

Extracting Change patterns from CVS Repositories

Salah Bouktif, Yann-Gaël Guéhéneuc, and Giuliano Antoniol

WCRE 2006

Benevento, Italy



Summary

■ Our work

- Introduces the concept of **change patterns** to analyse evolution information
- Defines and identifies the **Synchrony** change pattern using the DTW technique
- Evaluates the extraction of the Synchrony change pattern wrt. a **golden standard**
- Eases tracking changes in time- and geographically-distributed software development processes, *e.g.*, **open source**



Context and Problem

- Time- and geographically-distributed software development is common
- Difficulty in sharing timely information about changes among developers
- Version control systems contains millions of fine-grained language-independent changes difficult to track and to organise

Definitions

(1/2)

■ Change patterns

(Similar in the idea to design patterns)

- Common and recurring changes during the evolution of a software
- Reification of explicit and implicit relationships among software artefacts in the version control system

Definitions

(2/2)

■ Synchrony change patterns

- Files that change at **almost** the same moments in time
- Co-changes that happened in the past are likely to occur in the near future
- Explicit and implicit dependencies among files remain stable over time

“If this particular file changes, what other files should change?”

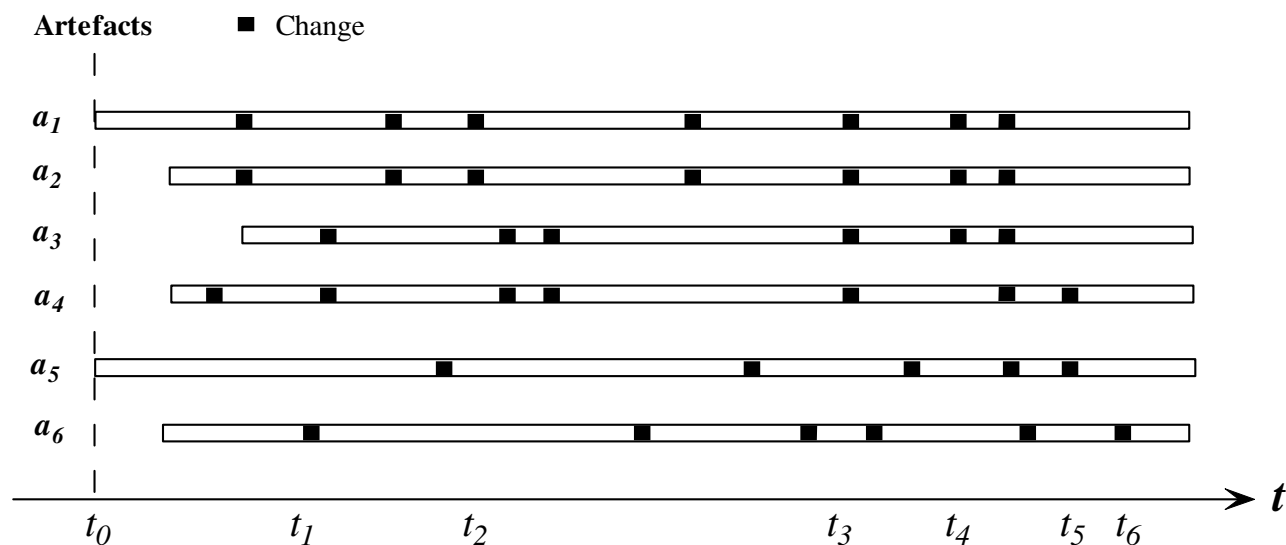


Related Work

- Gall
 - Evolution patterns
 - Growth and change of behaviour
- German
 - Modification requests, interrelationships
- Beyer
 - Interpretable graph layouts
- Zimmermann
 - Co-changing fine grain entities using data mining
 - Limited precision and recall

Difficulties

- Version configuration systems, *i.e.*, CVS, record change
 - Line of code
 - 1/1,000th of seconds

