



From imagination to impact



Australian Government  
Department of Broadband, Communications  
and the Digital Economy  
Australian Research Council

NICTA Members



Department of State and  
Regional Development



The University of Sydney



Queensland  
Government



NICTA Partners

# Detection of Consistent Patterns from Process Enactment Data

by

Ming Huo, (Jason) He Zhang and Ross Jeffery



# Outline



- Introduction, motivation and research aims
- Related work
- Process recovery approach
- Case study and findings
- Discussion
- Conclusion and future work

## Predefined process models:

- A planned process model that a development team intends to follow during project development.
- Traditional process models:
  - waterfall model, spiral model, etc
- ISO standards, such as ISO 12207



## Process enactment data:

- the data on project implementation
- the data records on how the development team follows a process model

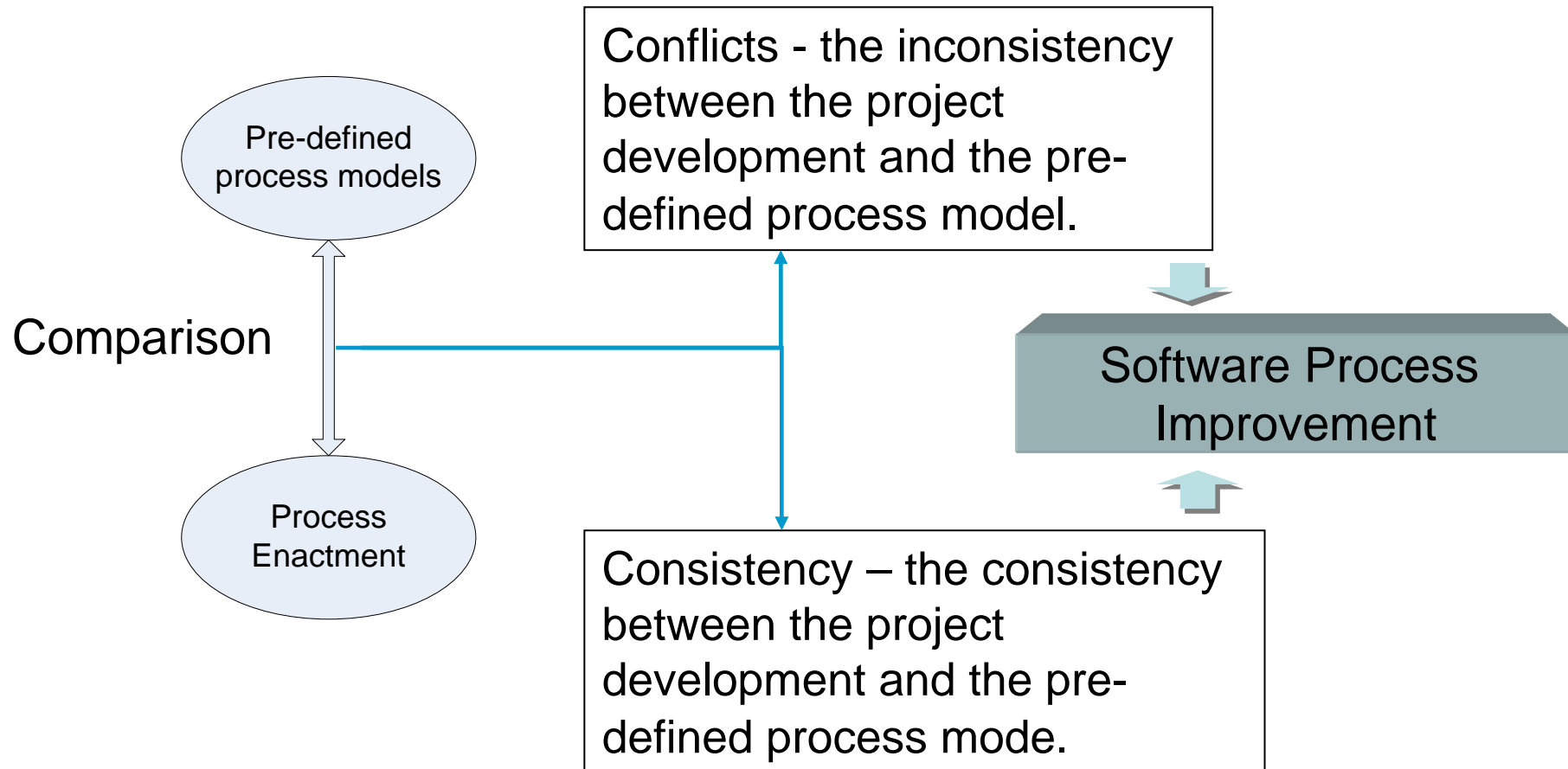
Project managers and researchers are curious as to:

