Social Networks API

Alban Galland

INRIA Saclay

January 26, 2009

Outline

- Different kinds of API
  - Micro-formats : XFN
  - Ontologies : FOAF
  - Interactions specifications : OpenSocial
  - Applications specifications : FBML
- About the design of SN APIs

Mirco-formats (1)

- XFN = “XHTML Friends Networks”
- Specification : http://www.gmpg.org/xfn/11
- Principe : for each hyperlink <a>, use the rel attribute to add social meta-data to the link.
- Usage : mainly on blogs and blogrolls
- Attributes types : friendship (contact,acquaintance,friend), physical (met), professional (co-worker, colleague), geographical (co-resident, neighbor), family (child, parent, sibling, spouse, kin), romantic (muse, crush, date, sweetheart), identity (me)

Mirco-formats (2)

- Plenty of other mirco-formats linked to SN
  - hCard : contact information
  - Adr : postal address
  - geo : localization
  - rel-tag : tagging
  - …

Mirco-formats (3)

- *Micro-formats are cool for SN in P2P because they are naturally distributed*
- *Micro-formats are bad for SN in P2P because they are hard to explore as a whole*

Ontologies (1)

- FOAF = Friend Of A Friend
- Specification : http://xmlns.com/foaf/spec/
- Ontology expressed with RDF and OWL
- Classes: Agent, Document, Group,Image, OnlineAccount, OnlineChatAccountm OnlineEcommerceAccount, OnlineGamingAccount, Organization, Person, PersonalProfileDocument, Project
- Properties: accountName, accountServiceHomepage, aimChatID, based_near, birthday, currentProject, depiction, depicts, dnaChecksum, family_name, firstName, fundedBy, geekcode, gender, givenname, holdsAccount, homepage, icqChatID, img, interest, isPrimaryTopicOf, jabberID, knows, logo, made, maker, mbox, mbox_sha1sum, member, membershipClass, msnChatID, myersBriggs, name, nick, openid, page, pastProject, phone, plan, primaryTopic, publications, schoolHomepage, sha1, surname, theme, thumbnail, tipjar, title, topic, topic_interest, weblog, workInfoHomepage, workplaceHomepage, yahooChatID

Ontologies (2)

- A large number of way of identification : mail, chat account, DNA, openId...
- Most of the properties are in test or unstable
- Social Graph API from Google use both XFN and FOAF

Ontologies (3)

- *Ontologies are cool for SN in P2P because they could be distributed and are naturally designed for reasoning on data**
- *Ontologies are bad for SN in P2P because they are so complex that they lead to instability*

Interactions specifications (1)

- OpenSocial = google + linkedIn, mySpace, Friendster, hi5 ...
- Principle : a couple (viewer,application) queries a platform (container) about the social graph centered on a distinct user (owner)
- Three APIs : JS, Restful and RPC protocols
- People : rich profile (direct translation of what you find in the profile page of Orkut)
- Relationship : two set of friends : owner_friends and viewer_friends + distance parameter

Interactions specifications (2)

- Activities : log of what user has done
- Persistence : data of the application
- Some notions of UI due to the “widget” approach of the JS API
- Capabilities discovery
- See also FaceBook APIs : FBJS, REST-like, Connect...

Interactions specifications (3)

- *Interaction specifications are cool for SN in P2P because they help transferring the Social Graph and using it to do something useful*
- *Interaction specifications are bad for SN in P2P because it is impossible to control what an application do with your data*

Applications Specifications (1)

- FBML = FaceBook Markup Language
- A subset of HTML + some FBML tags
- FBML is translated by FaceBook in HTML embedded in a frame, an email...
- The FBML is produced by the application server. Some interactions are taken in charge by FaceBook, other are passed to the application
- User Interface : classical HTML UI tags + some display functions (embedded media,)
- Programming structure : if-then-else, switch ...

Applications Specifications (2)

- Accessing data : FQL or functions
- FQL (FaceBook Query Language)
  - Query allowed by FBML
  - SN graphs see as a set of tables
  - It is queried by SELECT-FROM-WHERE queries
  - Constraints on the query insure privacy and prevent too wide crawling (notion of indexability)
- Some complex tags are wrapping of FQL with some translations (like your fb:name is “you”)

Applications Specifications (3)

- *Applications Specifications are cool for SN in P2P because the data are not send to the application but directly to the viewer*
- *Application Specifications are bad for SN in P2P because they limit what you can do with the data*

Design of SN APIs (1)

- What is cool ?
  - Distribution,
  - Reasoning,
  - Transfer,
  - Protection
- What is bad?
  - Dispersion,
  - Complexity,
  - Lake of privacy,
  - Lake of expressiveness

Design of SN APIs (2)

- Define people
  - Define an ID
  - Define an authentication method
  - Define a profile
- Define relationship between people
  - Define type of relationship
  - Define instance of relationship
  - Define groups

Design of SN APIs (3)

- Define Applications
  - Define Id and authentication method
  - Define activities (“log”, monitoring)
  - Define persistence (data of the application)
- Define access to social graph
  - Define model of a the graph
  - Define right access to the graph
  - Define query on the graph