

University of Castilla-La Mancha



A publication of the
Department of Computer Science

Estudio del Estado del Arte en Modelado y Ejecución de Procesos Interorganizacionales

by

Ángel Martínez, José Hilario Canós, Jesús Damián
García-Consuegra

Technical Report **#DIAB-01-06-21** October 2001

DEPARTAMENTO DE INFORMÁTICA
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE CASTILLA-LA MANCHA
Campus Universitario s/n
Albacete – 02071 – Spain
Phone +34.967.599200, Fax +34.967.599224

Este trabajo ha sido parcialmente financiado por los proyectos CICYT "DOLMEN", TIC 2000-1673-C06-06 y "METODOLOGIA DE ANALISIS DE SISTEMAS DE INFORMACION GEOGRAFICA", TIC 2000-1106-C02-02.

Contenido

Contenido	I
Figuras	II
1 Introducción	1
2 Conceptos básicos	2
2.1 <i>Descripción del problema</i>	3
2.1.1 Un ejemplo del problema.....	4
2.2 <i>Clasificación en función del campo de aplicación. Diferencias en los requisitos</i> 5	
2.3 <i>Requisitos para el soporte de la interoperabilidad</i>	7
2.3.1 Requisitos de modelado.....	7
2.3.2 Requisitos de los sistemas de soporte.....	8
2.3.3 Requisitos conjuntos de modelo y sistema.....	9
2.3.4 Evolución en procesos interorganizacionales.....	9
3 Modelos de procesos e interoperabilidad semántica	11
3.1 <i>Principales modelos propuestos</i>	14
3.1.1 Modelo general propuesto por la WfMC.....	15
3.1.2 Modelo IPO.....	17
3.1.3 Modelo <i>Language Action</i>	17
3.2 <i>Estándares de definición e intercambio de información sobre procesos</i>	18
3.2.1 WfMC interfaz de interoperabilidad y Wf-XML.....	18
3.2.2 SWAP.....	21
3.2.3 BPML y BPQL.....	22
3.2.4 PIF.....	22
3.2.5 The Process Specification Language Project, PSL.....	24
3.2.6 Workflow Management Facility.....	26
3.3 <i>Influencia de otras tecnologías en el modelado de procesos</i>	28
3.3.1 Modelado orientado a objetos.....	28
3.3.2 Técnicas formales.....	28
3.3.3 Arquitectura software y metanivel.....	29
3.3.4 Bases de datos federadas.....	30
3.4 <i>Discusión</i>	30
4 Sistemas de soporte de procesos interorganizacionales	32
4.1 <i>Clasificación</i>	32
4.2 <i>Principales sistemas y arquitecturas propuestas</i>	35
4.2.1 Marco de interoperabilidad propuesto por la WfMC.....	35

4.2.2	Interacción basada en eventos.....	36
4.2.3	Sistemas basados en metáforas.....	37
4.3	<i>Discusión</i>	37
5	Conclusiones y trabajos futuros.....	39
6	Referencias.....	40

Figuras

Figura 1.	Diagrama de Actividades que representa el proceso interorganizacional de ejemplo. Simplificación del propuesto en [van der Aalst y Weske, 2001].	5
Figura 2.	Esquema de los principales requisitos aplicables al modelado y ejecución de procesos interorganizacionales.....	10
Figura 3.	Ejemplo de representación gráfica de grados de implementación de interoperabilidad. Los círculos agrupan requisitos a diferentes niveles de importancia mientras que los sectores los subdividen dependiendo de su campo de aplicación. Esquema general (a) y ejemplo de implementación de interoperabilidad entre sistemas de flujo de trabajo y aplicaciones de reingeniería (b).....	14
Figura 4.	Terminología utilizada por la WfMC y sus relaciones [WFMC, 1999]......	16
Figura 5.	Visión general de un proceso según el paradigma IPO.....	17
Figura 6.	Ejemplo de proceso definido usando el paradigma Language Action. New Product Approval es el diálogo ActionWorkflow (ActionWorks® Process Builder).	18
Figura 7.	Ejemplo de mensaje codificado en Wf-XML para la creación de una instancia de proceso.	20
Figura 8.	Ejemplo de codificación de una petición de instanciación.....	21
Figura 9.	Ontología general de PIF. Todo es una ENTIDAD, que se especializa en ACTIVIDAD, OBJETO, PUNTO TEMPORAL y RELACION. Un proceso en un conjunto de ACTIVIDADES conectadas entre ellas o con otros OBJETOS mediante RELACIONES con PUNTOS TEMPORALES.	23
Figura 10.	Extracto de la ontología PSL Core expresada en KIF.....	25
Figura 11.	Modelo propuesto en la Joint Workflow Management Facility.	27
Figura 12.	Estilos arquitecturas dependiendo del elemento utilizado para la interconexión: (a) control (b) estado.....	33
Figura 13.	Clasificación propuesta por [Ben-Shaul y Kaiser, 1998] para los entornos de desarrollo de software. Verticalmente se representa el grado de heterogeneidad y horizontalmente el grado de acoplamiento entre los componentes del grupo de colaboradores. Se han incluido flechas para mostrar la equivalencia con la definida en bases de datos.....	34
Figura 14.	Modelo de referencia de la coalición [WFMC, 1999].....	35
Figura 15.	Esquema de interacción mediante nodos de evento. Ejemplo de ejecución de una secuencia de actividades insertando un nodo de evento de envío (org. 1) y recepción (org. 2) para la interacción mediante el evento E.....	36

1 Introducción

En este trabajo se presentan los resultados de la investigación sobre el estado del arte en interoperabilidad entre sistemas de soporte de procesos. La aceptación de las tecnologías de soporte de procesos está llevando a que surjan numerosos productos comerciales para el soporte de procesos organizacionales. Por otro lado, el proceso de globalización de la información está haciendo evolucionar rápidamente tecnologías como el comercio electrónico (B2B y B2C) y los servicios electrónicos, cuya implementación suele soportarse sobre sistemas orientados a procesos. Progresivamente, aumenta la necesidad de entendimiento entre las organizaciones que usan esas tecnologías, lo que implica crear procesos globales que articulen la interacción entre organizaciones colaboradoras. Pero estos procesos globales requieren grandes esfuerzos de estandarización o acuerdos puntuales para poder ser aceptados y comprendidos por cualquier participante. Para facilitar estas tareas se puede optar por establecer mecanismos de comunicación flexibles que no impongan obligaciones o restricciones notables a las organizaciones que los utilizan. Este trabajo trata de estudiar los problemas a los que se enfrentaría este hipotético “sistema de soporte de la interoperabilidad”.

En un primer trabajo de investigación [Martínez, 2000] se realizó el estudio de la evolución dinámica de flujos de trabajo. Como continuación, se presenta también este estudio sobre los problemas de interoperabilidad, con el fin de centrar la investigación en los problemas de evolución en ambientes interorganizacionales y heterogéneos. La característica necesariamente abierta de la interoperabilidad llevó a ampliar el campo de estudio a la interoperabilidad entre cualquier sistema orientado a procesos. Por ello, el objetivo de este trabajo se ciñe al problema de la interoperabilidad en ambientes heterogéneos, pero con la intención de servir como base para analizar la evolución en procesos interorganizacionales.

El trabajo se estructura de la siguiente forma. En el capítulo 2 se da una visión general del problema, una relación de los requisitos que le afectan y los conceptos principales manejados en el trabajo. En los capítulos 3 y 4 se hace un estudio de los modelos y sistemas actuales para la solución del problema. En el capítulo 3 se hace un especial hincapié en los estándares y tipos de modelos para discutir la posibilidad de interacción entre sistemas de soporte de procesos, mientras que en el cuarto se detallan los tipos de sistemas y los requisitos que satisfacen. Finalmente en el capítulo 5 se presentan las conclusiones y trabajos futuros.

2 Conceptos básicos

Desde antiguo se ha asumido la existencia de procesos en entornos de manufacturación o de oficina, existían una serie de actividades relacionadas que en su conjunto daban respuesta a una necesidad de la organización, pero no se describían explícitamente. Así por ejemplo en una referencia clásica para introducir el tema [Georgakopoulos et al., 1995] no se define explícitamente la noción de proceso, sino que simplemente se evidencia su existencia y se describen sus características. Independientemente del grado de automatización y de orientación a humanos o a sistemas, en [Georgakopoulos et al., 1995] un proceso se caracteriza por separar claramente tareas, roles, reglas y procedimientos que regulan el trabajo en manufacturación o en la oficina. Así, los procesos pueden clasificarse en tres categorías dependiendo del objetivo del trabajo que articulen:

- *Proceso material*: el objetivo es el ensamblado y distribución de productos físicos. En algunos casos se hace referencia a ellos como procesos de producción.
- *Proceso de información*: en el que el trabajo consiste en crear, procesar y proporcionar información. Las bases de datos son el principal soporte a este tipo de procesos.
- *Proceso de negocio*: a un nivel de abstracción mayor que los anteriores, denota el conjunto de actividades a realizar para satisfacer necesidades de clientes. Un proceso de negocio comprende tanto procesos materiales como de información

En este trabajo se ofrece una visión global del estado del arte en la automatización de cualquier tipo de proceso, pero se dedica mayor atención a los procesos de negocio, dada su importancia y la existencia de muchos más trabajos relacionados con este campo de aplicación.

Así, un entorno cuyo trabajo esté definido en base a procesos se llamará *entorno centrado en procesos* (Process-Centered Environment, PCE). El modelo de procesos será la especificación de cómo se define un proceso en un PCE determinado. En base a este modelo se crearán representaciones de procesos que se utilicen en el PCE, a los que llamaremos simplemente *procesos* (resultados del modelado). En la bibliografía también se le da el nombre de *tipo* o *plantilla de proceso* [Kradolfer, 2000][Weske, 1999] a esta representación del proceso real en un modelo concreto, por lo que con *proceso* se hacen referencia a la entidad real. En este trabajo no se hace necesario distinguir entre el proceso real y su representación en una herramienta de modelado, ya que siempre trabajamos con representaciones. Cada vez que se interprete y ejecute un proceso para satisfacer una necesidad diremos que se ha creado una instancia del proceso. En este trabajo se llamará *sistema de soporte de procesos* (Process Support System, PSS) a un sistema que ofrezca la siguiente funcionalidad básica: modelado de procesos, ejecución/gestión de sus instancias y monitorización.

Los campos a los que debe ceñirse nuestro estudio son básicamente aquellos en los que se requiere la consecución de un objetivo, y el medio para conseguirlo es la utilización de un modelo de coordinación que permite realizar las actividades necesarias para alcanzarlo. Los procesos de negocio (gestión de flujos de trabajo, [Georgakopoulos et al., 1995]), los procesos de desarrollo de software [Sa y Warboys, 1995], la experimentación científica [Alonso y Hagen, 1997], las empresas de manufacturación o la automatización de las cadenas de suministro [Hull et al., 2000] son algunas de las áreas de las que surge nuestra idea de proceso. Aunque con características y requisitos diferentes, lo común a todos estos campos de aplicación es la necesidad de definir un mecanismo de coordinación, siendo por tanto el área de coordinación de

actividades de trabajo (tal y como se define por primera vez en [Sheth, 1996]) la que nos proporciona una nueva definición de proceso. Concretamente, un proceso será un conjunto de servicios ofertados a un usuario (humano o no). En general se ha definido un proceso como un conjunto actividades realizadas para obtener un producto, pero nuestra visión está más cerca de la expuesta en [Warboys et al., 1999], donde se define un proceso como un proveedor de servicios.

2.1 Descripción del problema

Muchos vendedores de software están desarrollando nuevos productos y tecnologías de soporte de procesos [SAP, 2001][Navision, 2001][Microsoft, 2001][Oracle, 2001], a lo que acompaña una continua introducción de productos en el mercado. La disponibilidad de un gran número de productos ha provocado que vendedores individuales se centren en una funcionalidad o característica particular y los usuarios adopten productos para satisfacer necesidades específicas. Aun así no existe un marco suficientemente aceptado, a excepción quizá del propuesto por la Workflow Management Coalition (WfMC) para flujos de trabajo, para permitir que los diferentes PSS puedan trabajar juntos. El resultado es la proliferación de “islas” de automatización de procesos sin mecanismos definidos que las aune.

