Web Service Adaptation in Presence of Business Protocol Evolution

Maryam Eslamichalandar

Doctorate Candidate

Cedric - CNAM, Paris

January 18, 2013

Overview of Web Services

- Web service technologies
- Web service "description"
  - *Behavior*
  - *Capability*
  - *Quality* (QoS)
  - *Interface*
    - Service *signature*
      - operations and the inputs/outputs messages
      - message types and error parameters
    - Set of ports enabling the message exchange
    - WSDL is an XML language for interface description

Web Services Composition

- When we need to compose Web services?
- Combining the functionality of heterogeneous Web services
  - A value-added service as a *composite* service
  - Seamless and dynamic communication
    - regardless their locations or platforms
    - implementing the reusability of services
    - increasing quality and efficiency
    - satisfying the dynamic evolution of user’s demands

Service Composition Verification

- Ability of composing services regarding
  - ❑ Heterogeneous structure
  - ❑ Autonomous and independent designing by different organizations
  - ❑ Satisfaction of different customers

Makes the challenge of necessities for

- ❑ Evaluation and verification of behavior of composite service
- ❑ Efficient solutions and appropriate tools to solve arising problems

Compatibility of Service Composition

- Compatibility: A fundamental challenge on service composition verification

“ *two services are compatible if and only if their interaction works properly* ”

- Compatibility checking and verification approaches
  - ❑ Partner services can successfully interact with each other?
  - ❑ And two services can reach their expected results?

- Compatibility types
  - ❑ *Interface compatibility*
  - ❑ *Semantic compatibility*
  - ❑ *Behavioral compatibility*
  - ❑ *QoS compatibility*

Service Incompatibility

- Taxonomy of service mismatches at higher leyer of abstraction:
  - Mismatches in service *interface* level
    - message and operation name, number
    - type of input/output parameters of operations
    - parameter value constraint

    by such a classification:
      - *syntactical*
      - *structural*
      - *semantic*
      - *extra / missing messages*
      - *messages split / merge*

  - Mismatches in *service business protocol* (*behavior*) level
    - *deadlock*
    - *unspecified reception*

Service Adaptation

- **Adaptation**: typical approach for an incompatible service collaboration
  - service *interface* incompatibilities
  - service *business protocol* incompatibilities

- Service adaptation methodology:
  - *Synthesizing a stand-alone adapter protocol*
    - To adjust message exchange ordering
    - To perform message type transformation
  - *Service modification*
    - Applying some *tuning actions* to support partner service’s specifications

Service Adaptation & Service Evolution

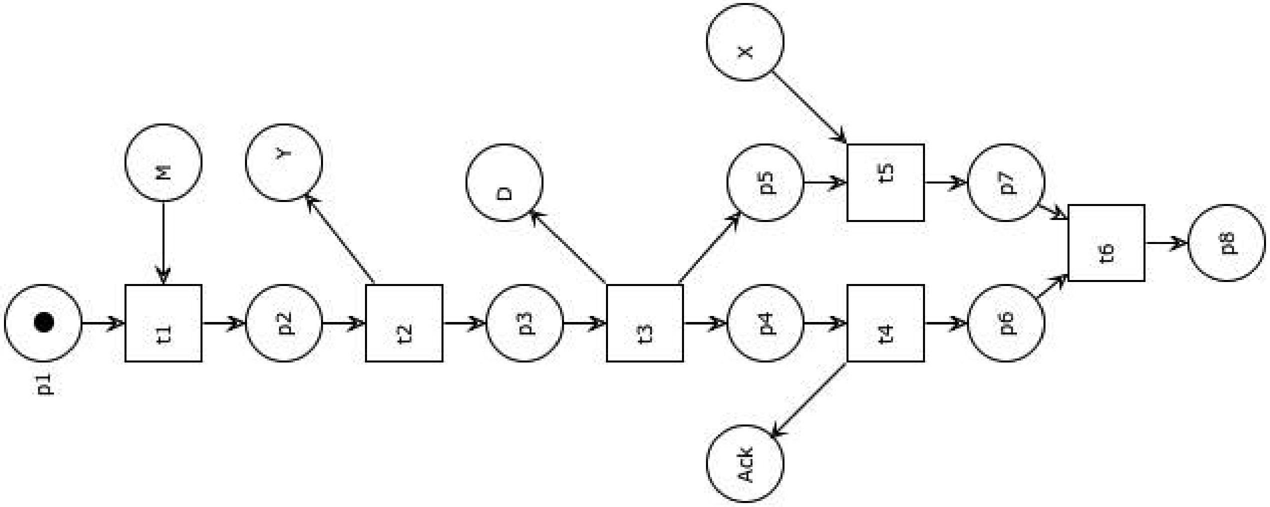
- The intention of Web service evolution:
  - To improve the business functionalities
  - To change the service interface, business protocol and policy

