Social Network Profile and Policy

Juliana Mitchell-Wong, Ryszard Kowalczyk, Bao Quoc Vo  
Centre for Information Technology Research  
Swinburne University of Technology  
Hawthorn, Victoria, Australia  
{jmitchellwong, rkowalczyk, bvo}@ict.swin.edu.au

Abstract

A social network describes an entity’s relationship with other entities for interacting. The digital world currently consists of many isolated communities or domains that enable entities to form social networks within each community but not across communities because of the undisclosed, incomplete or incompatible connectivity protocols and data representation. A specification for the social network profile and policy is presented in this paper with the aim to address these issues by providing a common representation for defining the entity’s relations and their social attributes, and its connectivity expectations and preferences. Furthermore, this profile and policy specification paves the path for supporting entities in automating their actions and managing their social networks.

1 Introduction

A social network represents the social connectivity of an entity to other entities, where an entity is an individual or community. An entity also develops links with the social applications it uses such as chat, e-mail and games.

Social networks in the current digital world are poorly modelled. Relationships to entities and links to applications are typically formed within isolated communities that are formed on different social platforms such as Facebook\(^1\) and Orkut\(^2\) where they use their proprietary identity system and applications. The issue that arises from this is that an entity’s social network that is established in one community is not transferable to another community, and an application used in one community is not available in another community. Often this is due to the undisclosed, incomplete or incompatible data representation.

In recent times, research in the area of digital social networks has focused on developing open standards. OpenID has become the industry accepted standard for an identity system while the battle for an application development standard is being fought between Google’s OpenSocial\(^3\) and Bebo’s Open Application Platform\(^4\). However at present, there are no standards for managing social networks.

An entity’s relationship to other entities forms its social network. Typically relationships are formed based on trust, and other information about the entity such as interests, locality, and occupation. Trust of the entity’s relations is available from its social network profile whilst other information of the relations are available from the relations’ personal profile. However, relationships in a social ecosystem often do not cross community boundaries because of undisclosed, incomplete or incompatible data representation.

A social ecosystem is the relationship mesh of all entities. Typically connectivity between relations are manually executed; or if automated, they are not managed or monitored. This is because the entity’s expectations and preferences are undisclosed, incomplete or incompatible with others. Requiring entities to explicitly define their expectations and preferences in a policy facilitates the entity to negotiate a connectivity protocol, automate its actions, and manage its social network.

The social networking profile and policy are customised by the entity to reflect its relations, and its expectations and preferences for connecting to them, respectively. The social network may be adapted or revised independently by the entity, or according to the positive or negative feedback from other entities.

This paper presents the social networking profile and policy specification in Section 2; and its use for achieving interoperability between entities, automation of actions and management of social networks in Section 3. Future work is described in Section 4 and a conclusion presented in Section 5.

2 Social Network Profile and Policy

The social network profile and policy specification is represented in XML. XML provides extensibility to the profile and policy through schemas where multiple information domains can be included. This flexibility enables the entity to configure their profile and policy to their needs. The profile and policy can also be adopted or referenced to eliminate data redundancy, thus also reducing the effort required to modify data to one location instead of multiple locations.

The repository of an entity consists of multiple profiles and policies. A profile and a policy for each of its relations and itself.

\[ \text{EntityRepository} = (SNProfile}^+, \text{SNPolicy}^+) \]

where

- \( E \) represents one instance of the element \( E \)
- \( E^+ \) represents one or more instance of the element \( E \)
- \( E^* \) represents any number of instance of the element \( E \)

2.1 Profile

The social network profile consists of the local name of the entity, identifiers of the profile, related entities, linked applications, managed content, and the social attributes of its related entities. The identifier represents the location the profile can be accessed; the related entity represents the entity’s different relationships such as friends, colleagues and clubs; linked applications represent the different applications that the entity uses such as chat and email, and the actions they provided to the entity; managed content represent data the entity has control over such as profile and policy files; and the social attributes are the entity’s assessed attributes of its related entities or linked applications such as trust. This profile is specified in set notations and an example is provided using the XML syntax.