1. what is really happening in the implementation of the model?
2. is the process model applied to a project and how appropriate it is? and
3. is there a distance between a planned process and the project enactment ?

Main research aims are:

- 1) to re-cover a process model fully or partially from its enactment data,
- 2) to find out the gap between the process enactment and pre-defined process as input to software process improvement.
- 3) to evaluate the appropriateness of a predefined process model

# Introduction, Motivation and Research Aims



# Outline



- Introduction, motivations and research aims
- **Related work**
- Process recovery approach
- Case study and findings
- Discussion
- Conclusion and future work

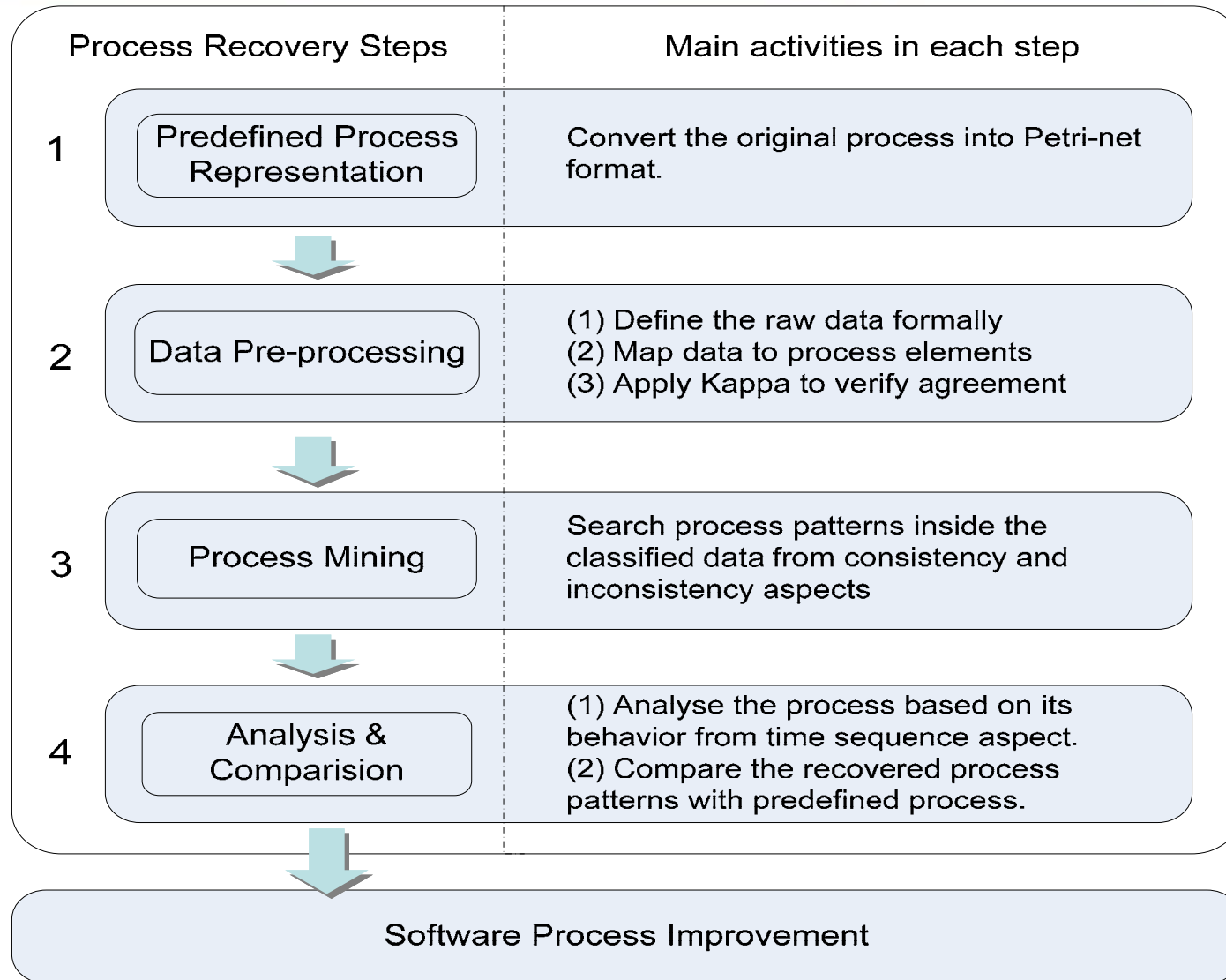
- Related work
  - Process mining intends to re-discover a process model from people's daily activities. (See also APSEC 2006 paper on process conflicts and patterns)
  - Limitations of current work in process mining:
    1. Large data amount are required.
      - Hundreds of process instance data have to be collected to discover a process model.
      - Hence, it is very difficult for us to have sufficient data for software process mining.
    2. The discovered model is a very low level process model
      - Discovered model is defined generally in the terms of daily activities.
      - Pre-defined software process models are the higher-level models, such as the waterfall model, spiral model or ISO standard models
      - A distinct distance between the low level recovered model and the higher-level pre-defined software process models.