- Existing service adaptation approaches:
  - Business protocols do not change after adapter generation

- New interesting challenge in service adaptation
  - to cope with the problem of service adaptation
    - when one or both of services' business protocols evolve

Definitions

- Approach supports the *open Workflow Nets* (oWFN) formalism
- oWFN N = (P, T, F, I, O, m_0, M_F) is an extension of Petri net comprising
  - *workflow net* - to describe the internal process
  - two disjoint sets of *input / output interface places* - to interact with other participant services.



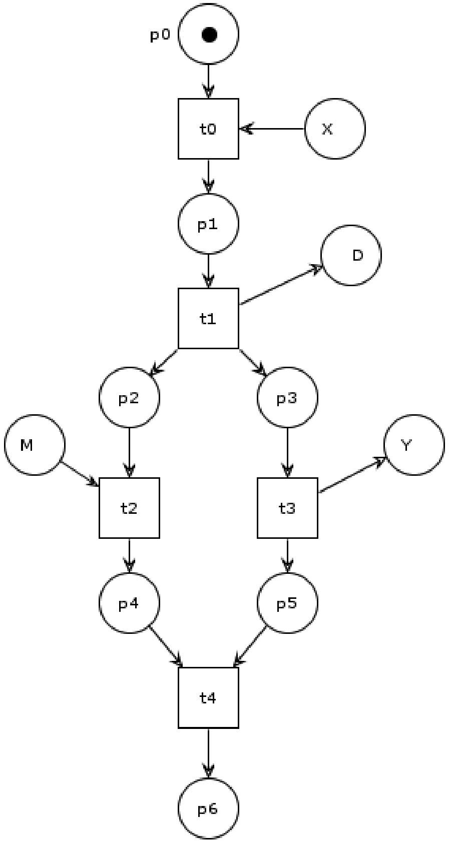
Definitions

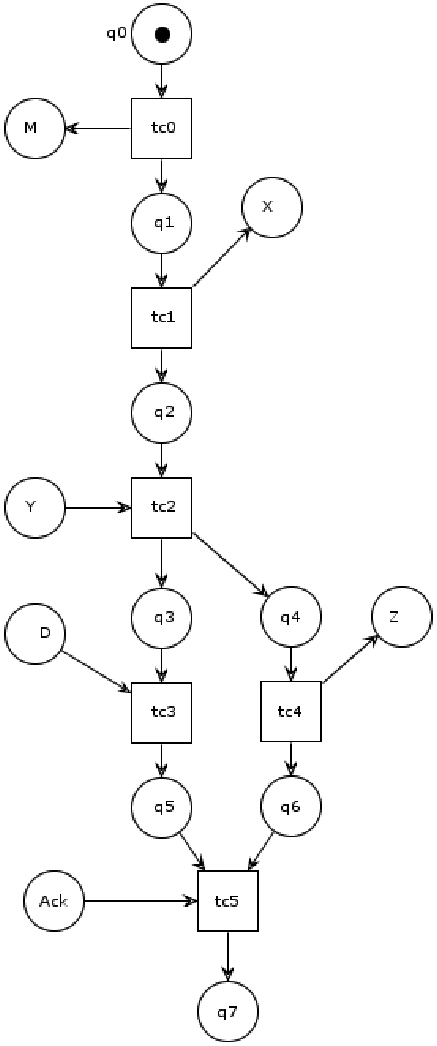
- **Definition 2 (Adapter Protocol).** Adapter protocol, AP = (P_A, T_A, F_A, I_A, O_A, m_{0A}, m_{FA}) between two business protocols N and M
  - ❑ P_A : a finite set of internal places;
  - ❑ T_A : a finite set of adapter transitions;
  - ❑ F_A : a set of arcs as $F_A \subseteq (P \times T) \cap (T \times P) \cap (I_A \times T) \cap (T \times O_A)$;
  - ❑ $I_A = \{O_N \cap O_M\}$: a set of input interface places;
  - ❑ $O_A = \{I_N \cap I_M\}$: a set of output interface places.

