POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

Effective Identification and Reuse of Model Patterns in Service Orchestration Modeling

by Florian Daniel, florian.daniel@polimi.it

SummerSOC, Crete, Greece, June 26, 2017

Goal

Assisting developers by
recommending development knowledge

Recommendation and Weaving of Reusable Mashup Model Patterns for Assisted Development

SOUDIP ROY CHOWDHURY, INRIA Saclay
FLORIAN DANIEL and FABIO CASATI, University of Trento

With this article, we give an answer to one of the open problems of mashup development that users may face when operating a model-driven mashup tool, namely the *lack of modeling expertise*. Although commonly considered simple applications, mashups can also be complex software artifacts depending on the number and types of Web resources (the components) they integrate. Mashup tools have undoubtedly simplified mashup development, yet the problem is still generally nontrivial and requires intimate knowledge of the components provided by the mashup tool, its underlying mashup paradigm, and of how to apply such to the integration of the components. This knowledge is generally neither intuitive nor standardized across different mashup tools and the consequent lack of modeling expertise affects both skilled programmers and end-user programmers alike.

In this article, we show how to effectively assist the users of mashup tools with contextual, interactive recommendations of composition knowledge in the form of reusable mashup model patterns. We design and study three different recommendation algorithms and describe a pattern weaving approach for the one-click reuse of composition knowledge. We report on the implementation of three pattern recommender plugins for different mashup tools and demonstrate via user studies that recommending and weaving contextual mashup model patterns significantly reduces development times in all three cases.

S. Roy Chowdhury, F. Daniel and F. Casati. Recommendation and Weaving of Reusable Mashup Model Patterns for Assisted Development. *ACM Transactions on Internet Technology* 14(2-3), Article 21, 2014.

Idea

Development knowledge -> model patterns
Proactively assist development -> recommend patterns
Don’t distract developers -> context + speed
Make knowledge operational -> weave patterns

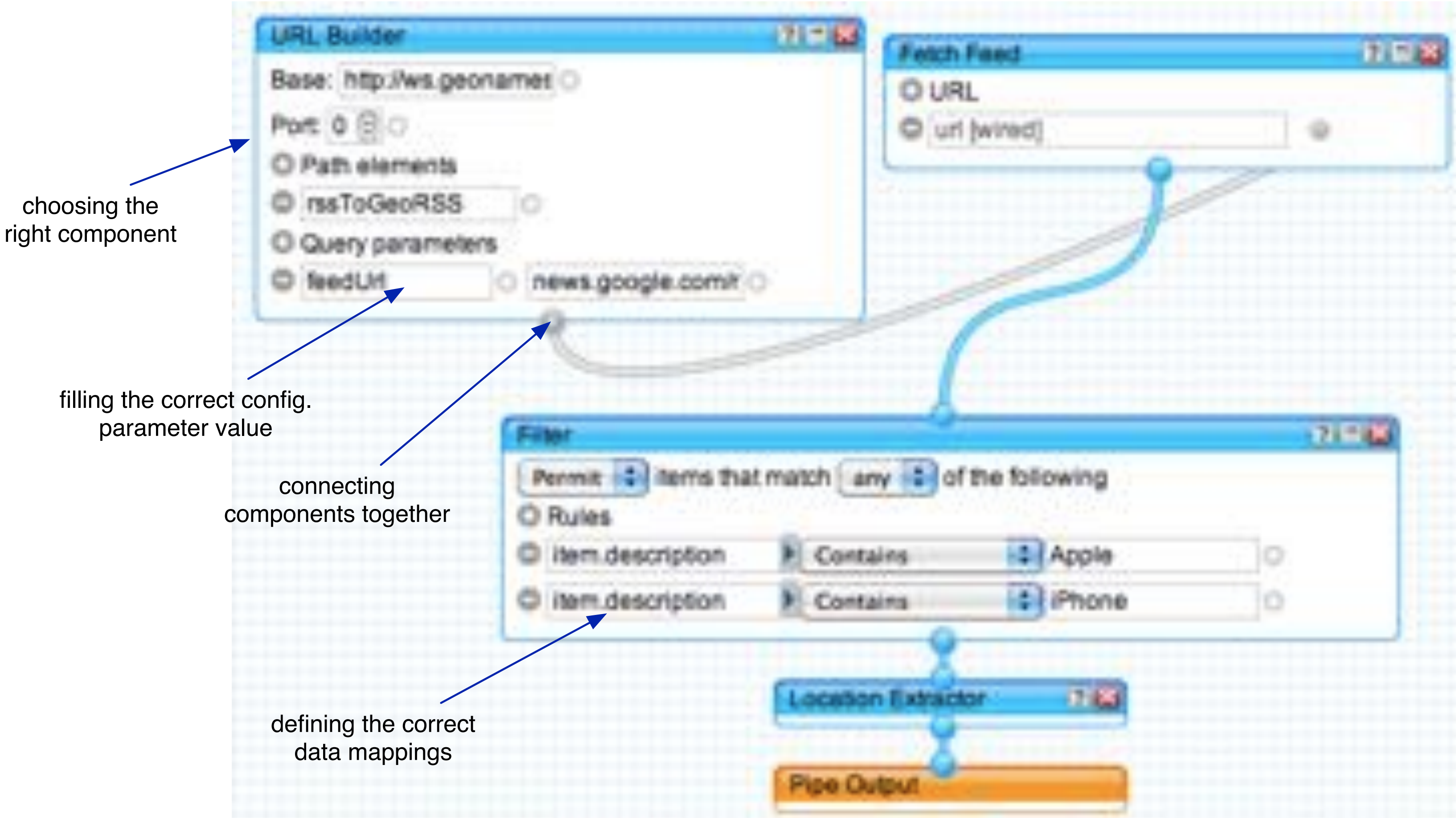
Research question

Does recommending model patterns really help developers model faster / better?

A typical **mashup model pattern** (Yahoo! Pipes)

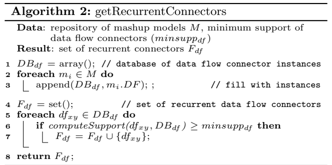
Mashup model: $m = \langle name, C, F, M, P \rangle$

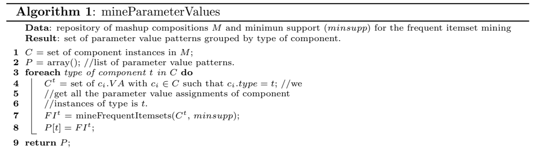
Composition pattern model: $cp = \langle C, F, M, P, usage, date \rangle$

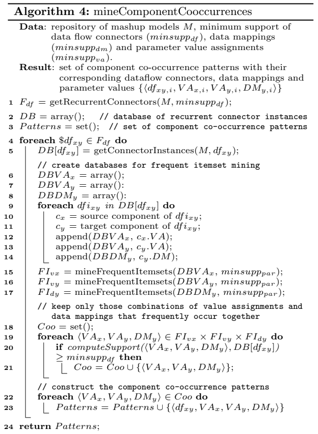


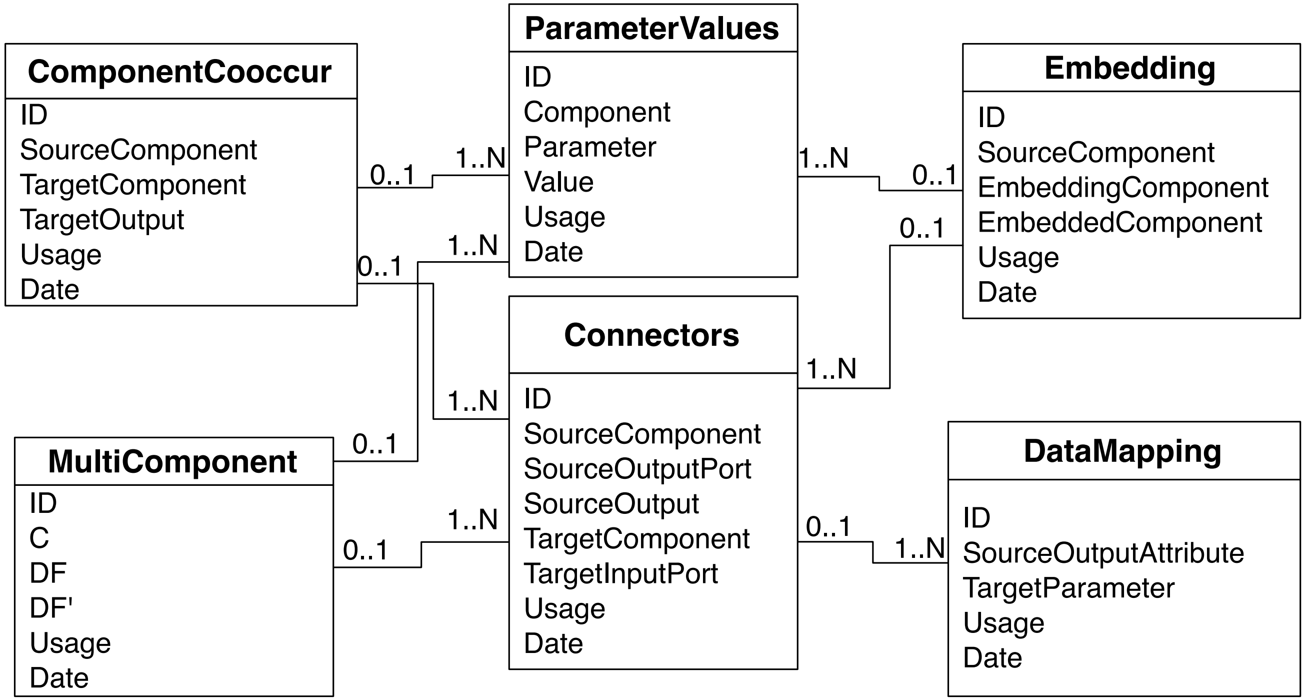
The pattern tells how to enrich an RSS feed with geo-coordinates and plot its items on a map

