

# Welcome

---

## PLUG (Perth Linux User Group) seminar:

### ebXML introduction

## Perth, September 9th, 2003

@ Lecture Theatre TAFE CMC - 25 Aberdeen St, Perth

Presentation is available @ <http://www.schlegel.li/ebXML/plug-presentation/index.html>

Sacha Schlegel @ <http://www.schlegel.li>

# Presentation Overview

---

## Outline of presentation

- Short bio
- Electronic Business
  - Enterprise Resource Planning System (ERP)
  - Enterprise Application Integration (EAI)
- ebXML Introduction (incl. ebXML scenario)
- ebXML CPPA
- my ebXML project
- Discussion

# Short bio

---

My name is Sacha Schlegel born in 1971 in Switzerland, grown up in the Principality of Liechtenstein (in the heart of Europe).

After obligatory school I finished a commercial apprenticeship in a Bank in Liechtenstein followed by a Bachelor in Computer Science (CS) and Business in Switzerland.

Came to Australia in 2001 for a postgraduate diploma in CS and a masters by research in CS.

Hobbies are Soccer, Windsurfing, Linux and lately (9 months) my little daughter Tawha.

homepage: <http://www.schlegel.li>

email: [sacha@schlegel.li](mailto:sacha@schlegel.li)

# Business and Computing

---

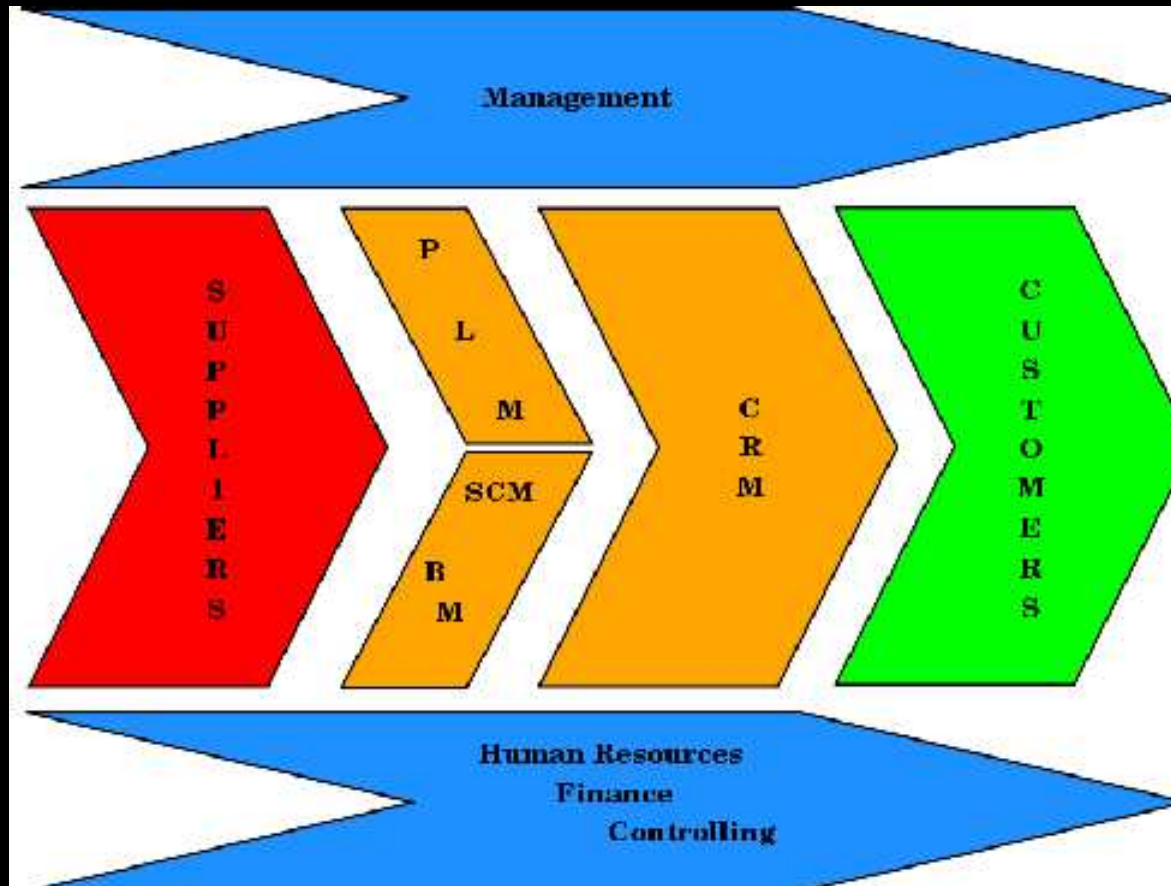
- Business is hundreds of years old (so is architecture, mathematics, physics, philosophy etc).
- Computing is still very young (first computers in the 1940's).