2.1.1 Set Representation

\[
\begin{align*}
\text{SNProfile(name)} & = (\text{Identifier}^+, \text{Entity}^*, \text{Application}^*, \text{Content}^*, \text{SocialAttribute}^*) \\
\text{Entity(name)} & = (\text{Identifier}^+) \\
\text{Application(name)} & = (\text{Identifier}^+, \text{Service}^*, \text{Action}^*) \\
\text{Content(name)} & = (\text{Identifier}^+) \\
\text{SocialAttribute(name)} & = (\text{Entity}^*, \text{Application}^*, \text{Context}^*, \text{Attribute}^*) \\
\text{Service(name)} & = (\text{Identifier}^+) \\
\text{Action(name)} & = (\text{Identifier}^+) \\
\text{Attribute(name, datatype)} & = \text{value} 
\end{align*}
\]

2.1.2 An Example XML Representation

```xml
<SNProfile name="Tom">
  <Identifier>example.com/tom/snprofile</Identifier>
  <Entity name="Harry">
    <Identifier>example.com/harry</Identifier>
  </Entity>
  <Entity name="Friends">
    <Identifier>example.com/tom/friends</Identifier>
  </Entity>
  <Application name="MSN">
    <Identifier>msn.com</Identifier>
    <Service>chat</Service>
    <Action name="SendChat">
      msn.com/function/sendMessage
    </Action>
    <Action name="ReceiveChat">
      msn.com/function/receiveMessage
    </Action>
  </Application>
  <Content name="HarryMSNChat">
    <Identifier>example.com/tom/content/MSNChat/harry</Identifier>
  </Content>
  <SocialAttribute type="application_trust">
    <Application>MSN</Application>
    <Context>chat</Context>
    <Attribute name="trustvalue" datatype="example.com/ontology/social/datatype">
      0.9
    </Attribute>
  </SocialAttribute>
</SNProfile>
```

Tom's social network profile defines an identifier. He has two entity relations, Harry and a group of friends; and a link to the MSN application which provides a chat service that has actions send and receive message. He keeps his past MSN chats with Harry; and his trust of the MSN application as a chat service is 0.9.

2.2 Policy

The social network policy consists of the local name of the entity, and the policy set, policy and rule elements of the entity. The combining algorithm of the policy set and policy elements defines the method to combine multiple social network policy elements. The target of the social network policy elements describes the applicability of the policy to a social network request; and obligation describes the action when the request evaluates to the specified result. This policy is specified in set notation, and an example is provided using the XML syntax.

This social network policy specification is not specific to any policy language and can use the different existing languages such as Protune, Ponder, Rei, KAoS, and XACML. Each of these languages have their own set of features. In this paper the social network policies are based on the
XACML language which is an OASIS standard. It has been mapped to description logic in the web ontology language (OWL-DL) where it uses DL reasoners to provide analysis services for policy comparison, redundancy and verification [3]. XACML has been used by numerous web specifications such as the web services policy (WS-Policy) and the security assertion markup language (SAML). The social network policy is represented using XACML policy elements.

2.2.1 Set Representation

\[
\begin{align*}
\text{SNPolicy}(name) & = (\text{PolicySet}^*, \text{Policy}^*, \text{Rule}^*) \\
\text{PolicySet}(name, alg) & = (\text{Target}, \text{PolicySet}^*, \text{Policy}^*, \text{Obligation}^*) \\
\text{Policy}(name, alg) & = (\text{Target}, \text{Rule}^+, \text{Obligation}^*) \\
\text{Rule}(name, effect) & = (\text{Target}, \text{Condition}) \\
\text{Target} & = (\text{Entity}^*, \text{Resource}^*, \text{Action}^*) \\
\text{Obligation}(fulfilOn) & = (\text{Action}) \\
\text{Resource}(name, type) & = (\text{Identifier})
\end{align*}
\]

2.2.2 An Example XML Representation

```xml
<SNPolicy name="Tom">
  <Policy name="Chat" alg="permit-override">
    <Target>
      <Action name="MSN chat"/>
      <Identifier>example.com/tom/action/chat</Identifier>
    </Target>
    <Rule name="All" effect="deny"/>
    <Rule name="Harry" effect="permit"/>
    <Obligation fulfilOn="permit">
      <Action name="MSN initiate chat">
        <Identifier>msn.com/function/init_chat</Identifier>
      </Action>
    </Obligation>
    <Obligation fulfilOn="deny">
      <Action name="Add to blacklist">
        <Identifier>example.com/function/add_blacklist</Identifier>
      </Action>
    </Obligation>
  </Policy>
  <Rule name="All" effect="deny"/>
  <Target/>
  <Condition/>
</SNPolicy>
```