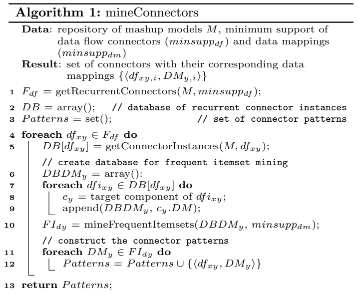
Pattern **knowledge base**

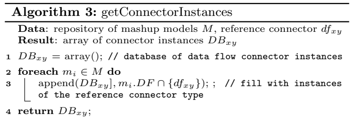












Patterns **mined** from a dataset of 970 “most popular” pipes models of Yahoo! Pipes (association rule mining, frequent itemset mining + predefined topologies)

C. Rodriguez, S. Roy Chowdhury, F. Daniel, H.R. Motahari Nezhad and F. Casati. Assisted Mashup Development: On the Discovery and Recommendation of Mashup Composition Knowledge. In *Web Services Foundations*, Springer, 2014, Pages 683-708.

Recommendation **algorithms**

Contextual: candidate patterns contain the object of the last modeling action; exact and approximate matching

Personalized: ranks contextual recommendations according to users' past component preferences

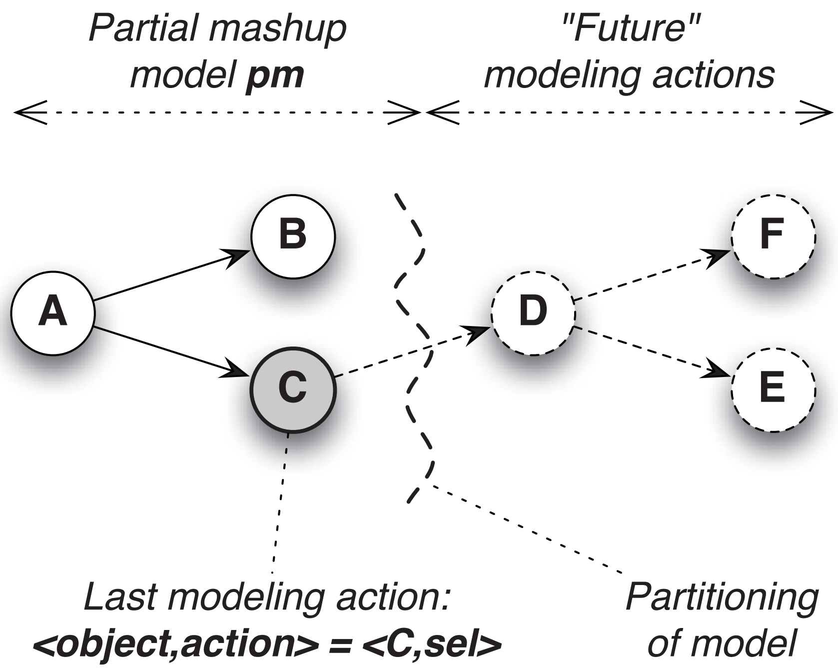
Expert: ranks contextual recommendations according to experts' past component preferences; cloning h-index

Modeling **test cases**

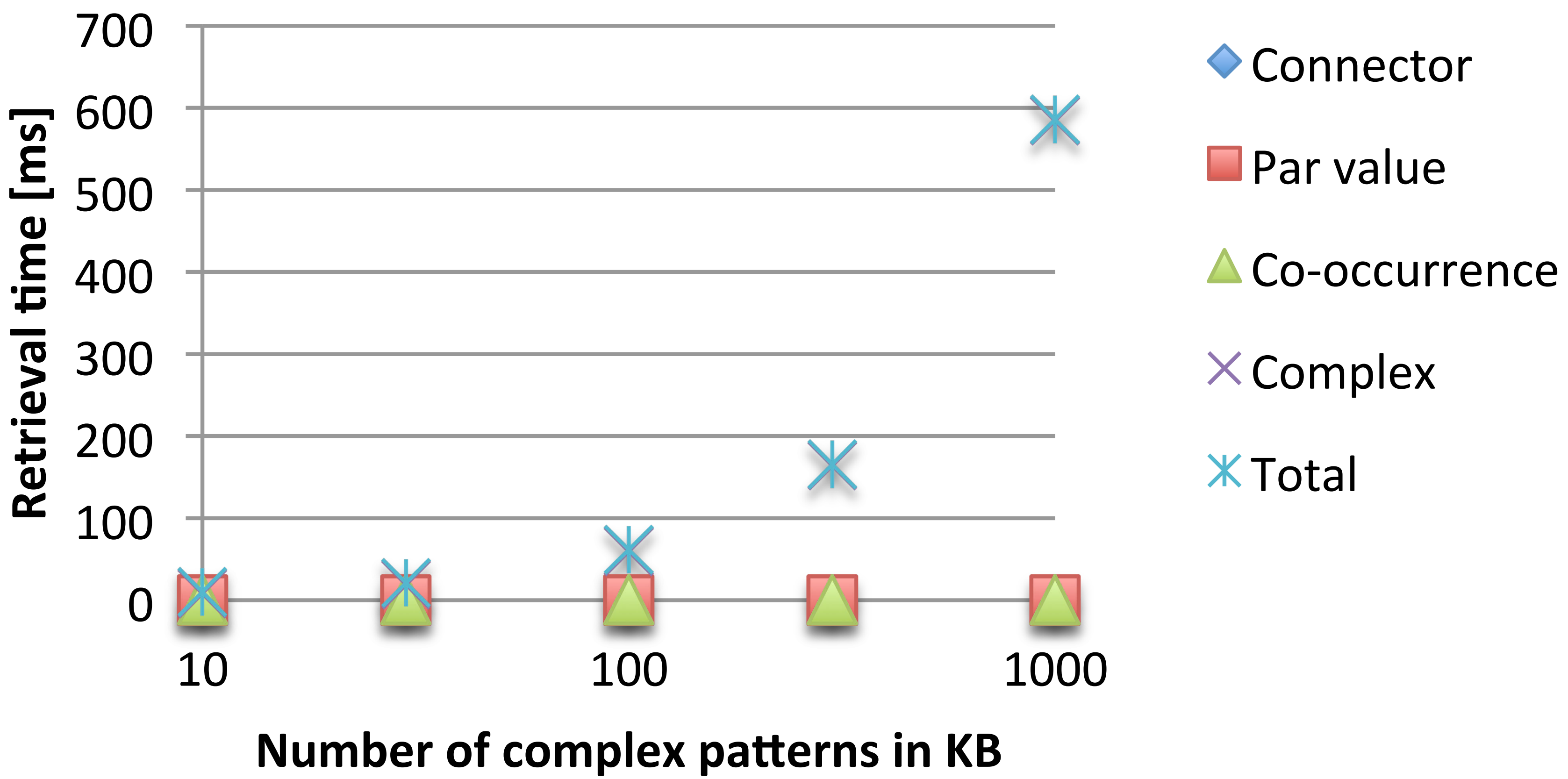
100 pipes models, different from the ones used to mine patterns

Generated **856** test cases with different object sizes:

