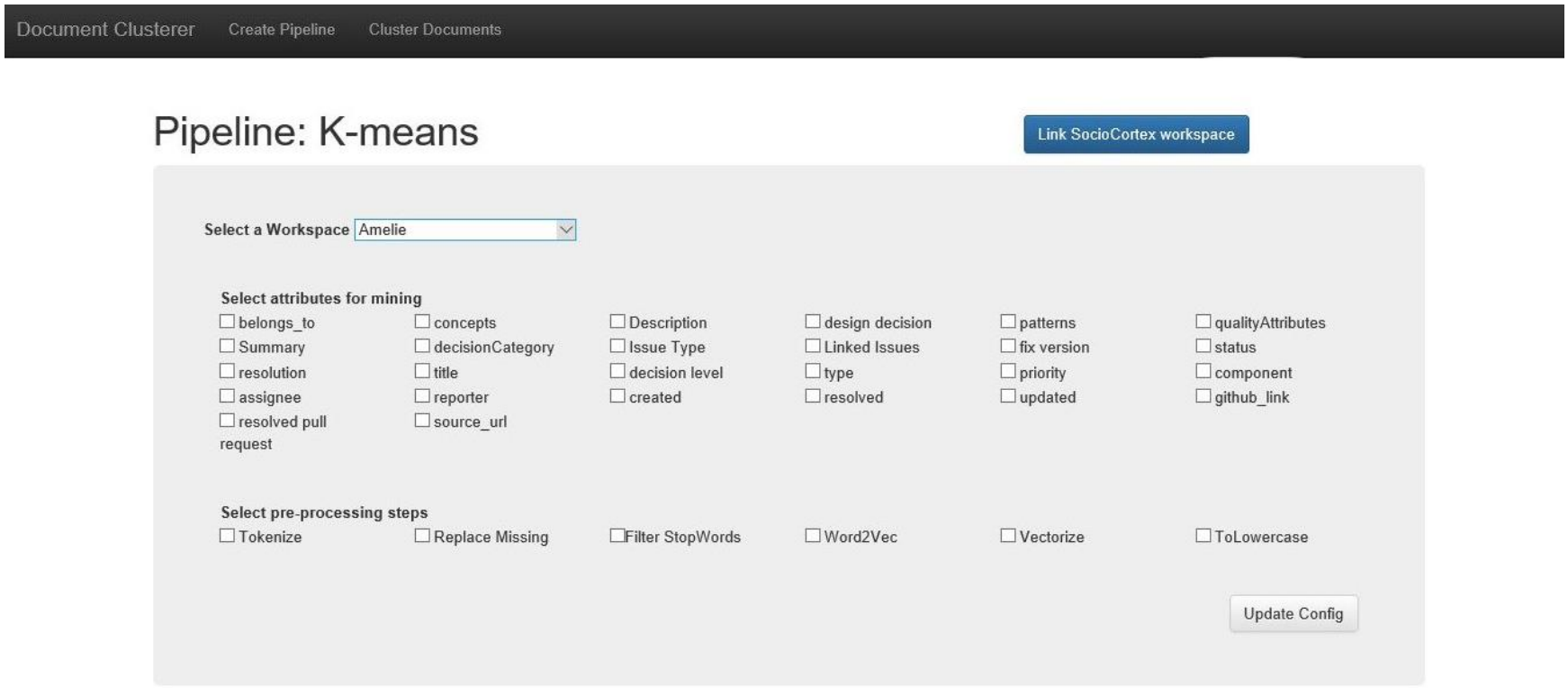
- ❑ Uniform clusters
- ❑ a more uniform spread for 20 clusters
- ❑ Best Assumption: 8 < K < 20
- ❑ However, some cluster have <= 7 members,
  - ➔ include member from other cluster in the results ?
  - ➔ Fuzzy C-Means

Further Research

- *K-Means **vs** Fuzzy C-Means ➔ Mutually exclusive clusters* **vs** *clusters with membership weights*
- Finding optimum k value
- Ranking within clusters ➔ Compare using context similarity measure