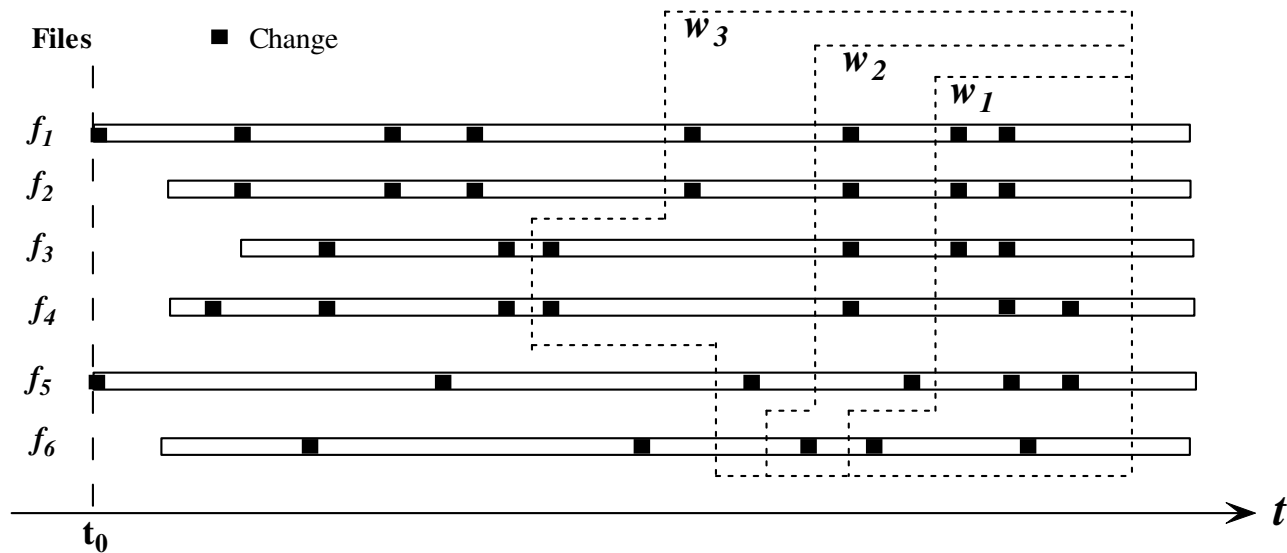
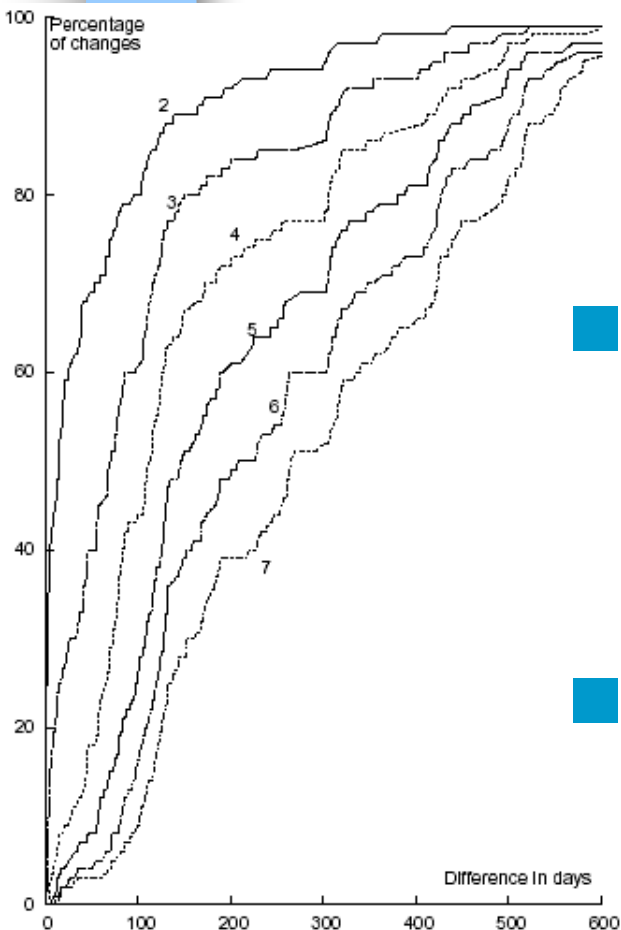




