Formal Consistency Verification between BPEL Process and Privacy Policy

Yin Hua Li, Hye-Young Paik and Boualem Benatallah
School of Computer Science and Engineering
University of New South Wales
Sydney NSW 2052, Australia
{yinhual,hpaik,boualem}@cse.unsw.edu.au

Salima Benbernou
LIRIS-University Claude Bernard Lyon1
43, boulevard du 11 Novembre 1918
69622 Villeurbanne France
sbenbern@liris.cnrs.fr

Abstract

Despite the increased privacy concerns in the Internet, not much attention has been paid into enforcing privacy policies of organisations who collect and consume personal data using automatic means (e.g., Web services). In this paper, we propose a graph-transformation based framework to check whether an internal business process (implemented using a standard Web service composition language such as BPEL) adheres to the organisation's privacy policies. The graph-based specification formalism combines the advantages of an intuitive visual framework with rigorous semantical foundation that allows consistency checking between a business process and privacy policy. The privacy consistency verification framework is defined by a set of rules to build the system state and sets of constraints (positive and negative) to specify the wanted and unwanted substates.

1 Introduction

Privacy is defined as the right of entities (e.g., a mortgage applicant) to determine for themselves when, how and to what extent information about them is communicated to others[23]. Privacy policies are developed so that organisations (e.g., a mortgage broker) that collect personal information can express their *"code of conduct"* in handling privacy issues. With the advent of the Internet, such policies are now often encoded as P3P (Platform for Privacy Preferences[16]), an XML application that is designed to capture the privacy policies in a machine-understandable format.

As more and more organisations shift their core operations to the Web services platform within which various interactions between the autonomous services occur, it has become important that the services are able to *understand* and *adhere* to the privacy policies laid out by the organisation.

One of the widely accepted standards in the Web services platform is BPEL (Business Process Execution Language for Web Services)[4]. BPEL defines a language for creating composite Web services in the form of business processes following the service orchestration paradigm. This paper introduces a formal framework through which the business operations in an organisation are checked against the organisation's privacy policies to see whether the operations conform to the policies. In our paper, we consider BPEL process description as representing the automated internal business operations of an organisation. Also, we assume that the policies are expressed in the P3P language.

We argue that checking the conformance of a business operation against the relevant privacy policy is the first important step in enforcing the privacy policies of the organisation. Such enforcement is required due to the fact that the group of employees who develop BPEL process specifications to implement business operations, and the group of employees who propose the organisation's privacy policies have different skill background, tasks and goals. For example, privacy policy officers would be more concerned with high level abstraction of personal data usage in their organisation and the relevant rules and government regulations. On the other hand, BPEL specification designers would be interested in the messages exchange and functionality of the operations. A framework is needed to ensure that these separately developed specifications are indeed coherent. We propose a formal approach based on graphs and graph transformation to address the problem of checking whether the business process conforms to the privacy policy. A system state is represented by a graph and a set of graph transformation rules describing how a system state evolves. The specification framework contains declarative information on what a system must contain (positive constraints) and what it should not contain (negative constraint).

The remainder of the paper is structured as follows. In section 2 we introduce BPEL and Platform of Privacy Preferences (P3P). Section 3 gives a running example to motivate the consistency verification between BPEL process and privacy policy. Section 4 explains the approach to build

the semantic relationships between BPEL and privacy policy. The architecture of privacy consistency verification system is present in section 5. In section 6, we introduce the graph transformation framework through our motivating example. Section 7 presents the privacy consistency verification framework. An overview of related work is given in section 8. Section 9 concludes the paper.

2 A short introduction to BPEL and P3P

Before we proceed further, we introduce the two technologies used in this paper; BPEL and P3P.

2.1 Business Process Execution Language

Business Process Execution Language for Web Services (BPEL4WS or BPEL for short), developed by Microsoft, IBM, BEA, Siebel Systems and SAP, is an XML-based language used to implement a process-oriented form of service composition. Each BPEL composition is a BPEL process that interacts with a set of Web services to achieve a certain goal (e.g., processing a home loan request). The services with which BPEL process interacts are called partners. BPEL uses $< partnerLinkType >$ to describe the relationship between two services and the role that each service plays, and $< partnerLink >$ to specify how logically tie a BPEL process to an existing Web service. Every Web service used in a BPEL process requires a $< partnerLink >$. There are a set of $< partnerLink >$ in a BPEL process to define services with which the BPEL process interacts. Each partner link has a partner link type.

Activities are the building blocks of BPEL processes. Primitive activities are conceptually simple behaviors, such as receiving a message, invoking a Web service, and assigning values to variables. BPEL also provides primitive activity structuring. The *sequence* activity contains one or more activities that are performed sequentially. The *switch* structured activity supports conditional behavior. The *while* activity supports repeated performance of a specified iterative activity. The *pick* activity awaits the occurrence of one of a set of events and then performs the activity associated with the event that occurred. The *flow* construct provides concurrency and synchronization. A BPEL process, like any Web service, supports a set of WSDL [8] interfaces that enable it to exchange messages with its partners.

2.2 Platform for Privacy Preferences

Platform for Privacy Preferences (P3P), developed by the World Wide Web Consortium (W3C), is the most significant effort to enable the Internet users to gain more control over their disclosed personal information. Written in XML, P3P can be programmatically accessed by any 'privacy-aware' applications[9]. P3P privacy policy consists of a set of statements, and each statement has four different components. The `data-group` specifies which personal information will be collected. The `purpose` defines what the collected data will be used for. P3P pre-defines twelve possible purposes, such as *current*, *contact*, *admin*, etc. The `recipient` specifies who will consumed the data and P3P provides 6 possible recipients(e.g., *ours*). Finally, the `retention` states how long the collected data will be kept by the service provider.

