Workflow Management Coalition

Interface 1: Process Definition Interchange

Organisational Model

Document Number WfMC TC-1016-O

Document Status - Draft 6.99

(limited distribution, under revision)

Issued on July, 15, 1998

Author:

Work Group 1

Send comments to: Workflow Management Coalition
WfMC@wfmc.org or
kdkreplin@empirica.de "Klaus-Dieter Kreplin"

The Workflow Management Coalition - Confidential

Copyright © 1994-1998 The Workflow Management Coalition

All rights reserved. Material from this publication may be reproduced by electronic, mechanical, photographic or other means for non-commercial purposes, providing acknowledgement is made to the Workflow Management Coalition as the original source.

This document is intended as an input to the Organisation Model subgroup of WG1. It ducuments the status of work performed so far in the context of WG 1.

1. INTRODUCTION <UNDER REVISION> **3**

1.1. Purpose **3**

1.2. Audience **3**

1.3. Overview <under revision> **3**

1.4. Conformance **3**

1.5. References **4**

2. OVERVIEW OF PROCESS DEFINITION INTERCHANGE <UNDER REVISION> **5**

3. META-MODEL AND INFORMAL DESCRIPTION **6**

3.1. Overview <under revision> **6**
3.1.1. Entities Overview 6
3.1.2. Attributes Overview <under revision> 7
3.1.3. Common Attributes 7
3.1.3.1. Extended Attributes 7
3.1.3.2. Formal Parameters 8
3.1.4. Vendor-specific Extensions 9

3.2. Organisation Model (Workflow Participant Definition) **10**
3.2.1. Overview 10
3.2.2. Meta-Model 10
3.2.3. Attributes 11
3.2.4. Participant Entity Types 12
3.2.4.1. Organisational Units 14
3.2.4.2. Human 14
3.2.4.3. Role 15
3.2.4.4. Resource 16
3.2.5. Built in OM Library Functions and Procedures 16
3.2.6. Minimal Organisation Model Declaration 17
3.2.7. Extended Library 17
3.2.7.1. Attributes 17

4. PROPOSED WPDL GRAMMAR **19**

4.1. WPDL at a glance **19**

4.2. WPDL Grammar and Language Constructs **20**
4.2.1. WPDL Description Method 20
4.2.1.1. Metalanguage 20
4.2.1.2. Special Symbol Conventions for Tokens 20

4.3. WPDL **22**
4.3.1. Workflow Participants 22

5. ANNEX **25**

5.1. Annex 1: Proposed Extensions and Open Issues 25
5.1.1. Organisational modeling (ch. 4.3.1) 25
5.1.1.1. Organisation Modelling separation 25
5.1.1.2. Performer-Participant Relationship Requirements 25

5.2. Annex 2: Analysis of PDLs 26

5.3. Annex 3: Representative Business Example 28
5.3.1. Informal Description 28
5.3.1.1. Company Organisation 28
5.3.2. Representative Business Example in WPDL Notation <*to be aligned*> 31
5.3.2.1. Workflow Participants 31

5.4. Annex 4: Organisation Modelling Impact on WfMC Specifications 33
5.4.1. Goal 33
5.4.2. Organisation modelling 33
5.4.3. Relationships with workflow features 34
5.4.3.1. Organisation administration 35
5.4.3.2. Process definition 35
5.4.3.3. Application invocation 37
5.4.3.4. Inter process communication 37
5.4.4. Relationships with WfMC interfaces 37
5.4.4.1. Workflow Process Definition 37
5.4.4.2. Workflow inter operation 37
5.4.4.3. Application call 38
5.4.4.4. Client Interface 38
5.4.5. Administration 38
5.4.6. External standards to be considered 38
5.4.7. Recommendations 38

5.5. Annex 5 Document History 39

6. INDEX 40

6.1. Index of Figures 40

6.2. Index of Tables 40

1. Introduction <under revision>

This is one part of the WfMC Process Definition Interchange. It is complemented by the document: “Interface 1: Process Definition Interchange. Process Model. Document Number WfMC TC-1016p”.

The Organisational Model described in this document can be used as an integral part of a Process Definition, or it can be used as a stand-alone definition of an Organisational Model, which may be used by a Process Model via an External OM Definition.

1.1. Purpose

The purpose of this document is to describe a common interface for the exchange of workflow process definitions. This interface is based on a standardised language - the Workflow Process Definition Language, "WPDL" - which can be supported by vendors of workflow management products to allow the exchange and documentation of workflow process definitions.

This paper describes the syntax and content of information exchanged across the interface.

1.2. Audience

The intended audience of this document includes all participants in the workflow industry. Comments should be addressed to the Workflow Management Coalition.

1.3. Overview <under revision>

1.4. Conformance

A vendor can not claim conformance to this or any other WfMC specification unless specifically authorised to make that claim by the WfMC. WfMC grants this permission only upon the verification of the particular vendor’s implementation of the published specification, according to applicable test procedures defined by WfMC.

A conforming implementation of this Functional Area of the Workflow Management Coalition specification includes the implementation of the relevant portions of the other functional areas: Client Application, Tool Invocation, Interoperability, Administrating and Monitoring.

But: Given the wide variation of capabilities in modelling tools, it is reasonable to assume that an individual tool might conform to the Interface 1 specification but not be able to swap complete definitions with all other conforming products. There is a two-level view of conformance:

1. Syntax, where on output the tool must generate valid, syntactically correct WPDL, on input, the tool must be able to read all valid WPDL. In this case, the translator should flag those expressions not applicable, and create appropriate descriptions so the modeller on the import side understands the nature and intent of the untranslated expressions.

2. Structure, where there is a mandatory set of objects and attributes. The suggestion is, to define a minimum set of objects and attributes needed to create a functioning model.

1.5. References

The following documents are associated with this document and should be used as a reference.

- WfMC Glossary
- WfMC Reference Model
- WfMC WPDL - Process Model

2. Overview of Process Definition Interchange <under revision>

3. Meta-Model and Informal Description

3.1. Overview <under revision>

3.1.1. Entities Overview

The meta-model identifies a basic set of entities and attributes for the exchange of process definitions.

- **Workflow Process Definition**
  describes the process itself, i.e. ID and further descriptions of a process, etc.. A workflow type definition reflects the header information of a process definition and is therefore related to all other entities in that process..

- **Workflow Process Activity**
  A process definition consists of many logical steps resp. descriptions of pieces of work: so called workflow process activities. Each of them is defined through 4 dimensions, the who, the what, the when and the how. First: An Activity is assigned to workflow participants, who are permitted to play the role in this activity (resp. are allowed to perform the activity). Second: The activity is assigned to an application, which will be invoked during runtime. Decision data and reference values to be passed along refer to process relevant data. Third: Activities are related to one another via flow control conditions (transition conditions; pre- and post-conditions). Flow control conditions are usually based on decision data. Activities are atomic in that sense that they are the smallest processable unit in a workflow. An Activity may be implemented as ***atomic*** (a logical working unit, the workflow participant handles during an activity) or as a ***sub process*** (a couple of logical working units).

- **Workflow Participant Declaration**
  allows to describe the performer of an activity in terms of a reference to elements of an organisational model. The declaration of such a participant does not necessarily refer to a single person, but also to a function or any other organisational entity.

- **Transition Information**
  describes the navigation between different process activities, which may involve sequential or parallel operations. Thus activities are connected to each other by transition information. A couple of transitions may span a subflow resp. a sub process.

