

# E-Business Coordination, Collaboration, and Integration

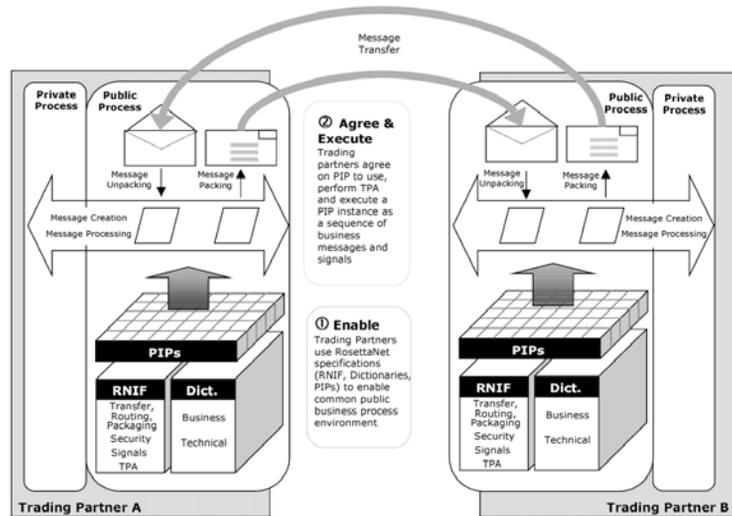
## Workflows and Web Services Kapitel 11

Workflows und Web Services  
WS 2003/2004

## RosettaNet

- Goal: Develop standard e-commerce interfaces to align the processes between IT supply chain partners
  - consortium founded in 1998
  - "vertical" coordination protocols
- Main standardization areas
  - Business processes
    - coordination protocols for trading partners
    - Partner Interface Processes (PIPs)
      - business documents, vocabulary, choreography of message exchanges
  - Data format
    - establishment of a common vocabulary
      - business directory
      - technical dictionary
  - Message services
    - RosettaNet Implementation Framework
      - reliable, secure execution of the protocol specifications
      - transfer, routing, packaging of encrypted and authenticated messages between business partners

# RosettaNet Trading Partner Implementation



## PIP Definitions

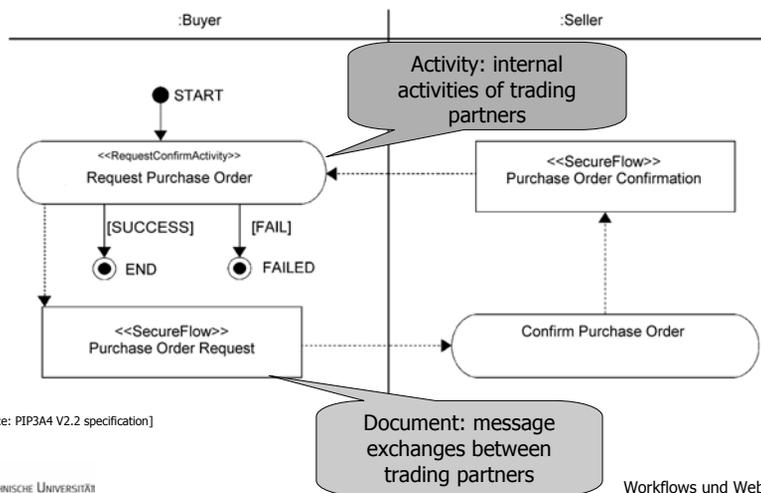
- Standardized PIP definitions are arranged into clusters, further broken down into segments
- Clusters:
  1. RosettaNet Support
    - administrative functionality
  2. Partner Product and Service Review
    - collect, maintain, distribute product or service information
      - account setup, product info subscription, ...
  3. Product Information
    - distribute, update product information
      - query technical product info, ...
  4. Order Management
    - request quote, request purchase order, query order status, ...
  5. Inventory Management
    - distribute inventory report, ...
  6. Marketing Information Management
    - exchange of marketing information
  7. Service and Support
    - request warranty claim, ...
  8. Manufacturing
    - "virtual manufacturing"
      - notify of manufacturing work order, ...

## Partner Interface Process (PIP) Specifications

- Describes how to implement a collaborative coordination protocol
  - technical dictionary describes components that are exchanged
  - message guideline document
    - business actions, business signals (ack receipt of action message)
- Major PIP specification sections
  - Business Operational View (aka Action Layer)
    - flow of business interactions, based on
      - partner roles
      - partner role interactions
  - Functional Service View (aka Transaction Layer)
    - derived from the business operational view
    - business transactions between entities in the form of message exchanges
      - coordination protocols
      - message control information
        - time limits for acknowledgements
        - security requirements
  - Implementation Framework View (aka Service Layer)
    - based on functional service and business operational views
    - defines communication protocol and message format requirements
      - e.g., SSL, encryption, XML DTDs for messages, ...