El problema general de la interoperabilidad se define como “la habilidad de dos o más sistemas o componentes para intercambiar información y hacer uso de la información que se ha intercambiado” [IEEE 90]. La parte realmente importante de la definición es la final, donde se especifica la necesidad de hacer uso de la información, con lo que el problema no se reduce sólo a la comunicación, sino también a la comprensión, de la información intercambiada. La interoperabilidad ha ganado interés en dominios como las arquitecturas cliente-servidor, las bases de datos o los procesos de negocio ya que la propia naturaleza de estos sistemas provoca la necesidad de comunicación entre productos que probablemente hayan sido desarrollados separadamente. En un PSS el problema de la interoperabilidad se define como “*el balanceo entre autonomía y colaboración entre múltiples procesos, tanto en la fase de diseño como ejecución, como base para la colaboración entre diferentes grupos*” [Ben-Shaul y Kaiser, 1995]. Además de cumplir con los objetivos planteados en la primera definición, aquí se añaden dos nuevos requisitos: “balanceo entre autonomía y colaboración” y “colaboración entre procesos”:

- **Autonomía y colaboración.** Las organizaciones y sus sistemas necesitan establecer relaciones de colaboración, pero preservando su autonomía y ocultando información a posibles competidores. Este será un problema recurrente en interoperabilidad entre PSS, provocando el rechazo de soluciones que no contemplen la publicación selectiva de información.
- **Procesos globales o particulares.** La existencia de procesos globales está muy relacionada con la autonomía. Llamamos *proceso global* (o interorganizacional) a aquél que está definido en el marco de la colaboración y establece el trabajo que ha de realizar el conjunto de colaboradores. En ocasiones este proceso será el único elemento de información que ha de intercambiarse, con lo que únicamente se conocerán otras organizaciones en relación al papel que juegan en la colaboración. Un caso particular de esta situación ocurre cuando el proceso global es propiedad de una organización, convirtiéndose el problema de la interoperabilidad en un problema de interconexión de

un coordinador (*outsourcer*) con organizaciones subcontratadas (*insourcers*) que únicamente conocen la actividad encomendada.

2.1.1 Un ejemplo del problema

El ejemplo que vamos a utilizar para ilustrar el problema al que nos enfrentamos será una simplificación del utilizado en [van der Aalst y Weske, 2001]. En la Figura 1 se muestra el diagrama de actividades UML [OMG, 2000] que define el proceso. Aunque éste tipo de diagramas no es apropiado para especificar procesos, se ha utilizado aquí por ser de uso general, facilitándose así su comprensión sin necesidad de introducir una notación desconocida para el lector.

El ejemplo presenta un posible proceso interorganizacional para la compra de libros en una librería electrónica. En él intervienen tres organizaciones (librería, editor y distribuidor) y el usuario. Puede interpretarse como un acuerdo entre la librería, el editor y un distribuidor para suministrar libros a un cliente, que a su vez se modela como otro agente más dentro del proceso. Solamente se describirá el flujo de actividades básico, pues una especificación real incluirá muchas más actividades y tratamiento de condiciones especiales. Además, el hecho de ser un proceso interorganizacional implica que es conocido por los participantes en el negocio, lo que provoca que únicamente se publiquen las actividades relevantes desde el punto de vista de la coordinación, dejando a cada participante la posibilidad de definir sus propios subprocesos concretos, autónomos y conocidos únicamente por la organización propietaria. Por ejemplo, la actividad “*Enviar Libro*” implicará gran cantidad de actividades para el distribuidor, pero el editor únicamente debe conocer que el distribuidor realiza el envío y su notificación a la librería, lo que posiblemente evita hacer la notificación al editor.

El diagrama está dividido en cuatro calles correspondientes a las cuatro organizaciones que intervienen en el proceso. Las actividades se han situado en la calle correspondiente a la organización que debe llevarlas a cabo. Las instancias del proceso se inician por parte de un cliente que realiza un pedido (*Pedido*), que debe ser gestionado por la librería (*Gestionar Pedido*). Como la librería es electrónica y no tiene stock debe realizar a su vez un pedido al editor (*Realizar Pedido*). El editor evalúa el pedido (*Evaluar Pedido*) y notifica el rechazo o aceptación del pedido dependiendo de sus existencias (*Notificar Rechazo* y *Notificar Aceptación* respectivamente). Con esta información la librería notifica al usuario la viabilidad del pedido (*Confirmar Pedido* y *Declinar Pedido*). Nótese que esta serie de actividades podrían realizarse en tiempo real: el usuario realiza la actividad de solicitud y la confirmación implica la ejecución de parte del proceso interorganizacional. Dependiendo del grado de automatización de las diferentes organizaciones, podría darse el caso, por ejemplo, de que la evaluación del pedido por parte del editor requiriera actividades manuales. Este tipo de información debería poder incluirse en el proceso de forma que la implementación de las actividades de confirmación del pedido fueran diferidas. De cualquier forma, si todo fue bien el editor se encargará de enviar el libro por medio de una empresa de distribución (*Enviar Libro* en el editor y *Enviar Libro* y *Notificar* en el distribuidor). Por último la librería enviaría la factura (*Enviar Factura*) que el cliente debería pagar una vez recibido el libro (*Recibir Libro*, *Recibir Factura* y *Pagar*).

Por último discutiremos la posibilidad de que el cliente sea otra organización más. Si tenemos en cuenta que el cliente tiene capacidad para automatizar procesos, las peticiones de libros deberían tener en cuenta el proceso global. La organización cliente podría definir subprocesos que realizaran por ejemplo las actividades de pago, lo que contribuiría también a automatizar las actividades del cliente sin más que conocer el proceso global.

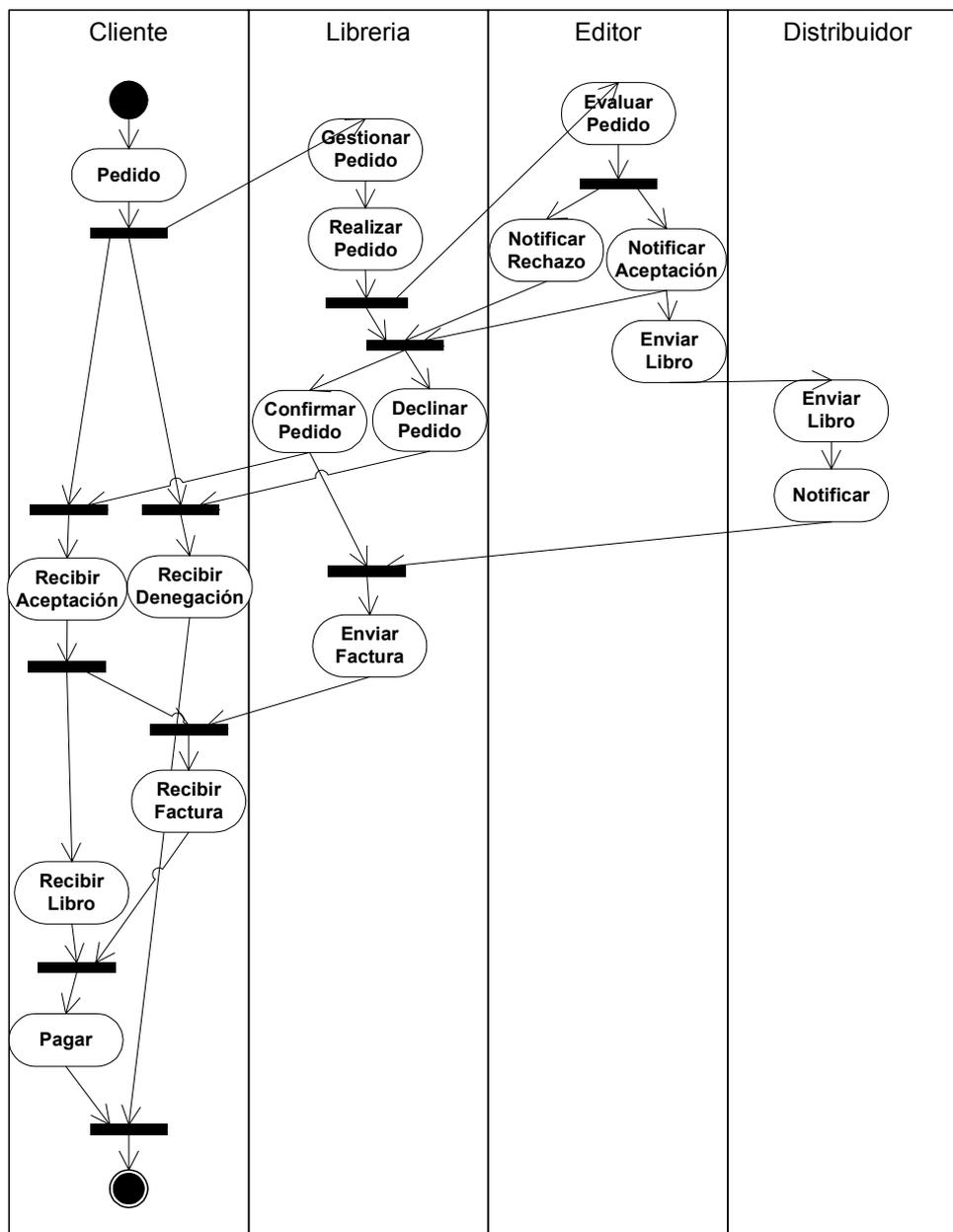


Figura 1. Diagrama de Actividades que representa el proceso interorganizacional de ejemplo. Simplificación del propuesto en [van der Aalst y Weske, 2001].

2.2 Clasificación en función del campo de aplicación. Diferencias en los requisitos

Los conceptos de soporte de procesos o coordinación de actividades tienen un alto número de campos de aplicación [Wainer, 2000]. Aunque poseen características comunes, cada uno de estos campos de aplicación precisa requisitos concretos que deben tenerse en cuenta tanto en el modelo de proceso como en los sistemas y arquitecturas que lo soportan. Veremos

una relación de los campos más importantes y algunas de sus características principales. El análisis de las diferencias permitirá extraer conclusiones sobre el nivel de flexibilidad y las características de los PSS en general. Además, ayudará a comprender la dirección en que se avanza en cada uno de esos campos cuando analicemos el estado del arte. Los principales campos de aplicación y donde se ha realizado un mayor esfuerzo de investigación son [Wainer, 2000]:

- **Procesos de manufacturación:** la principal aplicación del modelado de procesos es como planificadores de trabajos. Aquí por lo general los recursos necesarios para realizar las tareas son escasos (máquinas, trabajadores, materiales, etc.). El objetivo es optimizar el rendimiento. El PSS debe controlar los momentos en que los actores comienzan o terminan las actividades. La coordinación se produce entre procesos, decidiéndose en cada momento qué actividades de qué procesos deben realizarse para completar el producto en el menor tiempo y con el menor coste.
- **Aplicaciones de negocio:** se asume que no existe una carencia de recursos tan fuerte como en procesos de manufacturación, no existiendo un control tan férreo sobre las actividades que deben realizarse. En muchos casos los actores son libres de realizar las actividades en el orden que prefieran e invirtiendo el tiempo necesario (no conocido a priori). Los procesos se definen mediante flujos de trabajo (workflows) que asignan actividades a actores y establecen las restricciones o recursos necesarios para que cada actividad pueda llevarse a cabo. En ese momento la actividad se coloca en una lista de actividades pendientes del actor, perdiendo así el control sobre cuando se realizará la actividad. En este caso la coordinación se produce principalmente entre las actividades de un mismo proceso, y la función principal del PSS es de dispensador, seleccionando actividades que pueden realizarse y asignándolas al actor que pueda realizarlas.
- **Desarrollo de software:** el principal condicionante es que generalmente no existen suficientes recursos para realizar todas las actividades. Se debe seleccionar qué actividad y de qué proceso debe realizarse a continuación, pero a diferencia de los procesos de manufacturación la planificación no es automatizable, no se dispone de información de rendimiento. De nuevo la coordinación se realiza entre procesos. Las principales funciones del PSS son ayudar a tomar decisiones de planificación y el cálculo de consecuencias a la toma de decisiones. Aquí el PSS es un asistente que lista actividades preparadas para ser realizadas y que deben ordenarse (planificarse) o también puede ser un simulador que calcula las consecuencias de una determinada decisión de planificación (por ejemplo la ampliación de la etapa de prueba en el desarrollo de una aplicación)
- **Procesos científicos:** aunque es menos común, la gestión de procesos se puede aplicar para imponer políticas o reglas a los procesos científicos. En este dominio existen reglas que deben comprobarse durante o después de completar el proceso, por ejemplo el cálculo de correlaciones o pruebas. El PSS puede servir como avisador de ciertas condiciones, como la necesidad de realizar alguna prueba o la verificación que todas las actividades se han llevado a cabo. Un ejemplo típico es aquel en el que se realiza un experimento y el cambio de una condición (proporción de un producto) provoca la necesidad de reevaluar ciertos resultados, con lo que el PSS debería de encargarse de avisar sobre la realización de las actividades asociadas. Por todo ello podemos concluir que en éste dominio el PSS realiza funciones de ayudante o verificador.