- **Workflow Application Declaration**
  usually one to n applications are assigned to an activity within a process definition. These applications will be invoked during run time by the workflow management system. The workflow application definition reflects the interface between the workflow engine and the application.

- **Workflow Relevant Data**
  data, that is to be made available to a subsequent activity resp. related application and/or transition information and thus may affect the choice of the next activity to be executed.

Activities, invoked applications and/or transition conditions may refer to workflow process relevant data.

These entities contain attributes which support a common description mechanism for processes. They span the **Minimal Process Model**.

On top of the Minimal Process Model another entity is identified.

- **Workflow Model**
  allows to combine multiple Process Models and contains further descriptions of the included process like WPDL version, vendor ID etc.. The workflow model definition optionally allows to define entities that may be used by several Process Models and may contain references to an external organisational model or to other (external) Workflow Models, providing some sort of inheritance mechanism.

The Workflow Model also allows to define an

- **Organisation Model (Workflow Participant Definition)**
  that allows to define the hierarchical structure of an organisation in terms of ist Participants and their relationships. The Participants not only refer to single persons, but also to (business) functions or any other organisational entities. WPDL allows to define a basic set of organisation entities and references.

3.1.2. Attributes Overview <under revision>

3.1.3. Common Attributes

3.1.3.1.Extended Attributes

Informal Description

Extended Attributes can be used in all entities and in Library Functions and Procedures and External declarations.

Attributes

Attribute Name	M/O	WPDL Keyword	Data Type	Description
Identifier	M	EXTENDED_ATTRIBUTE	IDENTIFIER	Used to identify the Extended Attribute
Attribute type	M		IDENTIFIER	Datatype, valid types are: *simple and complex data types*

Attribute Name	M/O	WPDL Keyword	Data Type	Description
Attribute value	M		*(a value of appropriate type)*	Preassignment of data for run time, *an initial or a function access of appropriate type*
Attribute description	O		STRING	Textual description of the attribute

Table 3-1: Extended Attributes

3.1.3.2.Formal Parameters

Informal Description

Formal parameters can be used as attributes in Workflow Process and Workflow Application entities and in Library Functions and Procedures. These are the invocation parameters.

Attributes

Attribute Name	M/O	WPDL Keyword	Data Type	Description
Parameters				Formal Parameters that are passed during invocation and return of control (e.g. of an invoked application).
	O	IN_PARAMETERS	List of IDENTIFIER	The Input Parameters, e.g. for the invoked application
	O	OUT_PARAMETERS	List of IDENTIFIER	The Output Parameters, e.g. of the invoked application

Table 3-2: *Formal Parameters*

Parameter passing semantics

The **parameter passing semantics** is defined as:

(a) Any read-only formal parameters (parameter in the list of IN_PARAMETERS but not in the list of OUT_PARAMETERS) are initialised by the value of the corresponding actual parameter in the call (an expression). This is pass-by-value semantics.

(b) Any read/write formal parameter (same parameter in the list of IN_PARAMETERS as well as in the list of OUT_PARAMETERS) are initialised by the value of the corresponding actual parameter which must be the identifier of a workflow relevant data entity. On completion the value of the formal is copied back to the original actual parameter. This is copy-restore semantics.

(c) Any write-only formal parameters (parameter in the list of OUT_PARAMETERS but not in in the list of IN_PARAMETERS) are initialised to zero (strings will be set to the empty string, complex data will have each element set to zero). On completion, the value of the formal is copied back to the original actual parameter which must be the identifier of a workflow relevant data entity. This is zero-restore semantics

Concurrency semantics

Copying and restoring of parameters are treated as atomic operations. I.e. to avoid access conflicts due to concurrency these operations are serialised. Between copy and restore of (c) no locking is assumed.

3.1.4. Vendor-specific Extensions

The WPDL provides means to augment the defined WPDL by missing features, thus allowing to customise the language.

The primary means do customise the language are the Extended Attributes described in the previous chapter.

Besides the Extended Attributes there are some other parts of the WPDL that might need a vendor customisation. For these "generic" parts the WPDL either provides a generic symbol, beginning with "extended", together with some generic productions, or a hint is made that further description has to be provided.

Extended parameter mapping

Also parameter mapping may need an extension by a vendor. The mechanism how this is provided is not presented here.

Extended Library

Vendor/user provided functions and procedures may be added using an extended library declaration.

3.2. Organisation Model (Workflow Participant Definition)

3.2.1. Overview

This is the static definition of the Workflow-relevant part of an Organisation Model. The Organisation Model is defined as a list of Workflow Participants and the relationships between them. The Workflow Participant is defined by a type and the related information, which is a set of type specific attributes. This definition contains a basic set of Workflow Participant types: an organisation unit, a human, a role, and a resource. A role and a resource are used in the sense of abstract actors. During run time these abstract definitions are evaluated and assigned to concrete human(s) and program(s).

In addition to this structural definition built-in Library Functions and Procedures are provided that may be used in a Workflow Process Definition. Further specific Extended Library Functions may be defined by a vendor or OM definer.

The interface to the Organisation Model is used in the Activity Definition (describing the performer of an activity) and in the Process Definition (describing the responsible of a process).

3.2.2. Meta-Model

The meta-model for an Organisational Model identifies the basic types of Workflow Participants and their relationships.

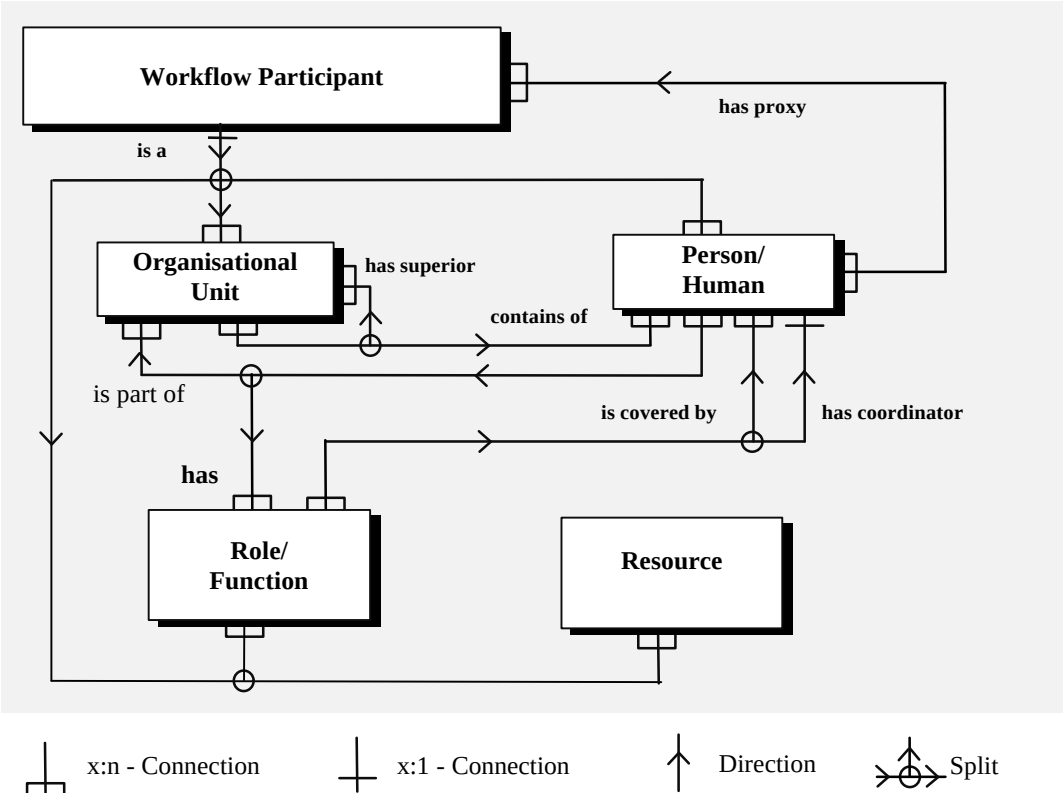

Figure 3-2: Organisation Model Definition Meta Model

3.2.3. Attributes

The attributes of a Workflow Participant characterise the Participant Type and allow to specify simulation-relevant data.