Motivating Example – Original Services

Two services P1 and P2 with behavioral Incompatibilities

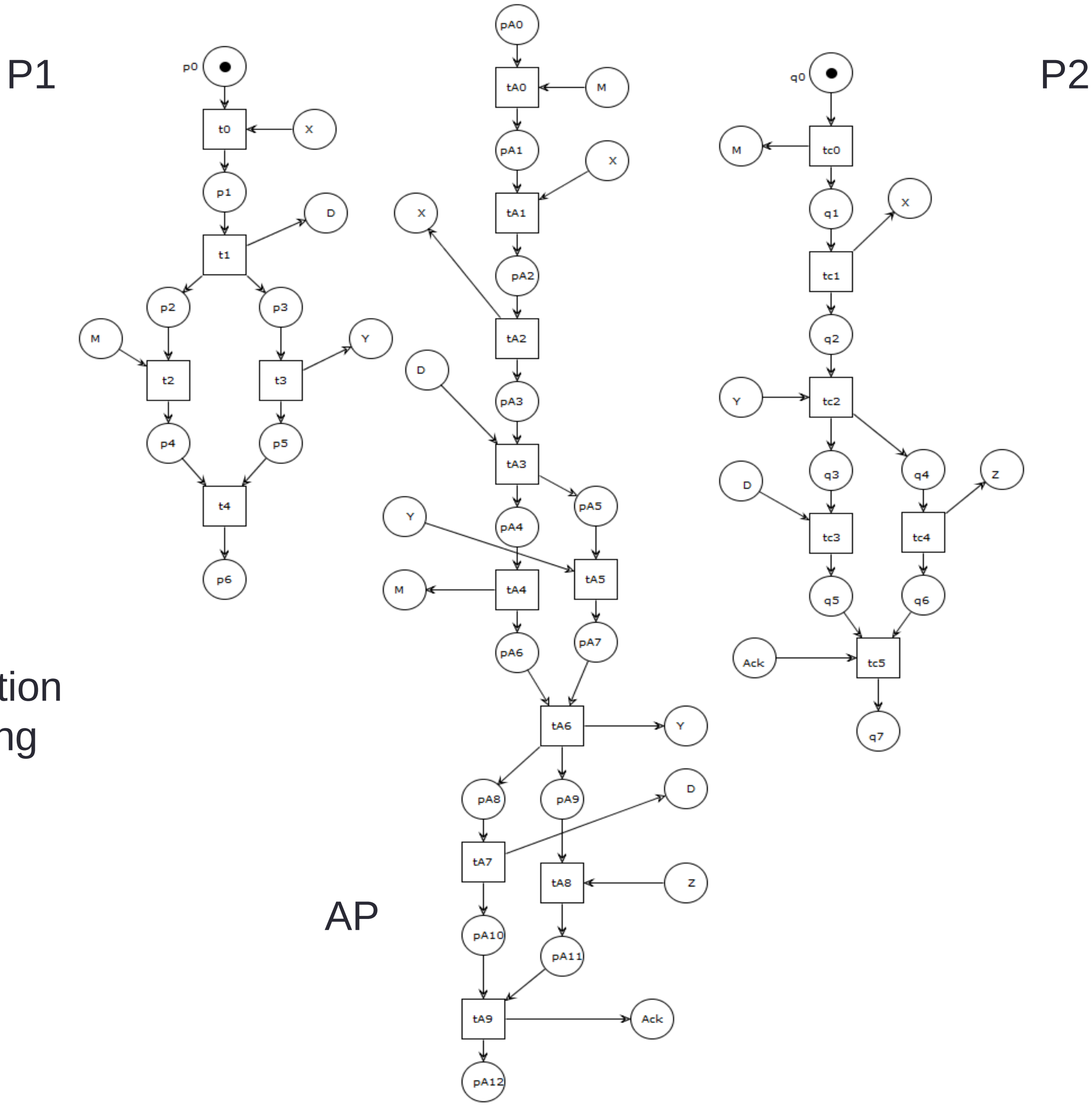
P1





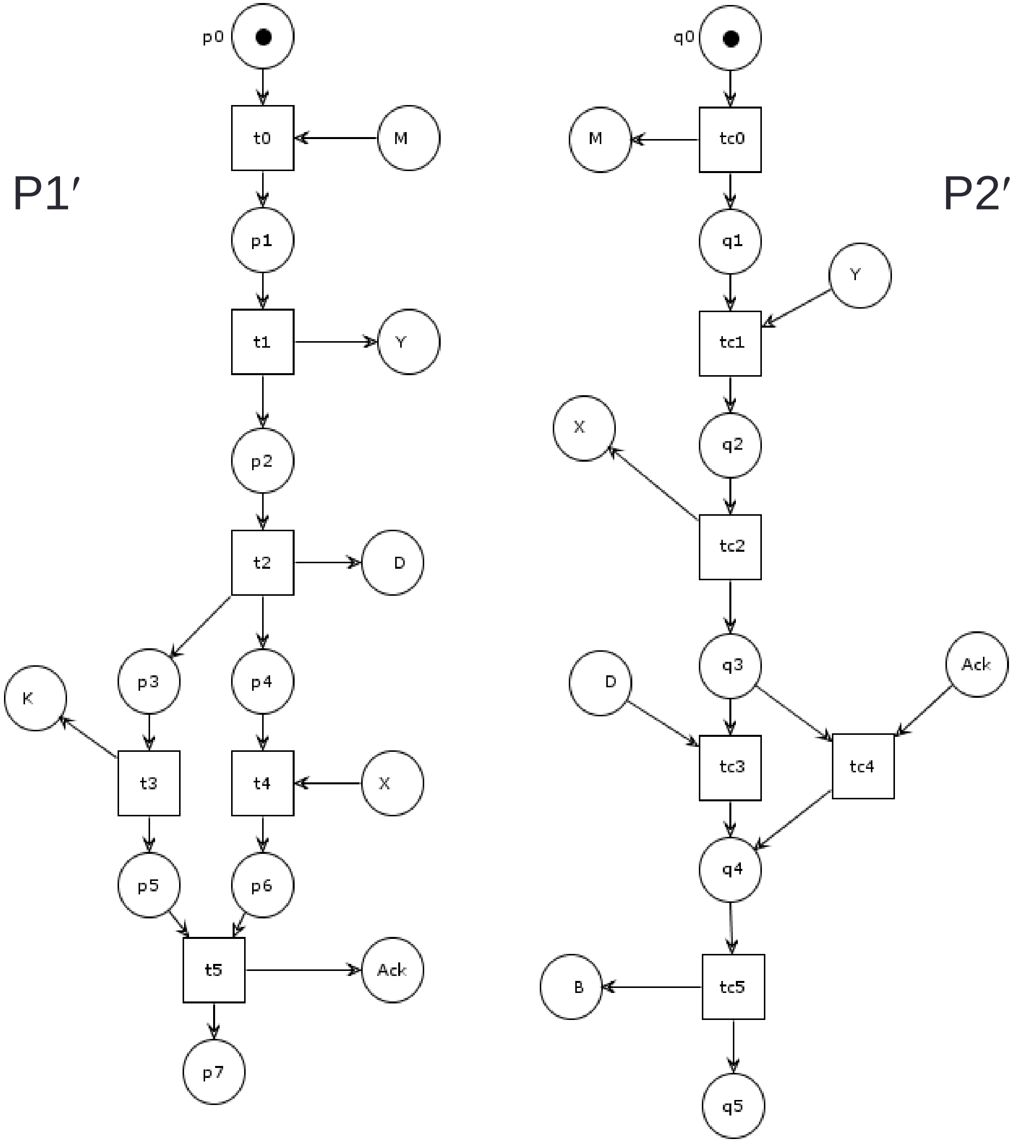
P2

Motivating Example – Original Services with Adapter



Compatible collaboration between P1 & P2 using an Adapter Protocol

Motivating Example – Evolution model



Service Adaptation in the Context of Business Protocol Evolution

- What is the impact of evolution of business protocols on the adapter
  - ❑ Is it possible to update the adapter specification dynamically?

- Our proposed analysis approach
  - ❑ To identify the changes in evolved business protocols
  - ❑ To analyze the respective impacts of evolution on the adapter

- Our approach determines:

1. The evolution has no effect on the adapter,
2. The impact of evolution on the adapter can be treated dynamically,
3. The impact of evolution leads to complete adapter re-generation.

Proposed Evolution Analysis Approach

- The main goal is to provide enough information to adapter developers
  - ❑ How to deal with changes in the underlying interacting services?
  - ❑ How to handle the specification of existing adapter protocol?