Como se ha podido observar cada campo de aplicación posee unos requisitos diferentes, siendo quizá la diferencia más importante la distinción entre dominios orientados a la coordinación inter-proceso o intra-proceso. Esta distinción obliga, en el primer caso a definir primitivas de coordinación entre procesos reduciendo el problema a uno de concurrencia, mientras que el segundo obliga a definir un modelo de proceso rico que permita expresar los artefactos necesarios para el modelado de los procesos en ese dominio.

De cualquier manera los requisitos en cada dominio no se definen en conjuntos disjuntos. Así por ejemplo puede aparecer la necesidad de cambiar el proceso general para adaptarse a cambios en ejecuciones particulares. El PSS puede actuar como replanificador, permitiendo la modificación del proceso para adaptarse a condiciones especiales y asegurando que no se incumplan las demás restricciones establecidas en el proceso.

2.3 Requisitos para el soporte de la interoperabilidad

Existen numerosos requisitos y condiciones que intervienen en la construcción de PSS y sus modelos asociados. Algunos serán aplicables a cualquier dominio de aplicación, como por ejemplo la necesidad de monitorizar instancias. También existen requisitos aplicables únicamente a problemas interorganizacionales, como podría ser el modelado de la interoperabilidad. No se tratará de hacer una relación exhaustiva de requisitos, sino de mostrar algunos requisitos que serían deseables en una colaboración “ideal”, es decir, centrando nuestro interés en el aspecto interorganizacional y de interoperabilidad.

2.3.1 Requisitos de modelado

- **Modelo de la interoperabilidad/evolución.** Puesto que nuestro objetivo es proporcionar interoperabilidad y evolución a un nivel interorganizacional es necesario establecer un modelo para reflejar las capacidades de interoperabilidad y evolución que posee cada organización o proceso. La existencia de esta información permite a cualquier organización exportar, además de unos determinados servicios, la capacidad de interacción o evolución con las que dota a esos servicios (metainformación). La satisfacción de este requisito posibilita la comunicación entre procesos.
- **Modelado interorganizacional.** En la bibliografía se ha afirmado que el modelado organizacional no es necesario para el caso de la colaboración entre PSS [Ben-Shaul y Kaiser, 1998]. Esta afirmación es cierta si se utiliza una visión encapsulada de los subsistemas, pues no necesitamos conocer detalles internos de una organización para solicitar sus servicios, pero ¿tiene sentido conocer la estructura interorganizacional?. Asumiendo un entorno de colaboración plano, donde no existe distinción entre organizaciones ni importa su tipo podríamos rechazar la idea de que existiese un modelo interorganizacional. Sin embargo generalmente existirá una jerarquía o tipología en las organizaciones, que juegan ciertos papeles en la agrupación, lo que permite crear una analogía con las causas que llevan a la necesidad de crear un modelo organizacional en el caso intraorganizacional. El ejemplo más claro es el problema de la recursividad en la creación de empresas virtuales. Cuando se hace uso de esta metáfora cualquier empresa ofrece servicios que son utilizados sin necesidad de conocer la empresa o empresa virtual que lo oferta. Si se comete el error de utilizar un servicio de una empresa ofertado por sí misma podríamos estar utilizando una vía inadecuada para invocar un servicio, pues por ejemplo no necesitamos establecer ningún contrato para utilizar recursos de la propia empresa. En este caso, la existencia de un modelo

interorganizacional facilitará la detección de estos problemas y la optimización de procesos.

- **Riqueza semántica.** Este es un requisito cuyo cumplimiento es difícil de precisar. La semántica de un modelo o lenguaje de modelado de procesos debe ser rica y rigurosa, debe cubrir un amplio rango de semánticas así como soportar el razonamiento sobre procesos. Ejemplos prácticos son las redes de Petri [van der Aalst, 2000], que disponen de una semántica precisa y completa para especificar dependencias de control entre actividades, pero tendrían difícil aplicación por ejemplo a la hora de modelar recursos o metainformación.
- **Múltiples paradigmas.** Puesto que los procesos son entidades multi-faceta, un modelo que soporte todas sus facetas debe proporcionar varios tipos de semánticas interrelacionadas. Por ejemplo, para modelar un proceso podemos utilizar precondiciones y postcondiciones para modelar el flujo de control, pero también debemos modelar control reactivo (disparos) en respuesta a ciertos eventos. De igual manera podemos necesitar plasmar control proactivo para programar de forma imperativa procesos en que las actividades a desarrollar sean muy independientes. Por todo ello se debe soportar la coexistencia entre paradigmas o lenguajes de modelado [Conradi y Liu, 1995].

2.3.2 Requisitos de los sistemas de soporte

- **Transparencia:** las organizaciones que deseen formar parte de una comunidad no deben verse afectadas por su adhesión. Los trabajadores de dicha organización o los equipos software deben verse afectados en lo mínimo posible, siendo la transparencia total el objetivo final. La razón más importante para ello es que las comunidades pueden surgir y destruirse dinámicamente y la organización no debe sufrir, a nivel intraorganizacional, el dinamismo o las decisiones a nivel interorganizacional.
- **No discriminación y evolución organizacional.** Un error importante es pensar que todas las organizaciones tienen una capacidad tecnológica suficiente como para soportar procesos complejos, modelarlos, monitorizarlos, etc. Una aproximación flexible también debe tener en cuenta el diferente grado de desarrollo de cada una de las organizaciones participantes en un negocio. Existe todo un rango de organizaciones, desde las que tienen un sistema de soporte de procesos asentado hasta las que solo están compuestas por un grupo de trabajo, pasando, por ejemplo, por empresas con portales organizacionales. Para ver una posible clasificación y su utilización en pro de una mayor flexibilidad consulte [Pernici y Mecella, 2000]. Por supuesto, cualquier solución no solo debe adaptarse al grado de evolución, sino que debe posibilitar también cambios tecnológicos una vez operativa. Todo esto se resume en un requisito: no discriminación de organizaciones por razón de desarrollo. El hecho de que una organización no posea el avance de las otras con las que colabora no es razón para excluirla, sino que colaborará en el grado en el que le sea posible en la comunidad, soportándose un nivel de heterogeneidad mayor.
- **Monitorización:** al igual que cualquier proceso intraorganizacional, un proceso interorganizacional debe poder ser accedido y monitorizado en tiempo de ejecución (tiempo real). Además, si queremos poder realizar simulación antes de lanzar procesos interorganizacionales debemos conocer las características de los servicios, tanto dinámicas como estáticas. Para posibilitar la monitorización, las organizaciones deben

ofrecer servicios de monitorización asociados a los servicios objetivo. Estos servicios deben ofrecer información tanto de las características estáticas del servicio (modelo) como de los procesos en ejecución (instancias). Por ejemplo, se haría necesario conocer la tasa de servicio, retardo medio, tipo en cola asociado al servicio, etc. para una simulación de un proceso interorganizacional basada en teoría de colas. Los datos de tiempo de servicio durante la ejecución de una determinada instancia permitiría reajustar los cálculos de la simulación o comprobar el estado actual del proceso en ejecución.

- **Comunicación asíncrona/síncrona.** Con este requisito hacemos referencia a la necesidad de establecer mecanismos de interconexión flexibles. La distribución que debe soportarse nos lleva a la necesidad de un sistema capaz de enviar información desde un sistema a otro de forma síncrona o asíncrona.
- **Diferentes niveles de abstracción.** Al igual que en el caso intraorganizacional [Bichler et al, 1997] la posibilidad de definir niveles de abstracción permite definir procesos con diferente nivel de detalle. En el caso interorganizacional, la utilización de niveles de abstracción permite introducir la definición de procesos globales con el nivel de detalle deseado [van der Aalst y Weske, 2001], lo que contribuye a la flexibilidad permitiendo definir la colaboración basada en políticas y restricciones mínimas o en procesos definidos exhaustivamente.
- **Soporte de desviaciones.** Durante la ejecución de instancias de un proceso es común que se den situaciones inesperadas. Estas pueden producir cambios en el flujo de control que necesitan ser tratadas en tiempo de ejecución por el PSS. Puesto que esas situaciones también pueden darse en procesos interorganizacionales el sistema debe ser capaz de tratar estos problemas y ofrecer herramientas al usuario para gestionarlas.

2.3.3 Requisitos conjuntos de modelo y sistema

- **Reutilización.** Por un lado el modelo debe soportar la composición o cualquier otro mecanismo que permita tomar partes de un proceso y reutilizarlo en otro. En el sistema también debe existir una forma de almacenar los modelos o partes de ellos, es decir, un repositorio con información reutilizable. Varios autores han identificado este problema y han realizado propuestas para la introducción de la reutilización en el campo de soporte de procesos [Kradolfer, 2000] [Phios, 2001].
- **Adaptabilidad.** Con este requisito hacemos referencia a la necesidad ajustar el modelo de proceso o el PSS a las necesidades particulares de un dominio o caso concreto de aplicación. Por ejemplo podríamos necesitar representar tareas críticas si estamos realizando un proceso de planificación, o insertar propiedades estocásticas si estamos creando un modelo simulable, lo que también conllevaría la existencia de herramientas de simulación y ejecución apropiadas por parte del PSS.

2.3.4 Evolución en procesos interorganizacionales

Dada su importancia y por ser uno de los objetivos principales de este trabajo se dará un trato especial a este requisito. En un trabajo de investigación anterior [Martínez, 2000] se establecieron los requisitos y características de la evolución en flujos de trabajo. En este trabajo se plantea la evolución de procesos interorganizacionales como un requisito más para la

interoperabilidad entre PSS. Para clasificar el nuevo problema al que nos enfrentamos en este trabajo se clasificará la evolución en:

- **Evolución intraorganizacional:** la que surge de la necesidad de adaptación de una organización ante cambios externos. Implica cambios en los procesos para contemplar nuevos requisitos o incluir resultados derivadas de la mejora continuada de procesos. En [Martínez, 2000] se realizó un análisis de este tipo de evolución, cuyos resultados se aplican en el modelo o PSS de una organización particular.
- **Evolución interorganizacional:** la que surge de la necesidad de una comunidad de organizaciones de cambiar alguna política, método de coordinación o proceso global. Aunque básicamente el problema de la evolución es el mismo que en el caso intraorganizacional, llevar a cabo los cambios requiere no solo la modificación de un modelo o proceso sino que implica a su vez un problema de coordinación para informar/realizar los cambios en cada una de las organizaciones y sus modelos particulares. Se puede asumir que el ciclo de vida de un proceso interorganizacional es equivalente al ciclo de vida de un proceso intraorganizacional, es decir, el proceso puede encontrarse en estado suspendido, inactivo, terminado, etc. aunque la semántica es diferente.

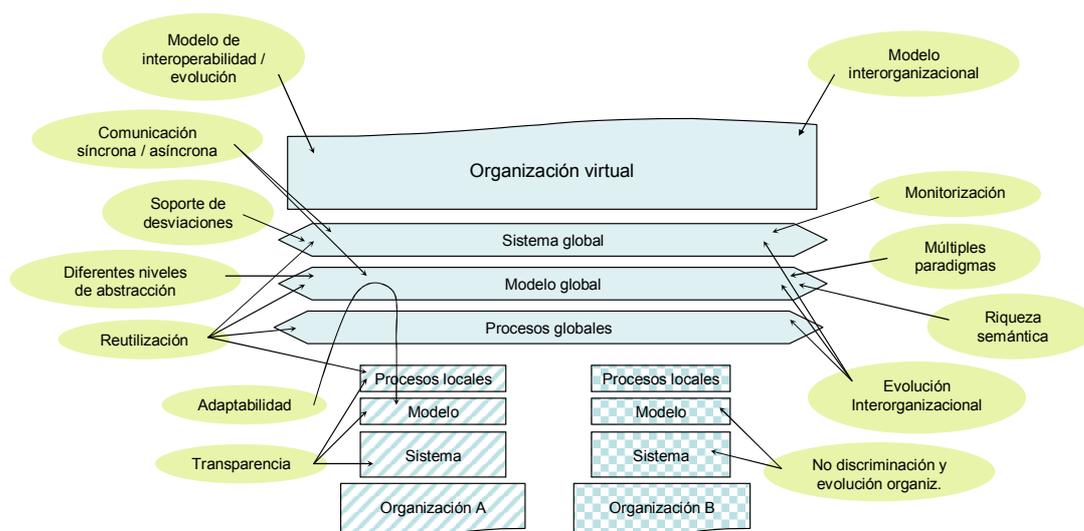


Figura 2. Esquema de los principales requisitos aplicables al modelado y ejecución de procesos interorganizacionales.

3 Modelos de procesos e interoperabilidad semántica

En este apartado vamos a realizar un estudio de los diferentes modelos para la representación de procesos. No se realizará un estudio exhaustivo de cada uno de los paradigmas o formalismos, ya que la gran cantidad de aproximaciones harían inviable el trabajo. Lo que se presentará será principalmente una clasificación de los diferentes formalismos y paradigmas más importantes, con el fin de poder analizar la potencia de las diferentes soluciones a la interoperabilidad.