3 A running example: Mortgage Broker

As a running example, we will consider a Mortgage Broker named BestMortgage. Their core business operation is processing a home loan request. Its privacy policies are written in P3P whose excerpt is shown in Table 1.

```
<POLICY>
  <STATEMENT>
    <PURPOSE><current><contact></PURPOSE>
    <RECIPIENT><ours><same></RECIPIENT>
    <RETENTION><legal_requirement></RETENTION>
    <DATA-GROUP>
       <DATA ref="\#user.Name"> </DATA>
    </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
    <PURPOSE><current><contact></PURPOSE>
    <RECIPIENT><ours></RECIPIENT>
    <RETENTION><legal_requirement></RETENTION>
    <DATA-GROUP>
      <DATA ref="\#user.HomeAddress"> </DATA>
      <DATA ref="\#user.HomePhone"> </DATA>
      <DATA ref="\#user.Mobile"> </DATA>
    </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
    <PURPOSE><current><pseudo_analysis></PURPOSE>
    <RECIPIENT><ours><same></RECIPIENT>
    <RETENTION><business_practices></RETENTION>
     <DATA-GROUP>
       <DATA ref="\#user.GrossMonthlyIncome"> </DATA>
       <DATA ref="\#user.PropertyAddress"> </DATA>
       <DATA ref="\#user.PropertyPurchasePrice"> </DATA>
     </DATA-GROUP>
</STATEMENT>
</POLICY>
```

Table 1. Privacy policy of BestMortgage

For example, according to the P3P statements, BestMortgage collects the mortgage applicant's home address, phone and mobile number and use them for processing the mortgage application (i.e., `current` purpose) and contacting the applicants (i.e., `contact` purpose).

BestMortgage implements the mortgage application process as shown in table 2 (i.e., the $.wsdl$ listing) and table 3 (i.e., the $.bpel$ listing). Its graphical representation is shown in Figure 1. In short, the process receives (via `receive` activity) a home loan request from an applicant, then it interacts with *RiskAssessor* who will assess the risk level of the request via `invoke` activity. After *RiskAssessor* replied, the process subsequently communicates with *LoanApprover* who will decide whether the request should be accepted. Finally, the `approvalMessage` is sent (via `reply` activity) to the applicant.

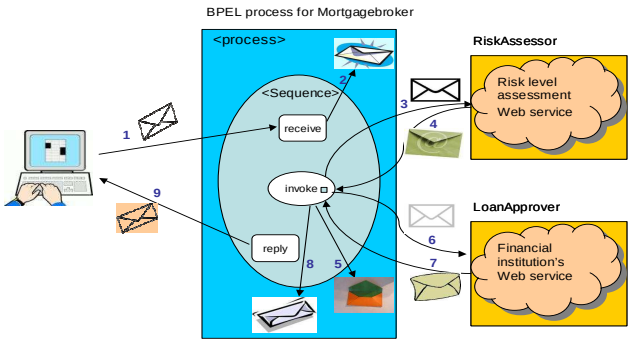

Figure 1. BPEL specification of BestMortgage **for mortgage applications**

3.1 Identifying P3P Data Items from a BPEL Process

As shown in Table 2, every message consumed and produced by an operation is defined in the WSDL document of the BPEL process. It is clear that some personal information collected by the process will be communicated to its partners(e.g., `RiskAssessor`) through these messages. In particular, we should pay attention to the messages that contain personal data that appears in the P3P statements. For example, the following message, which is consumed by the operation *riskLevelAssess*, contains parts that appear as P3P data group items (i.e., Name, MonthlyGrossIncome, PropertyAddress and PropertyPurchasePrice).

```
<message name="creditInformationMessage">
 <part name="Name" type="xsd:string"/>
 <part name="MonthlyGrossIncome" type="xsd:integer"/>
 <part name="PropertyAddress" type="xsd:string"/>
 <part name="PropertyPurchasePrice" type="xsd:integer"/>
 <part name="amount" type="xsd:integer"/>
</message>
```

Not all messages contain parts that appear as data items in the P3P statements. Such messages can be ignored when checking for privacy conformance. Based on this observation, we "extract" the following message parts from the BPEL WSDL document; name, email address, home address, home telephone number, mobile, monthly gross income, purchased property address and property price. For each of the identified data item, we construct the following:

- **From the BPEL process**: determine operations that consume or produce the messages that contain the personal data item. We also need to consider who performs these operations (i.e., the recipients of the data item). Table 4 shows the result based on the running example.
- **From the P3P statements**: determine the associated purpose and recipient. Table 5 shows the result.

```
<definitions>
......
<message name="homeLoanRequest">
 <part name="Name" type="xsd:string"/>
 <part name="Email" type="xsd:string"/>
 <part name="HomeAddress" type="xsd:string"/>
 <part name="HomePhone" type="xsd:string"/>
 <part name="Mobile" type="xsd:string"/>
 <part name="MonthlyGrossIncome" type="xsd:integer"/>
 <part name="PropertyAddress" type="xsd:string"/>
 <part name="PropertyPurchasePrice" type="xsd:integer"/>
 <part name="amount" type="xsd:integer"/>
</message>
<message name="creditInformationMessage">
 <part name="Name" type="xsd:string"/>
 <part name="MonthlyGrossIncome" type="xsd:integer"/>
 <part name="PropertyAddress" type="xsd:string"/>
 <part name="PropertyPurchasePrice" type="xsd:integer"/>
 <part name="amount" type="xsd:integer"/>
</message>
<message name="RiskLevel">
 <part name="riskLevel" type="xsd:string"/>
</message>
<message name="ApprovalMessage">
 <part name="accept" type="xsd:string"/>
</message>
......
<portType name="homeLoanServicePT">
 <operation name="request">
  <input message="homeLoanRequest"/>
  <output message="approvalMessage"/>
 </operation>
</portType>
<portType name="riskAssessmentPT">
 <operation name="riskLevelAssess">
  <input message="creditInformationMessage"/>
  <output message="riskLevel"/>
 </operation>
</portType>
<portType name="loanApprovalPT">
 <operation name="loanApprove">
  <input message="riskLevel"/>
  <output message="approvalMessage"/>
 </operation>
</portType>
......
<partnerLinkType name="homeLoanPartnerLinkType">
 <role name="MortgageBroker">
   <portType name="homeLoanServicePT"/>
 <role>
</partnerLinkType>
<partnerLinkType name="homeLoanAssessorPartnerLinkType">
 <role name="MortgageBroker">
   <portType name="homeLoanServicePT"/>
 </role>
 <role name="RiskAssessor">
   <portType name="riskAssessmentPT"/>
 </role>
</partnerLinkType>
<partnerLinkType name="homeLoanBankPartnerLinkType">
 <role name="MortgageBroker">
   <portType name="homeLoanServicePT"/>
 </role>
 <role name="LoanApprover">
    <portType name="loanApprovalPT"/>
  </role>
</partnerLinkType>
......
</definitions>
```