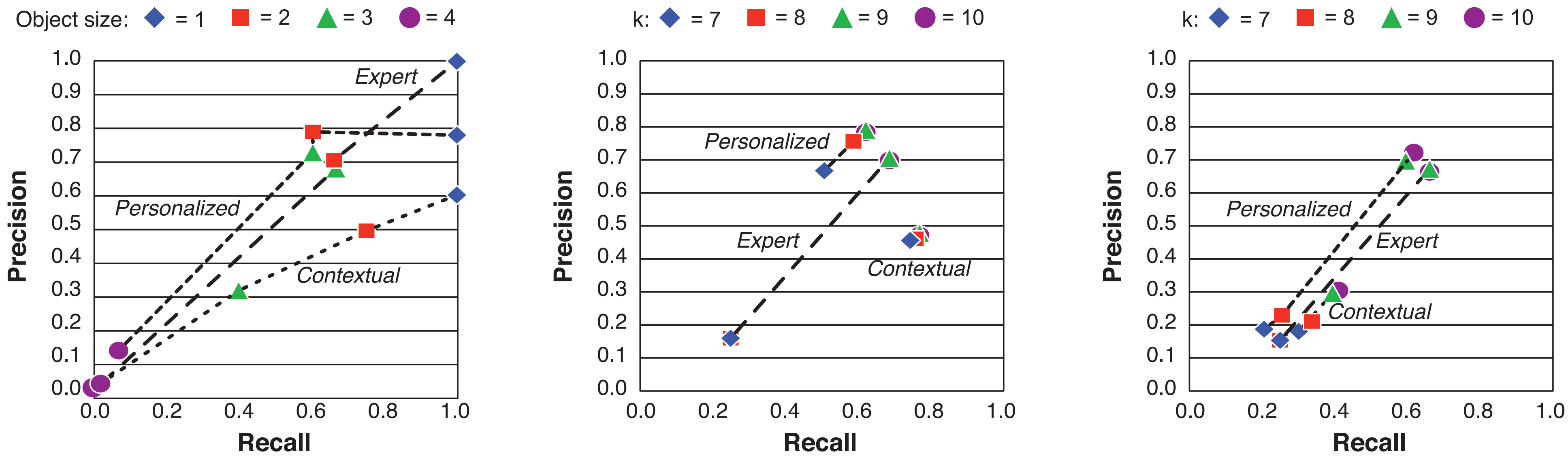
- 356 with object size 1
- 227 with object size 2
- 212 with object size 3
- 61 with object size 4



In-browser **performance** of recommendations



Precision and recall: $P = \frac{|TP|}{|TP|+|FP|}$ $R = \frac{|TP|}{|TP|+|FN|}$

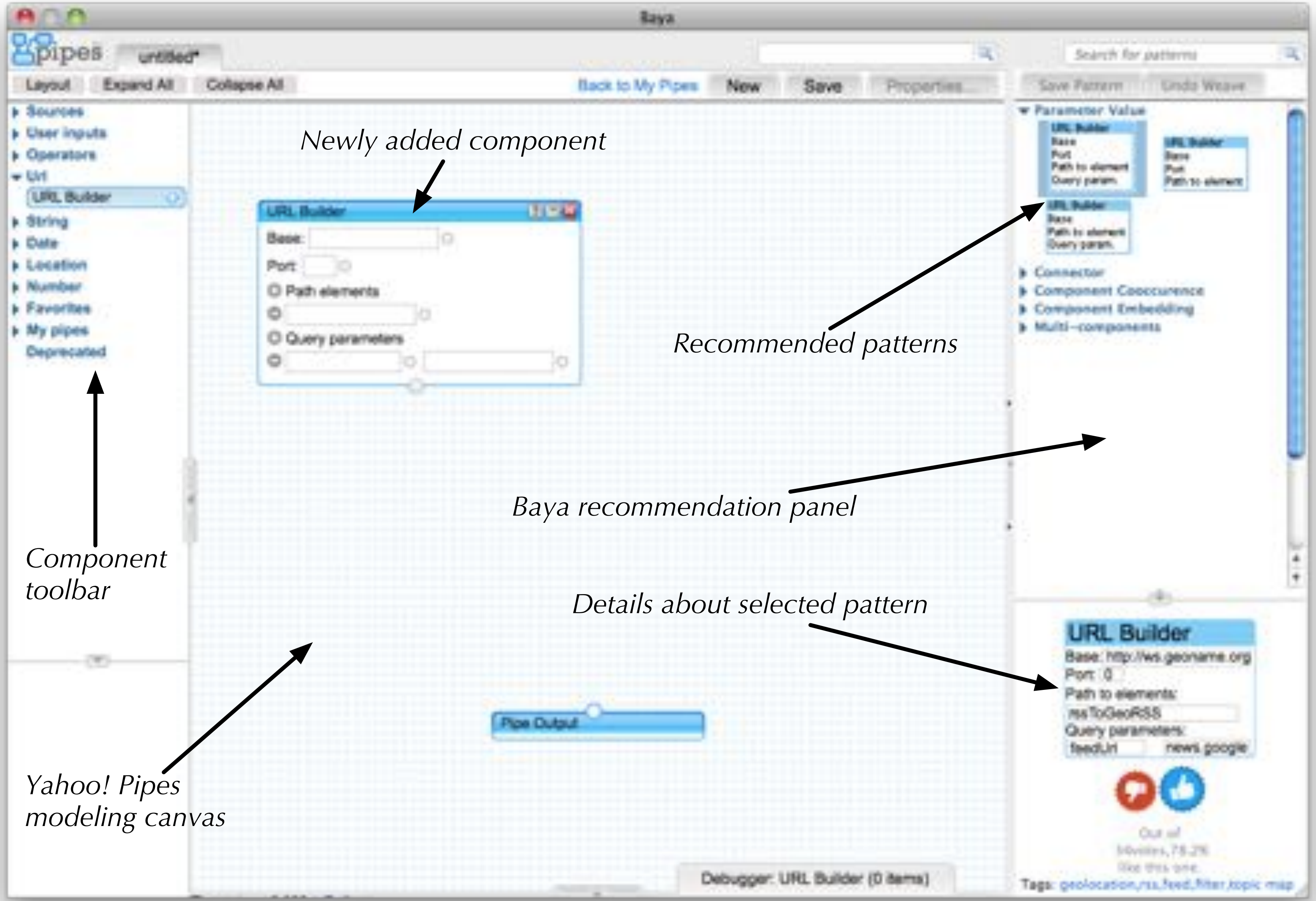


(a) P/R for **varying object size** (k=10) (b) P/R for **varying k** (object size is 2) (c) P/R for **varying k** (object size is 3)

High performance in response to stepwise modeling actions

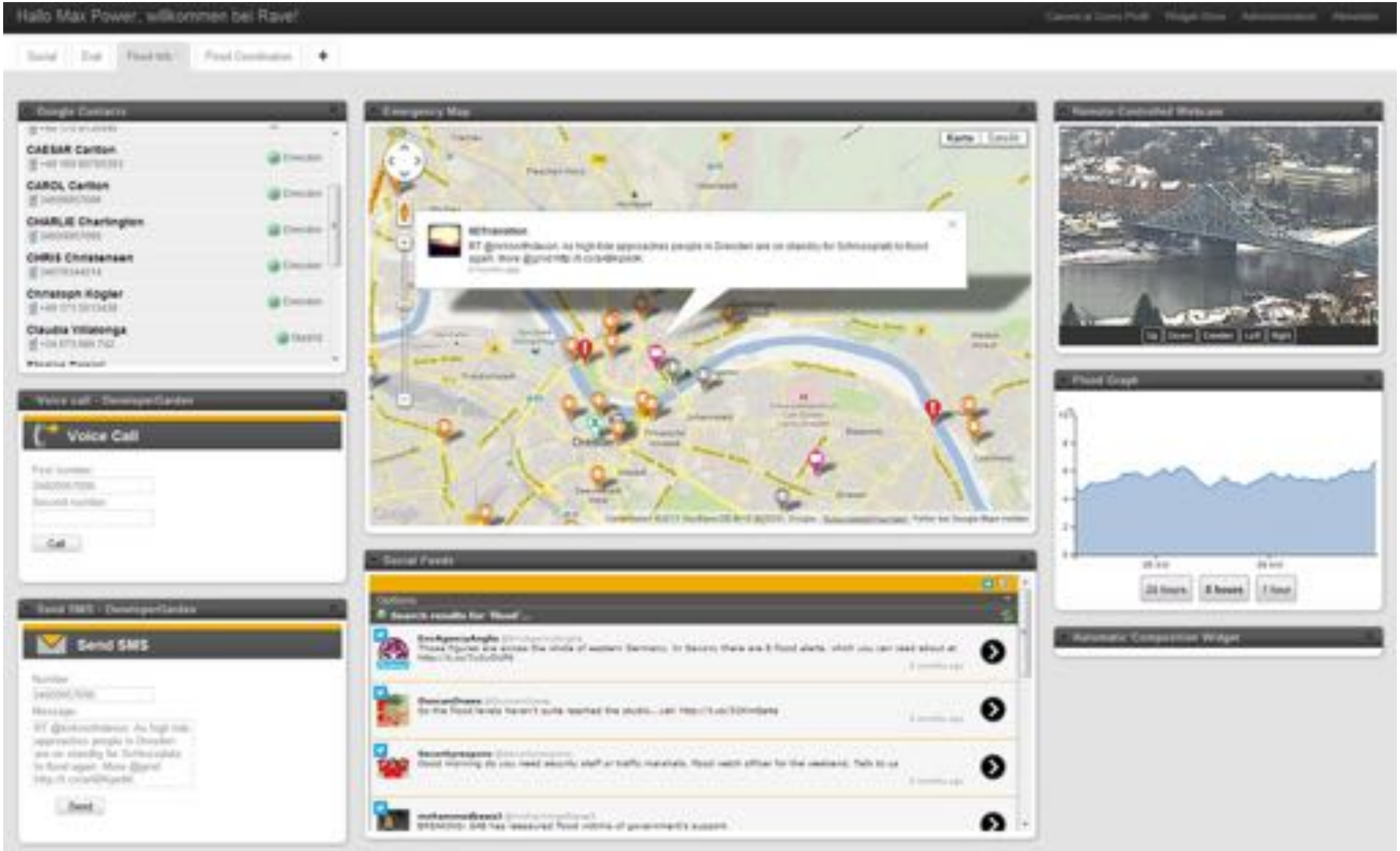
Retrieve at least 8-9 recommendations

baya assisted mashup development as a service = assisted development in Yahoo! Pipes