Aunque puede interpretarse que el modelo de proceso debe contener únicamente las primitivas que permitan expresar el flujo de control, generalmente la realización de una tarea conlleva la necesidad de comunicar información (o productos), conocer los recursos sobre los que se ejecutará o la estructura organizativa que lo soporta. Por ello, tanto nuestro estudio debe tener en cuenta varias facetas del modelado de procesos. Así, Conradi y Liu [Conradi y Liu, 1995] establece la necesidad de diferentes modelos:

- **Modelo de actividad**, para expresar las actividades y sus dependencias.
- **Modelo del producto**, para expresar elementos pasivos manejados por el proceso.
- **Modelo organizacional**, para expresar los recursos, roles, restricciones globales, coordinación, etc. disponibles para ejecutar el proceso.

También establece la necesidad de un modelo de herramienta, que puede ser incluido dentro del modelo de actividad. Por otra parte clasifica los modelos de actividad según sus diferentes aproximaciones, clasificación que sigue teniendo vigencia hoy en día y que define cuatro categorías: descriptivos o basados en disparos ([Joeris y Herzog, 1999]), basados en redes ([Grefen et al., 2000][van der Aalst y ter Hofstede, 2000]), lenguajes de modelado imperativos [Hagen y Alonso, 1999] e híbridos [Jaccheri et al., 1992].

De forma similar otros autores [Weske, 1999][Kradolfer, 2000] han propuesto la utilización del concepto de aspecto para introducir cierta modularidad y ortogonalidad en el modelado de procesos. Un aspecto se define simplemente como un conjunto de propiedades de un proceso lo más independiente posible de otras. La división en aspectos más común es (para una discusión más detallada consultar [Martínez, 2000]):

- **Aspecto Funcional**: descripción de qué se debe hacer durante la ejecución de los procesos.
- **Aspecto de comportamiento**: descripción de las condiciones para que se ejecuten las diferentes partes de un proceso.
- **Aspecto de información**: descripción de los tipos de datos involucrados en el proceso y de su flujo entre tareas.
- **Aspecto organizacional**: descripción la organización y el entorno técnico donde ha de ejecutarse el proceso.
- **Aspecto operacional**: describe las herramientas que han de ser invocadas por el proceso, facilitando así la ejecución de tareas por parte de sistemas externos.

En [Conradi y Liu, 1995] se exponen las diferentes causas y niveles de interoperabilidad que hacen necesaria la coexistencia de distintos paradigmas lingüísticos para soportar tanto un lenguaje de definición de procesos general como sublenguajes adaptados a necesidades particulares. Aunque su trabajo se centra únicamente en los procesos de producción de software,

identifican claramente lo que es importante respecto de la elección de un lenguaje de modelado de procesos:

- **Estandarización:** se deben usar tecnologías estándares. La clave es el soporte de la reutilización.
- **Interoperabilidad:** hacer interaccionar los componentes de un PSS. La clave es la creación de sistemas abiertos.
- **Acoplamiento de PSS:** el uso de sistemas de soporte debe perturbar lo mínimo posible los procesos objetivo. La clave es ofrecer un soporte flexible al nivel apropiado (control, automatización, razonamiento, etc.).
- **Interfaz de usuario:** los agentes que intervienen en un proceso deben tener una visión comprensible del contexto de trabajo, con conexiones a los trabajadores en otras actividades. Claramente este punto viene impuesto por la alta necesidad de colaboración humana en los procesos de producción software.
- **Fácil evolución del modelo de proceso a nivel de usuario:** al igual que en el punto anterior se identifica como requisito que los agentes participantes en el proceso comprendan el modelo, se debe dar libertad: la flexibilidad suficiente como para que este se cambie cuando sea necesario.

A modo de resumen se puede consultar [Schlenoff et al., 1996], donde se realiza un análisis de los requisitos necesarios para representar procesos. Se realiza una exploración de las aplicaciones que necesitan representar procesos para determinar los requisitos de cada uno. Para agrupar y poder abordar con cierta lógica el gran conjunto de requisitos se definen cuatro categorías: Núcleo, Núcleo Externo, Extensión y Aplicación. En la Tabla 1 se muestra la descripción y ejemplos de algunos de los requisitos de cada categoría. Además, los requisitos se dividen en representacionales y funcionales, es decir, requisitos que inciden sobre la expresividad del modelo y requisitos que inciden sobre la funcionalidad del PSS respectivamente.

Podemos utilizar esta categorización para analizar el grado de interoperabilidad al que un sistema puede o quiere llegar. En función de los requisitos que soporte, un sistema tendrá más o menos potencia expresiva, y dependiendo de esta potencia podrá representar y manejar uno u otro tipo de procesos. En la Figura 3 se presentan gráficamente las categorías de requisitos definidas en [Schlenoff et al., 1996]. Puede observarse un ejemplo de implementación de un sistema como una zona sombreada del conjunto de sectores en que se dividen los requisitos. Esta visión contrasta con la idea de interoperabilidad mostrada hasta el momento. Generalmente el problema de la interoperabilidad se ha afrontado desde el punto de vista de un único dominio, pero ¿por qué no han de colaborar una empresa de desarrollo de software con otra que haga comercio electrónico? En este caso no solamente aparece el problema de la interoperabilidad entre procesos de un mismo dominio, sino que también aparece el de interoperabilidad entre representaciones muy diferentes de procesos. Volviendo al ejemplo de la Figura 3, la categorización del espacio de requisitos nos permite definir los dominios y extensiones que cubre un sistema. Así, un sistema capaz de representar los procesos de tipo flujo de trabajo y de reingeniería del negocio será capaz de actuar como intermediario entre sistemas de estos dos dominios. De igual manera si un sistema o modelo no contempla las extensiones presentes en algún dominio será incapaz de representar los posibles procesos definidos en ese dominio. Una conclusión más, y quizá la más importante, a la que podemos llegar es a la imposibilidad de crear a priori un PSS o sistema de interoperabilidad universal. Al no conocer todos los requisitos de todos los dominios actuales o futuros no es posible definir un modelo o sistema

que los satisfaga, debiendo añadir algún mecanismo de extensión como garantía para la adaptación a nuevos requisitos.

Tabla 1. Resumen de requisitos para la representación de procesos según [Schlenoff et al., 1996].

Categoría	Descripción	Requisitos del modelo	Requisitos funcionales
Núcleo	Requisitos básicos inherentes a todo proceso. Se consideran tan comunes que cualquier aplicación, explícita o implícitamente los utiliza.	(12 requisitos) Datos de coste. Nivel de esfuerzo. Características de productos. Recursos. Necesidades de recursos para una tarea. Grupos simples de tareas. Capacidades/características simples de recursos. Secuencias simples. Características y representación de tareas simples. Duración de tareas. Ejecutor de tareas.	(3 requisitos) Extensibilidad. Asignación de recursos para una o más tareas. Precedencia simple.
Núcleo externo	Requisitos dominantes, pero no esenciales, comunes a la mayoría de aplicaciones.	(29 requisitos) Abstracción. Secuencias complejas (alternativas, concurrencia, paralelismo, etc.) Restricciones (temporales, pre y post-condiciones, existencia de estados, físicas, etc.) [...]	(12 requisitos) Inserción de anexos o hitos. Precedencia compleja Gestión de tiempos límite. Gestión de excepciones y recuperación. [...]
Extensión	Grupo de requisitos relacionados que juntos proporcionan funcionalidad añadida. Se definen seis extensiones: <ul style="list-style-type: none"> • Administrativa/Negocios • Planificación/Calidad/Análisis • Tiempo Real/Dinamismo. • Intención del proceso. • Agregados Recursos/Procesos • Estocásticos/Estadísticos 	Particular de cada extensión. Administrativa: Prácticas de negocio, reglas y restricciones. Información de gestión de configuraciones y procesos. [...] Planificación: Características del análisis. Tareas críticas. [...] [...]	Particular de cada extensión. Planificación: Coexistencia de planes y resolución de conflictos. Modificación dinámica del modelo. Optimización [...] Tiempo real: Modificación dinámica del modelo. Notificación de eventos y señalización. [...] [...]
Aplicación	Requisitos relevantes únicamente a aplicaciones concretas. Dentro de esta categoría estarían por ejemplo las aplicaciones de flujos de trabajo o las de reingeniería del negocio.	Particular de cada extensión, por ejemplo en flujos de trabajo se toman la mayoría de requisitos funcionales del núcleo externo, requisitos de las extensiones administrativa, de planificación y de tiempo real. Además se añaden dos requisitos propios: capacidades de las herramientas invocadas y ejecución manual/automática de actividades.	

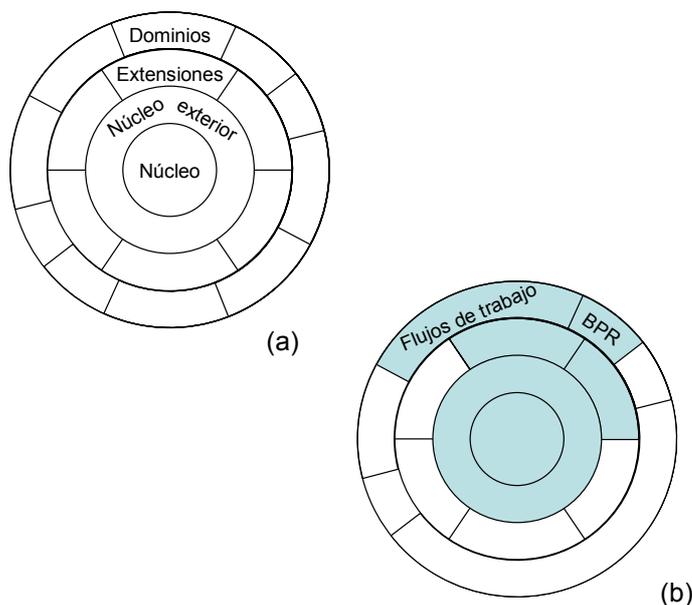


Figura 3. Ejemplo de representación gráfica de grados de implementación de interoperabilidad. Los círculos agrupan requisitos a diferentes niveles de importancia mientras que los sectores los subdividen dependiendo de su campo de aplicación. Esquema general (a) y ejemplo de implementación de interoperabilidad entre sistemas de flujo de trabajo y aplicaciones de reingeniería (b).

3.1 Principales modelos propuestos

Existe un gran número de lenguajes de modelado de procesos, con sus respectivos modelos y herramientas asociadas. Carlsen [Carlsen, 1997] identifica al menos cinco paradigmas para la construcción de procesos, considerando la tradicional definición IPO (Input Process Output) y Language Action como las más relevantes (estos paradigmas se estudiarán con detalle más adelante). Realiza también una comparación de la tecnología e identifica como uno de los retos de investigación para la tecnología de soporte de flujos de trabajo proporcionar una mayor flexibilidad al manejo de modelos conceptuales y un incremento de la comprensibilidad.

Podemos establecer una clasificación inicial de los modelos de procesos en función de los formalismos que los sustentan. Esta clasificación se debe interpretar desde un punto de vista de bajo nivel, pues generalmente un formalismo no contempla las facilidades para estructurar, reutilizar o hacer evolucionar los procesos descritos. De esta forma el modelado de procesos puede clasificarse en cinco grupos generales:

- **Formalismos no ejecutables.** Son aquellos en los que el modelado es informal y con el único fin de documentar o describir procesos. Ejemplos típicos son: los diagramas de estado o de actividad de UML [OMG, 2000] utilizados o extendidos para definir procesos, los diagramas IDEF0 [IDEF0, 1993], algunos modelos utilizados en BPR [isModeler, 2001], los diagramas de flujo de datos, etc.

- **Basados en Redes.** A este grupo pertenecen modelos tan importantes como las redes de Petri [van der Aalst, 1998] [Raposo et al., 2000] o las máquinas de estados [Harel et al., 1990]. Son quizá los más utilizados debido a su fácil representación gráfica y comprensión por parte de usuarios no especializados, ya que asemejan formalismos no ejecutables pero con una semántica definida y sin ambigüedad.
- **Declarativos.** Bajo este grupo englobamos aquellos formalismos que en vez de especificar imperativamente las actividades y sus dependencias, declaran restricciones sobre cómo debe realizarse el proceso. Generalmente se utiliza algún tipo de lógica [Kifer, 1996], permitiendo definir sistemas fácilmente modificables al poder añadir o quitar restricciones al modelo declarativo de proceso [Hull et al., 1999]. En este grupo se incluirían también los modelos basados en reglas [Joeris y Herzog, 1998], disparos, o gestión del conocimiento en general.
- **Imperativos.** Los procesos expresados en un lenguaje imperativo, ya sea específico de modelado de procesos o no, caerían en este grupo. Ejemplos de estos formalismos son los lenguajes de modelado de procesos (PML) como el propuesto por [Keane et al., 1994], el OCR propuesto en [Alonso y Hagen, 1997] o extensiones a lenguajes clásicos como Ada o C.
- **Híbridos.** Los grupos anteriores no son excluyentes. Pueden definirse sistemas en los que se haga uso de combinaciones de formalismos con el fin de aumentar su potencia [Kammer, 2000]. Por ejemplo podría utilizarse algún tipo de red para modelar procesos, añadiendo algún método declarativo para expresar restricciones globales o formalismos no ejecutables como vistas para facilitar la comprensibilidad de los procesos.