Business is similar among different companies.

# Business

---

Companies look similar such as:



Legend:

P - Product/Service    SCM - Supply Chain Management    L - LifeCycle    M - Maintenance  
B - Buy    M - Make    CRM - Customer Relationship Management (Marketing, Sales)

# electronic Business

---

Doing common business with the support of computing systems.

- Different software systems support different activities.
- Resulted in problem of "island" applications, individual applications distributed within the company.

A centralised databases/application was needed. e.g. Customer address should only and only be stored in one place. Otherwise address maintenance gets out of hand.

# ERP (Enterprise Resource Planning)

---

ERP systems can be seen as backbone of electronic business.  
Typical activities among most companies are:

- accounting
  - finance
  - marketing
  - production (operations)
  - human resources
  - and more.
- 
- Big commercial players are SAP, PeopleSoft/JD Edwards, Baan, Oracle
    - SAP system licence for 10 users is about US\$ 125'000
  - Smaller commercial players MyYOB, Quickbook, Quicken
  - Lots of applications for middle range
  - Free and Open Source Software ... on later slide

# Enterprise Application Integration (EAI)

---

A small sample company using Free Software / Open Source Software with the following applications:

- GnuCash (Accounting)
- Central Database (Customers, Suppliers, Product Descriptions)
- OpenOffice (Invoice templates)
- SalesStatistics (custom made)
- WebServer with online store (Sales)
- MailServer



# EAI (cont.)

---

Without EAI:

- A customer does log in with its ID (in sync with the database) on the webpage, browses the products and places an order. A person gets the message from the webserver with the details of the order. He/she has then to
  - a) do the accounting,
  - b) write and send an invoice,
  - d) update the sales statistics.
  - c) send the product,

# EAI (cont.)

---

How to integrate different applications?

How to automate these internal business process?

One possibility:

- Message Brokers

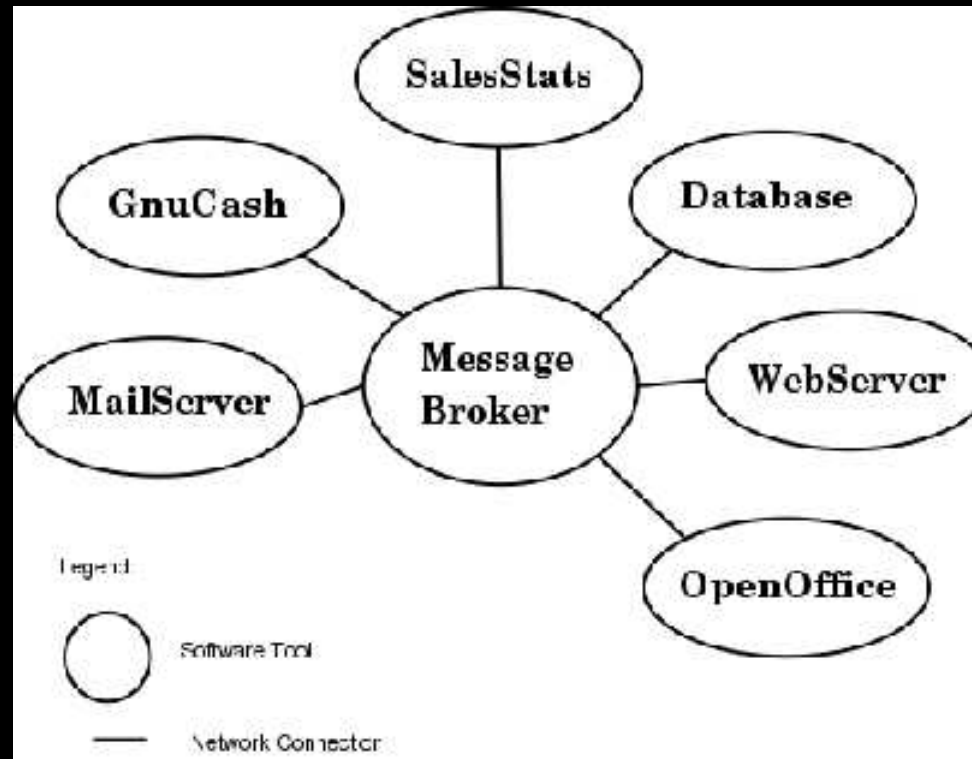
Based on the idea of publish and subscribe. An application must provide a message interface (eg incoming, outgoing messages)