Tom’s policy consists of a policy and two rule elements. The policy element determines the chat permission using the two rule elements. One rule element denies all requests and the other permits Harry provided he uses the MSN application. The combination algorithm is used to set preferences to the evaluation result, in this case permit takes precedence over deny. For example, when both rules are applicable, the rule permitting Harry to chat is applied before the rule that denies all requests. There are two obligations for the policy element, one for a permit result and the other for a deny result. The obligation for a permit result requires that Tom initiates a MSN chat with the entity that requested the chat, whilst the obligation for the deny result requires that Tom adds to the blacklist the entity that requested the chat.

2.2.3 XACML Representation

The policy set, policy and rule elements of the proposed SNPolicy can be directly mapped to those of the XACML language. One difference in syntax is the use of Entity in the SNPolicy instead of Subject in XACML. The schema for a social network policy is presented below.

```xml
<xs:element name="SNPolicy" type="sn-xacml:SNPolicyType"/>
<xs:complexType name="SNPolicyType">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="xacml:PolicySet"/>
    <xs:element ref="xacml:Policy"/>
    <xs:element ref="xacml:Rule"/>
  </xs:sequence>
</xs:complexType>
```

3 Interoperability, Automation and Management

The social network profile and policy provides an explicit and structured representation of an entity’s relation, and its expectation and preferences for connectivity. It facilitates the entity’s interoperability with others, automation of its actions and the management of its social network.

3.1 Interoperability

The common social networking profile and policy specification of an entity enable them to understand the social
network of others. This understanding enables the entities to integrate their policies to form a mutually agreed connectivity protocol. A number of different strategies to integrate policies exist including:

- forming a new policy from the intersection of the predicates of the rules in the two policies where the target of the rules and policies match exactly [1].
- entities negotiate by sending policy proposals [2, 5].
- entities define their flexibility towards the enforcement of their policy in the protocol [4].

3.2 Automation

The social networking policy is the explicit specification of an entity’s expectations and preferences for connecting to one another. Within these expectations and preferences, the requirements and obligations of an action are defined, thus creating a chain of actions.

To illustrate the automation of actions that are defined in policies, the example policy from Section 2.2.2 is referenced. Harry decides he want to chat with Tom and thus sends a request. This request reaches Tom where it is evaluated against his policy. Tom’s policy permits Harry to chat with him provided he uses the MSN application; and as a consequence of the permission, Tom initiates a session.

3.3 Management

Managing a social network involves monitoring connectivity and setting rewards or penalties for the monitored actions. This can be specified in a policy whose monitored target is specified in the target section of the policy and the reward or penalty is defined as the obligation.

To illustrate the management of the social network, the example policy from Section 2.2.2 is referenced. Sally decides she wants to chat with Tom and thus sends a request. This request reaches Tom where it is evaluated against his policy. Tom’s policy denies Sally from chatting with him; and as a consequence of the denial, Tom adds Sally to his blacklist.

4 Future Work

This profile and policy specification only covers entities in the social environment. Future work includes specifying the profile and policy of other social elements such as applications and agents.

The profile and policy specification of these three social networking elements will enable social networking systems to be automated and managed. The entities and applications are able to state their expectations and preferences, and agents are able to specify their decision making and learning behaviour.

5 Conclusion

A profile and policy specification for social networking agents has been presented. Both the profile and policy are representable in XML making them machine and human readable, scalable, customisable and portable.

The entity’s profile and policy reflects the entity and its interaction expectations and preferences. Together they provide the ability to automate the execution of interactions and to manage social networks by monitoring and enforcing policies.

It is envisaged that this social network profile and policy will be used in conjunction with the web service technology to provide humans with the experience that is assistive, time effective, and complements the user’s social connectivity in the physical world.

6 Acknowledgements

This work has been supported by the Agent-Enabled Social-Networks project in collaboration with VastPark under the Australian Research Council’s Linkage funding scheme (ARC Linkage Project LP0562500).

References