Attribute Name	M/O	WPDL Keyword	Data Type	Description
Participant Identifier	M	PARTICIPANT	INTEGER	Used to identify the workflow participant definition.
Participant Name	O	NAME	STRING	Text used to identify a performer
Participant Description	O	DESCRIPTION	STRING	Short textual description of a workflow participant..
Participant Type Description				Characterisation of the type of a workflow participant entity.
*	M	TYPE	*keyword*	Type of a workflow participant.
*	O	*(see below)*	*(see below)*	Participant type related information.

Attribute Name	M/O	WPDL Keyword	Data Type	Description
Simulation Data	O	CAPACITY	INTEGER	Number of parallel activities the participant is able to manage. Default: 1
	O	COST	STRING	Cost *(no currency ?)*
	O	PREPARETIME	INTEGER	Time to set up the workflow participant. Default: 0
*	O	STRATEGY	*keyword*	Strategy how the participant handles intray requests. Default: 0

Table 3-3: Attributes of Entity Workflow Participant

Scope

The scope of the identifier of an Workflow Participant entity is the surrounding entity in which it is defined. For a regular OM definition (see below chapter 3.2.6) this is the Process Model Definition.

Simulation Attribute Strategy

For simulation purposes a set of keywords is defined that map to predefined strategies.

WPDL Keyword	Value Keyword	Description
STRATEGY	FIFO	First in, first out. Default
	LIFO	Last in, first out
	SJF	Shortest job first
	LJF	Longest job first
	RD	Random
	UD	Undefined

Table 3-4: Entity Workflow Participant: Strategy Attribute

3.2.4. Participant Entity Types

The Participant entity type attribute characterises the participant to be an individual, an organisation unit or an abstract resource such as a machine.

WPDL keyword
ORGANISATIONAL_UNIT
HUMAN
ROLE

RESOURCE

Table 3-5: *Types of Workflow Participants*

Besides the keywords for the type the type description allows to define optionally type-specific extended attributes, and Type associated lists. The latter allow to specify the context of an organisational entity (type) such as human, role, organisational unit and so forth:

members of an organisational unit
humans with identical roles
direct superior organisational units
memberships in organisational units
roles a human has
proxies a human has.

Table 3-6: *Description Lists associated with Types of Workflow Participants*

The associated lists may have additional type-dependent attributes that also allow to define the relationship of this Participant entity to other Participant entities. This is depicted in Figure 3-3.

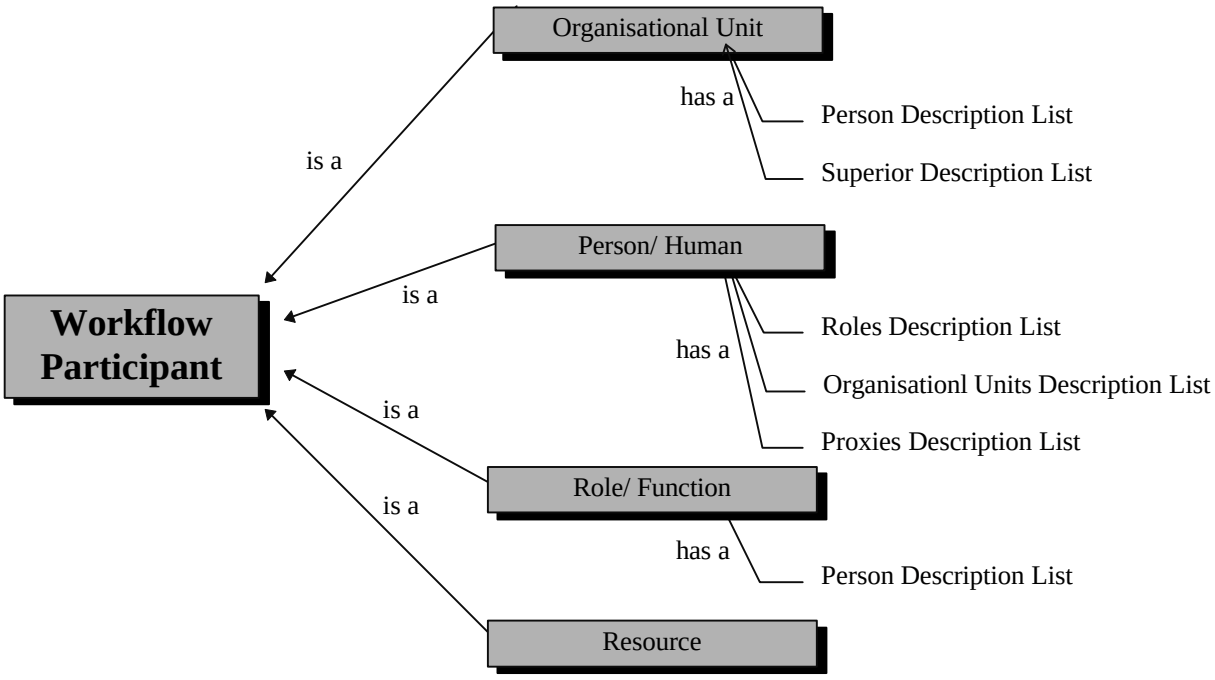

Figure 3-3: Types of Workflow Participant Assignment

3.2.4.1.Organisational Units

The manager (representing the organisational unit) or all members of an organisational unit get the work item if an organisational unit is addressed. The superior description lists describes the bottom up hierarchy and contains a hierarchical ordered list of superior organisational units. The person description list contains a list of all members of an organisational unit.

Attributes for ***Organizational Unit:***

Attribute Name	**Description**
Function	Description of a task or competence of the Organisational Unit
Manager	Manager of the Organisational Unit
Superiors	List of direct Superiors of the Organisational Unit, usually other Organisation Units
PersonList	List of human resources that belong to the Organisational Unit

Table 3-7: *Attributes of Organisational Unit*

<table to be aligned>

3.2.4.2.Human

In most workflow processes a human is addressed indirectly by his role, by his organisational unit and so forth. The roles description list contains all roles of that human, the organisational units description list the list of all units, where he belongs to, and the proxies description list the list of all proxies of this human.

Attributes for ***Human:***

Attribute Name	Description
UserID	User Id
SurName	Participants last name
ForeName	Participants first name
PhoneNumber	<??>
FaxNumber	<??>
Email	<??>
X400	X.400 identification
OrgUnits	List of Organization Units the participant belongs to
Alternates	List of participants that may substitute the participant
Roles	List of Roles that are played by the participant

Table 3-8: *Attributes of Human*

<table to be aligned>

3.2.4.3.Role

This type allows to address a performer by a role. A role in this context is a function a human has within an organisation. Because a function isn't necessarily unique, a coordinator (for administrative purposes or in case of exception handling) may be defined and a list of humans the role is related to. Attributes which qualify the role have to be defined within extended attributes (e.g. loan approval dependent on the amount, manager of team).