Table 2. excerpt of BestMortgage**'s** $.wsdl$ **listing**

The above information forms the basis for the important mapping technique that we propose in the next section. The

```
<process name="HomeLoanService"
 ......
<partnerLinks>
   <partnerLink name="customer"
      partnerLinkType="homeLoanPartnerLinkType"
      myRole="MortgageBroker"/>
   </partnerLink>

   <partnerLink name="Assessing"
      partnerLinkType="homeLoanAssessorPartnerLinkType"
      myRole="MortgageBroker"
      partnerRole="RiskAssessor"/>
   </partnerLink>

   <partnerLink name="Approving"
      partnerLinkType="homeLoanBankPartnerLinkType"
      myRole="MortgageBroker"
      partnerRole="LoanApprover"/>
   </partnerLink>
</partnerLinks>
......
<sequence>
   <receive partnerLink="customer"
            portType="homeLoanServicePT"
            operation="request"
            ......>
   </receive>
   ......
   <invoke partnerLink="Assessing"
           portType="riskAssessmentPT"
           operation="riskLevelAssess" ......>
   </invoke>
   ......
   <invoke partnerLink="Approving"
           portType="loanApprovalPT"
           operation="loanApprove" ......>
   </invoke>
   .......
   <reply  partnerLink="customer"
           portType="homeLoanServicePT"
           operation="request"
            ......>
   </reply>
   ......
</sequence>
......
</process>
```

Table 3. The *.bpel* excerpt of home Loan approval process

Personal Data	BPEL activity	WSDL Operation	BPEL Role
Name	receive	request()	MortgageBroker
	invoke	riskLevelAssess()	RiskAssessor
MonthlyGrossIncome	receive	request()	MortgageBroker
	invoke	riskLevelAssess()	RiskAssessor
PropertyAddress	receive	request()	MortgageBroker
	invoke	riskLevelAssess()	RiskAssessor
PropertyPurchasePrice	receive	request()	MortgageBroker
	invoke	riskLevelAssess()	RiskAssessor
Email	receive	request()	MortgageBroker
HomeAddress	receive	request()	MortgageBroker
HomePhone	receive	request()	MortgageBroker
Mobile	receive	request()	MortgageBroker

Table 4. Privacy related information in the home loan approval BPEL process

mapping is performed to *link* the messages and operations in BPEL process to purposes and recipients in P3P.

Personal Data	Purpose	Recipient
Name	current, contact	Ours, Same
MonthlyGrossIncome	current, pseudo-analysis	Ours, Same
PropertyAddress	current, pseudo-analysis	Ours, Same
PropertyPurchasePrice	current, pseudo-analysis	Ours, Same
Email	current, contact	Ours
HomeAddress	current, contact	Ours
HomePhone	current, contact	Ours
Mobile	current, contact	Ours

Table 5. Related information in P3P privacy policy

4 Relationship between BPEL process and Privacy Policy

Generally speaking, an inconsistent model does not have any correct implementation because the requirements expressed by different submodels are in conflict. To deal with such problems, it is essential to understand the relationships of submodels at semantic level. More specifically, in this paper, we are referring to BPEL processes and Privacy policies as submodels. BPEL processes and privacy policies are written by different people with their own perspectives on the internal business operations, based on their skills, responsibilities, knowledge and expertise. In order to check whether the BPEL process designed is consistent with the advertised privacy policy, the relationships between BPEL process terminologies and privacy policy terminologies need to be defined. Based on the relationships, the consistency checking process can *understand* these two submodels at the semantic level. Figure 2 illustrates the relationships between privacy policy and relevant components in BPEL. The top half of the diagram represents a model of a BPEL process which consists of activities(e.g., $receive$, $invoke$, $reply$, etc.). Each activity is accomplished by calling an operation which is offered by one of partners(i.e., defined by partnerLink). The operation may consume or/and produce a message. The model captures that an operation is performed by a partner who plays a certain role in the process. The bottom half of the diagram shows that each personal data item in P3P has purpose, retention and recipient. Based on this abstraction, we draw dashed line with arrow to show the relationships between privacy related parts of BPEL model and its corresponding component in P3P model. For example, we define the relationship between "messages" in BPEL and "personal data items" in P3P *(i.e., messages may contain personal data items)*. Also, the relationship between operations in BPEL and purposes in P3P can be described that *operations has purposes (i.e., by a way of simply answering the question "why operation A needs to consume message A that may contain personal data items?")*. A role who performs an operation is considered to be the recipient of the personal data item.

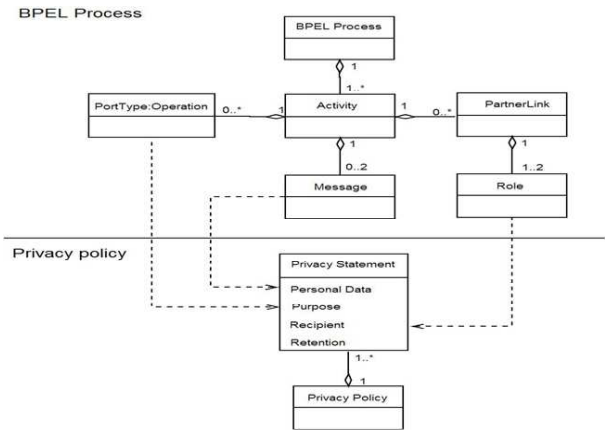

Figure 2. Relationship between BPEL and privacy policy components

In summary, we consider the following three types of semantic relationships between BPEL processes and privacy policies:

1. **messages and personal data items**: BPEL messages may contain personal data items

2. **operations and purposes**: BPEL processes use *receive*, *reply* and *invoke* activities to communicate with its partners by exchanging messages. An operation has its associated purpose for consuming message (especially the ones that may contain personal data items).

3. **partners and recipients**: the role(i.e., *partnerRole*) of *partnerLink* identifies the external (not necessarily outside of an organisation) parties that the BPEL process needs to interact with. We link each role in BPEL to recipients defined in P3P as an entity that consumes the personal data item.