Our proposed approach comprising:

1. Business protocol evolution
   - ❑ Introducing common patterns of evolution
2. Business protocol evolution identification
   - ❑ Presenting an Algorithm to detect evolution patterns
3. Protocol evolution impact analysis on the adapter
   - ❑ Presenting an algorithm to analyze the impact of evolution on current Adapter
4. Prototype Implementation
5. Experimental Results

1. Business Protocol Evolution Pattern

- **Pattern #1: new input data is needed.**
  An Interface transition t is updated by an input interface place p.

- **Pattern #2: new output data is delivered.**
  An Interface transition t is updated by an output interface place p.

- **Pattern #3: an existing input data is updated.**
  An Interface transition t is updated when the associated input place has been updated.

- **Pattern #4: an existing output data is updated.**
  An Interface transition t is updated when the associated output place has been updated.

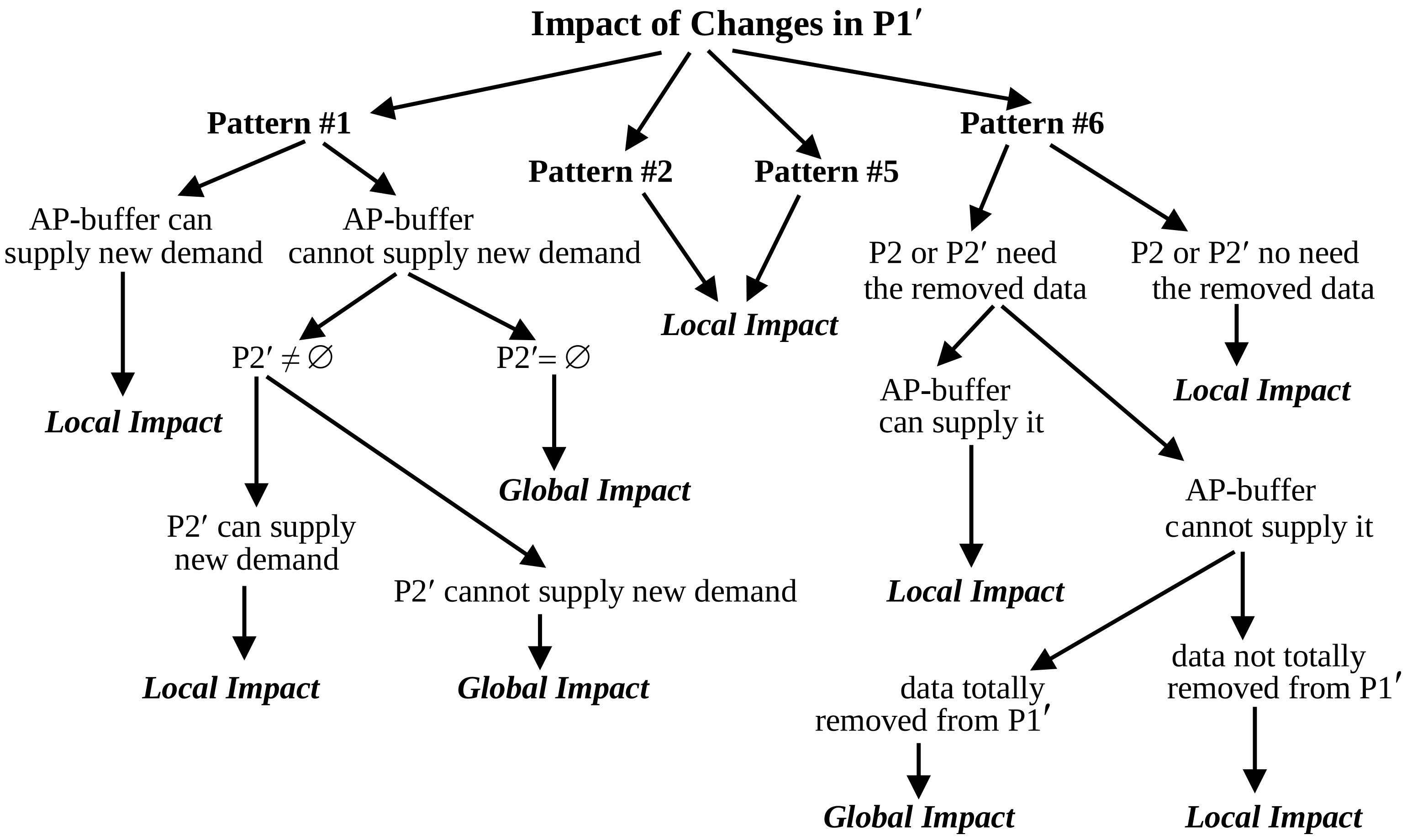
- **Pattern #5: an existing input data is removed.**
  An Interface transition t is updated when the associated input place has been removed.