Accessing an application through an API (Application Programming Interface) needs some expensive programming.

## EAI (cont.)

---

The picture:



All applications are connected to the MessageBroker, networked.

## EAI (cont.)

---

With EAI:

- Webserver (webstore) publishes "order" information.
- GnuCash subscribes to "order" message.
- SalesStatistics tool subscribes to "order" message.
- OpenOffice tool subscribes to "order" message.

Each time an order is placed by the webserver, the order is published and the subscribed applications will receive the order message in their specific format (XML, binary, exotic).

The Message Broker supports rules to compose an outgoing message from an incoming message (think XSLT).

## EAI (cont.)

---

With EAI:

- GnuCash receives the "order" message and triggers a function which will credit the "sales product x" account and debit customer y.
- SalesStatistics updates statistics.
- OpenOffice (isnt there a SDK?) generates a new invoice (address is retrieved from the database) and prints it or even better emails (including security) it to the customer.

Ooops, dont forget to bring product x to the post office.

# Next logical step forward.

---

The next logical step is to

- exchange business documents electronically.

example customer (can be company) receives the invoice by post or email and has to re-enter it into their gnucash system (data entry)

See where this leads to ... ???

Software systems communicate beyond company boundaries.

# EDI

---

In the 1960's EDI (Electronic Data Interchange) got momentum.

- ANSI (American National Standards Institute) X12
- UN (United Nations) TDI (Trade Data Interchange)

Both got merged into

- EDIFACT (Electronic Data Interchange For Administration, Commerce and Transport) as ISO Standard 9735

The "paper" based business documents got translated into electronic business documents which got exchanged over a private network.

# Pitfalls of EDI

---

A couple of reasons hindered the widespread of EDI.

- Expensive private networks
- Expensive EDI systems
- Expensive message conversion (as EDI2ProprietaryFormat)
- Cryptic messages
- EDI message setup between two companies mostly bilateral

On the other hand, companies (very big ones) who could afford the high entrance costs seemed to be happy with EDI.



# ebXML Introduction

---

ebXML is a joint initiative by the Organisation for the Advancement of Structured Information Standards (OASIS) and the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT).

- Initially a 18 months project ending in May 2001. Some groups keep working on ebXML.
- Some call ebXML the next generation of EDI.
- ebXML is based on proven technologies such as XML and HTTP

ebxml @ <http://www.ebxml.org> ;; oasis @ <http://www.oasis-open.org> ;; un/cefact @ <http://www.uncefact.org>

# ebXML Introduction (cont.)

---

## The ebXML mission

- "To provide an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties." (from ebXML website)

# ebXML Introduction (cont.)

---

- Collaborative Business Processes are very important to ebXML.

What is a business process (also a buzzword)?

- A precisely choreographed sequence of activities (actions) from a start state to an end state. Business rules control the sequence of activities. (from ebXML Professional)

What is a collaborative business process?

- Business process which happens between different partners (companies).