Attributes for ***Role:***

Attribute Name	Description
Function	Description of a task or competence of the role
Coordinator	Coordinator of the Role, usually a human participant
PersonList	List of human resources that belong to the Role

Table 3-9: *Attributes of Role*

<table to be aligned>

3.2.4.4.Resource

This type allows to address a performer by a resource. A resource in this context is either a program or a machine. A program may be imagined as an agent, a machine as an automatic scanner.

There are ***no predefined attributes*** for Resource.

3.2.5. Built in OM Library Functions and Procedures

The following Library Functions and Procedures are supported by the OM. They allow to evaluate OM relationships.

Functions

function	function id	M/O	result type	NoOf Pars.	Parameter type	description
manager of	MANAGER_OF	O	PARTICIPANT	1	PARTICIPANT	*<to be provided>*
coordinator of	COORDINATOR_OF	O	PARTICIPANT	1	PARTICIPANT	*<to be provided>*

Procedures

procedure	procedure id	M/O	NoOf Pars.	Parameter type	Parameter class	Description
alternate of ...	ALTERNATE_OF	O	2			*<to be provided>*
				PARTICIPANT	IN	*<to be provided>*
				list of PARTICIPANT	OUT	*<to be provided>*
superior of ...	SUPERIOR_OF	O	2			*<to be provided>*
				PARTICIPANT	IN	*<to be provided>*
				list of PARTICIPANT	OUT	*<to be provided>*

Process History Participants

The Process History is not part of the Organisation Model. Therefore the reference to previous Performers in the history of a process is handled by the Workflow engine (e.g. evaluating the Audit Data). Therefore Library Functions providing access to historical performers are no OM functions but Workflow functions (chapter **Error! Reference source not found.**). However, some vendors may implement these functions by OM functions. The specification of this possibility is outside the scope of this document.

3.2.6. Minimal Organisation Model Declaration

We distinguish a regular Organisation Model definition (called Workflow Participant Definition) described in the previous paragraphs and a Minimal Organisation Model declaration (called Workflow Participant Declaration) that declares a list of participant identifiers with an optional type characterisation., while a further description of the type and the type related information like organisation hierarchy etc. is missing. The Minimal OM declaration may be used in Workflow Process Definitions, while the regular OM definition may be used in Workflow Model Definitions. The Minimal OM commonly is used in connection with an external OM reference.

Attributes

For the Minimal Organisation Model Declaration the Participant Type Description attribute (see Table 3-3) differs from the OM definition in that the Type of a workflow participant is optional, and the Participant type related information is not present.

Scope

The scope of the identifier of an entity defined in a minimal OM Model Declaration is the surrounding Workflow Process Definition.

3.2.7. Extended Library

Informal Description

The Extended Library attribute may be used in the Workflow Process and the Workflow Model entity. It allows to declare Library Functions and Procedures. It may contain two parts, function and procedure declarations, which itself have further attributes including Extended Attributes.

3.2.7.1.Attributes

Attribute Name	M/O	WPDL Keyword	Value	Description
Library element identifier	M			Identifier. Used to identify the library element
		FUNCTION		Identifies a library function
		PROCEDURE		Identifies a library procedure
Result type	M	RESULT		A plain data type do denote the result type (for a Library Function only)
Name	O	NAME		*(...)*
Description	O	DESCRIPTION		Short textual description of the library element.

Attribute Name	M/O	WPDL Keyword	Value	Description
Parameters	O	(see chapter 3.1.3.2)		Parameters which are passed through to the Library Function or Procedure. For a procedure parameters may contain result values.

Table 3-10: *Attributes of Library Functions and Procedures*

4. Proposed WPDL Grammar

4.1. WPDL at a glance

The workflow process definition language (WPDL) is a language for describing workflows as an ASCII character stream (which may be a flat file or a string) using keywords (like WORKFLOW, ACTIVITY, DESCRIPTION etc.) for specifying objects, attributes and relationships and using variable parts in the grammar (string constants, and placeholders like process relevant data, etc.) for specifying their names and values.

In summary:

- The grammar is given in EBNF (Extended Backus Naur Form)
- Keywords are used to start an entity description (the entities represented in WPDL are contained in the minimum meta model)
- Keywords are written in uppercase letters
- Keyword - value pairs are used to specify attributes
- Keywords are used to specify relations to other entities
- Attributes and relations are optional
- Attributes and relations of entities are identified by keywords and don't have to appear in order
- Relations between two entities are defined on either side of the participating entities
- Tokens within an expression are separated by one or more whitespace characters
- Keywords are taken from the WfMC glossary
- Comments are supported between "/* and "*/" and after "//" (for the rest of the line)

Characteristics:

The WPDL language offers

- a minimum number of pre-defined entities (see minimum meta model)
- a minimum number of pre-defined relations between entities
- a number of pre-defined attributes (using keywords)
- additional generic attributes (for vendor specific attributes that are not pre-defined)
- additional generic relations (for vendor specific relations that are not pre-defined)
- additional generic data objects (for data objects that are not in the minimum meta model)

Recommendations:

- Only use a generic data object, if it cannot be mapped to a pre-defined entity.
- Only use a generic relation, if it cannot be expressed by a pre-defined relation.
- Only use a generic attribute, if it cannot be found in the table of pre-defined attributes.

4.2. WPDL Grammar and Language Constructs

On the following pages we define the grammar of WPDL. We do this by using a BNF-like format (BNF = Backus, Naur Form). Before introducing the syntax, we explain some general BNF rules which are used throughout this grammar, and introduce some generic and common language constructs.

4.2.1. WPDL Description Method

4.2.1.1.Metalanguage

The components of this grammar (the metalanguage) consist of:

```
·Symbols                <example_symbol>
·Keywords               EXAMPLE_KEYWORD
·Production sign        ::=
·Special characters     [ ] | / *
```

The Workflow Process Definition Language is defined as a set of productions.

On the left hand side of the production a symbol appears which is not part of the language. This symbol summarizes the components on the right hand side of the production sign. Therefore the right hand side of an production defines a rule for the development of the symbol on the left hand side.

If a symbol appears on the left hand side of a production, it can be substituted by the contents of the corresponding rule. The right hand side is a combination of symbols, keywords and special characters.

The keywords are the central parts of the language separated by blanks (white spaces). Keywords are case-sensitive, i.e. the usage of upper and lower case letters has to be considered.

A keyword or symbol (or combinations of both) appearing in square brackets ("[" and "]") indicates the construct is optional. The special character "|" implies exclusivity, i.e. one decides between the option before or behind the "|" character.

The special character combinations "/*" and "*/" indicate the part between these combinations and "//" that the subsequent part of the line up to the end are comments.

4.2.1.2.Special Symbol Conventions for Tokens

Some kinds of symbols, the tokens, are not further decomposed in the WPDL.

The basic symbols are those where descriptions is outside the WPDL.

Keywords characterize the WPDL process model description, their parts and attributes (model-relevant tokens).

Some kinds of tokens used for data types and expressions have a WPDL representation containing further keywords and special characters. To allow an easier distinction of these tokens from the meta language elements and the model-relevant keywords to provide for easier extensions we have introduced special terminal productions and added a chapter describing the WPDL representation of these special symbols.