The following subsections explain how we define these relationships.

4.1 Associating messages with personal data items

The first step in defining the semantic relationship between BPEL process and privacy policy is to identify all the messages in the BPEL process that contain personal data items. For example, table 6 shows the association between messages in BestMortgage and personal data items in P3P privacy policy. The messages that do not include any personal data item(e.g., *ApprovalMessage*) can be ignored.

messages in BPEL	data item in P3P policy
homeLoanRequest, creditInformationMessage	Name
homeLoanRequest, creditInformationMessage	MonthlyGrossIncome
homeLoanRequest, creditInformationMessage	PropertyAddress
homeLoanRequest, creditInformationMessage	PropertyPurchasePrice
homeLoanRequest	Email
homeLoanRequest	HomeAddress
homeLoanRequest	HomePhone
homeLoanRequest	Mobile

Table 6. Association between messages and personal data items

4.2 Associating Operation with Purpose

Purpose plays a major role in privacy policy. It states why a particular personal data is collected by using abstract level vocabularies(e.g., *contact*). The problem here is how to relate these concrete activities(i.e., operation invocation) in BPEL process to those purposes in privacy policy. We propose to build an operation-purpose connection tree to solve this problem.

Definition 1 ***(Operation-purpose connection tree)*** *Given a set of operations of BPEL process and a set of P3P predefined purpose vocabularies, we build a tree connecting purpose to operations. It is defined by* $Op2Purpose = \{\top, P \cup Op, E\}$*, where:*

- $\top$ *means general purpose and is the root of the tree.*
- P *is the set of P3P predefined purpose(e.g., current).*
- Op *is the operations of BPEL process.*
- E *is the set of edges, stating that if* (v_x, v_y) *are nodes connected by an edge* $E_{xy} \in E$ *then* v_y *is semantically a part of* v_x*.*

Proposition 1 *For any* $op \in Op$*, there exists a path from purpose* p *to operation* op *iff the aim of invoking* op *is to achieve the purpose* $p \in P$*. For any purpose* $p \in P$*, there is a path from root* $\top$ *to it.*

Figure 3 is an example of operation to purpose connection tree for home loan approval BPEL process. For instance, the purpose of *loanApprove()* is *current*, which means the aim of *loanApprove()* execution is to process the home loan request.

4.3 Associating partner roles to recipients

Any BPEL process has more than one partners and each partner provides specialized service(s). However privacy policy specifies who can use particular collected personal data beyond the service provider. It is necessary to establish the relationship between partner roles of BPEL process

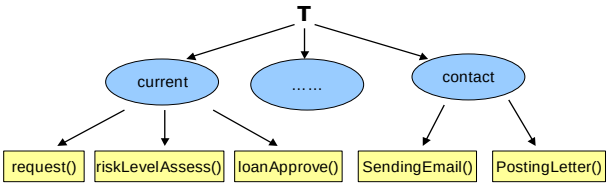

Figure 3. Operation-purpose connection tree for home loan approval BPEL process

and the recipients of privacy policy. Similarly we propose to build up a partner role to recipient connection tree to link partner roles of BPEL process to recipients of privacy policy.

Definition 2 ***(Role-recipient connection tree)*** *Given a set of roles which the BPEL process needs to interact with and the set of P3P predefined recipient vocabularies, we build a tree connecting recipient to roles. It is defined by* $Role2Recipient = \{\top, Role \cup R, E\}$*, where:*

- $\top$ *is the root of the tree.*
- R *is the set of P3P predefined recipient vocabularies(e.g., Ours).*
- $Role$ *is the party involving in the BPEL business process*
- E *is the set of edges, stating that if* (v_x, v_y) *are nodes connected by an edge* $E_{xy} \in E$ *then* v_y *is semantically a part of* v_x*.*

Proposition 2 *Given a role in the BPEL business process, there exists a path from recipient* r *to role* $role$ *in the Role2Recipient, iff the role involved to achieve an operation belongs to the recipient authorized to do that.*

Figure 4 is an example of role-recipient connection tree for home loan approval BPEL process.

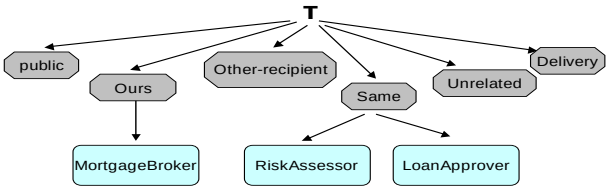

Figure 4. role to recipient connection tree for the home loan approval BPEL process

5 Overview of the verification System

Figure 5 shows the architecture of the privacy consistency verification system. The system consists of four components. First component (referred to as Model A) is a specification of an internal business process expressed in BPEL. The second one (referred to as Model B) is the privacy policy written in P3P. The third part is the relationship between BPEL process and privacy policy, which is described in the section 4, The core component of the system is the privacy consistency verification module which achieves the aim of checking whether the internal busines process of an organization conforms to the organization's privacy policy.

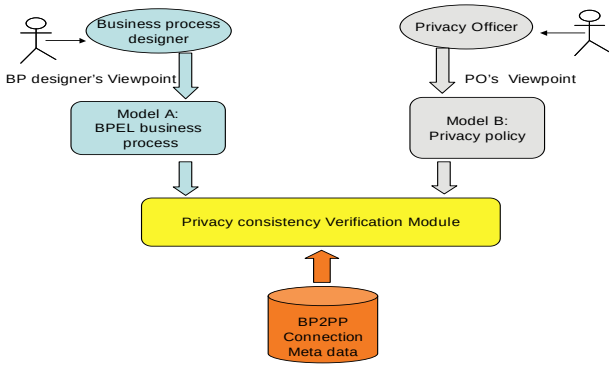

Figure 5. The architecture of privacy consistency verification system

In the following sections, we describe the core component in detail. The component is a consistency verification system based on graph transformation.

6 Graph transformation

This section introduces some basic definitions and notations for graph transformation[10]. Graph transformation has two main components, type graph and transformation rules.