# Outline



- Introduction, motivations and research aims
- Related work
- **Process recovery approach**
- Case study and findings
- Discussion
- Conclusion and future work

# Process Recovery Approach Overview



# Process Recovery Approach – Assumptions



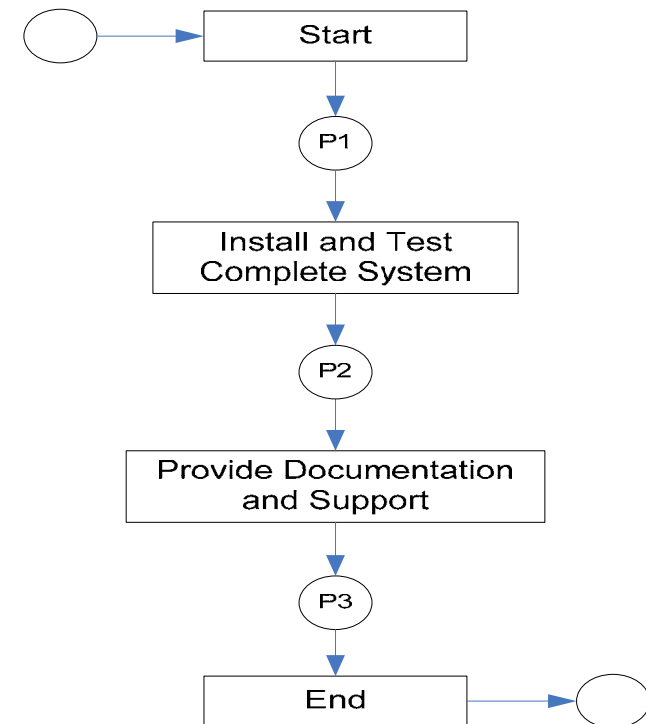
- Basic assumptions:

- A process model is set up for the development team to be followed in the company. It is the predefined process model.
- There should be a set of well defined steps in a process model, i.e., process **elements**; and the process **elements** might be sequentially connected;
- The moving action from one process element to another is a **transition**.
- The path presenting a transition is called a **place**; and there is only one place between two process elements
- One process element may have more than one transitions to multiple different process elements.