By convention these special symbol classes are:

Operator symbols terminate with letters "Op" (e.g. `<NotOp>`)

Constant symbols terminate with an upper case letter "C" (e.g. `<BooleanC>`)

Bracket symbols terminate with an upper case letter "B" (e.g. `<OpenArrayB>`)

Type symbols are written in uppercdase letters and terminate with an upper case letter "T" (e.g. `<INTEGER-T>`)

Other terminal symbols are written in uppercase letters (e.g. `<UPTO>`)

There are other special symbols for which further conventions exist:

Symbols that denote lists terminate with "list" or " List" (e-.g. `<Activity List>, <roles description list>`), and symbols denoting vendor-defined parts begin with "extended" (e.g. `<extended attribute list>`).

4.3. WPDL

In this chapter we describe the workflow model that is built up of meta model entity descriptions and attributes. Parts of the description are mandatory, while others are optional (included in square brackets). However, it has to be mentioned that omitting the optional parts completely does not provide a useful model.

4.3.1. Workflow Participants

The Workflow Participants are those elements of an Organisation Model that are either acting parties in a Workflow Process or responsible for it. The definition is an abstraction level between the real performer and the activity, which has to be performed. It may refer to an external organisational model. Actors may be defined by a membership to an organisational unit, by a function, role or competence, by relations to actors of already performed activities etc., we call it the type. WPDL supports a basic set of types: organisational unit, human, role, resource, relation to process history.

We distinguish an regular Organisation Model definition (called Workflow Participant Definition) describing the OM entities and their types and optionally their relationships to one another as far as they are workflow relevant, and a Minimal Organisation Model Definition (called Workflow Participant Declaration) that is only a list of participant identifiers with an optional type characterisation.

```
<Workflow Participant List>                  ::=
                  PARTICIPANT                <participant id>
                       [NAME                 <name>]
                       [DESCRIPTION          <description>]
                       [<extended attribute list>]  // for participant specific a.
                       <participant type description>
                  END_PARTICIPANT
                  [<Workflow Participant List>]

     <participant id>       ::=  <identifier>
     <capacity>             ::=  <integer>    // number of parallel activities a role
                                              // is able to manage for simulation
                                              // purposes
     <prepare time>         ::=  <integer>    // estimated time needed to prepare the
                                              // execution
                                              // of an activity in seconds, default: 0
     <strategy>             ::=  FIFO         // first in, first out, default
                                 | LIFO       // last in, first out
                                 | SJF        // shortest job first
                                 | LJF        // longest job first
                                 | RD         // random
                                 | UD         // undefined

<participant type description>
                            ::=  <declaration Ptype description>
                                 | <definition Ptype description>


<Workflow Participant Declaration>
                            ::=  <Workflow Participant List>
                                   // using <declaration Ptype description>
<declaration Ptype description>
```

```
                              ::=  [TYPE        <Ptype key>]

<Workflow Participant Definition>
                              ::=  <optional simulation data>
                                   <Workflow Participant List>
                                     // using <definition Ptype description>
<optional simulation data>    ::=
                              [CAPACITY         <capacity>]
                              [COST<cost estimation>]
                              [PREPARETIME      <prepare time> ]
                              [STRATEGY         <strategy> ]
<definition Ptype description>
                              ::=  TYPE         <Ptype related information>

<Ptype key>                   ::=  <ou key> | <hu key> | <ro key> | <resource key>
     <ou key>                 ::=  ORGANISATIONAL_UNIT             // an organisational
unit
     <hu key>                 ::=  HUMAN                     // a human
     <ro key>                 ::=  ROLE                      // a role
     <resource key>           ::=  RESOURCE                  // a resource

<Ptype related information>   ::=  <ou key>   <ou attributes> [<pteal>] <ou
relationships>
                                   | <hu key> <hu attributes> [<pteal>] <hu
relationships>
                                   | <ro key> <ro attributes> [<pteal>] <ro
relationships>
                                   | <resource key> [<pteal>]
     <pteal>                  ::=  <extended attribute list>
                                   // for participant type specific attributes

<ou attributes>               ::=  [MANAGER     <manager>]  // Manager
     <manager>                ::=  <participant id>         // a human
<ou relationships>            ::=  [<superior description list>]
                                   [<person description list>]

<hu attributes>               ::=  [USERID      <string>]
                                   [SURNAME     <string>]
                                   [FORENAME    <string>]
                                   [ADDRESS     <string>]
                                   [PHONENUMBER <string>]
                                   [FAXNUMBER   <string>]
                                   [EMAIL       <string>]
                                   [X400        <string>]
<hu relationships>            ::=  [<roles description list>]
                                   [<organisational units description list>]
                                   [<proxies description list>]

<ro attributes>               ::=  [FUNCTION    <function>]
                                   [COORDINATOR <coordinator>]     // Coordinator
     <function>               ::=  <string>     // description of a task or competence
     <coordinator>            ::=  <participant id>
                                             // a human or organisational unit
<ro relationships>            ::=  [<person description list>]
```

Participant type associated lists (productions above ending with "`relationships>`") allow to specify the context of an organisational entity (type) such as human, role, organisational unit and so forth. We use a simplified notation to define these lists, called `<Prelationship list>`:

```
<Prelationship list>        ::=  <Prelation key> <Prelation description list>
<Prelation description list>::=  <Prelation description> [<Prelation description
list>]
                                                  // all of the same type (see table below)
     <Prelation key>        ::=  PERSONDESCR | SUPERIOR | ORGUNITDESCR | ROLEDESCR
                                 | ALTERNATE
     <Prelation description>::=  <participant id>

<person description list>   ::=  <PRelationship list>     // for a person
<superior description list> ::=  <Prelationship list>     // for a superior
<organisational units description list>
                            ::=  <Prelationship list>     // for a orgunit
<roles description list>    ::=  <Prelationship list>     // for a role
<proxies description list>  ::=  <Prelationship list>     // for a proxy or alternate
```

The following combinations for Prelation keys and Prelation description list elements are valid:

Prelation key	**Prelation description list elements**	**Description of list**
PERSONDESCR	IDENTIFIER *of a human*	*List of workflow participants the organisation has or the role is related to*
SUPERIOR	IDENTIFIER *of an organisational unit*	*List of direct superior organisational units*
ORGUNITDESCR	IDENTIFIER *of an organisational unit*	*List of organisational units the workflow participant belongs to*
ROLEDESCR	IDENTIFIER *of a role*	*List of roles assigned to a human*
ALTERNATE	IDENTIFIER *of any participant (usually of a human)*	*List of proxies a workflow participant has*

Table 4-11: Workflow Participant entity: Relationships

5. Annex

5.1. Annex 1: Proposed Extensions and Open Issues

In the sequel we have collected a set of open points that are either issues in the current version or proposed extensions. These are of different level of detail.

5.1.1. Organisational modeling (ch. 4.3.1)

5.1.1.1.Organisation Modelling separation

There is a request from WG1 and others to separate Workflow Modelling and Organisation Modelling (see also Annex 6: Organisation Modelling Impact on WfMC Specifications ch. 5.4)

In the present document a first step has been made to separate these parts which are combined in the Workflow Model Definition (ch. **Error! Reference source not found.**) as far as possible, e.g. by distinction of Performer (used in the WM) and Participant (used in the OM). The next step would be to have two separate definitions, one for the WM and one for the OM, and to align the relationship declarations.

In addition the discussion of different organisational models and the positioning of an organisational model within the reference model should be taken into account.