A graph $G = (G_V, G_E, s_G, t_G, l_V, l_E)$ consists of disjoint sets of nodes V and edges E, two total functions $s, t : E \rightarrow V$ mapping each edge to its source and target node respectively, a function $l_V : V \rightarrow Lables$ assigning a label to each node, and a function $l_E : E \rightarrow Lables$ assigning a label to each edge. Lables are elements of a disjoint union of sets $Labels = X \cup C$, where X is a set of *variables* and C is a set of *constants*.

Definition 3 ***(A type graph)*** *TG represents the type information in a graph transformation system[10, 15, 14] and it specifies the node and edge types which may occur in the instance graphs modeling system states.*

For example, figure 6 depicts the possible type for our privacy consistency verification system. It has 5 node types and 4 edge types. The node types are $Data$, $Purpose$, Op, $Recipient$ and $Role$. Nodes of type $Data$ represent personal data, nodes of type $Purpose$ are purposes, nodes of type Op are operations, nodes of type $Role$ are roles, nodes of type $Recipient$ are recipients, respectively. The Data's corresponding information in privacy policy is modeled as two edges, one is the edge from $Data$ to $Purpose$ which means the Data will be used for the purpose. The other is the edge from $Data$ to $Recipient$ which means the recipient will use the Data. The Data's associated information in BPEL process is modeled as two edges as well, one is the edge from $Data$ to Op which means the operation will consume the message which contains Data. The other is the edge from $Data$ to $Role$ which means the data will be consumed by Op that is offered by $Role$, respectively.

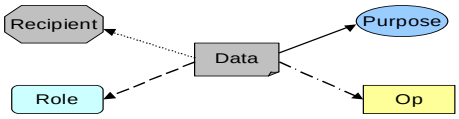

Figure 6. type graph for privacy consistency verification system

Definition 4 ***(A graph morphism)*** *$f : G \rightarrow H$ maps the nodes and edges of graph G to the nodes and edges of graph H while the shape of the graph G is preserved. In the labeled graph, the labeling of nodes and edges is preserved as well. More specifically, a graph morphism is defined as follows: Given two labeled directed graphs $G = (G_V, G_E, s_G, t_G, l_{V_G}, l_{E_G})$ and $H = (H_V, H_E, s_H, t_H, l_{V_H}, l_{E_H})$, then a graph morphism is a pair of functions: $\varphi = (\varphi_V : V_G \rightarrow V_H, \varphi_E : E_G \rightarrow E_H)$ with:*

- *$s_H \circ \varphi_E = \varphi_V \circ s_G$ and $t_H \circ \varphi_E = \varphi_V \circ t_G$, which denotes the preservation of the shape.*
- *$l_{V_G} = l_{V_H} \circ \varphi_V$ and $l_{E_G} = l_{E_H} \circ \varphi_E$, which denotes the preservation of the labeling of nodes and edges.*

Definition 5 ***(A transformation rule)*** *is formally given by a graph morphism $r : L \rightarrow R$, where both L and R are graphs, called left-hand side and right-hand side respectively. The graph L describes the elements that a graph must contain for the rule to be applicable. R is a new graph created by applying rule r [10, 21, 20].*

Example 1 *(Graph transformation rules)*

Figure 7 shows the schemes for the transformation rules of the privacy consistency verification system. The labels for the nodes $(Data_x, Purpose_y, Op_z ...)$ of the rules are variables taken from the set of variables in Labels.

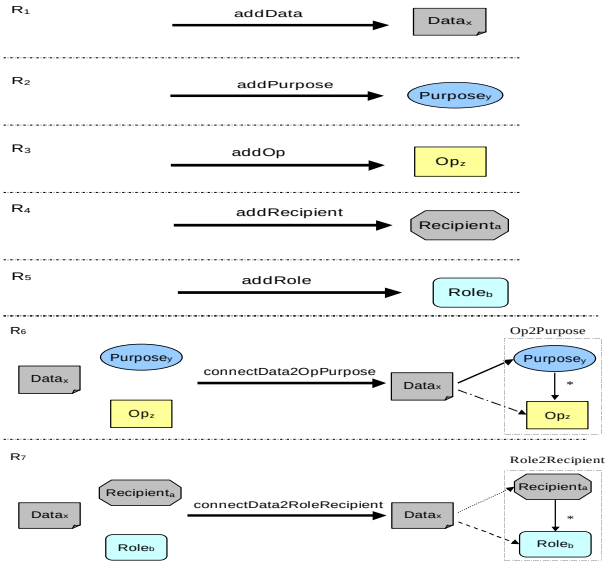

Figure 7. Transformation rules

There are seven rules in the privacy consistency verification system. The first five rules have empty left-hand side and they are used to add the obtained privacy related information(e.g., information in table 4 and table 5) into the privacy consistency verification system. More specifically, rule *addData* is used to insert a node $Data_x$ which is an instance node of $Data$ node type. Rule *addPurpose* is used to add a node $Purpose_y$, an instance node of $Purpose$ node type. This is allowed if and only if the node $Purpose_y$ does not exist in the system. Rule *addOp* is used to add a node Op_z, an instance node of Op node type, if the node Op_z does not exist in the system. Rule *addRole* is used to add a node $Role_b$, an instance node of $Role$ node type, if and only if the node $Role_b$ does not exist in the system. Role *addRecipient* is used to create a node $Recipient_a$, an instance node of $Recipient$ node type if and only the node $Recipient_a$ does not exist in the system. Rule *connectData2OpPurpose* is used to model the relationship between $Data_x$ and Op_z, and the relationship between $Data_x$ and $Purpose_y$ by inserting an edge from node $Data_x$ to node Op_z, and an edge from node $Data_x$ to node $Purpose_y$. Similarly, rule *connectData2RoleRecipient* is used to visualize the relationship between $Data_x$ and $Role_b$, and the relationship between $Data_x$ and $Recipient_a$ by adding an edge from node $Data_x$ to node $Recipient_a$, and an edge from node $Data_x$ to node $Role_b$.

The application of rule $r : L \rightarrow R$ to a graph G takes

following steps:

- Find the left-hand side L as a subgraph in G. The subgraph is denoted by $L(G)$
- Remove all nodes and edges from $L(G)$ that are not present in R
- Add all nodes and edges in R that are not present in L. The nodes occurring both in L and in R are used to connect the new parts to the old ones.