End User System
Extending AMELIE to Include Similar Design Decisions

Backend System
Configurable Pipelines



Evaluation Strategy
Qualitative & Quantitative

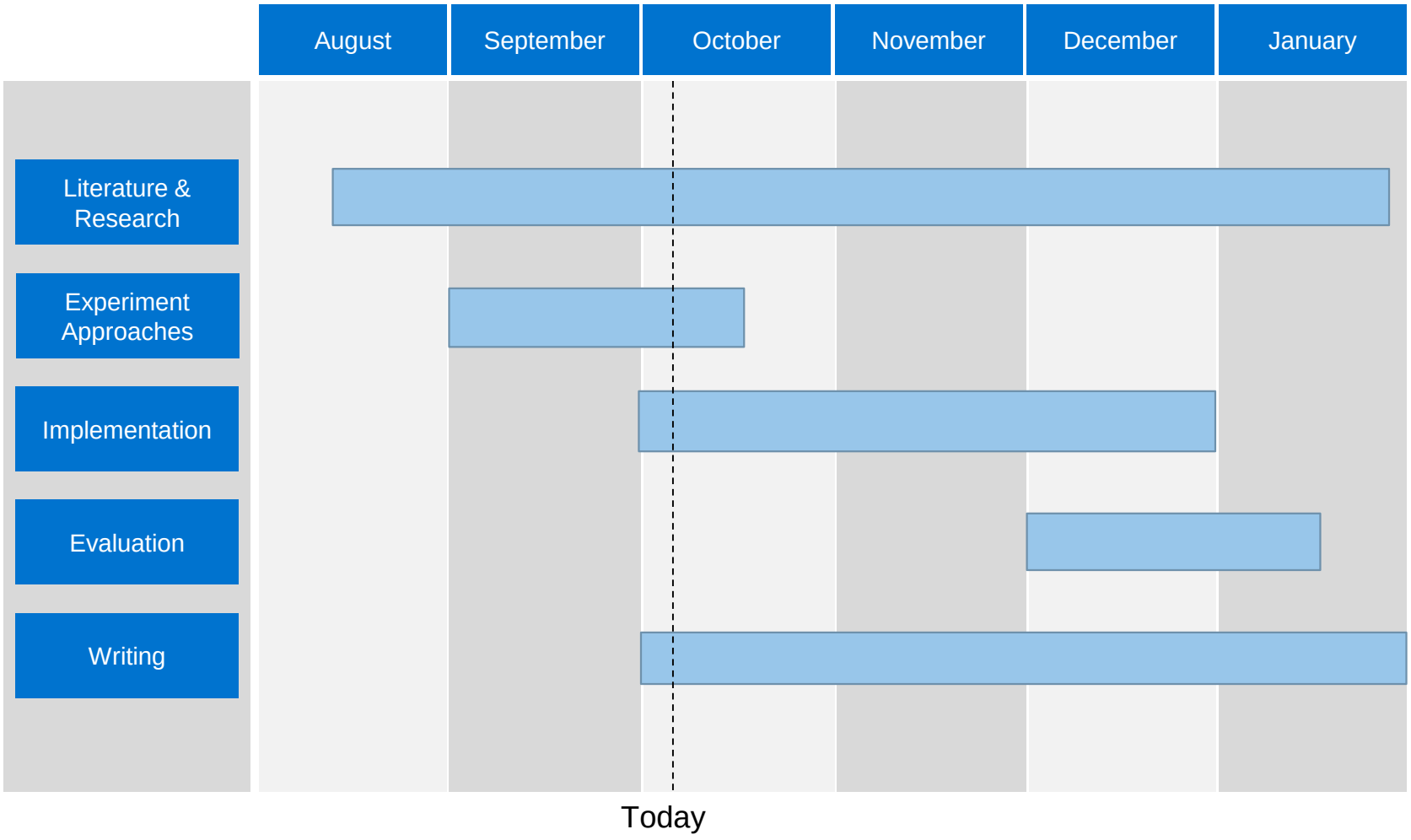
Qualitative Strategy

- ❑ Expert Evaluation by Employees of Siemens (Experiment Dataset provided by Siemens)