5.1.1.2.Performer-Participant Relationship Requirements

The definition of an organisation is out of the scope of this paper. Nevertheless the assignment of workflow participants to an activity refers to the organisational model. The assignment of workflow participants is not solved finally but has to include examples like:

- sales representative
- sales representative, releases sales contracts less than $50,000
- Mrs. Dastler, releases sales contracts over $250,000
- sales representative, but not performer of last activity
- boss of performer of last activity
- performer of activity x
- to be done automatically (by machine)
- sales department
- secretary in sales department
- ...

It has to be verified if by the present approach these possibilities can be described in an appropriate way. It has to be taken into account that solutions might be found inside an OM Model or by appropriate use of expression and Function Definition possibilities inside the WM.

5.2. Annex 2: Analysis of PDLs

This recommendation for a common WPDL is based upon a number of vendor specific PDLs. It was seen, that there are different methods to describe the organisational parts of a process:

- a procedural description based on workflow primitives (parallelism, alternative, loop, ...) and stepwise decomposition;
- a directed graph description with pre- and post-conditions for the activity nodes;
- a variety of petri net descriptions;
  - simple petri nets
  - predicate transition nets
  - coloured petri nets
  - funsoft nets
  - etc,
- a description of speech act networks (event driven nets).

The analysis of three different PDL's was undertaken to arrive at a "Minimum Meta Model".

In general one could say, that the discussed PDL's contain more or less the same information. However the following issues arise:

- the bundling of process definitions into sections is done differently (e.g. transition conditions are carried in an extra section versus the transition conditions are carried in pre- and post-conditions of an activity; data flow is described separately versus data flow described as input/output parameters of an activity)

It was agreed that WPDL should support two philosophies - to have all data described in one flat process definition file and, on the other hand, to allow references between separate files. Nevertheless the main focus is on processes, subprocesses and activity definitions. Organisation models are out of the scope for the first step.

In this document only the OM-related part is included.

To find a proper recommendation for WPDL we started by taking the Meta Model Entities and looked up the information corresponding to these entities in the three different PDL’s. As an example, we were able to break down the major entities as shown in the folowing table.

Meta Model Entity	**IBM Flowmark Terminology**	**Ley Cosa Terminology**	**SNI WorkParty Terminology**	**Recommended WPDL Term**
Workflow Participant	PERSON	(ref.)	(ref.) resp. part of activity	PARTICIPANT : END_PARTICIPANT

Table 5-1: Evaluating different PDLs

In the first column the table contains the entities of the Minimum Meta Model, the appropriate elements of the considered PDL's are shown in the next three columns and, in the last column, the recommendation for WPDL.

The analysis of the different PDLs brought up the aspects:

- Certain PDL's define the organisational aspects as integrated part of the process definition whilst others define attributes within the description of the activity, which points at a separate organisational management system.
- Many aspects of a process (not only the transition conditions) depend on runtime evaluations of process relevant data - input/output parameter, workflow participant definition, etc.

5.3. Annex 3: Representative Business Example

In this document only the OM-related part is included.

5.3.1. Informal Description

This section contains the methodology and results which will be used to verify that the standard being proposed for Interface 1 is in fact possible to implement.

The objective is to define a fictitious business case example which will contain most of the standard workflow entities and attributes (as defined in the Meta Model) for modelling a workflow process.

The FBN Sports Equipment Company, located in Luxembourg, manufactures a complete range of soccer, baseball, tennis and general athletic equipment. They only sell to European resellers, major sports outlets and North American Distributors. All sales are made by way of Purchase Orders.

The company has grown from a small organisation to the point where response to their customers is getting longer and longer. It has been determined to employ a workflow management system in order to improve their business processes. In the existing environment all mail comes into the mail room and is then distributed as shown in Figure 5-1. Since the company receives 80% of its sales_order activity by mail and FAX, it has been determined to have an image scanner in the mail room as one of the methods to enhance its workflow system.

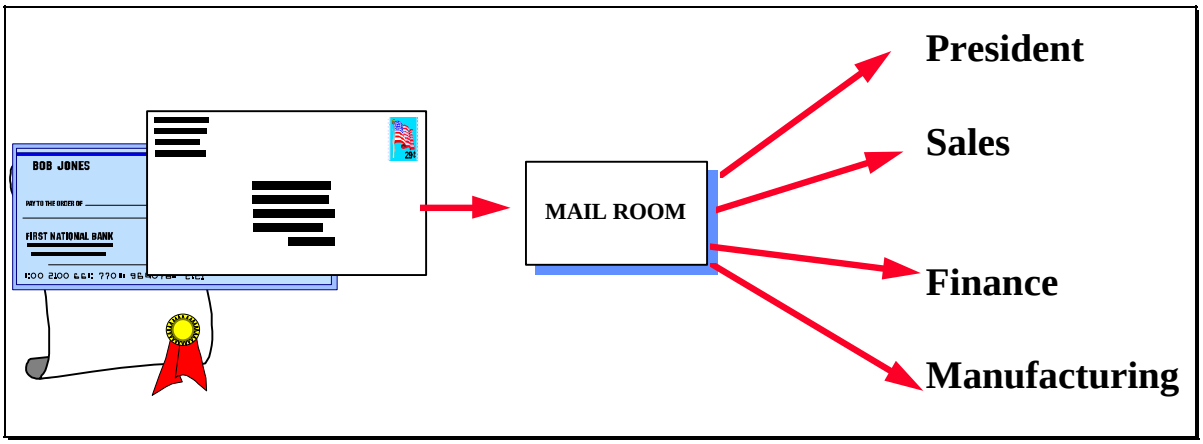

Figure 5-1: FBN Sports Equipment

5.3.1.1.Company Organisation

FBN is a small company with a President and three (3) departments; Sales, Finance and Manufacturing. The organisation is presented in Tables 5-3 and 5-4. There should be sufficient detail to determine the roles for each activity, a deputy (when required) and the specific name of an actor if needed.

In the definition of the business case, constraints have been defined which may be used as Workflow Process Relevant Data or Transition Conditions.

Position	Name	Roles
President	Adelle Dastler	1. Approves Purchase Orders over $100,000 2. Release Sales Contracts over $250,000
Sales Secretary	France Baroque	1. Schedules President's meetings 2. Arranges Travel 3. Deputy For Marketing Manager on Advertising 4. Prepares & Sends All Sales Quotations
Vice-President, Sales	Dick Meadows	1. Approves sales commission bonuses 2. Release Sales Contracts Less Than $250,000 3. Approves travel expenses less than $10,000 4. Can Approve Manufacturing Build Changes On Orders Greater Than $250,000
Channel Manager	Mark Smith	1. Handles Advertising 2. Handles all USA Distributor Sales 3. Release All Distributor Sales Under $250,000
Sales Manager	Jake Goodfellow	1. Deputy for Vice-President, Sales 2. Release Sales Contracts Less Than $100,000 3. Approves travel expenses less than $5,000 4. Can Approve Manufacturing Build Changes On Orders Less Than $100,000 5. Handles All Other European Sales
Sales Representative	Patrick LaFleur	1. Deputy for Sales Manager 2. Release Sales Contracts Less Than $50,000 3. Can Approve Manufacturing Build Changes On Orders Less Than $50,000 4. Handles All Sales For UK & France
Sales Representative	Marlene Braun	1. Release Sales Contracts Less Than $50,000 2. Can Approve Manufacturing Build Changes On Orders Less Than $50,000 3. Handles All Sales For Germany
Customer Service Representative	Joy Winter	1. Verifies Product Shipment Arrived & OK 2. Handles Direct Customer Inquiries