3.1.1 Modelo general propuesto por la WfMC

La Workflow Management Coalition (WfMC, o simplemente coalición), fundada en 1993, es una organización sin ánimo de lucro e internacional constituida por vendedores de sistemas de flujos de trabajo, clientes y usuarios. Su misión es establecer estándares para facilitar la interoperabilidad y conectividad entre productos de flujo de trabajo. Desde el punto de vista de la coalición un sistema de gestión de flujos de trabajo proporciona la automatización de procesos de negocio, gestionando las actividades e invocando los recursos necesarios asociados a los pasos definidos en el proceso. Al nivel más alto el sistema puede caracterizarse por soportar tres áreas funcionales:

- **Funciones de tiempo de construcción:** definición y posiblemente modelado del proceso y sus actividades.
- **Funciones de control en tiempo de ejecución:** gestión del proceso en un entorno operacional y secuenciación de las actividades de cada proceso.
- **Funciones de interacción en tiempo de ejecución:** con humanos o herramientas para realizar el proceso.

Define un marco general para la especificación de estándares de flujo de trabajo describiendo la terminología común y con ella un modelo de proceso de referencia. En base a este modelo se describen cinco categorías de interoperabilidad y comunicación estándar que permiten la coexistencia e interoperabilidad de los distintos sistemas de soporte de flujos de trabajo. En concreto, el modelo de proceso (flujos de trabajo) propuesto contiene los siguientes elementos principales:

- **Flujo de trabajo:** la automatización de un proceso de negocio, por completo o en parte, durante el que los documentos, información o tareas pasan de un participante a otro para actuar de acuerdo a un conjunto de reglas procedurales.
- **Proceso de negocio (proceso):** conjunto de uno o más procedimientos o actividades unidos que colectivamente realizan un objetivo del negocio o política global. Generalmente enmarcado en el contexto de estructura organizacional que define los roles funcionales y las relaciones.
- **Definición de un procesos:** la representación de un proceso de negocio. Es una red de actividades y sus relaciones, criterios de comienzo y fin, información sobre actividades, etc.
- **Actividad:** descripción de una parte de trabajo que forma un paso lógico dentro de un proceso. Una actividad puede ser manual (actividad manual) o automática (actividad automática o actividad de flujo de trabajo).
- **Instancia:** representación de una ejecución de un proceso o actividad dentro de un proceso, incluyendo sus datos asociados.
- **Participante en el flujo de trabajo:** recurso que realiza el trabajo representado por una instancia de actividad.
- **Elemento de trabajo:** representación del trabajo a procesar en el contexto de una actividad dentro de una instancia de proceso.
- **Lista de trabajo:** lista de elementos de trabajo asociados con un participante en el flujo de trabajo.

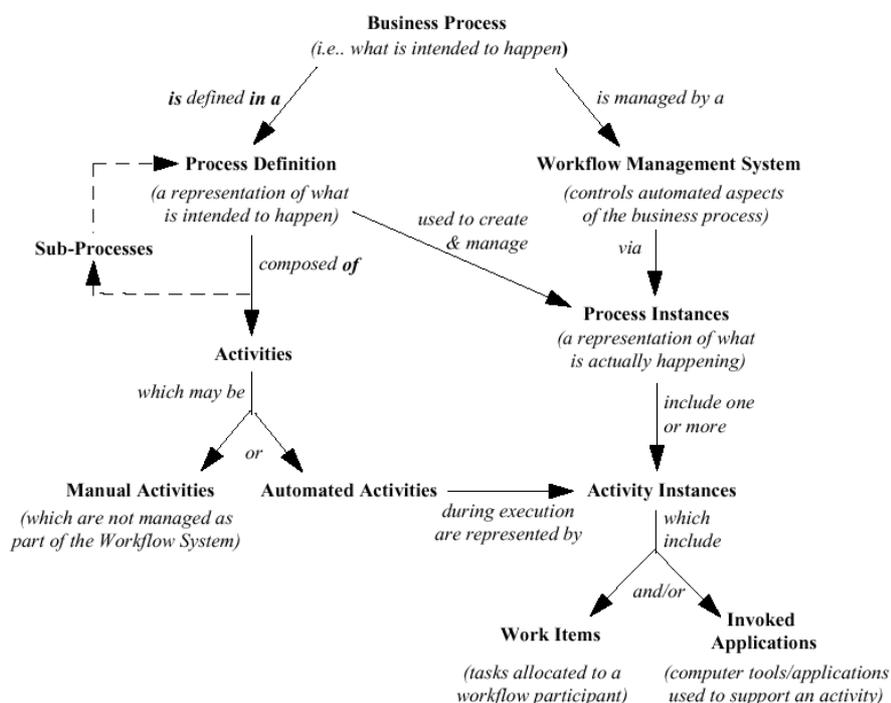


Figura 4. Terminología utilizada por la WfMC y sus relaciones [WFMC, 1999].

3.1.2 Modelo IPO

El paradigma IPO (Input-Process-Output) es predominante entre los productos comerciales de soporte de procesos. Se basa en el concepto simple de que un proceso es una caja negra que transforma una entrada en una salida. Generalmente se añade también algún control, al modo en que se define en los diagramas IDEF0. Los procesos (o actividades del proceso) pueden encadenarse juntos mediante flujos de información o materiales, y al mismo tiempo puede descomponerse en subprocesos. A través del modelado del flujo de información, materiales y control se adapta bien al modelado de procesos estructurados. Un beneficio es la descomposición jerárquica, que representa la posibilidad de ver los procesos a diferentes niveles de detalle o abstracción. El mayor inconveniente es el soporte inadecuado de la interacción y comunicación entre actores (humanos) que obliga a crear nociones como las “entidades externas”.

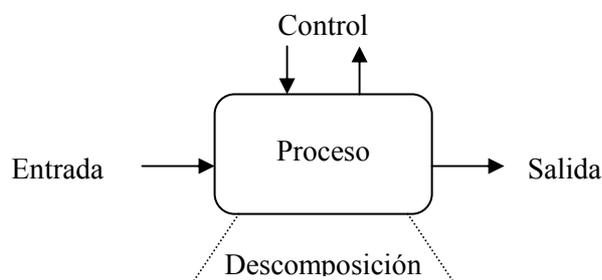


Figura 5. Visión general de un proceso según el paradigma IPO.

3.1.3 Modelo *Language Action*

El paradigma *Language Action* es una aproximación relativamente antigua que se ha utilizado para el modelado de sistemas de información en general y en particular para comercio electrónico [Covington, 1997], flujos de trabajo [Krogstie y Carlsen, 1996] o reingeniería de procesos [De Michelis, 1995]. Está basado en la teoría *speech-act* (acto de habla) [Searle y Vanderveken, 1985] y asume que el lenguaje no se usa simplemente para comunicar información sino que también se puede usar para planificar y coordinar acciones. Winograd y Flores [Winograd y Flores, 1986] propusieron que algunos tipos de conversaciones podían formar la base para el diseño de sistemas.

Para este trabajo el principal interés se centra en el sistema *ActionWorkflow* [Metro], donde se utiliza el paradigma *Language Action* para representar procesos de negocio como conversaciones entre dos roles. El análisis y modelado de procesos se realiza capturando las interacciones necesarias entre los actores del proceso para alcanzar el objetivo deseado. El modelo de proceso está orientado al usuario en el sentido de que los integrantes del negocio son capaces de entender y construir modelos (que posteriormente pueden ser revisados por expertos). El constructor principal en el modelo es el *ActionWorkflow* (o diálogo) que define una conversación con una estructura de bloques consistente en cuatro fases:

- **Apertura.** Cuadrante superior izquierdo (véase la Figura 6). Iniciado por una petición del cliente u oferta del proveedor.
- **Acuerdo.** Cuadrante superior derecho. El cliente y proveedor acuerdan unas condiciones de satisfacción para la instancia de diálogo.

- **Realización.** Cuadrante inferior derecho. Es donde se realiza la acción real, donde se realiza el trabajo deseado y se informa de la terminación.
- **Aceptación.** Cuadrante inferior izquierdo. En esta fase el cliente verifica el resultado y declara la aceptación/rechazo de acuerdo a las condiciones establecidas. Esta fase cierra el diálogo.

El objetivo de este paradigma es la construcción de redes de “conversaciones” donde los elementos de conexión son los roles de cliente y proveedor. Al contrario que en el paradigma IPO, donde la naturaleza del trabajo es procesar materiales o información, el objetivo es conducir el negocio y cumplir con las peticiones. Se adapta mejor a entornos cambiantes al proporcionar plantillas para acciones basadas en las mejores prácticas y no en la definición de procesos “acordados”.

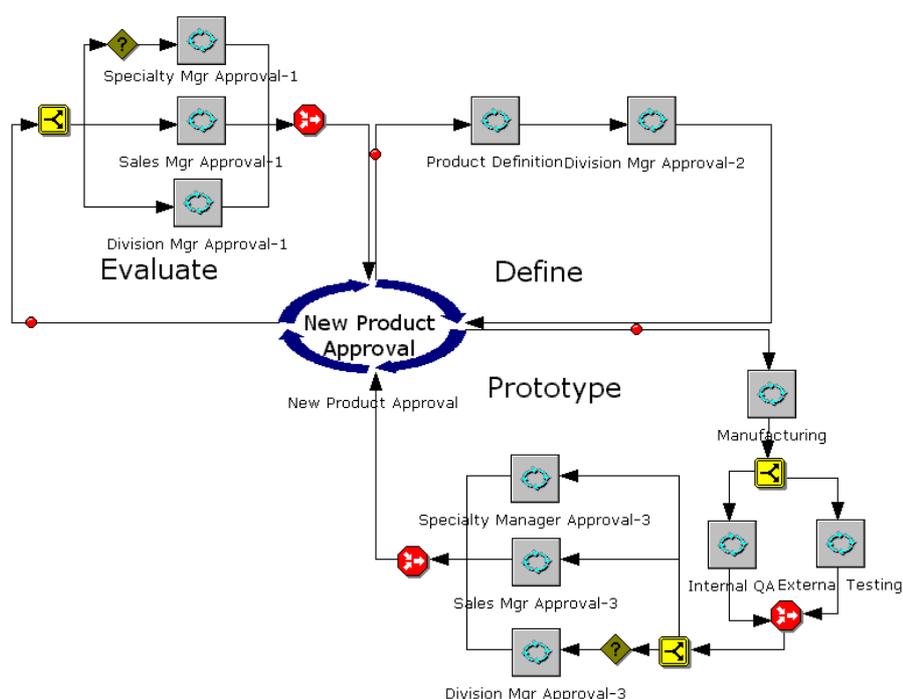


Figura 6. Ejemplo de proceso definido usando el paradigma Language Action. New Product Approval es el diálogo ActionWorkflow (ActionWorks® Process Builder).

3.2 Estándares de definición e intercambio de información sobre procesos

3.2.1 WfMC interfaz de interoperabilidad y Wf-XML

La WfMC define un modelo de referencia (véase el apartado 4.2.1) donde se definen los bloques funcionales principales de un sistema de soporte de flujos de trabajo. Entre algunos de estos bloques se definen interfaces que dotan de modularidad al sistema, permitiendo abstraer detalles de implementación o funcionamiento de cada uno de esos bloques funcionales. Aquí analizaremos el interfaz para la comunicación entre motores de ejecución, es decir, el mecanismo de interoperabilidad propuesto por la WfMC. La especificación abstracta del

interfaz puede encontrarse en [WFMC, 1999b], donde se define la funcionalidad que deben soportar los productos que quieran interoperar mediante estándares de la WfMC. Puesto que lo que nos interesa estudiar son las diferencias entre modelos utilizaremos una especificación concreta, la implementación utilizando XML de la especificación abstracta [WFMC, 2000]. Aunque la WfMC está trabajando en ello desde principios de 1999, y se dispone únicamente de una propuesta inicial, este estándar representa una evolución de los estándares existentes, pues ha utilizado trabajo originado en la OMG (Object Management Group) y las propuestas iniciales de la IETF (Internet Engineering Task Force) como SWAP (Simple Workflow Access Protocol). Wf-XML es un derivado de la especificación abstracta de interoperabilidad (interfaz 5) que puede trabajar con HTTP, correo electrónico, TCP/IP y cualquier otro mecanismo de transporte. Incluye una definición de un DTD (Document Type Definition) básico con la codificación de los mensajes para soportar la interoperabilidad.