# ebXML and XML

---

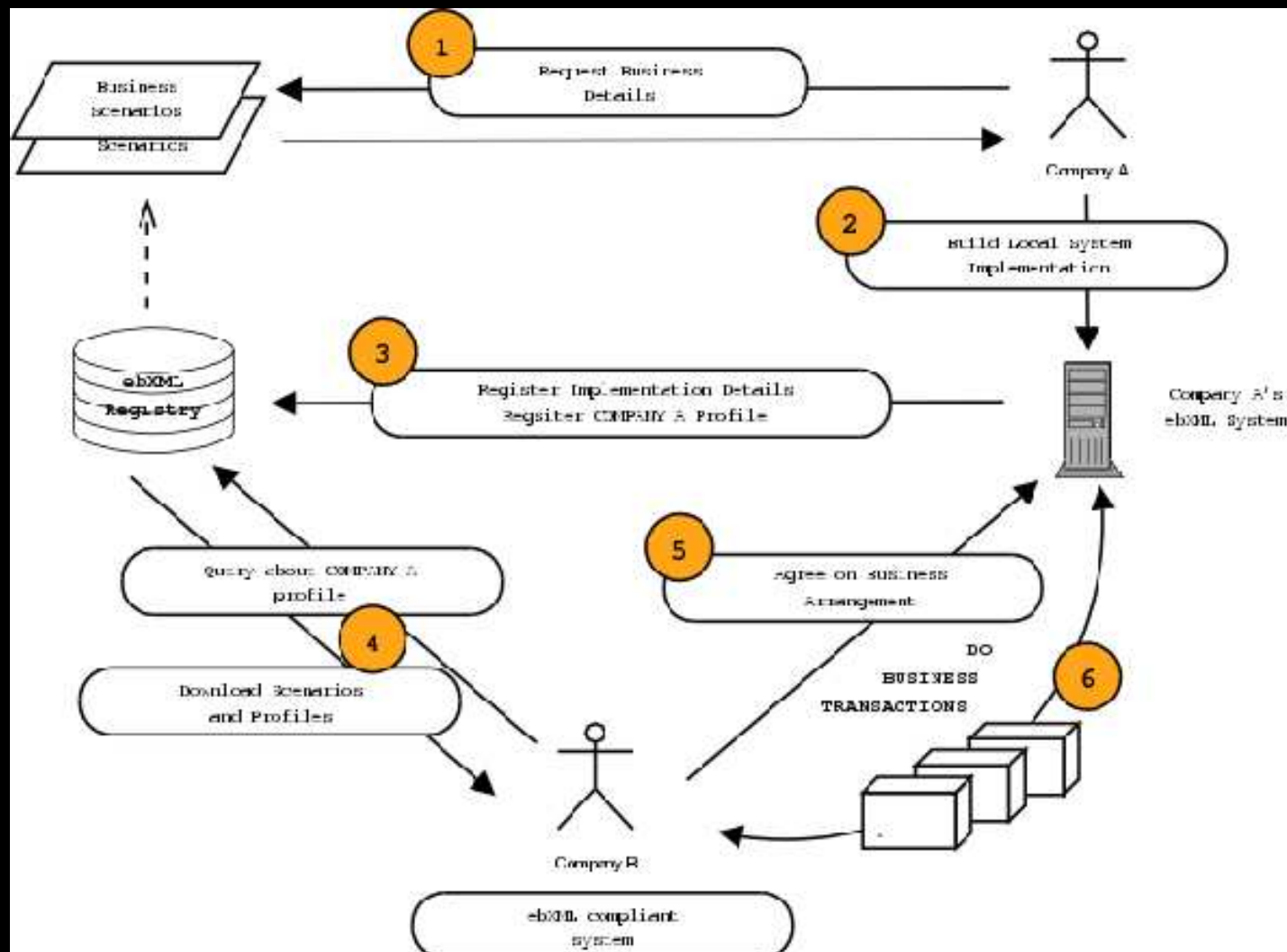
ebXML uses the Extensible Markup Language (XML) heavily because XML does structure documents very well and because XML parsers are widely available (free and commercial).

ebXML is not just an XML vocabulary for electronic business, but more like a framework for electronic business.

xml - <http://www.w3c.org/XML>

# The ebXML scenario

The sample ebXML scenario:



adapted from the ebXML Technical Architecture Specification

# The ebXML scenario (cont.)

---

comments:

- 1) Company A browses the ebXML registry to see what is available online. At best, company A can reuse all the existing business processes, documents, and core components common to its industry that are already stored in the ebXML registry. Otherwise company A designs the missing parts, stores them in the ebXML registry and makes them available for its industry partners.