Table 5-2: FBN Organisation and Roles (part 1)

Position	Name	Roles
Vice President, Manufacturing	John Hammer	1. Manages Product Cost 2. Expedites parts from suppliers 3. Plans Production Schedule
Manufacturing Secretary	Heidi Werner	1. Prepares shipping & customs forms 2. Schedules transport
Production Manager	Ralph Cramden	1. Plans Production Schedule 2. Expedites parts from suppliers 3. Deputy for Vice President, Manufacturing
Production Operator	Norman Bates	1. Builds product line-A 2. Packages all product shipments 3. Prepares shipping & customs forms
Production Operator	Dave Bennie	1. Builds product line-B 2. Schedules transport 3. Receives all returned goods 4. Receives & Stores all manufacturing materials
Shipper	Theo Christensen	1. Packages all product shipments 2. Prepares shipping & customs forms 3. Schedules transport 4. Receives all returned goods 5. Receives & Stores all manufacturing materials
Purchasing Agent	Bob Cratchett	1. Purchases all parts 2. Expedites parts from suppliers 3. Can approve escalation payments up to $10,000
Vice President, Finance	Rose Chutney	1. OK's Credit Approval on New Distributor Accounts 2. Approves all sales_orders less than $250,000 +20%
Finance Secretary	Elizabeth Blume	1. Handles all Human Resource functions 2. Prepares Purchase Orders 3. Prepares Invoices
Accountant	Anna Kopinski	1. Performs Credit Checks 2. OK's Credit Approval on New Reseller Accounts 3. Approves Invoices for Mailing 4. Deputy for Vice President, Finance

Position	Name	Roles
		5. Approves all sales_orders less than $100,000 +20%
Finance Clerk	Jim Samson	1. Approves all sales_orders less than $50,000 +20% 2. Deposits customer payments 3. Handles customer interface 4. Deputy for Accountant
Mail Room Clerk	Tim White	1. Opens and distributes all mail 2. Operates company FAX distribution

Table 5-3: FBN Organisation and Roles (part2)

5.3.2. Representative Business Example in WPDL Notation *<to be aligned>*

The example has not been outlined completely. The representative business example contains 4 processes, an organisational model, business example relevant data and a list of applications, which have to be invoked through the activities.

To show the structure of the file and the WPDL elements the example is broken down in a number of significant parts.

At this time WPDL does not support rendezvous linkages between processes and the capability to describe the FBN's Organisation. By using extended attributes or business case global data these restrictions may extinguished.

5.3.2.1. Workflow Participants

```
// <=Workflow Participant List>

PARTICIPANT             'p_1'
   NAME                 "France Baroque"
   TYPE                 HUMAN
   USERID               "fb123"
   SURNAME              "Baroque"
   FORENAME             "France"
   DESCRIPTION          "President's Secretary"
END_PARTICIPANT
PARTICIPANT             'p_2'
   NAME                 "Tim White"
   TYPE                 HUMAN
   USERID               "tw456"
   SURNAME              "White"
   FORENAME             "Tim"
   DESCRIPTION          "Mail Room Clerk"
END_PARTICIPANT

PARTICIPANT             'p_3'
   NAME                 "Presidents_Secretary"
   TYPE                 ROLE
   PERSONDESCR          "France Baroque"
```

```
   DESCRIPTION          "handles presidents mail"
END_PARTICIPANT
PARTICIPANT             '_4'
   NAME                 "Mail_Room_Clerk"
   TYPE                 ROLE
   PERSONDESCR          "Tim White"
   DESCRIPTION          "handles incoming mail"
END_PARTICIPANT
PARTICIPANT             'p_5'
   NAME                 "VP_Sales"
   TYPE                 ROLE
   DESCRIPTION          "handles sales leads"
END_PARTICIPANT

PARTICIPANT             'p_6'
   NAME                 "Sales_Department"
   TYPE                 ORGANISATIONAL_UNIT
   DESCRIPTION          "is involved in handling sales orders"
END_PARTICIPANT
PARTICIPANT             'p_7'
   NAME                 "Manufacturing_Department"
   TYPE                 ORGANISATIONAL_UNIT
   DESCRIPTION          "is involved in handling sales orders"
END_PARTICIPANT
PARTICIPANT             'p_8'
   NAME                 "Finance_Department"
   TYPE                 ORGANISATIONAL_UNIT
   DESCRIPTION          "is involved in handling sales orders,
   invoices for payments and payments"
END_PARTICIPANT
PARTICIPANT             'p_9'
   NAME                 "Customer Support"
   TYPE                 ORGANISATIONAL_UNIT
   DESCRIPTION          "handles product inquiries"
END_PARTICIPANT
```

5.4. Annex 4: Organisation Modelling Impact on WfMC Specifications

Organisation Modelling (Position Paper by Martin Ader)

Ader.M.67M@centraliens.fr (Martin Ader)

Forewords to version 1 10/11/96

This version of this document is the result of Martin Ader ideas (W&GS) corrected by Juergen Elden (dresdnerbank) it was also sent to Gerhard Wernke(SNI) and Fred.Van.Leeuwen. A new version will be distributed as soon as their comments will be received

5.4.1. Goal

Goal of this document is to propose, as a position paper, an analysis of the organisation modelling techniques used in workflow systems in order to attempt to identify :

- levels of concepts that could potentially be part of the WfMC= reference model, and
- possible consequences on WfMC specifications at the level of tasks handling.

From that point, it will be up to the WfMC to decide what parts of that position paper is a valuable target for inclusion in the reference model and to undertake necessary studies to update accordingly each interface

specification.

5.4.2. Organisation modelling

Actor. Modelling of the organisation is a mandatory feature in any workflow system since their goal is to assign activities to actors. The minimal model of an organisation is thus made of the actors potentially participating to the processes supported.

Role. Most workflow models add the role object. This enables to separate the allocation of task to actors into two parts :

- activities are assigned to roles at the process definition level, and

- a role is assigned to an actor by interpreting, the assignment made at process definition level, according to a mapping done in the organisation model between roles and actors, and in the context of a particular instance of a business process.
- it may also be useful to introduce hierarchies of roles, e.g. to describe that a certain role supervises another. In that way, the process description remains relatively independent of the actual organisation since, for example, changing actors that can hold a certain role does not impact the process description.

Grouping. By adding grouping object like organisation-unit, project, committee the model becomes capable of representing a real organisation. Grouping can be assembled in trees reflecting the various levels of organisation. This added level of representation offers

- scope
- features

in affectation process of activities to actors like : "any actor of the marketing unit the actor holding the manager role of the sales district 4 of the sales division".

Attributes. Each object type has its set of attributes that might vary in a large extend with models. They can be used also for actors selecting through query expressions of unlimited complexity at least in some products, example : buyer role attribute product family, buying process attribute product family, tenant of the process any holder of a buyer role with product family equal to the one of the process..

Security. With those objects can be associated security levels, or identifiers used to control access to processes, and to activities. The way this is achieved can vary and can include mechanisms associated to groupings that look like to inheritance with inclusion end exclusions mechanisms. Points of view differ on the usefulness of such features : (I would not introduce security as an independent feature of WF. From my point of view it is easier and sufficient to map security mechanisms to roles and limit the definition of security to the assignment of roles to actors. If you apply this separate feature, you will have to explain the system's behaviour if, e.g., an activity is associated with a particular role owned by only one actor who does not own the required security level) Juergen Elden.