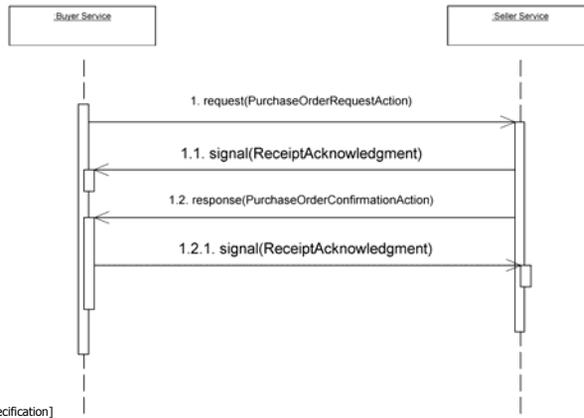
## Business Operational View - Example

- Business Process Diagram for PIP3A4: Request Purchase Order



## Functional Service View – Example

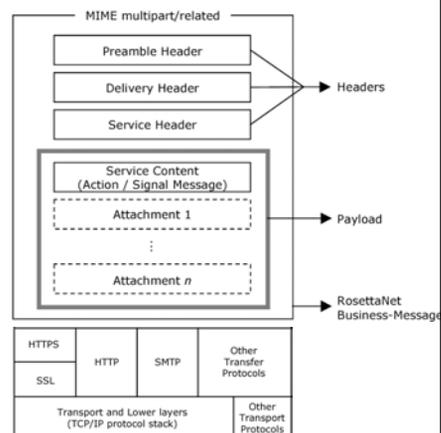
- Business Transaction Dialog Specification for PIP3A4: Request Purchase Order



[source: PIP3A4 V2.2 specification]

## RosettaNet Implementation Framework

- Defines
  - Business Message
    - packaging payload (incl. attachments), headers, ...
    - uses MIME, S/MIME
  - Protocol Stack
    - transport-independent
    - reliable messaging
      - support for HTTP, SMTP, ...
  - Security Mechanism
    - based on encryption, digital signatures
    - supports authentication, authorization, encryption, non-repudiation
- Designed before the time of SOAP
  - May likely be replaced by SOAP-based web service infrastructure in the future



[source: RosettaNet Implementation Framework Core Specification]

## BPEL – Abstract Processes

---

- Abstract Process = Role-specific view of a protocol
  - only public information
  - no private, implementation-specific aspects
    - branching conditions, activity realization, ...
  - not executable
  - can be used by a conversation controller to ensure protocol compliance (see chapter 5)
- Properties of BPEL abstract processes
  - handle only protocol-relevant data
    - message properties
  - variables
    - do not need to be fully initialized
    - variables for inbound or outbound messages may be omitted from invoke, receive, reply, if the intent is to just constrain the sequence of activities
  - opaque assignments
    - can correspond to creating a unique value for correlation properties
    - hide private behavior for providing the values

## Implementing Business Protocols

---

- Suggested path
  - protocol specification as a starting point
  - derive role-specific views of the protocol
    - includes all the message exchanges that involve a certain role
  - define abstract process for role-specific view
    - model interactions using receive, invoke, reply
    - represent additional public information, such as branching situations, parallelism
  - turn abstract process into an executable process to implement it

# Implementing RosettaNet PIPs

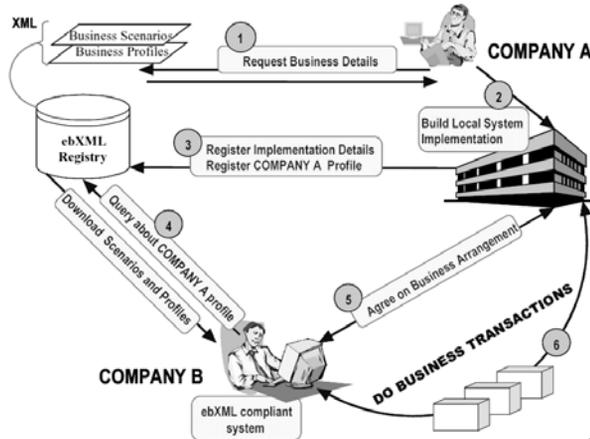
- Involves mapping PIP to WSDL, BPEL
  - types in message definitions -> types in WSDL
    - DTDs to XML Schema
  - message definitions -> WSDL message definitions
  - PIP actions -> operations in WSDL
  - PIP partner roles -> BPEL partners
  - PIP choreography: follow the "suggested path" on previous chart
- Additional aspects
  - realize time-outs, etc. using BPEL events and fault handlers
  - additional requirements regarding security need to be resolved
    - WS-Security support, not integrated in BPEL

# ebXML

- Supported by UN/CEFACT, OASIS
- Vision
  - single global electronic marketplace
  - based on exchange of XML messages
- ebXML architecture covers:
  - definition of business processes and their associated messages and content
  - registry and discovery of business process sequences with related message exchanges
  - definition of company profiles
  - definition of trading partner agreements
  - uniform message transport layer
- ebXML advantages
  - goes beyond generic protocols and specifications
    - e.g., ebXML registry is much more detailed than UDDI
  - captures the logic behind e-commerce exchanges
    - e.g., business arrangements
  - specifies how e-commerce exchanges should be specified, documented conducted

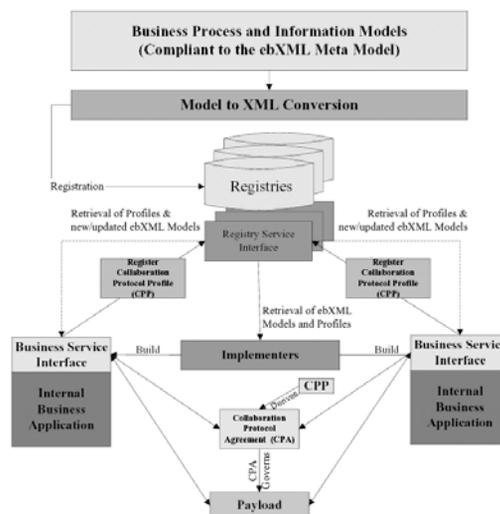
# Collaboration with ebXML

- Example



(source: ebXML Technical Architecture Specification)

# Technical Architecture



(source: ebXML Technical Architecture Specification)

# How Do These Standards Relate?