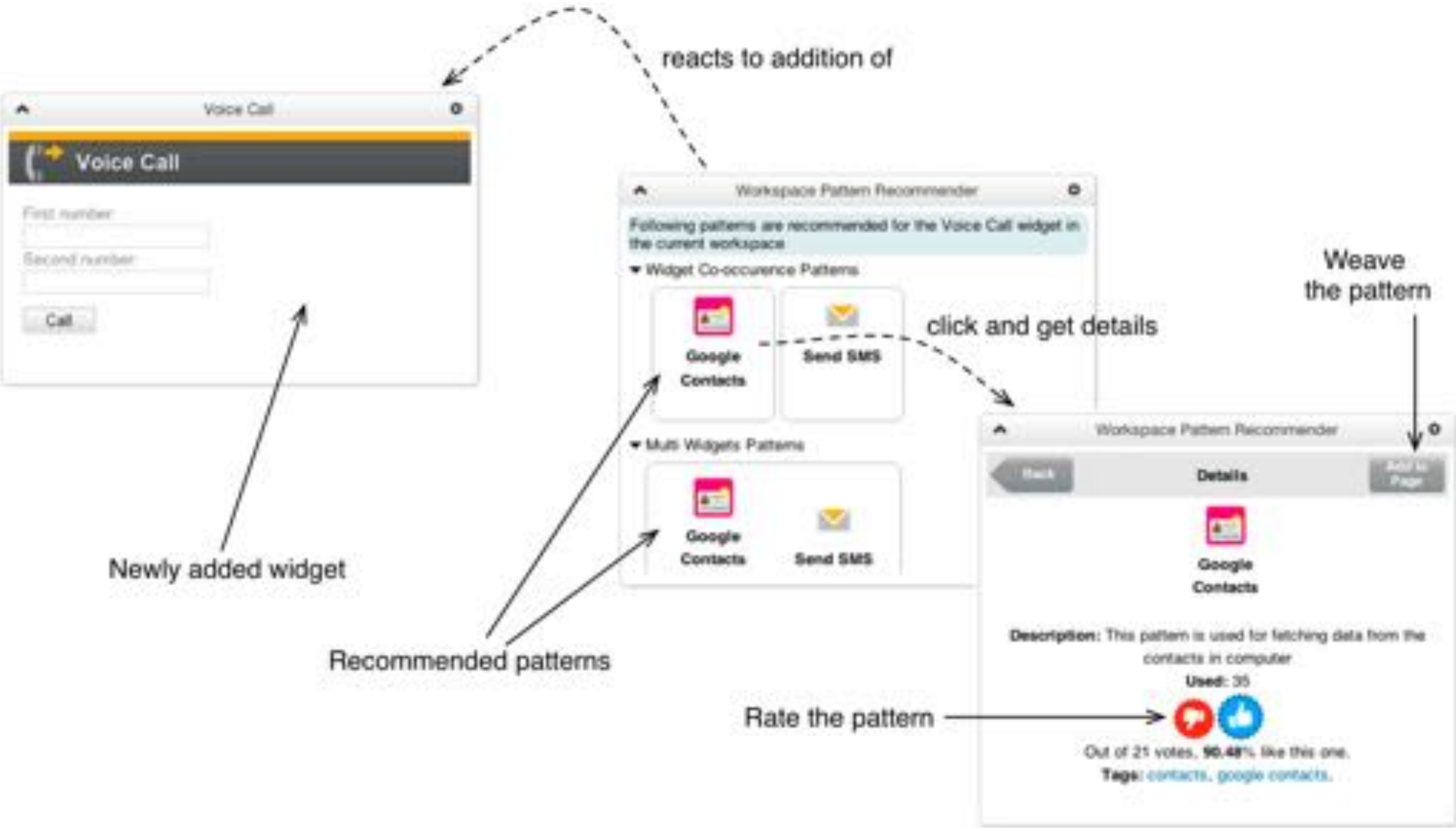


for developers

Omelette = extension of Apache Rave for UI mashups



for end-users

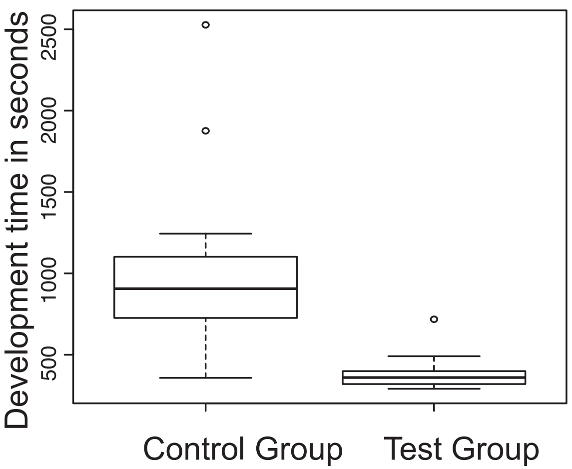


reacts to addition of
Voice Call
Voice Call
First number
Second number
Call
Workspace Pattern Recommender
Following patterns are recommended for the Voice Call widget in the current workspace
Widget Co-occurence Patterns
Google Contacts
Send SMS
Multi Widgets Patterns
Google Contacts
Send SMS
click and get details
Weave the pattern
Workspace Pattern Recommender
Back
Details
Add to Page
Google Contacts
Description: This pattern is used for fetching data from the contacts in computer
Used: 35
Out of 21 votes, 90.48% like this one.
Tags: contacts, google contacts.
Newly added widget
Recommended patterns
Rate the pattern

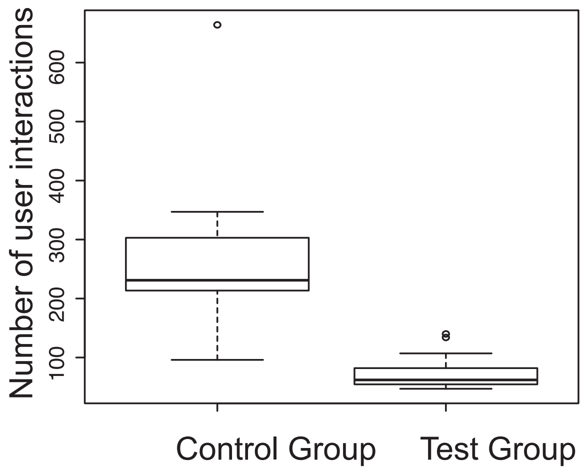
User studies

H1: Baya speeds up mashup development
H2: Development with Baya requires fewer user interactions
H3: Development with Baya requires less thinking time

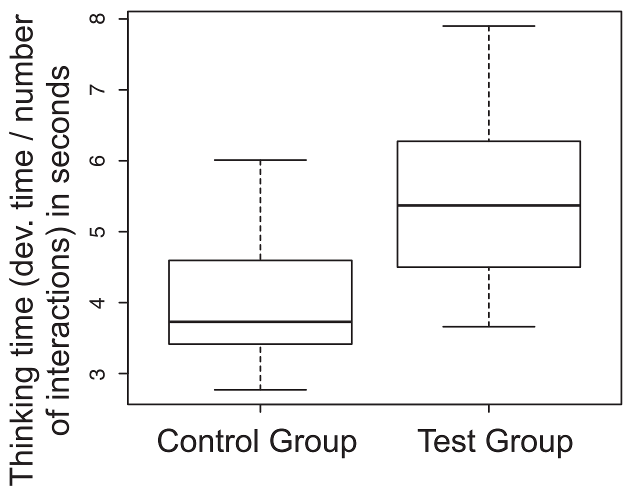
Crowdsourced user study (Amazon Mechanical Turk):
30 participants equally split into control and test group (developers)