Example 2 *(Application of a graph rule)*

Figure 8 shows how to apply the rule *connectData2OpPurpose* to visualize that $Email$ will be used for $current$ purpose, and the message which contains $Email$ will be consumed by the operation $request()$. Assume the node *Email*, the node `current` and the node *request()* are already in the system. The left-hand side L of the rule *connectData2OpPurpose* occurs several times in G. In one possible match, the node $Data_x$ in L is associated to the node $Email$ in G, the node $purpose_y$ to the node $current$, and node Op_z to node $request()$ respectively. The rule application adds two edges, one is from node $Email$ to node $current$, the other is from $Email$ to node $request()$. Figure 9 shows how to apply rule *connectData2RoleRecipient* to G and obtain graph H by inserting two edges, one is from node $Email$ to node Our, the other is from $Email$ to node $MortgageBroker$.

7 Privacy consistency verification Framework

This section presents our graph transformation based framework of privacy consistency verification between BPEL process specification and P3P privacy policy . The framework consists of four components: a type graph that provides the type information of verification framework, a set of graph rules(i.e., rewriting rules), which is used to visually specify how to build the mapping between privacy related elements of BPEL process and its relevant information in privacy policy, a set of negative constraints to specify graphs that shall not be contained in any system graph, and a set of positive constraints to describe graphs that must be explicitly constructed as parts of a system graph.

7.1 Graph-based Correctness of BPEL Process

In this section, we show how the graphical formalism is used to prove the correctness of BPEL process specification in terms of privacy policy. An example will be used to illustrate the reasoning mechanism.

Some properties of the system can be defined to specify the privacy requirement of BPEL process. A consistent constraint is a property of a graph that has to be preserved by application of rules. These requirements can be modeled as a set of graphical constraints.

Definition 6 *A graphical constraint is a graph that depicts a forbidden or a required structure. Both positive and negative constraints are formally specified by total graph morphisms $c : X \rightarrow Y$. It is the semantics of the morphism that distinguishes between positive and negative constraints.*

Definition 7 ***(Constraint satisfaction)*** *A total injective graph morphism $q : X \rightarrow G$ satisfies a positive (negative)constraint $c : X \rightarrow Y$ if there exists (does not exist) a total injective graph morphism $p : Y \rightarrow G$ such that $X \overset{c}{\rightarrow} Y \overset{p}{\rightarrow} G = X \overset{q}{\rightarrow} G\ i.e\ p\ o\ c = q$. A consistency constraint c is satisfied by a graph G, written $G \models c$, if for all morphisms p there is a total morphism q.*

Example 3 *(Constraints for privacy consistency verification framework)*

During the transformation, the system must ensure that the defined properties are preserved. Figure 10 shows two positive constraints for privacy consistency verification framework. Constraint c_1 specifies that there should exist at least one path from purpose $Purpose_y$ to operation Op_z. c_2 describes that at least one path from recipient $Recipient_a$ to role $Role_b$.

When the *connectData2OpPurpose* rule is applied, the graphical constraint c_1 must be satisfied. More specifically, if the message containing the personal data item $Data_x$ is consumed by BPEL process operation Op_z, the $Purpose_y$ should be one of the purposes in $Data_x$'s related P3P privacy statement. In other words, there should exist a path from node $Purpose_y$ to node Op_z in operation to purpose connection tree.

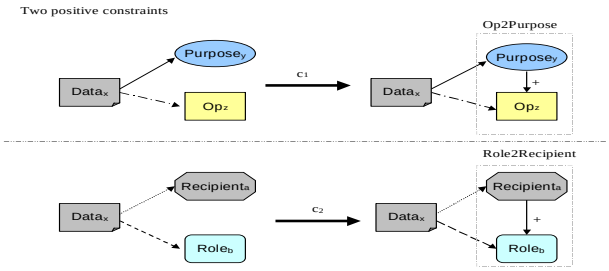

Figure 10. Two positive constraints for privacy verification framework

Figure 11 shows two negative constraints for privacy consistency verification framework. Both negative constraints c_3 and c_4 specify that if message containing personal date item $Data_x$ is consumed by internal business