- **Pattern #6: an existing output data is removed.**
  An Interface transition t is updated when the associated output place has been removed.

- **Pattern #7: a new interface transition is added.**
- **Pattern #8: an interface transition is removed.**

1. Business Protocol Evolution Pattern

Potential impact of evolution patterns on the Adapter
from point of view of P1′



Impact of Changes in P1′

Pattern #1

AP-buffer can supply new demand

Local Impact

AP-buffer cannot supply new demand

P2′ ≠ ∅

P2′ can supply new demand

Local Impact

P2′ cannot supply new demand

Global Impact

P2′= ∅

Global Impact

Pattern #2

Pattern #5

Local Impact

Pattern #6

P2 or P2′ need the removed data

AP-buffer can supply it

Local Impact

AP-buffer cannot supply it

data totally removed from P1′

Global Impact

data not totally removed from P1′

Local Impact

P2 or P2′ no need the removed data

Local Impact

2. Business Protocol Evolution Identification

- Introducing an algorithm to identify the evolution of business protocol
  - ❑ Comparing the oWFN model of original and new version

- The algorithm outputs:
  - ❑ Which elements have changed
  - ❑ Which type of changes has been occurred
    - ➢ *added*, *removed*, and *updated* interface elements

- Introducing the *interface incidence matrix* in terms of an oWFN model
  - ❑ Revising the *incident matrix* of a Petri net
    - ➢ the columns include only the interface transitions,
    - ➢ the rows includes only the input/output interface places.

2. Business Protocol Evolution Identification

Definition (Interface Incidence Matrix). For an oWFN $N = (P, T, F, I, O, m_0, m_F)$, an interface incident matrix **IM** is a $|R| \times |C|$ matrix wherein

- $R = (I \cup O)$;
- $C = \{t \in T \mid ({}^{\bullet}t \in I) \cup (t^{\bullet} \in O)\}$;

such that $\forall p \in R$, $t \in C$, $i \in I$, and $o \in O$;

- $M[p, t] = 1$ if $(i, t) \notin F$ and $(t, o) \in F$;
- $M[p, t] = -1$ if $(i, t) \in F$ and $(t, o) \notin F$;
- $M[p, t] = 0$ if $(i, t) \notin F$ and $(t, o) \notin F$.