## ▪ Process representations

- Using notations to present a pre-defined process model in Petri-net format
- The process model consists of a set of sequenced steps/tasks which we call process **elements**  $O$ .  $O$  is a finite set of elements.
- $V$  is subset of the set  $O$ :  $V = \{v_i / v_i \in O\}$
- $E$  is a set of relations on the set  $V$ , in the other words, the transitions among the process elements.
- $P$  is the place and it indicates the paths.
- At this stage, the process model will be represented in Petri-net format:

A net,  $N = (O, P, E)$ .



Note: This picture is not shown in the paper due to page limitation

# Process Recovery Approach – Data Pre-processing



## ■ Process data pre-processing

- Mapping tasks to process element in  $O$  is one of the important aspects in our approach.

  - the mapping results must be accurate as input to next stage of our approach and the verification of the results is required.

- Task classification can only be done manually

  - these tasks are fairly low level data with some detailed descriptions, e.g. start time, end time, its comments and etc.

- Discovering the relationship of tasks and their sequence

  - the relationship of predefined process elements can be reflected from the enactment data only after the tasks are classified to process elements,

  - because the process mining can only be applied to tasks.

## ■ Verification of task classification:

- Kappa is a chance-corrected measure of agreement among the 3 raters used. The Kappa value is computed as below.

- Process patterns mining

- We focus on the temporal relations to investigate whether there are two tasks that have temporal dependency.
  - For example, if one task is always followed by another task, it is likely that a temporal relation exists between them.
- Given two tasks  $t_i$  and  $t_j$  we say  $t_i$  and  $t_j$  have temporal relation  $t_i \rightarrow t_j$ :
  - iff  $\text{End}(t_i) < \text{Start}(t_j)$ , and we will add a directed line from  $t_i$  to  $t_j$  in the Petri-Net of tasks.
- The temporal relationship will be searched from the whole data set of one process instance.
  - While accumulating the relationship, the recovered process patterns are generated piece by piece.

# Process Recovery Approach – Measurement



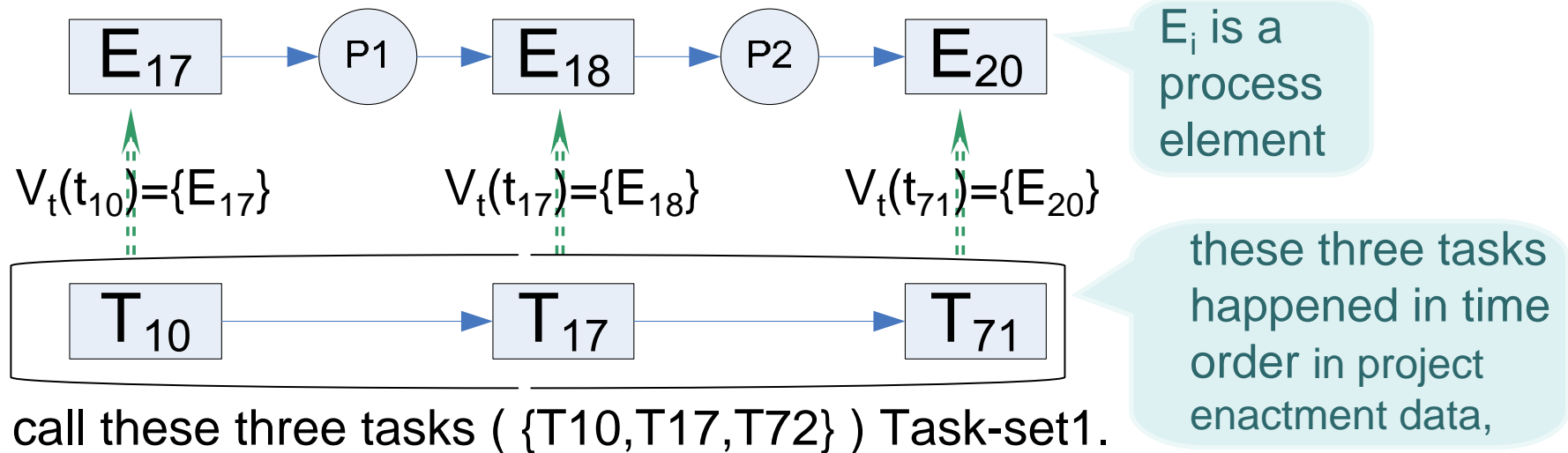
- Two types of patterns are mined out
  - conflicts: This has been studied in our previous paper (APSEC2006).
  - consistency patterns: This is studied here.
- Frequency of each pattern is measured
  - The frequency is determined by its unique supporting task-sets.
  - The frequency is the number of the unique supporting task-sets.
  - Counting the frequency assists us in identifying the heavily adopted parts of a pre-defined process model.
  - The frequency provides quantitative information on the usage of a pattern.
- An example of how a frequency is counted in next slide