# The ebXML scenario (cont.)

---

comments:

- 2) Company A decides to do electronic business the ebXML way and considers implementing a local ebXML compliant application. An ebXML Business Service Interface (BSI) provides the link between the company and the outside ebXML world. The company has to create a Collaboration Protocol Profile (CPP) which describes the supported business process capabilities, constraints and technical ebXML information such as choice of encryption algorithms, encryption certificates and choice of transport protocols.

# The ebXML scenario (cont.)

---

comments:

- 3) Company A submits its CPP to the ebXML registry. From that point on, company A is publicly listed in the ebXML registry and is likely to be discovered by other companies looking/querying for new trading partners.



# The ebXML scenario (cont.)

---

comments:

- 4) Company B is already registered at the ebXML registry and is looking for new trading partners. Company B queries the ebXML registry and receives the CPP of company A. Company B then has two CPP's: Company A's CPP and its own. The two companies have to come to an agreement on how to do business. This is called a Collaboration Protocol Agreement (CPA) in the ebXML terminology. Company B uses an ebXML CPA formation tool to derive a CPA from the requirements of the two CPP's.

# The ebXML scenario (cont.)

---

comments:

- 5) In this scenario company B communicates with company A directly and sends the newly created CPA for acceptance to company A. Upon agreement of the CPA by company A both companies are ready for electronic business.
- 6) The companies then use the underlying ebXML framework and exchange business documents conforming to the CPA. This means that both companies follow the business processes defined in the CPA.