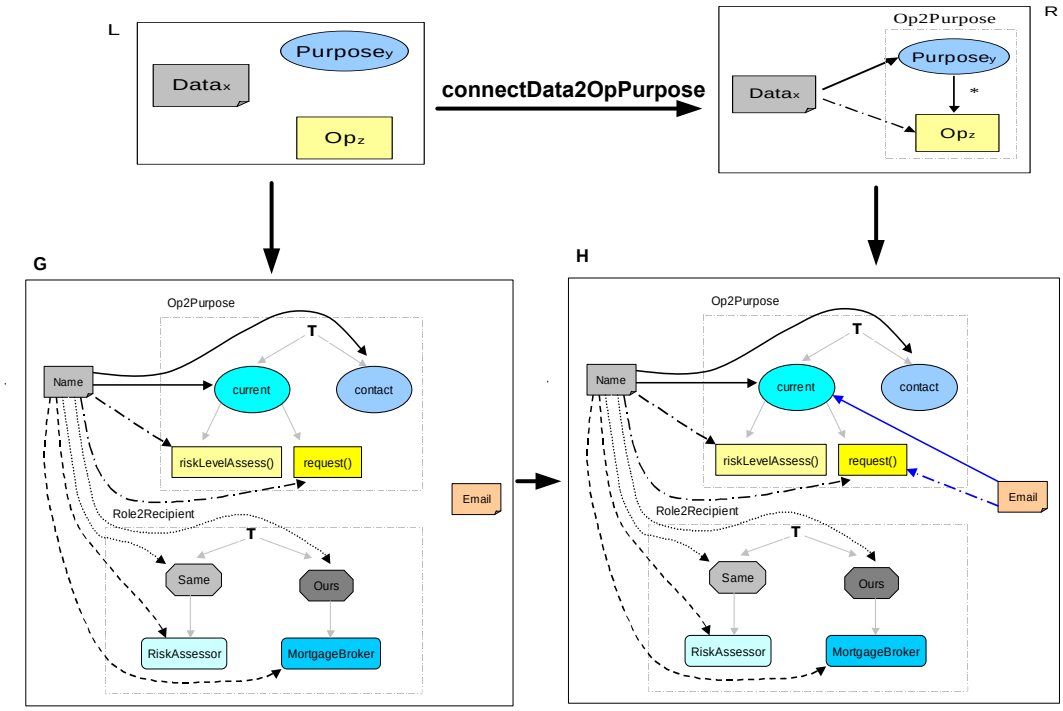

Figure 8. Application of rule *connectData2OpPurpose*

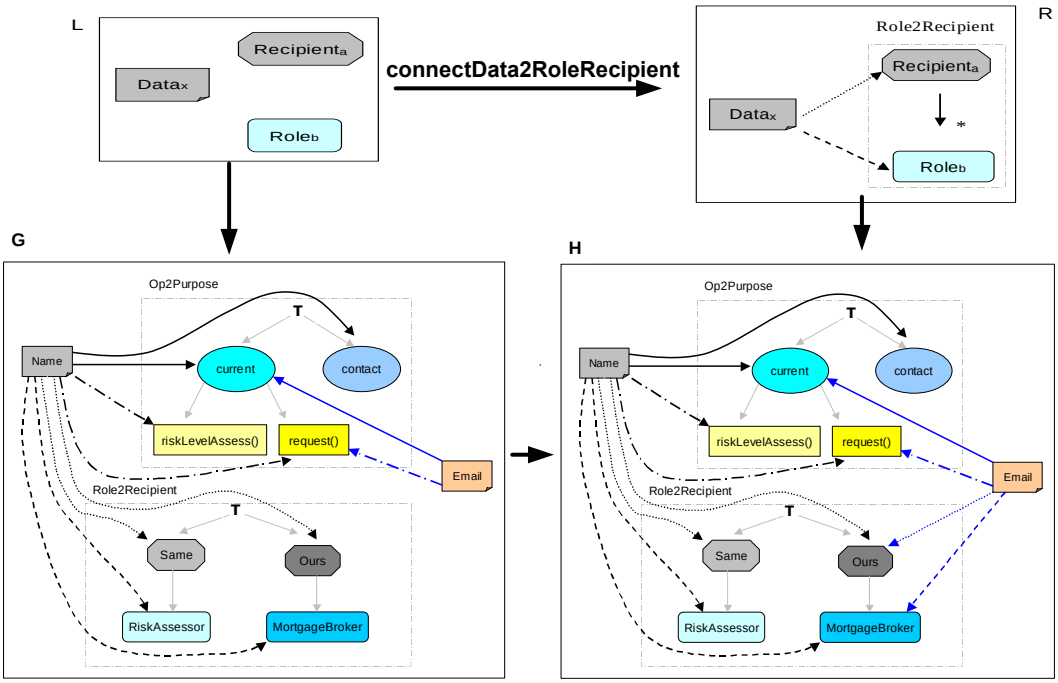

Figure 9. Application of rule *connectData2RoleRecipient*

operation Op_z offered by $Role_b$, however there is no P3P privacy statement exists whoes personal data item is $Data_x$.

Then the BPEL process violates the privacy policy.

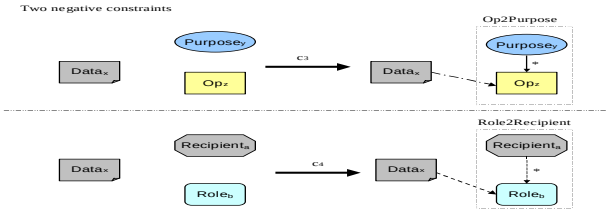

Figure 11. Two negative constraints for privacy verification framework

7.2 The Framework

Definition 8 ***(Privacy consistency verification framework)*** *A privacy consistency verification framework is a tuple* $BP_P = (TG, Rules, Pos, Neg)$*, where*

- TG *is a type graph*
- *Rules is the set of transformation rules*
- *Pos is a set of positive constraints*
- *Neg is a set of negative constraints.*

The graphs that can be constructed by the rules of a framework represent the possible system states. These graphs are called *system graphs* in the sequel. We defined the constraints to declare wanted and unwanted system graphs. Do the rules construct all the wanted system graphs required by the positive constraints and do the rules prevent the construction of unwanted system states expressed by the negative constraints? If the rules do so, the framework is *coherent*.

Definition 9 ***(Coherence)*** *A privacy consistency verification framework is positive coherent (resp negative coherent) if all system graphs satisfy the constraints in Pos (rep Neg). A privacy framework is coherent if it is both positive and negative coherent.*

Theorem 1 *Given a rule r, a graphical constraint c, a graph G consistent w.r.t. c, the graph H resulting from an application of r to G is consistent w.r.t. c.*

Proof sketch The proof consists in ensuring consistency by preconditions. We provide a general *construction* that transforms constraint into preconditions for individual rules. It is done in three steps . First a constraint c is specialized to a postcondition called $A_R(c)$ for a given rule. Then this postcondition is anticipated by an equivalent precondition called $r^{\alpha}(A_R(c))$. The third step is minimising the construction by using the assumption that the given graph G already satisfies the consistent constraint c.

8 Related Work

This section provides an overview of relevant work in privacy enforcement, BPEL verification and graph transformation based verification.

Naturally, privacy enforcement is always considered together with access control and combined with data management technologies. IBM's Enterprise Privacy Authorization Language(EPAL) is designed for organizations to specify their internal privacy policies[5, 24]. Schunter et al have showed how to derive P3P policy from EPAL in [13], and they have presented a framework to enable an enterprise to enforce policy promise across the whole IT system of the organization by extending access control in [7]. PARBAC model[11] has been proposed to enforce purpose component of privacy policy within an organization by extending role-based access control model. The eXtensible Access Control Markup Language(XACML) is an OASIS standard access control language[18, 3], and it already takes purpose, one of elements of privacy policy, into account[19]. All these works are focus on privacy enforcement within a closed system rather than an open, intranet-based and internet-based system(e.g., BPEL business process).