# Process Recovery Approach – Measurement



$P_{tn} = \{E_{17}, E_{18}, E_{20}\}$

The frequency of this pattern is 1.



We call these three tasks (  $\{T_{10}, T_{17}, T_{71}\}$  ) Task-set1.

There is another task set: Task-set2

1. contains three tasks
2. also support the pattern
3. not equal to Task-set1

There are two task sets (Task-set1 and Task-set2) that support this pattern  $P_{tn}$ .  
Therefore, the frequency of this pattern can go up to 2.

# Outline



- Introduction, motivations and research aims
- Related work
- Process recovery approach
- **Case study and findings**
- Discussion
- Conclusion and future work

# Case Study – Background



- Case study background

- An Australian software firm customized ISO 12207 and applied a web-accessible electronic process guide (EPG) based on it as their fundamental process model.

The typical information of one task consists of:

1. project the task belongs to,
2. task start & end date,
3. assignee, the person who executes it,
4. brief comments made by the operator.

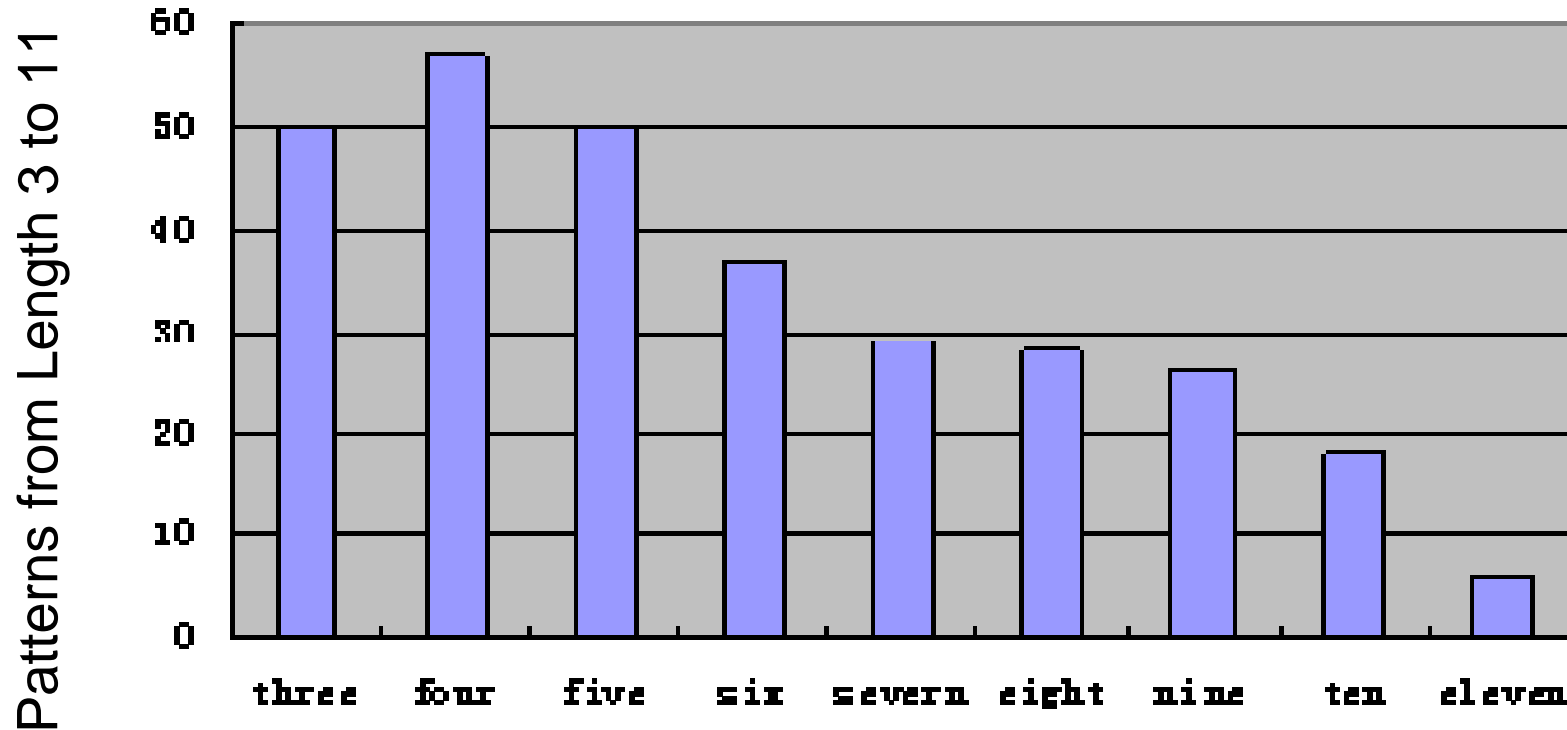


<b>Project No</b>	2298
<b>Start Date</b>	2000-9-7
<b>End Date</b>	2000-10-29
<b>Employee No.</b>	5
<b>Description</b>	Code: Create the HTML pages for the find tool.

- Pilot project information:

Project Information		
Team size	10 persons	
Total Effort	2212.25 man-hours	
Duration	717 days	
Closed year	2002	