Quantitative Strategy

- ❑ Creating a Test Dataset from Open Source Projects that contains duplicates
- ❑ Evaluate the trained model for precision and recall

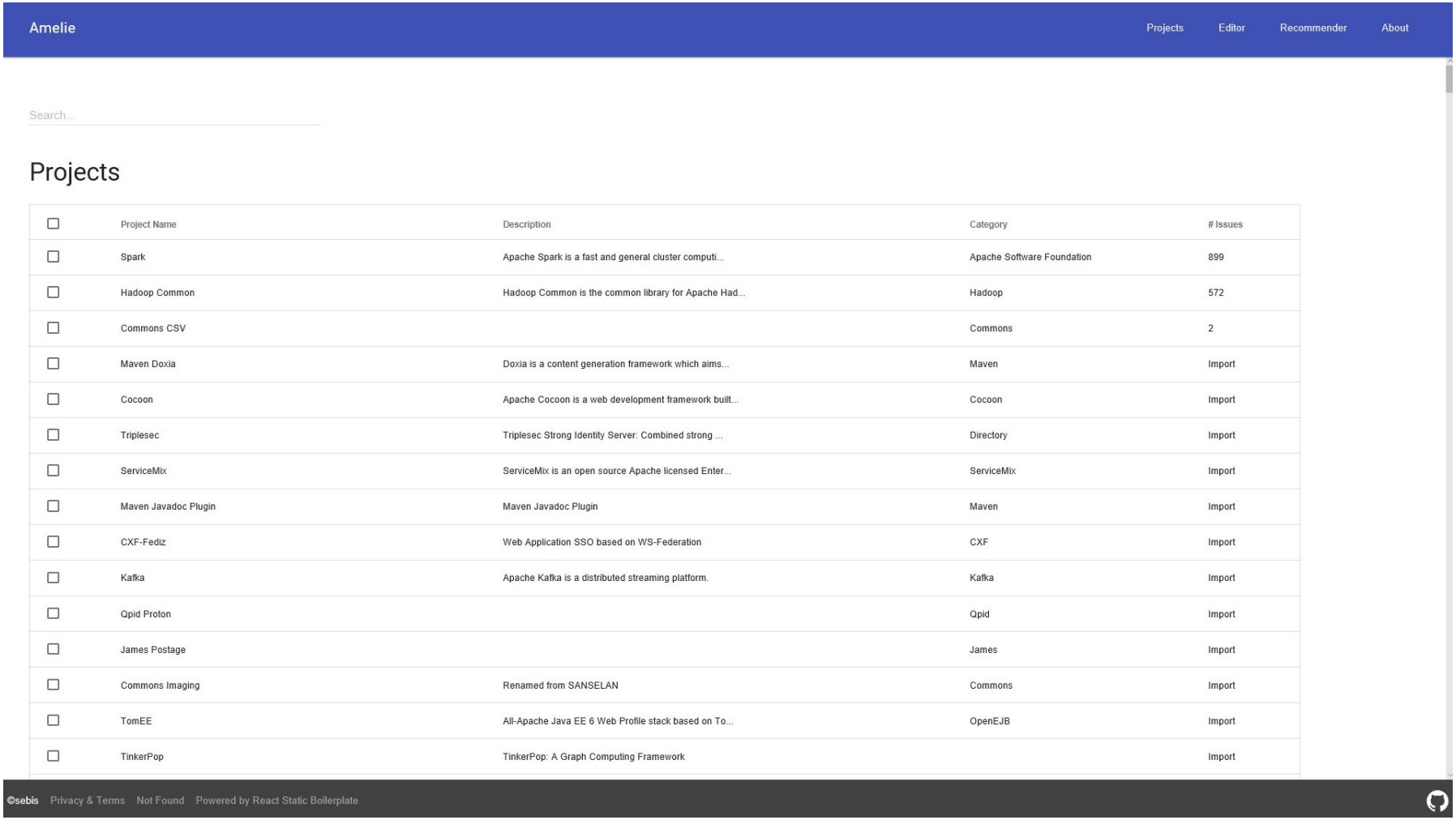
Thesis Timeline



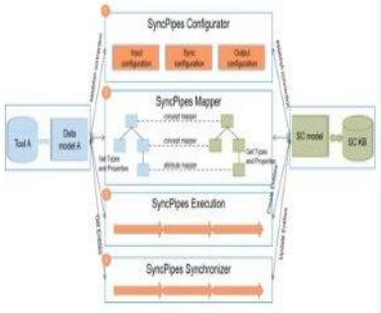
Official Start Date: 15.08.2017 Official End Date: 15.02.2018 Advisor: Manoj Mahabaleshwar

Thank you

End User System: AMELIE
The Project Explorer

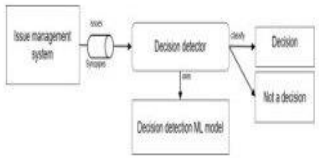


End User System: AMELIE
A Visual Frontend



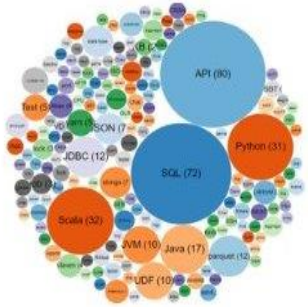
Step 1. Import issues from JIRA using

DEMO



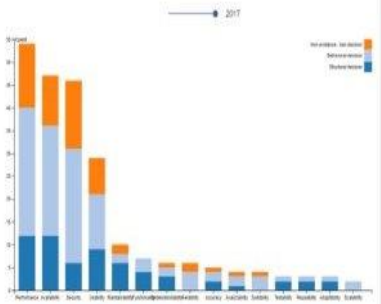
Step 2. Extract design decisions from issues.

DEMO



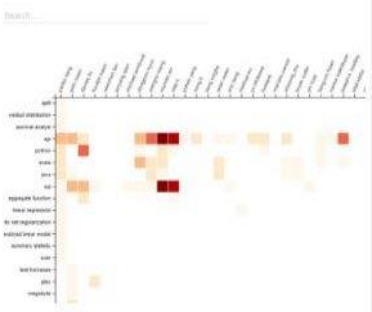
Step 3. Annotate architectural elements in

DEMO



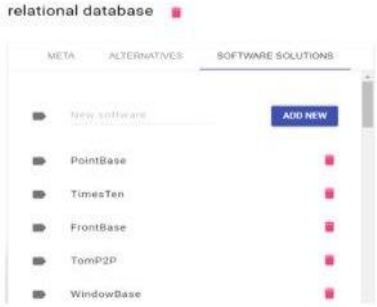
Step 4. Annotate design decisions with quality

DEMO



Step 5. Get recommendations about experts.

DEMO



Step 6. Get recommendations for alternative

DEMO

Observations Detailed
K-Means & Similarity

With lower K value (k <= 4 clusters) and no pre-processing

- If attribute “design decision” is included, cluster members are those with values 1 & 0 value for it.
- If quality attribute is included, cluster members are based on type of quality,
  - Not required, we already have classification based on this.
- Clustering based on the summary and description attributes of issues
  - Leads to inconsistent clusters with the initial assignment of a one document to each cluster and followed by the assignment of all documents to the first cluster.
- Clustering based on missing values