Solutions

- Choosing software artefacts
 - Files, for simplicity
 - Could be other artefacts, *i.e.*, class...
- Computing appropriate window size
- Grouping co-changing files with DTW

Window Size



- Different co-changes in time
 - Keeping all changes
 - Missing recent co-changes
- Most files are changed frequently
 - Window of **5 to 7** changes

Grouping Co-changing Files(1/3)

- DTW: Dynamic Time Warping

$$S_1(n), n = 1, 2..N, S_2(m), m = 1, 2..M.$$

$$W = w_1, w_2, \dots, w_P$$

Such as

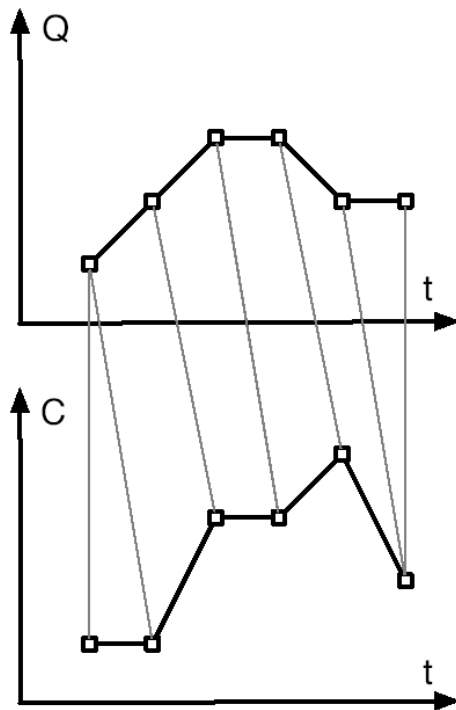
$$\max(N, M) \leq P < N + M \quad w_k = (i, j)$$

$$w_1 = (1, 1) \text{ and } w_K = (N, M)$$

$$Dist(W) = \sum_{k=1}^{k=P} Dist(w_{ki}, w_{kj})$$

Grouping Co-changing Files(2/3)

- Example with Euclidian distance
 - $Q=(1\ 2\ 3\ 3\ 2\ 2)$ and $C = (1\ 1\ 3\ 3\ 4\ 2)$



	1	1	3	3	4	2
1	0	0	4	4	9	1
2	1	1	1	1	4	0
3	4	4	0	0	1	1
3	4	4	0	0	1	1
2	1	1	1	1	4	0
2	1	1	1	1	4	0

Dynamic Time Warping
Path Cost = 2

Grouping Co-changing Files(3/3)

$threshold \leftarrow value$

for each $hist_1 = f_1, f_2, \dots$ **do**

$Set_{hist_1} \leftarrow \emptyset$

for each $hist_2 = f_1, f_2, \dots$ **do**

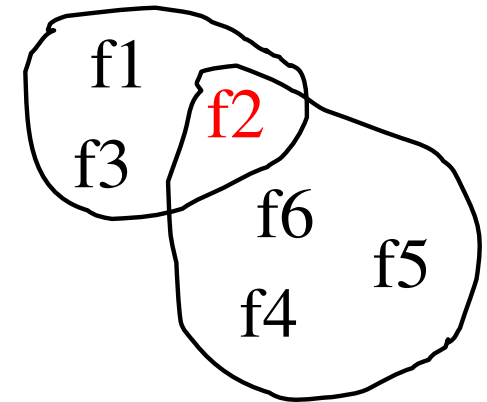
$dist \leftarrow DTW(hist_1, hist_2)$

if ($dist < threshold$) **then**

$Set_{hist_1} \leftarrow Set_{hist_1} \cup hist_2$

endfor

endfor



■ Threshold

- Between 43,200 sec. and 86,400 sec. \Rightarrow Balance between precision and recall
- Above 86,400 sec. \Rightarrow Recall over precision
- An artefact may belong to more than one group



Case Study

(1/8)