(a) mean development time in designing a pipe without and with Baya



(b) mean number of user interactions in designing a pipe without and with Baya



(c) mean thinking time for designing a pipe without and with Baya

$$\mu_{dev,ctrl} = 1027.1s$$
$$\mu_{dev,test} = 384.9s$$

$$\mu_{int,ctrl} = 258.9$$
$$\mu_{int,test} = 74.3$$

$$\mu_{th,ctrl} = 4.0s$$
$$\mu_{th,test} = 5.5s$$

Retain H1
(p=0.00045)

Retain H2
(p=0.00009)

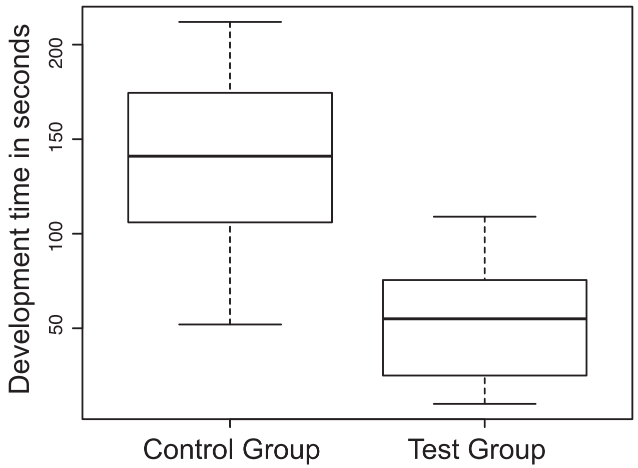
Reject H3
(p=0.00209)

2-sample t-test (Welch): normally distributed samples, unequal variances

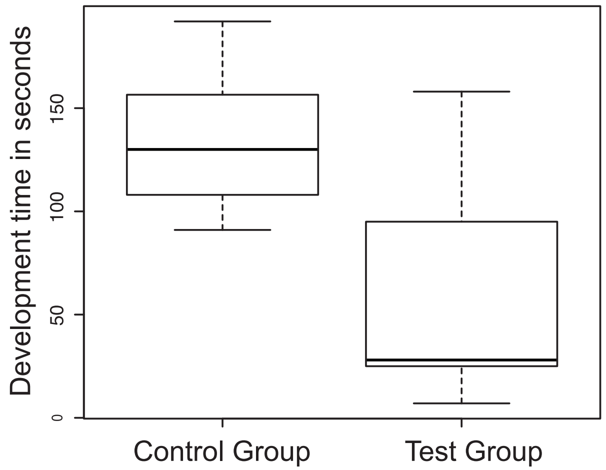
Independent user studies by partners in the EU FP7 project Omelette: Baya for Apache Rave, 44 participants (admins)

HUAWEI

T-Systems Multimedia Solutions



(a) user study with Apache Rave (China)



(b) user study with Apache Rave (Germany)

$$\mu_{dev,ctrl} = 139.8s$$
$$\mu_{dev,test} = 54.6s$$

Retain H1
(p=0.0001)

$$\mu_{dev,ctrl} = 132.1s$$
$$\mu_{dev,test} = 58.9s$$

Retain H1
(p=0.0007)

2-sample t-test (Welch): normally distributed samples, unequal variances

In conclusion

- Recommending and weaving model patterns can really make modelers **more efficient**!
- Baya is a concrete proof of concept and a flexible starting point for others

Mining and Quality Assessment of Mashup Model Patterns with the Crowd: A Feasibility Study

CARLOS RODRÍGUEZ, University of Trento
FLORIAN DANIEL, Politecnico di Milano
FABIO CASATI, University of Trento

Pattern mining, that is, the automated discovery of patterns from data, is a mathematically complex and computationally demanding problem that is generally not manageable by humans. In this article, we focus on small datasets and study whether it is possible to mine patterns with the help of the crowd by means of a set of controlled experiments on a common crowdsourcing platform. We specifically concentrate on mining model patterns from a dataset of real mashup models taken from Yahoo! Pipes and cover the entire pattern mining process, including pattern identification and quality assessment. The results of our experiments show that a sensible design of crowdsourcing tasks indeed may enable the crowd to identify patterns from small datasets (40 models). The results however also show that the design of tasks for the assessment of the quality of patterns to decide which patterns to retain for further processing and use is much harder (our experiments fail to elicit assessments from the crowd that are similar to those by an expert). The problem is relevant in general to model-driven development (e.g., UML, business processes, scientific workflows), in that reusable model patterns encode valuable modeling and domain knowledge, such as best practices, organizational conventions, or technical choices, modelers can benefit from when designing own models.

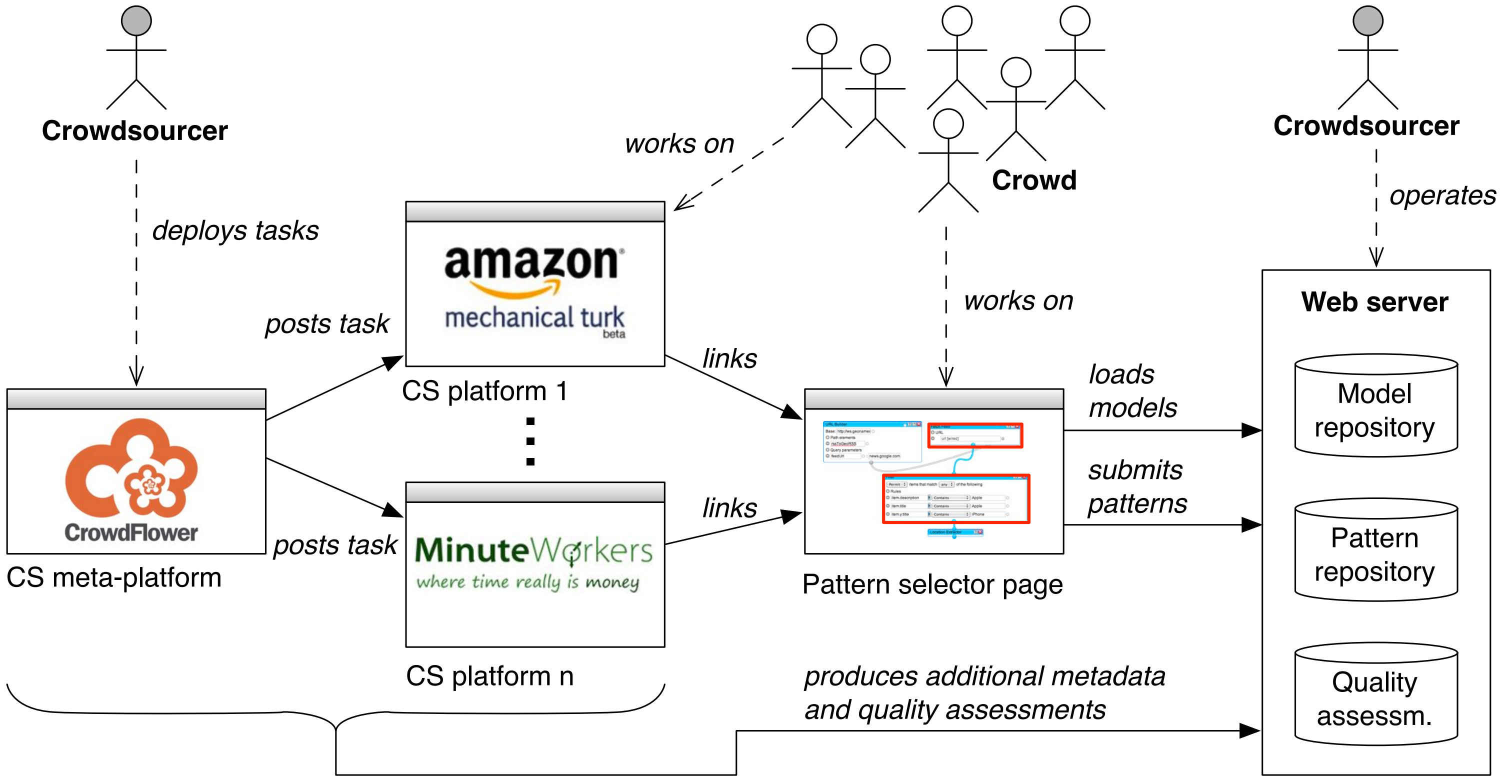


C. Rodríguez, F. Daniel, F. Casati. Mining and Quality Assessment of Mashup Model Patterns with the Crowd: A Feasibility Study. *ACM Transactions on Internet Technology*, 2016, in print.

Research questions

1. Is the crowd able to **discover** meaningful, reusable mashup model patterns?
2. Is it possible to crowdsource the **quality assessment** of identified patterns?