Tabla 2. Elementos del lenguaje Wf-XML

Elemento	Descripción
Espacio de nombres	La interoperabilidad basada en XML permite interactuar con varios vocabularios, mezclando elementos como sea necesario. Esta característica es clave para Wf-XML ya que cada sistema puede poseer datos específicos. El espacio de nombres permitirá distinguir entre elementos definidos por la especificación o por terceros.
Tipos de datos	Se proporcionan tipos de datos como binario, booleano, entero, flotante, fecha, URI, etc.
Estructura de los mensajes	Cada mensaje tiene un número de versión, una sección opcional de información de transporte, una cabecera y un cuerpo. La sección de transporte puede contener información de la implementación particular, la cabecera contiene información sobre enrutamiento y preprocesamiento mientras que el cuerpo contiene información específica de la operación.
Tipos de mensaje	Los mensajes son de tipo "Respuesta" o "Petición". Una petición inicia una operación en una fuente remota o proporciona una entrada a un recurso. Las respuestas sirven para enviar resultados.
Cabecera y cuerpo de los mensajes	Una cabecera contiene información sobre: si el mensaje es de petición o respuesta, si requiere respuesta, el lenguaje de respuesta y una clave identificando el destino (URI). Un cuerpo contiene información específica para cada operación: <ul style="list-style-type: none"> • Petición: parámetros de la petición. • Respuesta: parámetros devueltos por la operación
Representación del contexto de procesos y resultados	La información asociada a una instancia concreta de proceso es el contexto en un mensaje de petición y el resultado en un mensaje de respuesta. Puesto que esta información ha de ser accesible se declara una sección dentro del cuerpo del mensaje que contiene esta información. Puesto que esta información depende de la aplicación no está normalizada y puede contener cualquier tipo de elemento.
Estado de los procesos	Para almacenar el estado en un instante concreto la WfMC define un conjunto de estados posibles. Los estados posibles son abierto y cerrado, que a su vez pueden dividirse en ejecutándose o no ejecutándose y completado o completado anormal (terminado o abortado) respectivamente. Esta información se codifica como un elemento más en los cuerpos de mensajes.
Manejo de errores	Se identifican tipos de excepciones. Se pueden definir elementos de tipo <i>Exception</i> que almacenan esta información.

Puede usarse para tareas de interoperabilidad básicas: flujos de trabajo encadenados y anidados, comunicación síncrona y asíncrona. El estándar define los elementos mostrados en la Tabla 2.

El lenguaje se extiende con operaciones sobre flujos de trabajo. Las operaciones más importantes que se pueden realizar se muestran en la Tabla 3. Estas son las únicas operaciones normalizadas, por lo que el estándar proporciona un nivel de interacción muy restringido, pero serán extendidas para incluir operaciones de otros interfaces de la WfMC para formar una especificación completa para todas las funciones de un flujo de trabajo.

Tabla 3. Operaciones definidas para definición, gestión de instancias y observación

Grupo	Operaciones
Definición de procesos	<i>CreateProcessInstance</i> : instancia una definición de proceso conocida.
Instancias de procesos	<i>GetProcessInstanceData</i> : devuelve los valores de las propiedades definidas para una instancia de proceso. <i>ChangeProcessInstanceState</i> : modifica el estado de una instancia, por ejemplo de abierto.ejecución a abierto.noejecución.suspendido.
Observación	<i>ProcessInstanceStateChanged</i> : utilizada para observar cambios de estado a cerrado.completado y cerrado.completadoanormalmente.

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
  <WfMessageHeader>
    <Request ResponseRequired="Yes"/>
    <Key>http://www.XYZcompany.com/wfprocess/foo</Key>
  </WfMessageHeader>
  <WfMessageBody>
    <CreateProcessInstance.Request StartImmediately="true">
      <ObserverKey>http://www.ABCcompany.com/wfprocessor</ObserverKey>
      <ContextData>
        <Vehicle>
          <VehicleType>Car</VehicleType>
          <Specification>
            <Manufacturer>Mercedes</Manufacturer>
            <Model>450SL</Model>
          </Specification>
        </Vehicle>
        <Customer>John Doe</Customer>
      </ContextData>
    </CreateProcessInstance.Request>
  </WfMessageBody>
</WfMessage>
```

Figura 7. Ejemplo de mensaje codificado en Wf-XML para la creación de una instancia de proceso.

Se puede observar que las operaciones están definidas a nivel de protocolo, pues aunque incluyan información de la aplicación incluyen estos elementos sin normalización. El protocolo puede servirnos como medio de comunicación entre sistemas, podríamos definirlo a nivel de sesión en términos de arquitectura de red OSI, pero no a nivel de aplicación. Los modelos de proceso deben acordarse y ser conocidos por cada una de las partes, estableciendo la

codificación (extensión al esquema definido por el estándar) que se realizará de los procesos expresados en este modelo (contextos en el cuerpo de los mensajes).

3.2.2 SWAP

SWAP (Simple Workflow Access Protocol) fue uno de los primeros intentos por estandarizar los mecanismos de comunicación entre motores de flujos de trabajo [IETF, 1998]. Aunque éste trabajo no ha tenido continuidad, viéndose rebasado por otros estándares como WfXML, se incluirá una pequeña descripción por haber surgido también del trabajo realizado por la WfMC y a su vez haber servido a la coalición como prueba y reentrada para sus trabajos posteriores.

SWAP ofrece la posibilidad de comenzar una instancia de un servicio asíncrono genérico, monitorizar, controlarlo y ser notificado cuando se complete. El servicio (o proceso en nuestra terminología) puede realizar cualquier tarea, por lo que se relaciona con la tecnología de flujos de trabajo como mecanismo de comunicación, aunque no esté diseñado expresamente para este tipo de procesos. El estándar está muy relacionado con el protocolo HTTP, utilizando peticiones HTTP a una URI con la definición del proceso para generar una instancia (amplia las operaciones HTTP). El estado de un proceso puede obtenerse por medio de otras peticiones HTTP, así como realizar otras operaciones sobre él, como el envío o recepción de parámetros o datos de salida respectivamente.

Una característica importante de SWAP es la definición de tres interfaces fundamentales que forman el corazón del modelo de recurso que propone. Estos son:

- *ProcessInstance*: el ejecutor real del servicio. Cada vez que se invoca el servicio se crea una nueva instancia. Sólo puede usarse una vez y estará en estado comenzado, pausado, recuperado o terminado.
- *ProcessDefinicion*: puesto que cada invocación crea una instancia, con un URI diferente, se necesita un modelo para crearla. Para cada servicio se proporciona una URI única con la definición del proceso.
- *Observer*: las instancias conocen poco acerca de quién las invoca. Para poder comunicarse se le proporciona una referencia a un recurso observador (URI). Pueden existir tantos observadores como se deseen, observando no solo la terminación sino también cualquier cambio en el estado o los datos.

```

<swap>
  <interfaces>ProcessInstance</interfaces>
  <key>http://myServer/app1?proc=889</key>
  <validStates>
    <li>open.notRunning</li>
    <li>open.running</li>
  </validStates>
  <state>open.notRunning</state>
  <data>
    <d:city>San Francisco</d:city>
    <d:state>California</d:state>
  </data>

```

Figura 8. Ejemplo de codificación de una petición de instanciación.

Como puede observarse SWAP representó un gran paso hacia delante, pero su intencionalidad no incluía medios para describir procesos.

3.2.3 BPML y BPQL

La Iniciativa para la Gestión de Procesos de Negocio (Business Process Management Initiative, BPMI) [BPMI] es una organización sin ánimo de lucro con la misión de promover el desarrollo y uso de la gestión de procesos de negocio a través del establecimiento de estándares para el diseño, distribución, ejecución, mantenimiento y optimización de procesos.

BPMI define especificaciones abiertas como el Business Process Modeling Language (BPML) y el Business Process Query Language (BPQL) que posibilitan la gestión de procesos de negocio basada en estándares. BPMI considera que un proceso de e-Business está conducido por dos organizaciones y compuesto por tres partes: un interfaz público y dos implementaciones privadas (una para cada organización). El interfaz público es común y está soportado por protocolos como ebXML, RosettaNet y BizTalk. Las implementaciones privadas son específicas para cada participante y están descritas en cualquier lenguaje ejecutable, como BPML. Una vez desarrollada, la implementación necesita ser ejecutada, para lo cual se define BPQL, un interfaz para gestión de la distribución y ejecución de procesos.

BPML [BPMI, 2001] es un metalenguaje para el modelado de procesos de negocio. Proporciona un modelo de ejecución abstracto para procesos colaborativos y transaccionales basado en el concepto de máquina de estados finita transaccional. BPML considera un proceso como un conjunto compuesto por una interfaz pública y varias implementaciones, lo que posibilita publicar el proceso en XML independientemente de su implementación privada. De igual modo en que los documentos XML están expresados en un esquema XML, los procesos BPML pueden describirse basándose en el esquema XML de BPML. Representa los procesos como una mezcla de flujo de control, flujo de datos y flujo de eventos, añadiendo un diseño ortogonal de reglas de negocio, roles de seguridad y contextos transaccionales. Ofrece también soporte explícito para transacciones distribuidas sincrónicas y asíncronas.

Si BPML es el lenguaje de definición, BPQL es el lenguaje de gestión [BPMI, 2001b]. Define un interfaz estándar para los PSS existentes y venideros. Permite administrar el PSS y consultar las instancias en ejecución. Es un interfaz a una infraestructura de gestión de procesos de negocio que incluye un servidor de procesos y un repositorio para la distribución de procesos. El interfaz BPQL a un PSS posibilita consultar el estado y controlar la ejecución de las instancias de proceso en ese servidor. Este interfaz está basado en SOAP (Simple Object Access Protocol) [Box et al., 2000]. El interfaz BPQL al repositorio de procesos facilita al analista del negocio la gestión y distribución de procesos. Los procesos gestionados por el repositorio a través del interfaz BPQL pueden exportarse como servicios UDDI [UDDI] con el propósito de poder registrar, publicar o descubrir procesos.

Uno de los inconvenientes de estos estándares es su etapa de desarrollo actual, encontrándose en fase de borrador (BPML) o incluso en fase de discusión inicial (BPQL).

3.2.4 PIF

La especificación más reciente de Process Interchange Format (PIF) [Lee et al., 1998] tiene como objetivo desarrollar un formato de intercambio que facilite el uso de descripciones de procesos entre PSS como sistemas de flujos de trabajo, diagramadores, sistemas de simulación, planificadores y repositorios de procesos en general. No se trata por tanto de un

estándar de comunicación sino de un esfuerzo por soportar el intercambio de descripciones de procesos entre diferentes representaciones de procesos. El marco de PIF persigue el soporte de la traducción de procesos tal que:

- Una descripción de proceso pueda traducirse automáticamente entre PIF y otras representaciones con la mínima pérdida de significado. Si esta traducción no se puede realizar automáticamente el esfuerzo humano debe ser mínimo.
- Si un traductor no puede traducir parte de un proceso PIF, debe:
 - Traducir la mayor parte de descripción posible (y no informar simplemente de un error, por ejemplo).
 - Representar cualquier parte no traducible como tal y presentarla de modo que una persona pueda entenderla y completar la traducción manualmente si se desea.
 - Preservar cualquier parte no interpretable de forma que el traductor pueda añadirla a PIF cuando se realice una traducción inversa.

PIF es una de las especificaciones más antiguas (el proyecto PIF comenzó a finales de 1993), pero a pesar de ser simple y comprensible no parece haber tenido el éxito esperado por tener una sintaxis basada en KIF/LiSP. Precisamente es KIF (Knowledge Intergchange Format), un lenguaje diseñado para facilitar el intercambio de conocimiento, el que dota a PIF de una potencia superior a otras especificaciones. KIF proporciona semántica declarativa formalmente definida, potencia expresiva y una estructura que facilita traducciones semiautomáticas.

El proyecto PIF se ha mezclado con el proyecto PSL (Process Specification Language) del NIST. El PIF CORE y sus extensiones se han incorporado al PSL CORE y sus extensiones, por lo que en el siguiente apartado se introducirá éste proyecto

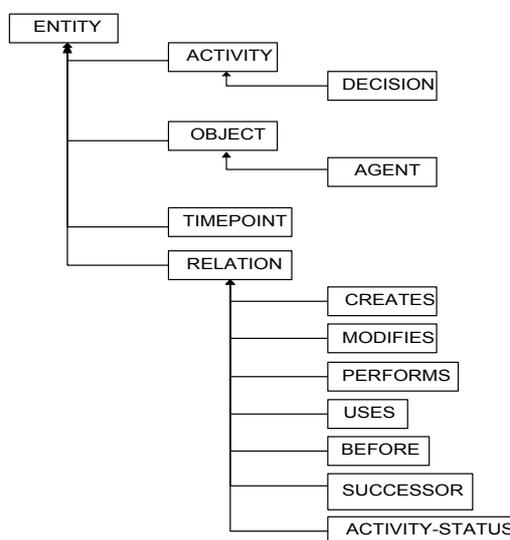


Figura 9. Ontología general de PIF. Todo es una ENTIDAD, que se especializa en ACTIVIDAD, OBJETO, PUNTO TEMPORAL y RELACION. Un proceso es un conjunto de ACTIVIDADES conectadas entre ellas o con otros OBJETOS mediante RELACIONES con PUNTOS TEMPORALES.

3.2.5 The Process Specification Language Project, PSL

Originalmente el proyecto Process Specification Language (PSL) [PSL] se centró en el desarrollo de una representación neutral del intercambio de procesos de manufacturación, mientras que PIF se centró en los procesos de negocio. Rápidamente se detectó que los conceptos eran similares, por lo que PSL y PIF están en proceso de fusión. PSL pretende incluir un lenguaje para la comunicación entre aplicaciones de manufacturación, pero los esfuerzos actuales de PSL se centran en el desarrollo del modelo de datos (también referenciado como la ontología) que define el significado de los términos del lenguaje relacionados con procesos. La ontología PSL fue creada por personas representativas del mundo de los flujos de trabajo, la manufactura, los negocios y la planificación militar. Se alcanzó un consenso para definir el contenido del “PSL Core” y se hizo trabajo adicional para extender esos conceptos a varios tipos de procesos. La ontología de PSL está organizada como una serie de módulos, todos ellos construidos a partir del PSL Core. Podemos decir, por tanto, que PSL Core captura conceptos de alto nivel inherentes a la especificación de procesos. Cada módulo refina PSL Core, pudiendo construirse también en base a otros módulos, lo que hace que la ontología PSL sea una telaraña de módulos todos descendientes de PSL Core.

Esta especificación formal, basada en ontologías, de la semántica del modelo de proceso permite un fundamento riguroso para el diseño, análisis y ejecución de procesos. Una ontología es una descripción formal de las entidades en un dominio dado: sus propiedades, las relaciones en que participan, las restricciones a las que están sujetos y los patrones de comportamiento que exhiben [Uschold y Gruninger, 1996]. Todo ello proporciona una terminología común que ayuda a capturar distinciones entre conceptos en diferentes dominios, lo que ayuda en el proceso de traducción.