■ Objectives

- Compute internal precision and recall
 - k-fold cross validation
- Compute external precision and recall
 - Comparison with a golden standard provided by an expert
 - The expert was not aware of the goal nor of the applied process and methodology
- Assess predictive power
 - Prediction of changes with the golden standard
- Study the scalability of the approach
 - k-fold cross validation on large program



Case Study

(2/8)

- Measures
 - Precision
 - Recall
- Query = File to be changed
- Retrieved documents = Co-changing files
- Average and weighted precision/recall



Case Study

(3/8)

- Object of the case study
 - PADL
 - Meta-model to describe OO programs
 - 3 years of development
 - 91 files with a history of a least 5 changes

Case Study

(4/8)

■ Internal precision and recall

– Building disjoint test and training sets

- Maximum length of 7
- Testing sets a–b–c ($c < 5$, $a > 1$, $b > 1$, $c > 1$)
- a, b, and c are numbers of changes in histories of length 5, 6, and 7 used to build test sets
- 2-3-3 means test sets with the 2 most recent changes for history of length 5, 3 for length 6...

	$Prec_w$	$Prec_a$	Rec_w	Rec_a
Internal Evaluation	84.81% (7.02)	85.91% (2.47)	71.86% (6.98)	59.32% (0.26)

Case Study

(5/8)

■ External precision and recall

– Golden standard from PADL

- Conservative approach

```
./src/padl/kernel/impl/MemberClass.java
```

```
-> ./src/padl/kernel/impl/MemberClass.java
```

```
./src/padl/kernel/impl/MemberInterface.java
```

```
./src/padl/kernel/impl/ContainerAggregation.java
```

	$Prec_w$	$Prec_a$	Rec_w	Rec_a
External Evaluation	78.98% (6.52)	91.10% (2.54)	76.89% (1.66)	64.77% (7.03)

Case Study

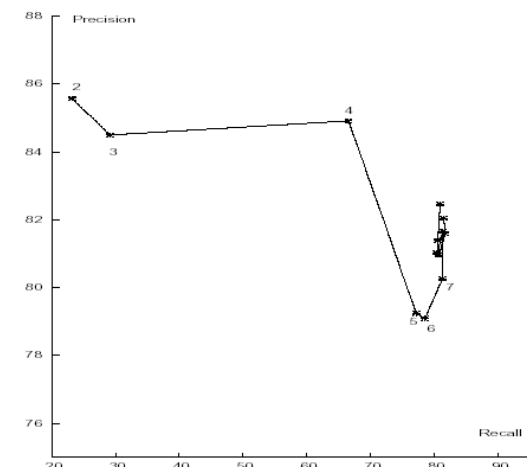
(6/8)

■ Predictive power

– Real application for prediction

- Take the last 7 changes
- Ignore the most recent change
- Use the 6 changes to predict the recent change

Windows	$Prec_w$	Rec_w
4	66.54%	84.92%
5	77.20%	79.20%
6	78.00%	78.00%



Case Study

(7/8)

■ Scalability of the approach

– PADL

- 91 files, 3 years

– Mozilla Web browser

- Mirrored on July 15, 2005
- Versions of the next 5 days
- More than 20,134 files (in 2,480 directories)
- 3 minutes

$Prec_w$	$Prec_a$	Rec_w	Rec_a
73.58%	82.74%	67.61%	52.18%
(14.52)	(7.89)	(9.14)	(16.14)



Case Study

(8/8)

- Threat to the validity
 - Expert did not participate in the building of the DTW-based groupings
 - Use of same files to identify window size and precision/recall
 - Generalisation



Conclusion

- Change patterns
- Synchrony change patterns
- Dynamic Time Warping
 - Increased precision and recall over previous work, in particular Zimmermann
 - External: 78.98% precision, 76.89% recall
 - Prediction: 77.20% precision, 79.20% recall



Future Work

- Analysis of the strength of the groups
- Study of fluctuations in precision/recall
- Perform more external validations
- Apply clustering techniques
- Compare time intervals with windows based on the number of changes