5.4.3. Relationships with workflow features

In workflow systems, the organisation model is related to several important features : organisation administration, process definition, process installation, user agent features, statistics, and application invocation.

5.4.3.1.Organisation administration

A tool enabling creation and maintenance of the organisation model. This tool is very dependent on the model itself and of its complexity. However, from the level of model that will be retained by the WfMC as the one that must be part of any workflow system it can be derived a set of actions that the administration interface must or can support. A special point is the administration of actors substitutes (see later) It is important to note that such tools already exist at least in large organisations and are applied independent of WF. They handle e.g. personnel data bases and security levels resp. access rights. In this area it is especially important to map the concepts applied within these tools and the objects handled by them to the needs of WF systems. For practical use, the Coalition will also have to define means to make use of this existing data, perhaps by augmenting the administration spec. by import interfaces.

5.4.3.2.Process definition

One essential part of the process definition is the expression of how the engine will select the actor to which the activity should be assigned at run time for each process. Let us call that assignment expression. This expression can take several forms described here in natural English (non exhaustive):

1. direct designation of an actor by its name
2. one of the members of a grouping
3. one actor having a designated role
4. assign to the actor that was previously selected for the designated activity.
5. assign an actor which is not the actor selected for a specific activity, but using the expression defined for that activity (for double action required in banks)
6. assign to the manager of the grouping to which belongs the actor assigned for a specific activity. (do, then approve by manager)
7. assign to the actor that initiated the process
8. assign to the actor that has the responsibility of the whole process, can be defined by an expression
9. assign to one actor inside a list being the result of the evaluation of a query expression, the expression can use attributes of the activity (input) or of the process, as well as attributes of the organisation objects. This expression can be in addition qualified by the rule used for selection the actor if the expression returns several actors. The most frequently used rules are : fifo, Random, Load Balancing

Position : From a technical point of view it is sufficient to introduce logical expressions like well known from programming languages to calculate an actor by introducing variables, Boolean operators like =, <>,= relational operators like AND, OR, NOT. For sake of simplicity a front end might map them to operators like SAME_AS. Special roles like INITIATOR could be generalised to e.g. ACTOR_N being a variable describing the actor of activity N. (INITIATOR is then equal to ACTOR_1) Juergen Elden.

Comment by Juergen Elden There are much more things to be defined at this stage. At least the programs associated with an activity, perhaps the objects to be processed within that activity and often deadlines, priorities and so on will have to be added.

Process installation

When a new process is installed for exploitation in a workflow engine, its triggering conditions are specified. More precisely restrictions are declared on who can start such a process, and on who is responsible of the handling of exceptions in that process (responsible). This is usually defined using the organisation model. Some examples follows.

1. The sales process can be initiated by any actor with rule salesman. The responsible is the manager of the initiator.
2. The buy process can be initiate by any actor with role manager and authority RequestInvestments, the responsible is any actor with role Buyer.
3. The Mission order process can be initiate by any actor, the responsible is the actor with role secretary of the grouping to which the actor belongs to.

Subject for debate : Juergen Elden: I am not sure if process installation has to be a separate step. I guess all these definitions undertaken here can be sufficiently associated with the Process definition phase. Martin Ader: goal is to be able to re-install a procedure with different parameters without updating it ,

Client applications features

From the client, the actor to which the activity was assigned has several possibilities :

1. transfer the activity to another actor (that should satisfy the selection rules used to select the initial actor) then a list of possible actors should be presented to the user.
2. Ask the system to select another actor to handle the job, using the initial expression.
3. Send the activity to someone else for advice and process it only when back to him.
4. In addition, when absent, an actor has a substitute, or a rule to compute a substitute. The activity is then redirected automatically to its substitute. This assignment must be handled and administered.
5. postpone an activity to a later date
6. make annotations to a process for later use by subsequent actors
7. cancel a process
8. Modify the process by insertion of a new activity resp. removal of an unneeded activity ...
9. Statistics

5.4.3.3.Application invocation

When an application is automatically invoked, all the problems of adequate transfer of access rights from the workflow engine to the application must be handled this can be done by

1. trusting the workflow from others applications, the workflow is seen as a super user from those applications and gives its own identification and password
2. transferring user identification password to the application (problem of global administration)
3. asking the user a new login by the application (boring for the user).
4. trusting other access control mechanisms within workflow; the WF management system is seen as a service accessible only by previously authorised user

5.4.3.4.Inter process communication

When a process invokes another process, there is a problem of verification of rights and of ownership of the newly created process. Again there are here several solutions :

1. The workflow engine invoking the other one is the owner of the new process regardless for whom the invoking process is being run (Trust strategy)
2. If it is an activity that creates the sub process, it is the actor of the activity that becomes the initiator of the sub process. If the two processes are in different workflow engines this requires that he actor is present in the two organisation models.
3. If it is a process that creates a sub process, it is the initiator of the process that becomes the initiator of the sub process.
4. At process creation time, the assignment of the actor to the initiator role is done according to a rule part of the call.

5.4.4. Relationships with WfMC interfaces

5.4.4.1.Workflow Process Definition

1. Definition of objects depending on organisation model chosen
2. Definition of assignment expressions, depending on desired coverage

5.4.4.2.Workflow inter operation

1. Added parameters form security and ownership continuity across workflow engines.
2. Secure transmission activation (encryption, signature, non repudiation) Although this is beyond WfMC (it is rather the application of an independent service) the Coalition will need at most an interface to that.

5.4.4.3.Application call

1. How to ensure security by the called application

5.4.4.4.Client Interface

1. Ways to redirect an activity
2. Commands to ask list of adequate actors

5.4.5. Administration

1. Administration of the organisation model
2. Assignment of substitute to actor.
3. Installation of new procedures with security parameters

5.4.6. External standards to be considered

1. X500 for its representation of an organisation
2. Common login mechanisms
3. CORBA conventions for secure calls
4. Standards for electronic signatures (Internet, Kerbos, ...)

5.4.7. Recommendations

An architecture working group sub group should be created to address those concerns and decide which one can and should be taken into account as part of

1. the reference model
2. each interface specification.

5.5. Annex 5 Document History

Changes Draft 6.94i dated April 18 1997to Draft 6.94k dated July x 1997

- Split document into part -O (Organisation Modelling) and -P (Process)modelling)
- <optional simulation data> nonterminal introduced, shifted to position after extended attributes

Changes Draft 6.94k dated July x 19971997to Draft 6.95 dated Octber 15 1997

- Small structural changes of layout

6. Index

6.1. Index of Figures

Figure 3-2: Organisation Model Definition Meta Model.....11
Figure 3-3: Types of Workflow Participant Assignment.....13
Figure 5-1: FBN Sports Equipment.....28

6.2. Index of Tables

Table 3-1: Extended Attributes..... 8
Table 3-2: Formal Parameters..... 8
Table 3-3: Attributes of Entity Workflow Participant.....12
Table 3-4: Entity Workflow Participant: Strategy Attribute 12
Table 3-5: Types of Workflow Participants 13
Table 3-6: Description Lists associated with Types of Workflow Participants 13
Table 3-7: Attributes of Organisational Unit 14
Table 3-8: Attributes of Human.....15
Table 3-9: Attributes of Role.....15
Table 3-10: Attributes of Library Functions and Procedures 18
Table 4-11: Workflow Participant entity: Relationships 24
Table 5-1: Evaluating different PDLs.....26
Table 5-2: FBN Organisation and Roles (part 1).....29
Table 5-3: FBN Organisation and Roles (part2).....31