# The ebXML specifications

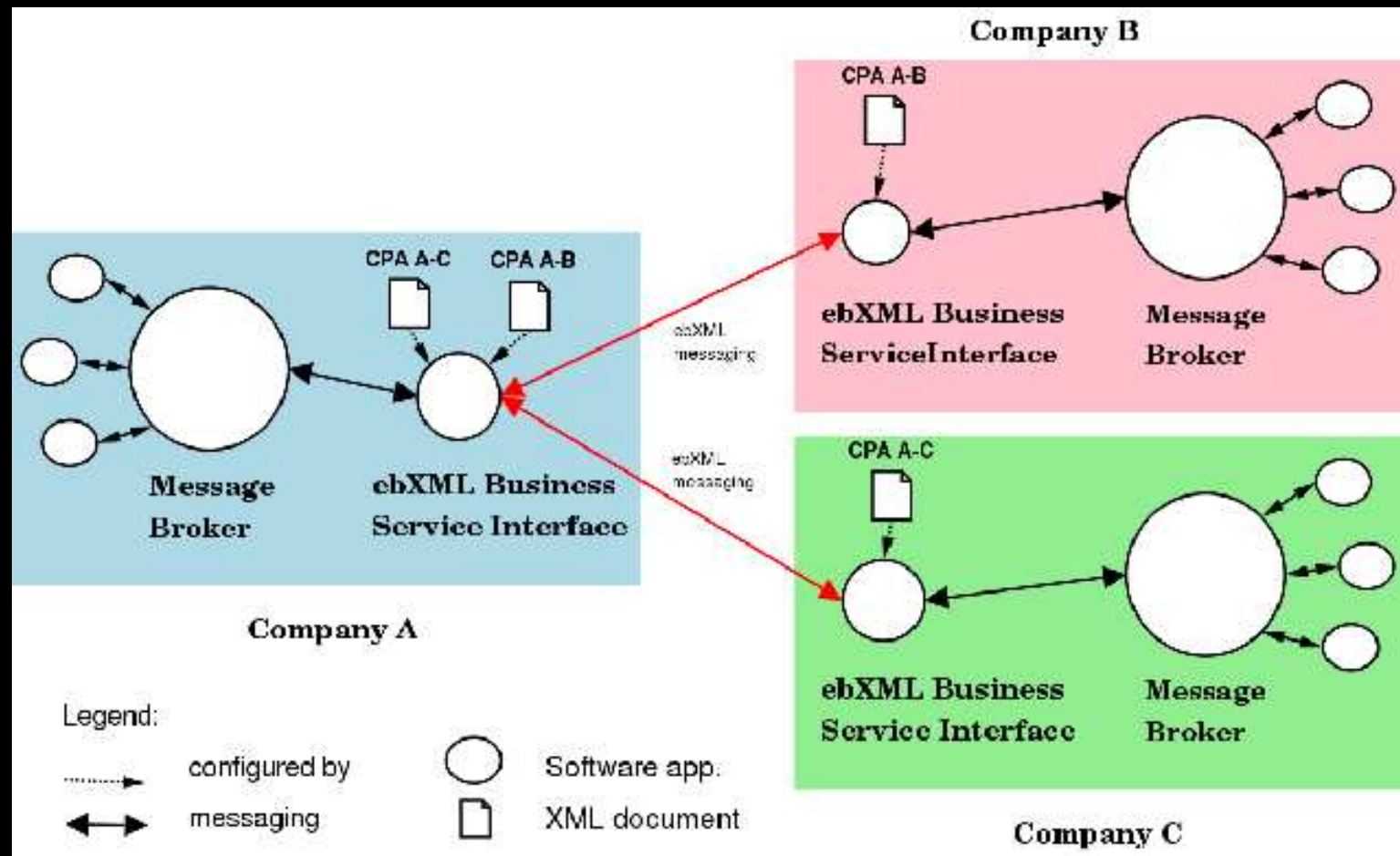
---

The outcome of the ebXML effort were several specifications such as:

- ebXML Technical Architecture Specification (v1.04)
- Registry Service Specification (v2.0)
- Business Process Specification Schema (v1.01)
- Collaboration Protocol Profile and Agreement Specification (v2.0)
  - Automated Negotiation of a Collaboration Protocol Agreement Specification (v0.04)
- Message Service Specification (v2.0)
- more plus technical reports

# ebXML in in action

ebXML system to many ebXML systems.



# ebXML CPPA

---

## What is inside a Collaboration Protocol Profile (CPP)?

```
<CollaborationProtocolProfile
  <!-- Namespaces -->
  version="1.1">
<PartyInfo> <!--one or more-->
  ...
</PartyInfo>
<Packaging id="ID"> <!--one or more-->
  ...
<Packaging>
<ds:Signature> <!--zero or one-->
  ...
</ds:Signature>
<Comment>text</Comment> <!--zero or more-->
</CollaborationProtocolProfile>
```

# ebXML CPPA

---

What is inside a Collaboration Protocol Profile (CPP)?

Content of a CPP is

- PartyInfo (one or more)

- CollaborationRole

- ▶ ProcessSpecification (reference to Collaborative Business Process)
    - ▶ Role (eg buyer role, seller role)
    - ▶ ServiceBindings (CanSend, CanReceive of each message)
    - ▶ Certificates (for encryption, signature etc)
    - ▶ DeliveryChannel (bundling of Document Exchange and Transport)
    - ▶ Document Exchange (properties of messaging such as reliability, non-repudiation)
    - ▶ Transport (eg HTTP, FTP, SMTP plus characteristics such as security)

- Packaging (Message, eg header, body, attachement structures)

# ebXML CPPA

---

## What is inside a Collaboration Protocol Agreement (CPA)?

```
<CollaborationProtocolAgreement
  <!-- Namespaces -->
  cpaid="YoursAndMyCPA"
  version="1.2">
  <Status value = "proposed"/>
  <Start>1988-04-07T18:39:09</Start>
  <End>1990-04-07T18:40:00</End>
  <!--ConversationConstraints MAY appear 0 or 1 times-->
  <ConversationConstraints invocationLimit = "100"
    concurrentConversations = "4"/>
  <PartyInfo>... </PartyInfo>
  <PartyInfo>... </PartyInfo>
  <Packaging id="N20"> <!--one or more--></Packaging>
  <!--ds:signature MAY appear 0 or more times-->
  <ds:Signature>any combination of text and elements
  </ds:Signature>
  <Comment xml:lang="en-gb">any text</Comment> <!--zero or more-->
</CollaborationProtocolAgreement>
```

# ebXML CPPA

---

What is inside a Collaboration Protocol Agreement (CPA)?