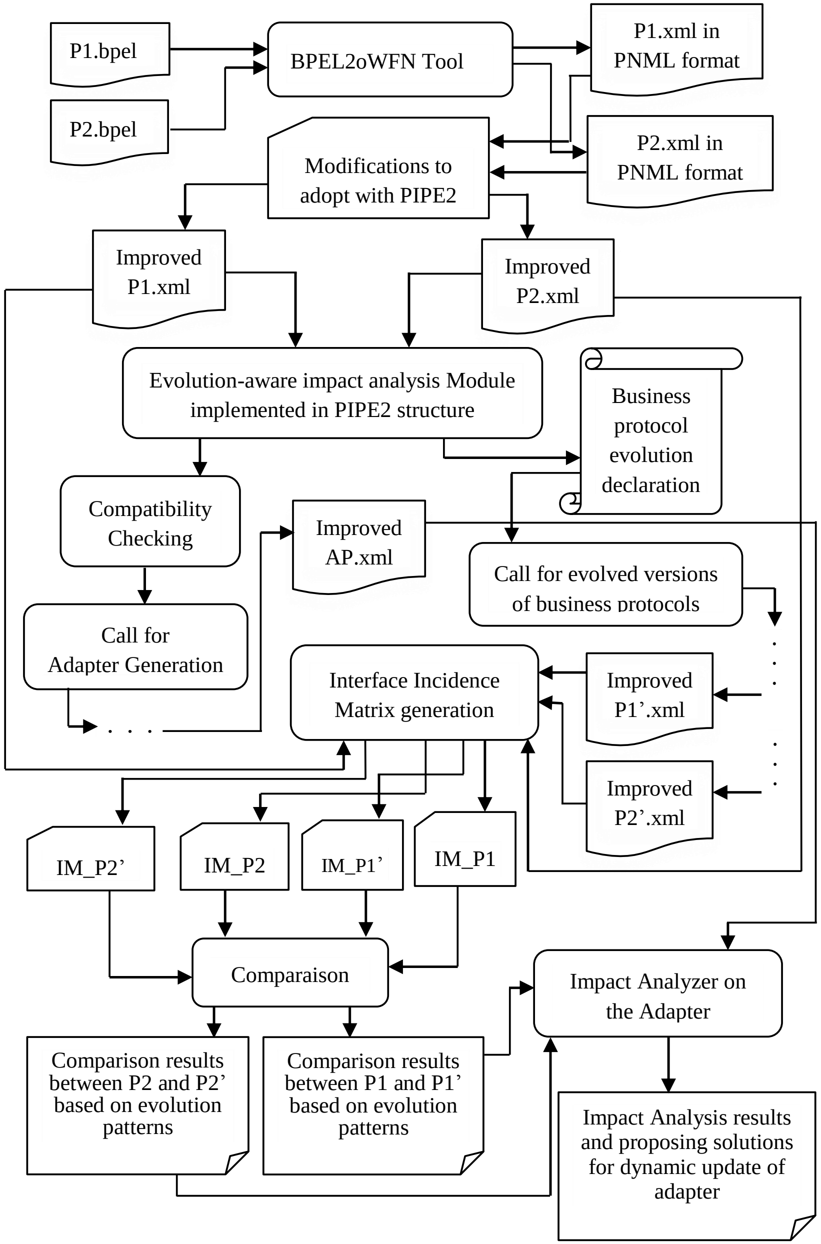
$$\text{IM\_P1}' \quad \begin{array}{c|cccccc|} & t0 & t1 & t2 & t3 & t4 & t5 \\ \hline X & 0 & 0 & 0 & -1 & 0 & 0 \\ M & -1 & 0 & 0 & 0 & 0 & 0 \\ Y & 0 & 1 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 1 & 0 & 0 & 0 \\ Ack & 0 & 0 & 0 & 0 & 0 & 1 \\ K & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

$$\begin{array}{c|cccc|} & t0 & t1 & t2 & t3 \\ \hline X & -1 & 0 & 0 & 0 \\ M & 0 & 0 & -1 & 0 \\ Y & 0 & 0 & 0 & 1 \\ D & 0 & 1 & 0 & 0 \end{array} \quad \text{IM\_P1}$$

3. Protocol Evolution Impact Analysis

- Proposing an Impact Analysis Algorithm
  - ❑ Evaluating the impact of evolution on current adapter specification
    - ➢ Through partial exploration of affected portions of business protocols
    - ➢ Applying the evolution patterns
    - ➢ Providing adaptation solution for *dynamic updating the adapter*

4. Prototype Implementation



P1.bpel

P2.bpel

BPEL2oWFN Tool

P1.xml in PNML format

P2.xml in PNML format

Modifications to adopt with PIPE2

Improved P1.xml

Improved P2.xml

Evolution-aware impact analysis Module implemented in PIPE2 structure

Business protocol evolution declaration

Compatibility Checking

Improved AP.xml

Call for evolved versions of business protocols

Call for Adapter Generation

Interface Incidence Matrix generation

Improved P1'.xml

Improved P2'.xml

Comparaison

Impact Analyzer on the Adapter

Comparison results between P2 and P2' based on evolution patterns

Comparison results between P1 and P1' based on evolution patterns

Impact Analysis results and proposing solutions for dynamic update of adapter

5. Experimental Results

The effectiveness of approach through some synthetic examples

n = is the number of activities (interface transitions)

- Evolution Impact Analysis Cost : **Γ = O(n)**
- Dynamic Adapter Re-configuration Cost: **μ = O(n)**
- Complete Adapter Re-generation Cost : **Ψ = O(n^2)**

Impact of Evolution on current Adapter	Complete Adapter Re-generation	Adapter Re-configuration via Evolution-aware Impact Analysis
Evolution without impact	Ψ	$\Gamma + \varnothing$
Dynamic update	Ψ	$\Gamma + \mu$
Hard change	Ψ	$\Gamma + \Psi$

le cnam

Thanks you so much for your attention!