Total task number	138	
Task Classification	EPG Tasks	106 (82%)
	None-EPG Tasks	23 (18%)
Kappa Agreement	0.810	

# Case Study – Discoveries from Pattern Analysis



- The detected patterns are grouped by their lengths.
- Each column presents the amount of patterns in different length.
- The frequencies of each pattern are also counted.

# Case Study – Discoveries from Pattern Analysis



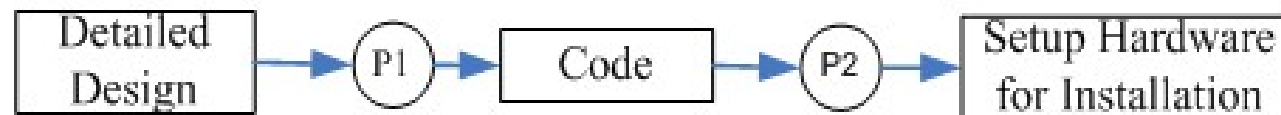
- We convert the frequency of each pattern from absolute numbers to percentage within the patterns of the same length
- In this case study, the patterns with frequency above 1% from length 3 to 11 are considered for further analysis
- We show the content of the top 5 high frequency patterns in length 3 as examples

Length 3 Pattern's Content and Frequency	
Content	Frequency
Ptn1={ E <sub>18</sub> , E <sub>17</sub> , E <sub>18</sub> }	27.8%
Ptn2={ E <sub>17</sub> , E <sub>18</sub> , E <sub>24</sub> }	23.3%
Ptn3={ E <sub>17</sub> , E <sub>18</sub> , E <sub>25</sub> }	10.7%
Ptn4={ E <sub>17</sub> , E <sub>18</sub> , E <sub>22</sub> }	8.8%
Ptn5={ E <sub>17</sub> , E <sub>18</sub> , E <sub>19</sub> }	6.2%
EPG Element ID	E <sub>17</sub> = 'Detailed Design' E <sub>18</sub> = 'Code' E <sub>19</sub> = 'Test'
	E <sub>22</sub> = 'Write User Documentation'
	E <sub>24</sub> = 'Setup Hardware for Installation'
	E <sub>25</sub> = 'Installation and Test'