```

Primitive Lexicon:
  Relations:
    (object ?x)
    (activity ?a)
    (activity_occurrence ?occ)
    (timepoint ?t)
    (before ?t1 ?t2)
    (occurrence_of ?occ ?a)
    (participates_in ?x ?a ?t)
  Functions:
    (beginof ?occ)
    (endof ?occ)
  Constants:
    inf+
    inf-
Defined Lexicon:
  Relations:
    (between ?t1 ?t2 ?t3)
    (beforeEq ?t1 ?t2)
    (betweenEq ?t1 ?t2 ?t3)
    (exists_at ?x ?t)
    (is_occurring_at ?a ?t)
  Axioms
    Axiom 1 The before relation only holds between timepoints.
    (forall (?t1 ?t2)
      (=> (before ?t1 ?t2)
        (and (timepoint ?t1)
              (timepoint ?t2))))
    Axiom 2 The before relation is a total ordering.
    (forall (?t1 ?t2)
      (=> (and (timepoint ?t1)
                (timepoint ?t2))
        (or (= ?t1 ?t2)
            (before ?t1 ?t2)
            (before ?t2 ?t1))))
  [...]
  Supporting Definitions
    Definition 1 Timepoint ?t2 is between timepoints ?t1 and ?t3 if and
    only if ?t1 is before ?t2 and ?t2 is before ?t3.
    (defrelation between (?t1 ?t2 ?t3) :=
      (and (before ?t1 ?t2) (before ?t2 ?t3)))
  [...]

```

Figura 10. Extracto de la ontología PSL Core expresada en KIF.

El reto es construir un marco donde el significado de la terminología de la ontologías sea explícito. Cualquier intuición implícita es una posible fuente de ambigüedad y confusión. Para la ontología de PSL debemos proporcionar una caracterización matemática rigurosa así como expresiones precisas de las propiedades básicas de la información contenida en el lenguaje. Para proporcionar la ontología se especifican tres nociones:

- **El lenguaje:** un léxico (un conjunto de símbolos) y una gramática (una especificación de cómo se pueden combinar esos símbolos para crear fórmulas bien formadas).
- **Una teoría de modelo.** La teoría del modelo de PSL proporciona una caracterización matemática rigurosa y abstracta de la semántica o el significado del lenguaje PSL. Esta representación es generalmente un conjunto con una estructura adicional (por ejemplo un orden parcial o un espacio vectorial). La teoría del modelo define el significado de la terminología y la noción de verdad para las sentencias del lenguaje en términos de este

modelo. El objetivo es identificar cada concepto del lenguaje con un elemento de alguna estructura matemática.

- **Una teoría de prueba** (proof theory). La teoría de prueba está compuesta por tres componentes: el núcleo PSL (PSL Core), una o más teorías fundacionales, y extensiones PSL.
 - PSL Core: un conjunto de axiomas escritos en el lenguaje básico de PSL.
 - Teorías fundacionales: teorías cuyo potencial expresivo es suficientemente potente como para dar definiciones precisas de conceptos primitivos de PSL, mejorando así la precisión de las traducciones entre esquemas. De las teorías fundacionales quizá la más familiar es la teoría de conjuntos.
 - Extensiones: que posibilitan la expresión de información mediante conceptos que no están en PSL Core. Un buen ejemplo de extensión es la teoría de duraciones temporales. PSL Core no proporciona recursos para expresar duraciones, aunque en muchos contextos puede necesitarse, por lo que se ha desarrollado una teoría que puede añadirse a PSL Core para proporcionar este poder expresivo.

3.2.6 Workflow Management Facility

Este estandar está basado en los definidos por la WfMC. En realidad se puede interpretar como un OMG IDL binding [OMG, 2000b] para el interfaz de interoperabilidad de la coalición. Se pretende que la especificación de la Workflow Management Facility IDL esté avalada por la WfMC y la OMG. En ella se especifican los interfaces para el control de la ejecución y monitorización de flujos de trabajo y la interoperabilidad entre flujos de trabajo definidos y gestionados independientemente. Los interfaces están basados en un modelo de flujos de trabajo orientado a objetos, que incluye sus relaciones y dependencias con solicitantes, asignaciones y recursos.

El modelo está representado en un diagrama de clases UML (véase la Figura 11) y diagramas de interacción y especificado en interfaces IDL.

La especificación de la OMG es más detallada que el estándar de la WfMC (véase el Apartado 3.2.1 WfMC interfaz de interoperabilidad y Wf-XML), incluyendo interfaces que no están ni siquiera presentes en Wf-XML como WfActivity, WfResource y WfAssignment, además de ampliar operaciones como ProcessInstance y ChangeProcessInstanceState con la definición de interfaz WfProcess. Esto puede contribuir a una mejor utilización práctica del estándar, pero a su vez define un modelo mucho más concreto, que restringe la libertad para definir modelos particulares.

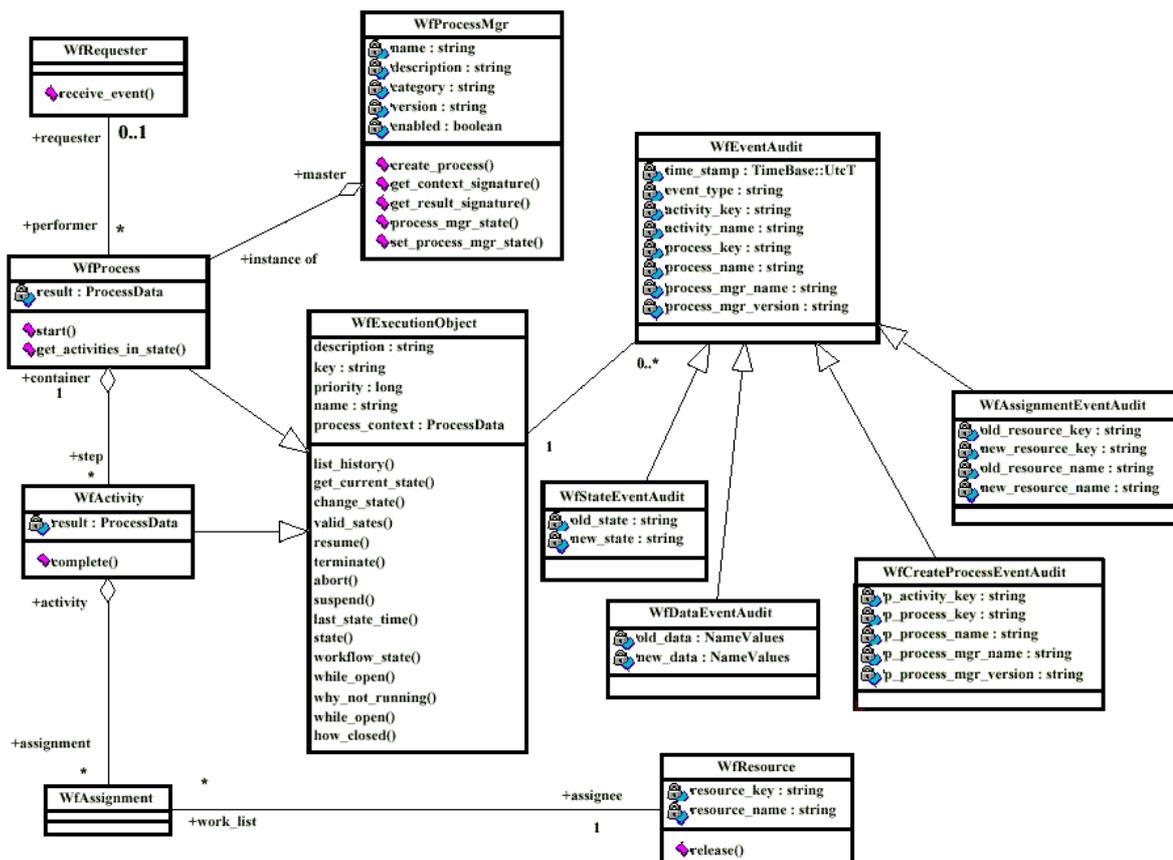


Figura 11. Modelo propuesto en la Joint Workflow Management Facility.

Tabla 4. Interfaces principales definidos en la Workflow Management Facility.

Interfaz	Descripción
<i>WfRequester</i>	Enlaza el propietario inmediato de una petición para un <i>WfProcess</i>
<i>WfProcessMgr</i>	Proporciona la factoría y el soporte de localización para <i>WfProcess</i>
<i>WfProcess</i>	El ejecutante de una petición de flujo de trabajo, hecha por un usuario o actor automático como una <i>WfActivity</i> actuando como <i>WfRequester</i>
<i>WfActivity</i>	Un paso en un <i>WfProcess</i> y que también puede ser un <i>WfRequester</i>
<i>WfExecutionObject</i>	Una clase base abstracta para <i>WfProcess</i> y <i>WfActivity</i> .
<i>WfAssignment</i>	Enlaza actividades con <i>WfResources</i> potenciales/actuales
<i>WfResource</i>	Una persona o cosa que puede realizar o aceptar una <i>WfActivity</i>
<i>WfEventAudit</i>	Un interfaz común para registrar eventos de flujos de trabajo. Se definen varios subtipos de este interfaz para almacenar cambios de estado; datos asociados con el, y cambios en la asicación de recursos a <i>WfActivity</i>

3.3 *Influencia de otras tecnologías en el modelado de procesos*

3.3.1 Modelado orientado a objetos

La aplicación del paradigma orientado a objetos ha tenido gran importancia en la ingeniería del software en general, pero en el modelado de procesos no existe, hasta lo que conocemos, una gran aplicación [Shams-Aliee y Warboys, 1995][Wang, 1998]. La principal causa de la utilización de modelos orientados a objetos es el tratamiento de la complejidad. En general los modelos simples, como redes de Petri o grafos de actividades contribuyen a formalizar o dar una representación sencilla a los procesos, pero tienen una limitación: no se adaptan bien a entornos en los que los procesos son complejos, tanto en su definición como en su uso [Wang, 1998]. La utilización de orientación a objetos no solo favorece el tratamiento de la complejidad mediante la modularidad, sino que favorece la reutilización, comprensión, etc. del modelo. Incluso en trabajos recientes se defiende la utilización de la orientación a objetos para realizar el metamodelo [Joeris y Herzog, 1998][Kradolfer, 2000], pero el modelo definido (lenguaje de modelado) no tiene características suficientes como para clasificarlo dentro del paradigma orientado a objetos. Por ejemplo se incluyen en el metamodelo nociones como la composición, pero no llega a definirse formalmente la agregación (o la herencia) como el mecanismo de creación de procesos compuestos. Podemos decir que la orientación a objetos se utiliza como herramienta pero no como objetivo. Condiciones mínimas para poder declarar un modelo de procesos como orientado a objetos deberían ser: uso de herencia para reutilizar comportamientos o estructuras, uso del polimorfismo para manejar cambios dinámicos, uso de la encapsulación para ocultar la implementación de procesos y el uso de la agregación para manejar subprocesos.

3.3.2 Técnicas formales

La aplicación de métodos formales al modelado de procesos en general ha sido extensa, pero si nos restringimos a la aplicación a tecnologías de soporte de procesos organizacionales esta aplicación se ve muy reducida, quizá en gran medida por el surgimiento de herramientas comerciales ad-hoc en respuesta a necesidades puntuales [Georgakopoulos et al., 1995]. Las redes de Petri son quizá el formalismo más ampliamente utilizado para representar procesos, dada su sencillez y adaptación natural a la definición de tareas y dependencias [Bandinelli y Fugetta, 1993][Shams-Aliee y Warboys, 1995][van der Aalst y ter Hofstede, 2000][Raposo et al., 2000].

Aunque hay otros formalismos, los que adquieren mayor relevancia en aproximaciones flexibles y que soporten los requisitos expuestos al comienzo del trabajo son los basados en lógica y ontologías. En [Sa y Warboys, 1995] se realiza la descripción de un paradigma basado en lógica temporal. Además se propone la agregación de la reflexión como vía para tratar el cambio durante la vida de un proceso. Esta base formal matemática permite definir un modelo de proceso reflexivo que ofrezca la oportunidad de razonar acerca de las propiedades de un proceso antes de que el modelo pase a utilizarse (ejecución). Otro intento de aplicar la lógica al modelado de procesos es [Davulcu et al., 1998]. En él se propone el uso de Concurrent Transaction Logic (CTR). Lo interesante de esta aproximación es la utilización de una especie de paradigma de programación automática aplicada a flujos de trabajo, lo que se puede trasladar a la posible transformación de una representación de procesos en el modelo canónico propuesto en CTR a cualquier paradigma de modelado de procesos. De igual forma la interoperabilidad basada en el intercambio de ontologías ha sido aceptada por la comunidad de negocios

[Ontology] y aplicada en la definición de modelos de intercambio de información sobre procesos con una total comprensión de la semántica por cada una de las partes [PSL].

3.3.3 Arquitectura software y metanivel