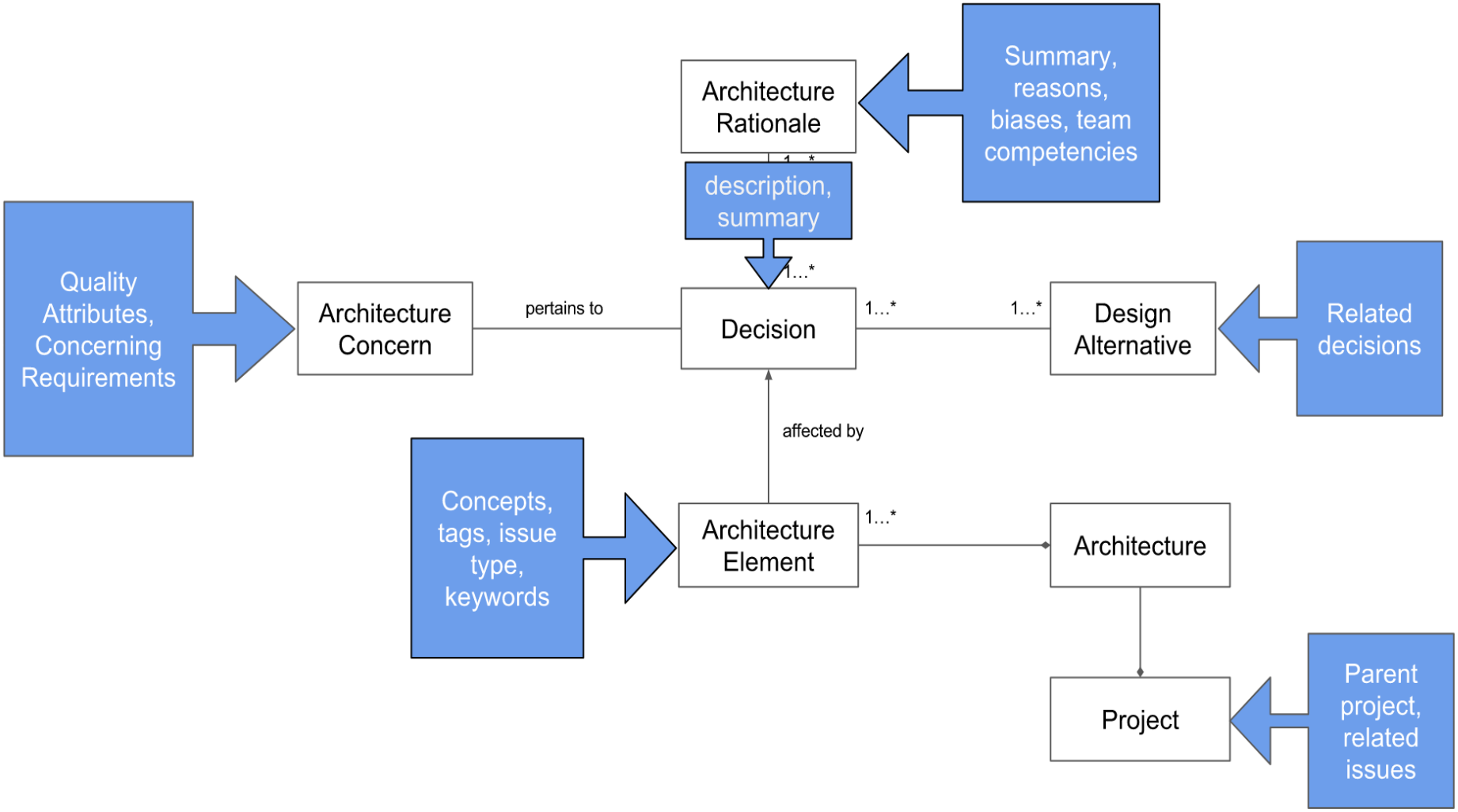
With higher K value (k = 8 & k = 20) and with pre-processing

- Uniform clusters (a more uniform spread for 20 clusters)
- However, some cluster have <= 7 members, include member from other cluster in the result ➔ Fuzzy C-Means

With Direct Similarity Measure

- Equidistant from eachother

Methodology: Where does context lie ?



Algorithmic steps for K-Means clustering

1) Set K – To choose a number of desired clusters, K.
2) Initialization – To choose k starting points which are used as initial estimates of the cluster centroids. They are taken as the initial starting values.
3) Classification – To examine each point in the dataset and assign it to the cluster whose centroid is nearest to it.
4) Centroid calculation – When each point in the data set is assigned to a cluster, it is needed to recalculate the new k centroids.
5) Convergence criteria – The steps of (iii) and (iv) require to be repeated until no point changes its cluster assignment or until the centroids no longer move.