Content of a CPA is

- PartyInfo (exactly two ; with some add-ons)
- Packaging



# My ebXML project

---

Title: "CPA formation process"

- The "CPA Formation Process" is the process to get a CPA from two CPP's.
- The "CPA formation Process" can be divided into
  - CPA composition
  - CPA negotiation

# My ebXML project (cont)

---

## CPA Composition

- based on the CPP and CPA Specification Version 2.0
- input: 2 CPP, 2 NDD's
- output: CPA template(s), new NDD(s)
- algorithm does pair building
  - A.send\_message\_x\_this\_way
  - B.receive\_message\_x\_this\_way
- matches/checks
  - can result in conflicts of type
    - equality (eg Transport Protocol)
    - non-equality (eg Roles)
    - cardinality (optional elements/attributes)
    - references: have to be followed (eg URI)

# My ebXML project (cont)

---

CPA Composition Demo

Demo

Web-based demo at [http://www.schlegel.li/ebXML/Msc\\_live.rhtml](http://www.schlegel.li/ebXML/Msc_live.rhtml)

# My ebXML project (cont)

---

## CPA Negotiation

- based on Automated Negotiation of CPA Version 0.04
- input: CPA template, NDD
- output: final CPA or updated CPA template

A negotiation system must support

- follow the Negotiation CPA,
- follow the Negotiation Business Process (state machine)
- follow the Negotiation Messages (business documents)
- uses a Messaging System

# My ebXML project (cont)

---

## CPA Negotiation (cont.)

A negotiation can be

- computer to computer (using AI for learning, strategy)
- computer to human
- human to human (any form: web, telephone)

# My ebXML project (cont)

---

## CPA Negotiation Demo

Demo of a human to human CPA negotiation following the Negotiation CPA

no demo at <http://www.schlegel.li/live.rhtml> available yet.

# ebXML Implementations (or parts of it)

---

Commercial ebXML implementations (see "tool" link on ebXML website):

- XMLGlobale, SterlingCommerce, SAP, SUN, ORACLE, CommerceOne, IONA, BEA, Sysbase, Tibco, more (see

<http://www.ebxml.org/tools/index.htm> )

Open Source ebXML implementations:

- <http://www.freeebxml.org>
- <http://ebxmlrr.sourceforge.net/> (ebXML Messaging System)
- <http://www.openebxml.org>
- more: [sourceforge.net](http://sourceforge.net) has some ebXML projects in planning and alpha status.

# Free Software / Open Source business applications

---

Business applications useful today:

- gnucash @ <http://www.gnucash.org> (personal and business finance)
- sql-ledger accounting @ <http://www.sql-ledger.org/> (finance)
- gnuenterprise @ <http://www.gnue.org> (erp and tools)
- arias @ <http://www.aria-erp.org> (ex nola)
- gnuledger @ [webaccountant.sourceforge.net/](http://webaccountant.sourceforge.net/)
- pago @ <http://papo.vialibre.org.ar/>



# Free Software / Open Source business apps

---

Other projects (java based):

- [www.jboss.org](http://www.jboss.org) - application server (transaction, persistence, load-balancing, messaging etc)
- [www.jbpm.org](http://www.jbpm.org) - business process management to manage bp and workflows
- [www.mec-eagle.org/](http://www.mec-eagle.org/) - b2b integration tool
- [www.xmlblaster.org](http://www.xmlblaster.org) - Message Oriented Middleware

# Free Software / Open Source business apps

---

Free Software, Open Source for business applications and ebXML

- Affordable for Small to Medium Organisations
- Affordable Foundations (OS's, Databases, etc)
- Open Standards based
- No vendor lock in (current problem in e-markets)
- More transparency, security
- Collaborative nature of ebXML
- it just rocks

# The end.

---

- Thanks

- Questions

- Discussion

# Finish

---

GO BACK.