WS-Privacy[1] is a specification to addresses how privacy practices can be stated and implemented by Web Services. By using a combination of WS-Policy[6], WS-Security[2] and WS-Trust[22], organizations can state and indicate conformance to stated privacy policies. This specification will describe a model for how a privacy language may be embedded into WS-Policy descriptions and how WS-Security may be used to associate privacy claims with a message. Finally, this specification will describe how WS-Trust mechanisms can be used to evaluate these privacy claims for both user preferences and organizational practice claims. Unfortunately, at present, there are very limited information available regarding this specification.

There are a few works on BPEL verification. In [17], Alex Martens proposed a petri net based framework to check whether an executable BPEL process is consistent with the predefined abstract BPEL process. VERBUS[12] is an automatic model-checking based verification system to ensure the correctness of the defined BPEL business processes in terms of a set of predefined properties. However none of these works address BPEL verification in terms of privacy.

[14] presented a graph transformation based formalization for role based access control to verify the properties of a given graph-based RBAC. In [15], Manuel Koch proposed a graph-based specification formalism for access control policies to detect and resolve conflicts.

9 Conclusion and Future work

In this paper, we present a graph transformation based approach to check whether an internal business process of an organization(e.g., implemented using a standard Web service composition language such as BPEL) conforms to the organization's privacy policy. Our approach is motivated by two main facts: the privacy concerns has become one of the major concerns for organizations and indiviuals, and it is important that any privacy policy of an organization is properly enforced within the business processes of the organization. We propose a formal consistency verification using graph transformations. Graph transformation is a graphical specification technique giving an intuitive visual description of privacy consistency conditions on business process and the dynamic structures that occur in business process.

One of our future work is to implement a tool for business process designer to build the relationship between business process vocabularies and P3P vocabularies. After that, a privacy-aware BPEL engine that reflects the proposed framework will be implemented. This engine can monitor the BPEL process execution to ensure that the privacy policy is not violated during BPEL process execution. The privacy enforcement in organization's information system will be considered as well(e.g., how to enforce *retention* component of privacy policy). Another interesting aspect for future work is to extend proposed framework for checking the compatibility between two BPEL processes in order to replace one service with the other. The replacebility should consider whether privacy policies are poreserved.

References

[1] Web Service Privacy. *http://www.serviceoriented.org/ws-privacy.html*.

[2] A.Nadalin, C.Kaler, R.Monzillo, and P.Hallam-Baker. Web service security: Ws-security specification. *http://docs.oasis-open.org/wss/2004/*, Feb 2006.

[3] A. Anderson. Comparison of two privacy policy languages: Epal and xacml, tech report. Technical Report TR-2005-147, Sun Microsystems Laboratories, 2005.

[4] T. Andrews and F. Curbera. Business process execution language for web services(1.1). *ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf*, May 2003.

[5] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter. Enterprise privacy authorization language(EPAL 1.2). *http://www.w3.org/Submission/EPAL/*, November 2003.

[6] S. Bajaj and D. Box. Web services policy framework(ws-policy). *http://specs.xmlsoap.org/ws/2004/09/policy/ws-policy.pdf*, Sep 2004.

[7] P. A. Calvin S. Powers and M. Schunter. Privacy promises, access control, and privacy management. In *Int'l Symp. Electronic Commerce (ISEC-02)*, 2002.

[8] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana. Web Services Description Language (WSDL) Version 2.0 . *http://www.w3.org/TR/wsdl20*, August 2005.

[9] L. Cranor, M. Langheinrich, and M. Marchirio. A P3P preference exchange language 1.0. *W3C working draft*, Feb 2001.

[10] H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg. *Handbook of Graph Grammars and Computing by Graph Transformation.* World Scientific, 1999.

[11] Q. He. Privacy enforcement with an extended role-based access control model. Technical Report TR-2003-09, NCSU Computer Science, 2003.

[12] C. D. K. Jess Arias Fisteus, Luis Snchez Fernndez. Formal verification of bpel4ws business collaborations. In *5th International Conference on Electronic Commerce and Web Technologies (EC-Web '04)*, LNCS, Zaragoza, Spain, August 2004. Springer.

[13] G. Karjoth, M. Schunter, and E. V. Herreweghen. Translating privacy practices into privacy promises -how to promise what you can keep. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy*, pages 135–146. IEEE Computer Society, June 2003.

[14] M. Koch, L. V. Mancini, and F. Parisi-Presicce. A formal model for role-based access control using graph transformation. In *Proc. of 5th ESORICS*, pages 122–139. Springer, 2000.

[15] M. Koch, L. V. Mancini, and F. Parisi-Presicce. Graph-based specification of access control policies. *J. Comput. Syst. Sci.*, 71(1):1–33, 2005.

[16] M. L. Lorrie Cranor, Giles Hogben, M. Presler-Marshall, J. Reagle, and M. Schunter. The platform for privacy preference 1.1(P3P 1.1) specification. *http://www.w3.org/TR/2004/WD-P3P11-20040720*, July 2004.

[17] A. Martens. Consistency between executable and abstract processes. In *2005 IEEE International Conference on e-Technology, e-Commerce, and e-Services (EEE 2005), 29 March - 1 April 2005, Hong Kong, China*, pages 60–67. IEEE Computer Society, 2005.

[18] T. Moses. extensible access control markup language (xacml) version 2.0. *http://docs.oasis-open.org/xacml/2.0/access-control-xacml-2.0-core-spec-os.pdf*, Feb 2005.

[19] T. Moses. Privacy policy profile of xacml v2.0. *http://docs.oasis-open.org/xacml/2.0/access-control-xacml-2.0-privacy-profile-spec-os.pdf*, Feb 2005.

[20] P.Baldan, B.Knig, and B.Knig. A logic for analyzing abstractions of graph transformation systems. In *In Proc. of SAS '03,International Static Analysis Symposium*, page 255272. Springer LNCS 2694, 2003.

[21] R.Heckel and A.Wagner. Ensuring consistency of conditional graph grammars: a constructive approach. *Theoretical Computer Science*, (1), 1995.

[22] a. e. Steve Anderson. Web Services Trust Language(WS-Trust) . *http://specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf*.

[23] A. F. Westin. *Privacy and Freedom*. Bodley Head, 1970.

[24] A. I. A. william Stufflebeam and Q. He. Specifying privacy polices with P3P and EPAL: Lessons learned. In *Proc. of Workshop on privacy in the Electronic Society(WPES'04)*, 2004.