Architecture



Experiment 1: pattern identification

Similar **dataset** as in previous study: 997 pipes models

- 40 randomly picked models for the crowd (*Crowd*)
- 997 for the automated algorithm (*Machine*)

Evaluation **metrics**

- Number of patterns identified
- Avg pattern size (# components)
- Distribution of pattern sizes
- Cost per pattern

Crowd task **designs**

Naive: shows one pipe and asks for a pattern

Random3: shows 3 pipes and asks for a pattern

ChooseN: shows 10 pipes and asks to choose N pipes and to identify a pattern

+ Automated mining **algorithm*** for comparison

* C. Rodriguez, S. Roy Chowdhury, F. Daniel, H.R. Motahari Nezhad and F. Casati. Assisted Mashup Development: On the Discovery and Recommendation of Mashup Composition Knowledge. In *Web Services Foundations*, Springer, 2014, Pages 683-708.

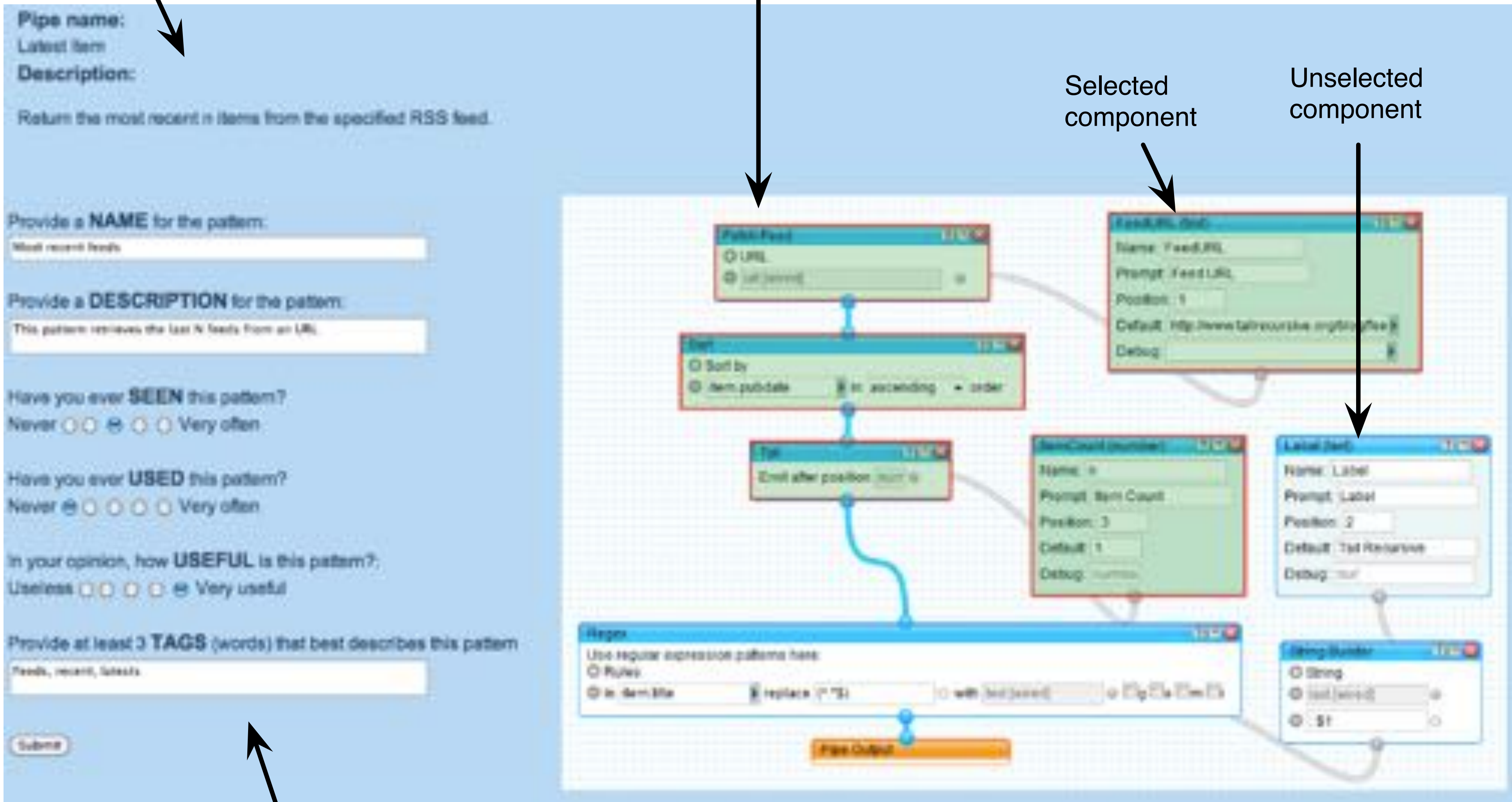
Screen shot of the **Naive** task design

Name and description of pipe sources from Yahoo! Pipes

The pipe model to be analyzed by the worker. The model is a clickable image map that allows the worker to define a pattern by selecting its components.

Selected component

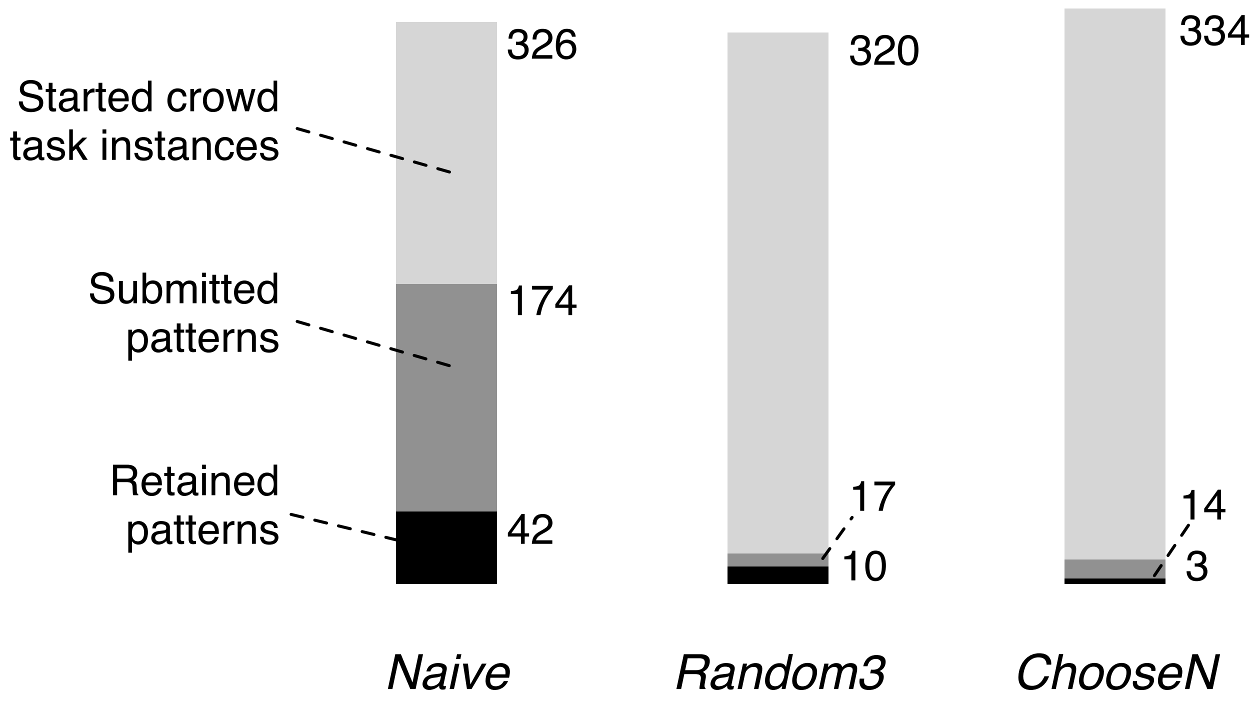
Unselected component



Additional input fields for the specification of pattern name, description and meta-data