# Case Study – Patterns and Problems



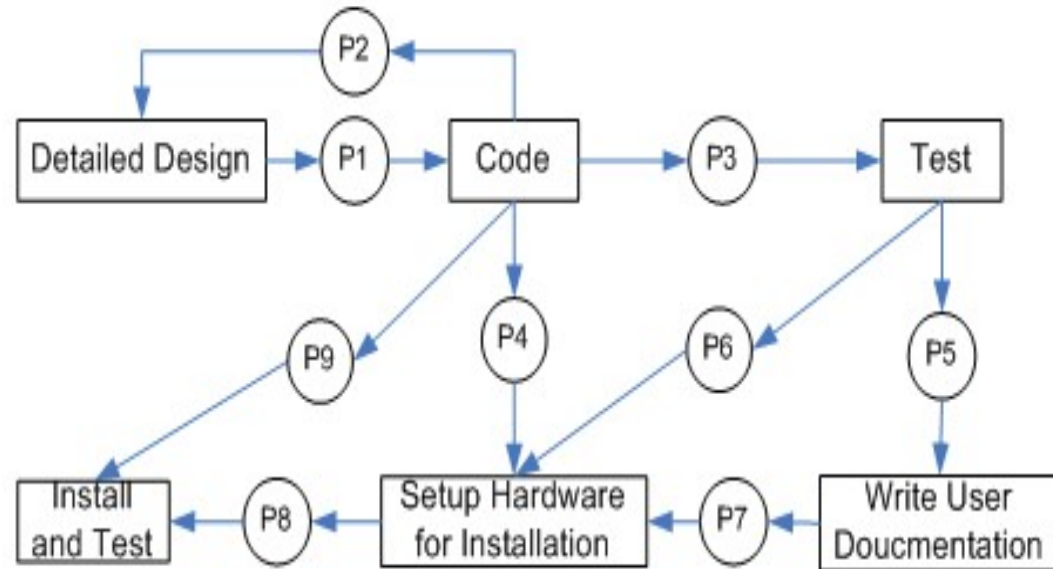
- The highest frequency pattern among length 3 patterns
  - $Ptn1 = \{E_{18}, E_{17}, E_{18}\}$ .
  - $E_{17} = \text{'Detailed Design'}$   $E_{18} = \text{'Code'}$
- In EPG model, there is a cycle between 'Detailed Design' and 'Code' exists. Ptn1 is consistent with EPG.
- Based on its frequency, it is apparent that Ptn1 is followed heavily during the development,
- This means the system design might not be completed in one time.
- It discloses that the team leans towards a short iteration that mainly focuses on design and coding, i.e. the team was doing a bit of design and then coding it.



# Case Study – Patterns and Problems



- Similar investigations were performed in patterns with length from 4 to 11.
- We summarize the patterns in Petri-net format



- There is a path directly from 'Test' to 'Setup Hardware for Installation'.
- This means 'Write User Documentation' might be skipped during project development.
- The path from 'Code' to 'Setup Hardware for Installation' shows that in some circumstance, the team skipped 'Test' and 'Write User Documentation' tasks.

# Case Study – Patterns and Problems



- Some findings from the analysis of the consistency patterns.
  - The development team did not follow the project preparation procedures designed in their pre-defined process model, because there is no consistency detected in project design stage of the pre-defined model.
  - However, no pattern found in this phase reveals that the development team may have their procedures for preparing the project. This phase might be not as useful as it was planned to this project.
  - The iteration of 'Detailed Design' ( $E_{17}$ ) and 'Code' ( $E_{18}$ ) is repeated heavily and up to four times.
  - A pattern with length 11 is found to have this iteration four times and its frequency is up to 28.6% among all the length 11 patterns
  - $Ptn = \{E_{18}, E_{17}, E_{18}, E_{17}, E_{18}, E_{17}, E_{18}, E_{17}, E_{18}, E_{22}, E_{24}\}$  ( $E_{22}$ ='Write User Documentation',  $E_{24}$ ='Setup Hardware for Installation').
  - The occurrence of the path from 'Code' to 'Test' is not high. Data in the 2006 paper supports the assertion that the team might follow a test-driven process.

