The Semantic Desktop for Application Developers

George Goldberg
Berlin Desktop Summit 2011

This talk is intended for:

- KDE Application Developers
- Not Nepomuk Experts
- The Curious

This talk will cover:

1) What can applications use Nepomuk for?

2) Why should they use Nepomuk?

3) How should they use Nepomuk?

4) A case study: KDE-Telepathy.

What is Nepomuk?

A place where applications can share information about things with each other.

1) What can applications use Nepomuk for?

Example: Photo Manager

- Tags
- Albums
- People in pictures
- Location of pictures

Example: Address Book

- Last call/chat times
- Recent messages
- Presence
- Pictures of contacts
- Interaction history

2) Why should applications use Nepomuk?

1) Digikam

Look for photo of friend

2) KAddressBook

Find friend's email

3) KMail

Email photo as attachment

Collabora

Look for photo of friend

Send photo to friend

Solution Requirements

- Common API for data interchange
- Must not be application-specific
- Must be a part of the environment in which applications operate

3) How should applications use Nepomuk?

Ontologies

- Describe the data so that any application can understand it
- Effectively an API for the data
- Classes, properties, relations etc

Ontologies: Example

```
nco:PersonContact
      a        rdfs:Class ;
      rdfs:comment "A Contact that denotes a Person. A pe
      rdfs:label "PersonContact" ;
      rdfs:subClassOf nco:Contact .

nco:IMAccount
      a        rdfs:Class ;
      rdfs:comment "An account in an Instant Messaging sy
      rdfs:label "IMAccount" ;
      rdfs:subClassOf nco:ContactMedium .

nco:PostalAddress
      a        rdfs:Class ;
      rdfs:comment "A postal address. A class aggregating
      rdfs:label "PostalAddress" ;
      rdfs:subClassOf nco:ContactMedium .
```

A Contact (much like a vcard)

An Instant Messaging account of a contact

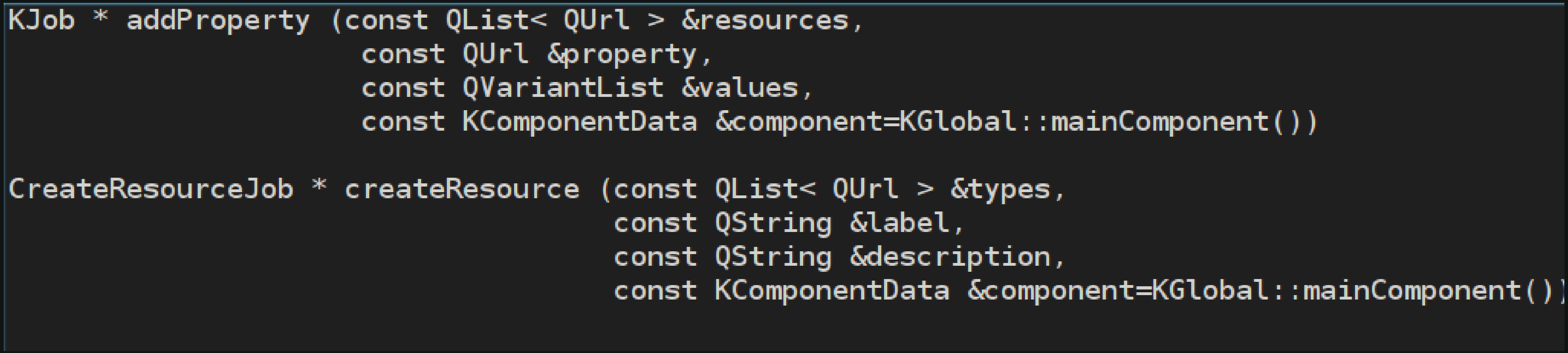
A postal address of a contact

Collabora

PIMO Ontology

- Two main sets of ontologies:
  - Nepomuk ontologies represent implementation-level constructs
  - PIMO ontology represents real-life entities

API



```
KJob * addProperty (const QList< QUrl > &resources,
                     const QUrl &property,
                     const QVariantList &values,
                     const KComponentData &component=KGlobal::mainComponent())

CreateResourceJob * createResource (const QList< QUrl > &types,
                                    const QString &label,
                                    const QString &description,
                                    const KComponentData &component=KGlobal::mainComponent())
```

- Asynchronous
- KJob based

Collabora

API

- Query Service
  - Allows you to query for specific items
  - Notifies you when new items match your query
- Resource Watcher
  - Notifies you when properties change

4) Case Study: KDE-Telepathy

What is KDE-Telepathy?

- Instant Messaging/VoIP for KDE
- Based on Telepathy Framework
- DBus based architecture
- Modular components
- Supports collaborative features (tubes)

What information is there?

- Contacts (name, avatar, etc)
- Presence (ours and our contacts')
- File-Transfer – who sent it?
- Details of collaborators and what they did

Where is it useful?

- System wide MetaContacts
- Address Book/PIM Integration (presence)
- Show who sent us a file in Dolphin
- Who worked on this document?

Collabora

Conclusion

- If your application has interesting data on just about anything, put it in Nepomuk
- Make use of data from Nepomuk to enrich your application user experience
- The more applications that use Nepomuk, the more we all benefit from it.

Collabora

Conclusion

If you want to do stuff with Nepomuk, but you're not sure how, or you find that nothing works properly, talk to the Nepomuk developers, or even *invite them to your sprints*.

#nepomuk-kde nepomuk@kde.org

Collabora

Questions?

Collabora

Links

- **Shared-Desktop-Ontologies Documentation**

  http://oscaf.sourceforge.net

- **Nepomuk DataManagement API**

  http://api.kde.org/4.x-api/kdebase-runtime-apidocs/nepomuk/html/namespaceNepomuk.html

- **Resource Watcher API**

  http://api.kde.org/4.x-api/kdebase-runtime-apidocs/nepomuk/html/classNepomuk_1_1ResourceWatcher.html

- **Query Service API**

  http://api.kde.org/4.x-api/kdebase-runtime-apidocs/nepomuk/html/namespaceNepomuk_1_1Query.html