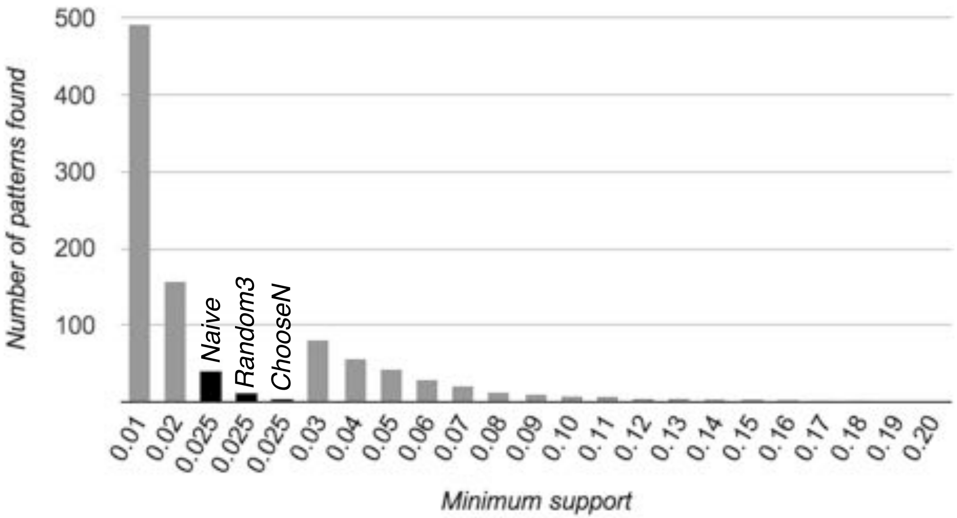
Results

Crowd task instances vs. patterns collected

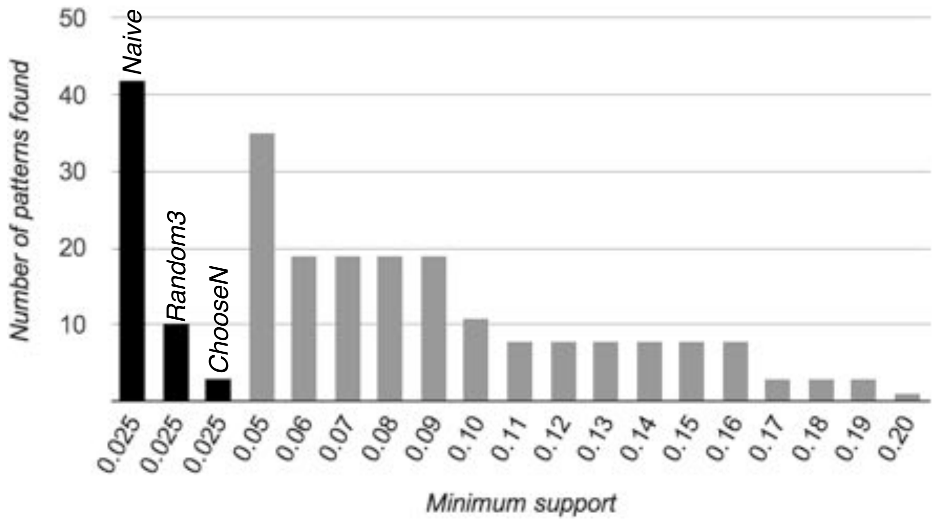


Retained patterns = **valid patterns**, manually checked

Number of patterns identified



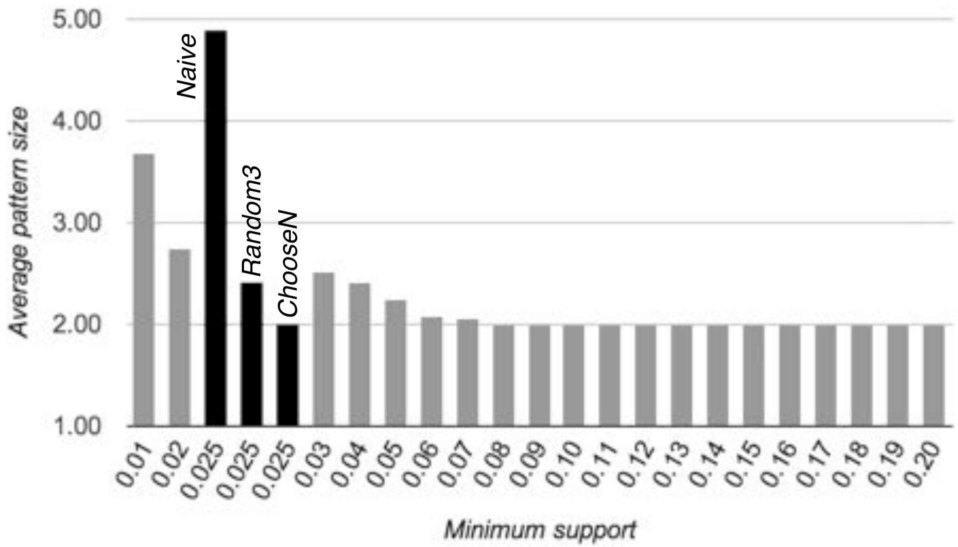
(a) *Machine*^{997}(gray) compared to the crowd (black)



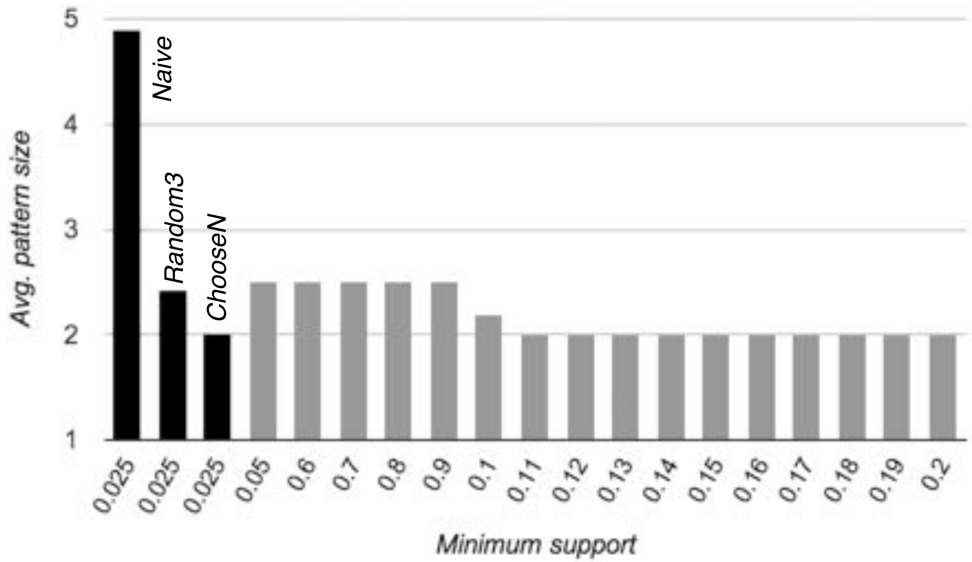
(b) *Machine*^{40} (gray) compared to the crowd (black)

—> Yes, it is possible to identify patterns with the crowd

Average pattern sizes



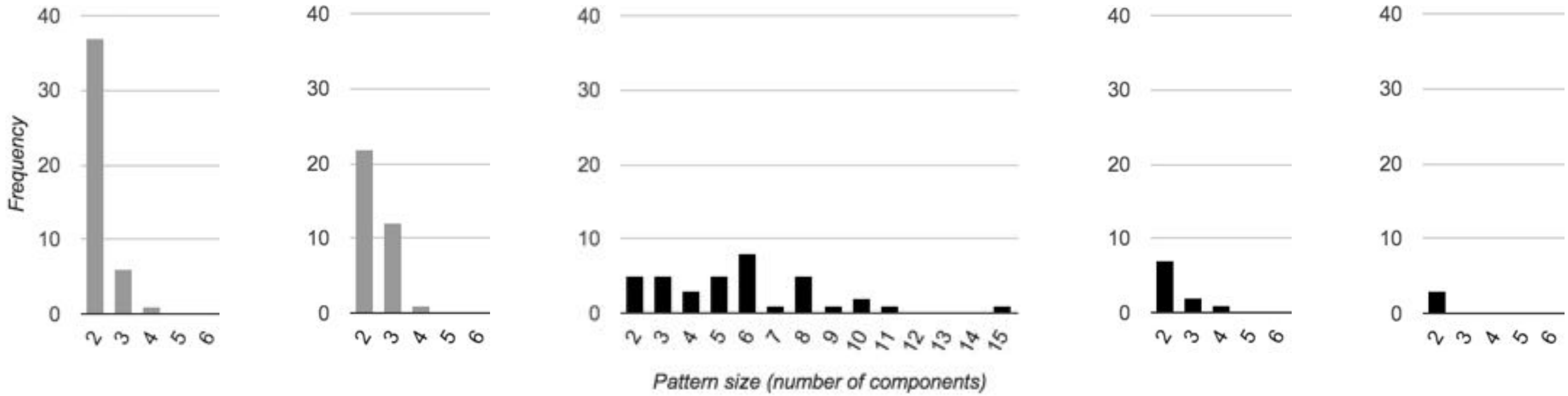
(a) *Machine*^{997}(gray) compared to the crowd (black)



(b) *Machine*^{40} (gray) compared to the crowd (black)