# Case Study – Patterns and problems 1



- *Iteration Composed by Project Design and Code:*
  - The analysis of consistency patterns shows that this project follows a short iteration in which the development team design a bit and then code a little.
- *Test Driven:*
  - There are patterns found, particularly in the 2006 paper, which show that the development team did testing before coding, and patterns in this study which show a coding, testing, coding loop.
  - The evidence from both studies suggests that the team was tending to follow a type of test driven practice in this project
- *Scope of EPG model:*
  - There are three phases in the pre-defined process model. However, most of the detected patterns are gathered in the development phase, phase 1.
  - It means that the project preparation and wrapping up stages have not been extensively adopted or recorded by the development team.
  - The project manager may need to consider re-designing these two phases, if necessary.

# Discussion 1



## ▪ Scalability:

▪ We have applied our approach on three different sized projects including the case study presented in this paper.

▪ Each case study is a typical representative of different sized software development projects.

<i>Project Information</i>	<i>Case Study 2</i>		<i>Case Study 3</i>
Team size	6 persons		14 persons
Total Effort	1652.25 man-hours		3524.5 man-hours
Duration	875 days		597 days
Closed year	2002		2003
Total task number	85		58
Task Classification	EPG Tasks	53 (62%)	57 (98%)
	None-EPG Tasks	32 (38%)	1 (2%)
Kappa Agreement	0.695		0.700
AC1 correction	0.708		0.743

▪ These case studies proved that our approach is applicable to different sized projects, including small to middle sized projects and even to project modules.

## Discussion 2



- Data type and required data size:

- The previous' methods require large data set, normally more than hundreds of cases. It is very difficult to gather such amount of data.
- After amending the algorithm and introduced task classification, the required data size of our approach could be fairly small.

- Mining method:

- We are trying to find the temporal relations from the data set and then to detect the connecting places from the causal relations.
- The mined out results can be process model or process patterns depends on if the data set is complete.

- Output

- In previous work, the data set mainly stays at the “activity” level. The recovered model with such details is normally very large and mainly reflects the low level process model. Meanwhile, the transitions sometimes are countless and complex.
- In our methods, the recovered process elements are in the same context as the pre-defined process model.

# Conclusion



- The value of our method can be summarized as:
  - 1) providing evidence of how a process executes during development,
  - 2) identifying the coverage of the process model,
  - 3) extracting the process patterns from live data,
  - 4) discovering the conflicts and consistence between the defined process model and its enactment model

- Future work:
  - 1) Some noise data, such as the tasks that can not be mapped to the process elements, cannot be handled at this moment,
  - 2) Our approach needs to provide clear instructions on which type of projects should be divided into modules and how when the project is a moderate-large sized project.
  - 3) Our work only deals with the sequential temporal relation at current stage. We need to investigate the relationships among process elements, such as concurrency or choices



From imagination to **impact**

# Backup Slides

# Process recovery approach - Measurement



- Our work on process mining
  - Our previous paper: APEC08, presents how the conflicts are minded out
  - The consistent patterns follow to the type 1 in the table.

# Process recovery approach – Measurement



- Measurement
- There are the seven possibilities in total.
- Except for the first condition, 2 to 7 are defined as conflicts.
- $V_t(t_A)$  should be finished before  $V_t(t_B)$ .
- $V_t(t_A)$  presents the process element that  $t_A$  belongs to.
- We applied our approach to a pilot case study.

Diagram	Sequence Name	Conflict type
	(1) A is directly followed by B	None
	(2) A is right overlapping with B	Right overlap
	(3) A is containing B	Containing
	(4) A is directly exceeded by B	Sequence confl
	(5) A is left overlapped by B	Left overlappin
	(6) A is contained by B	Contained
	(7) A and B are paralleled	Parallel