—> The patterns identified by the crowd are in average bigger

Distribution of pattern sizes



(a) *Machine*^{997}: 44 patterns with support of 0.05

(b) *Machine*^{40}: 35 patterns with support of 0.05

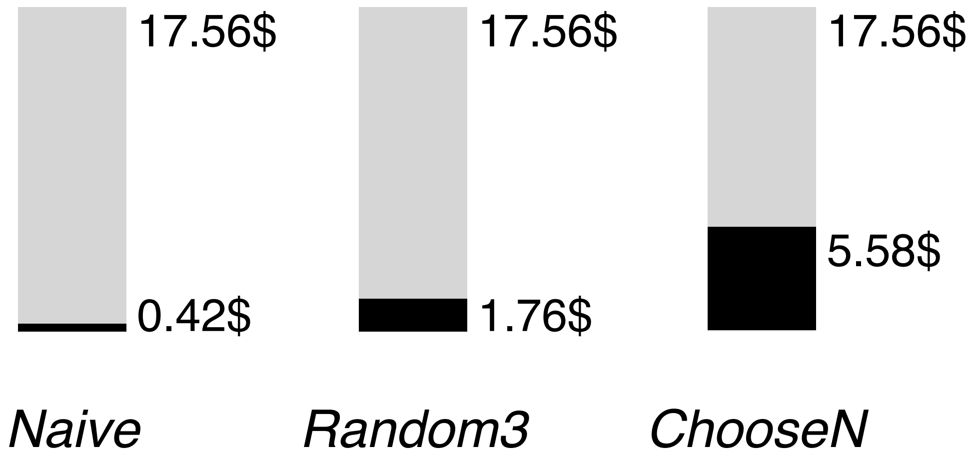
(c) *Naive* with 42 patterns

(d) *Random3* with 10 patterns

(e) *ChooseN* with 3 patterns

—> The domain knowledge captured by Naive is even complex

Cost



—> The approach is cost-effective

Experiment 2: quality assessment

Dataset = output of best crowd mining approach of Exp 1

Pattern assessment **metrics**

- Reusability
- Novelty
- Usefulness
- Understandability

Crowd task **designs**

Individual: asks for assessment of the for metrics, given one pattern

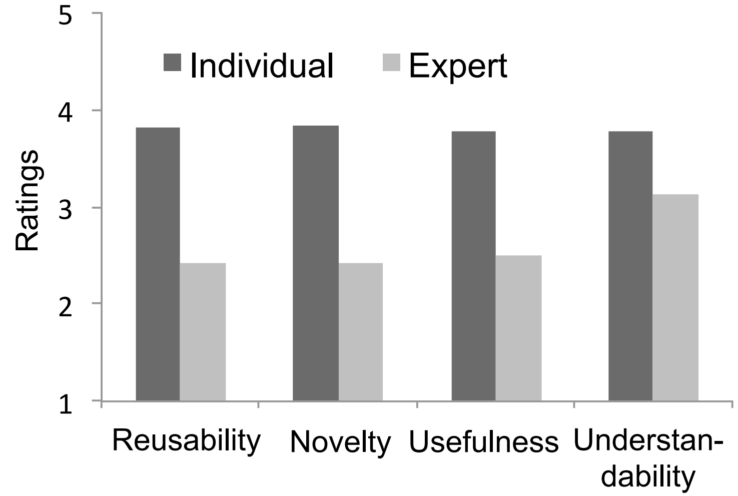
Pair-wise: asks for each metric to choose which of two given patterns is better

\+ **Expert assessment** for comparison

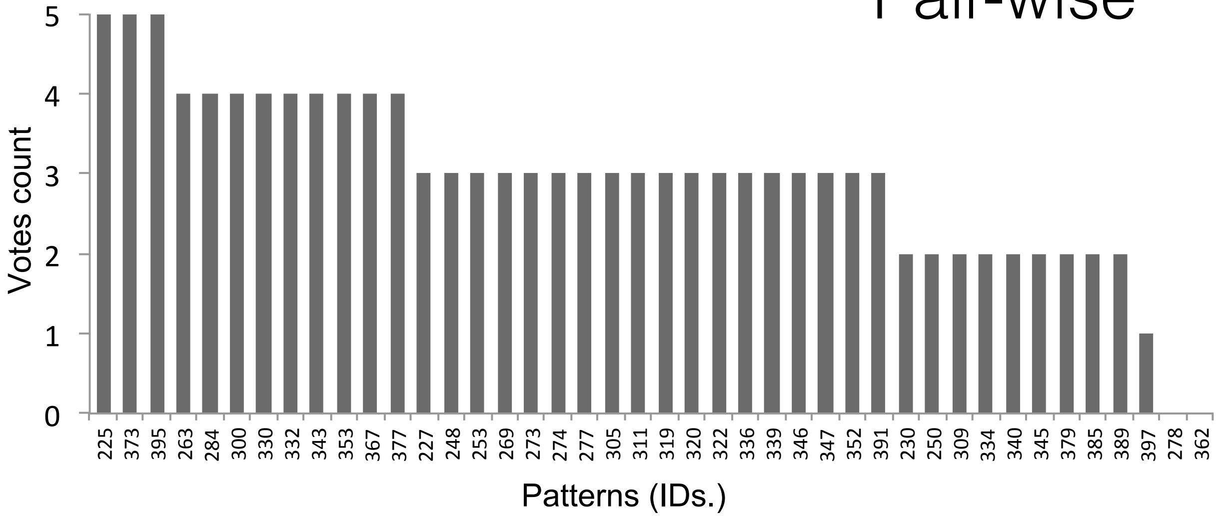
Results

Replaceability of experts

Pair-wise



(a) *Individual* vs. *Expert* rating (avg of Likert ratings)



(b) *Understandability* pattern ranking in decreasing order of aggregated votes

		Spearman's correlation coefficients ρ	
Criteria	**Mann-Whitney's test**	***Expert* vs. *Individual***	***Expert* vs. *PairWise***
Reusability	$p = 5.787 \times 10^{-9}; U = 8029$	-0.0783	0.1581
Novelty	$p = 3.287 \times 10^{-13}; U = 8741$	0.1212	-0.1017
Usefulness	$p = 6.392 \times 10^{-10}; U = 8197$	0.0257	0.1755
Understandability	$p = 5.744 \times 10^{-4}; U = 6870$	0.0732	0.1403

—> Individual does **not** produce anything like the experts

—> Pair-wise does **not** produce anything like the experts

Precision and recall

$$P = \frac{TruePos}{TruePos + FalsePos} \qquad R = \frac{TruePos}{TruePos + FalseNeg}$$

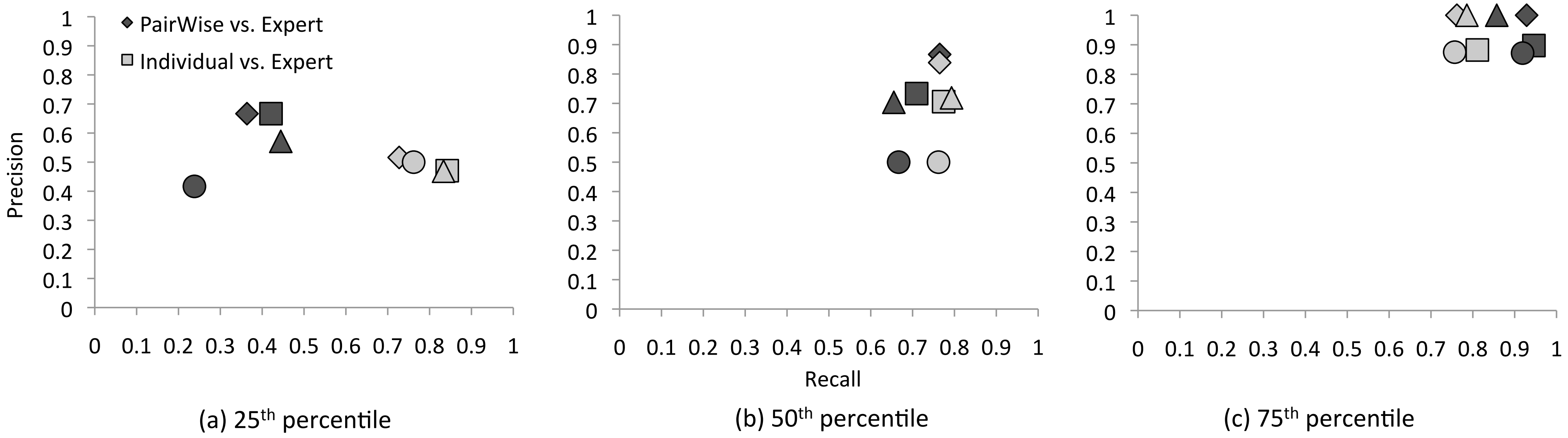


Fig. 11. Precision and recall of the *Individual* and *PairWise* assessment experiments with varying selectivity (top 25, 50, 75 percentiles) for understandability (□), usefulness (◇), reusability(△), novelty (○).

—> The approaches could be used to filter out the worst patterns

Conclusion

- Using suitable task designs, **the crowd is able** to identify meaningful model patterns.
- More visibility into the dataset (e.g., to spot **repetitions**) does not help, to the contrary.
- We were not able to obtain reliable **quality assessments** from the crowd.

The real conclusion

- Model patterns can **really help** if suitably recommended and used
- The key problem is **finding** good patterns
- The **crowd** may be a viable alternative (or complement?) to computational approaches in identifying patterns