Un metaproceso es simplemente un proceso responsable de crear y adaptar un proceso. Por tanto un metaproceso puede usarse para manejar la evolución, estados, inconsistencias de otro proceso. Si este proceso es ofertado al usuario, este tendrá un mecanismo para cambiar o eliminar inconsistencias, para tratar cambios excepcionales en una ejecución determinada, para establecer o asignar recursos específicos, y en general para realizar cualquier actividad cuyo fin sea tratar o modificar un proceso. En definitiva la existencia de un proceso capaz de manejar y organizar el tratamiento de otros procesos permite automatizar y asegurar la correcta ejecución de estas tareas, al igual que se hace con las tareas del negocio. Puesto que el metaproceso es también un proceso necesita ser creado, controlado o modificado, lo que nos lleva a la necesidad de un metametaproceso para gestionar el metaproceso. Si seguimos esta aproximación podríamos encontrarnos con una recursión infinita. Para evitar este problema debemos cerrar la recursión, y esto se hace proporcionando a un proceso la capacidad de definir actividades que manejen el propio proceso. Este proceso se llama proceso reflexivo, y un modelo de tal proceso se convierte en un modelo reflexivo.

Los trabajos más interesante sobre aplicación de la noción de metaproceso son los del IPG (Informatics Process Group, University of Manchester) aplicados al tratamiento del cambio en procesos software. Los trabajos iniciales pueden consultarse en [Sa y Warboys, 1995] [Robertson, 1996] y una aplicación más reciente e incluso demostraciones en el proyecto PIE [PIE, 2001]. Su objetivo es definir tanto un modelo de proceso (software) formal y reflexivo como definir un metaproceso implementable. En [Robertson, 1996] se define un metaproceso implementable como un proceso más en un PSS, el llamado Process for Process Evolution (PPE o P²E), cuyas características más importantes son: soportar la especialización para contemplar objetivos específicos y crear nuevos modelos de proceso ejecutable, poder implementar ese modelo, poder cambiar su propio comportamiento (reflexión) y poder reproducirse a sí mismo.

Aunque la utilización de este metanivel se ha aprovechado para definir las primitivas de evolución de los procesos, podría definirse también las primitivas de comunicación. Con ello se obtendría un metanivel que definiría las características del modelo global, de sus posibilidades de evolución y de su semántica con el fin de posibilitar su gestión y traducción.

La construcción de sistemas mediante la integración de subsistemas ha sido también abordado desde el punto de la arquitectura software [Papadopoulos y Arbab, 1998]. En esta campo se han propuesto metáforas consistentes en la idea de conectores software que realizan la función de “tuberías” por las que circula información de control o datos. Otras metáforas han sido la de bus software, encargado de proporcionar un mecanismo de comunicación entre los sistemas que desean interoperar, o la de puertos de comunicación. Estos últimos especifican ciertos puntos de entrada a los subsistemas que pueden ser interconectados mediante conectores software, asemejando así la tarea de composición que se realiza en juegos tipo lego. Otra metáfora con gran aceptación es la basada en la idea de contrato, quizá por su adecuación a la solución de interoperabilidad entre sistemas que requieren una etapa previa de establecimiento de las condiciones de colaboración, la negociación.

3.3.4 Bases de datos federadas

Al igual que en ocurre con los procesos existe la necesidad de cooperación y el acceso integrado a múltiples bases de datos autónomas y heterogéneas, es decir, acceder a los datos dispersos por varias organizaciones como si de una sola base de datos se tratase [Saltor y García-Solaco, 1993]. El acceso integrado implica interconectar los sistemas de bases de datos mediante una red de comunicaciones, superponiendo una capa de software sobre los sistemas existentes, respetando su autonomía a nivel de aplicaciones locales preexistentes, pero cediendo parte del control al nivel federado. Esta colección de bases de datos que cooperan, conocidas como Bases de Datos Componentes (BDCs) [Campderrich et al., 1996], forman una federación de bases de datos y el software encargado de gestionarlas recibe el nombre de Sistema de Gestión de Bases de Datos Federadas (SGBDF). Como puede comprobarse podemos establecer una analogía entre la federación de sistemas de datos y federación de PSS. Incluso se puede asumir que se requiere una federación de bases de datos para poder realizar una federación de procesos, pues en la mayoría de las ocasiones los procesos globales contemplan la gestión o el flujo de datos interorganizacionales, lo que representa la necesidad de una capa lógica que trate la heterogeneidad en la gestión de las bases de datos.

Otra noción importante que puede importarse de este campo es la idea de definir grados de acoplamiento. En bases de datos federadas se han propuesto diferentes arquitecturas y tipos de SGBDFs según los distintos niveles de integración de las BDCs y según la variedad de servicios globales ofrecidos. Podemos clasificar, según el grado de autonomía, los SGBDF en:

- **Débilmente acoplados**, (loosely coupled federated database system). Cuando es responsabilidad propia de cada usuario crear y mantener la federación y no existe ningún tipo de control desde el sistema federado ni de su(s) administrador(es). Estos sistemas también se conocen como sistemas de bases de datos interoperables y/o sistemas multibase de datos.
- **Fuertemente acoplados**, (tightly coupled federated database system). Cuando la federación y su(s) administrador(es) son responsables de la creación y mantenimiento de la misma y participan activamente en el control de acceso de las BDCs. Un sistema federado fuertemente acoplado puede ser de dos tipos:
 - Con federación única, si permite la creación y gestión de un único esquema federado.
 - Con múltiples federaciones, si permite la creación y gestión de múltiples esquemas federados.

Volviendo a la analogía con los PSS, un acoplamiento débil representa la capacidad de cada uno de los componentes de crear sus propios procesos globales para interactuar con su entorno. En el segundo caso, un acoplamiento fuerte asumiría la existencia de una entidad que define y administra los procesos interorganizacionales.

3.4 *Discusión*

Como se ha mostrado en este capítulo, existe una gran diversidad de modelos y estándares para la definición de procesos. Generalmente las diferencias entre modelos en un mismo dominio no son grandes, pues los requisitos son similares, mientras que las representaciones en distintos dominios pueden llegar a presentar diferencias notables. Esto hace

pensar en la definición de diferentes grados de heterogeneidad en el problema, pudiendo considerar como “heterogeneidad simple” la que considera diferencias en un mismo dominio y “heterogeneidad múltiple” la que incluye también diferencias entre dominios.

Un tema de discusión más es la conveniencia de crear modelos globales flexibles capaces de adaptarse a cualquier modelo en un campo de aplicación (heterogeneidad simple). En [Carlsen, 1997] se realiza el estudio de los modelos previos para intentar crear un modelo lo más flexible posible, se analiza y propone un nuevo paradigma, una nueva notación gráfica y se utilizan nuevos conceptos como los puertos para el modelado de procesos. Esto no contribuye en absoluto a la interoperabilidad, pues generalmente no podrá alcanzarse un modelo genérico, algo similar a lo que ocurre en bases de datos, donde a pesar de existir un lenguaje genérico (SQL) se necesitan extensiones para cada nuevo método de manejo de datos (relacional, orientados a objetos, multidimensionales, espaciales, etc), aún trabajando en un campo de aplicación reducido. Por ello se da por sentada la necesidad de coexistencia entre diferentes paradigmas, adaptados a cada una de las vistas del proceso [Conradi y Liu, 1995].

Asumiendo la discusión anterior el problema se concreta en cómo hacer interoperar a procesos expresados en diferentes paradigmas. La solución puede afrontarse definiendo modelos canónicos en los que se representa el modelo global, que es traducido a cada uno de los diferentes submodelos o simplemente definiendo mecanismos de comunicación entre procesos, lo que involucra la aceptación de los estándares de comunicación por parte de los actores. En cualquier caso algunas de las características más importantes, desde el punto de vista de la interoperabilidad, que deberían contemplarse para la definición general de procesos son las siguientes:

- **Extensibilidad.** Posibilidad de modificar construcciones existentes o añadir nuevas. Esta característica incluiría también la existencia de un metanivel que definiere la semántica de las extensiones.
- **Meta-información.** Los procesos pueden tener información asociada como nivel de esfuerzo, coste, notas, etc.
- **Secuencias complejas.** Además de la secuencia, la alternativa, la ejecución paralela, etc. deben contemplarse la existencia de relaciones complejas como “B ocurre después de A si C ocurrió en 3 horas”. También constructores para sincronización de secuencias paralelas.
- **Constructores generales.** Conceptos como actividad, rol, recursos, etc. y las relaciones asociadas.
- **Mecanismos de abstracción.** Encapsulación jerárquica, vistas y soporte de la ambigüedad o incompletitud. También agrupación y mecanismos de ordenación como colas, listas, etc.

Como ya hemos analizado anteriormente se hace necesaria la coexistencia de diferentes paradigmas de modelado, lo que a-priori imposibilita la utilización de un modelo formal como “modelo visible”. Creemos que las técnicas formales deben utilizarse como modelo integrador, usados para reificar un modelo abstracto y de comprensión sencilla, ya que no es posible restringir u obligar a los usuarios finales del modelo a conocer sus fundamentos teóricos. Por todo ello, la conclusión principal a la que podemos llegar es la utilización de técnicas formales como medio, pero no única base del modelo.

4 Sistemas de soporte de procesos interorganizacionales

4.1 Clasificación

En primer lugar podemos encontrar la clasificación en niveles de interoperabilidad propuesta por la WfMC [WfMC, 1999b]:

1. Sin interoperabilidad.
2. Coexistencia.
3. Uso de Gateways.
4. Subconjunto de APIs Común.
5. Conjunto completo de APIs Común.
6. Formatos de Definición Compartidos.
7. Compatibilidad de Protocolos.
8. Apariencia y Utilidades Comunes.

Esta clasificación está definida desde un punto de vista arquitectónico e intenta declarar un grado cada vez mayor de interoperabilidad entre motores de flujos de trabajo. Aunque esta clasificación presenta varios problemas (véase el apartado 4.3), sirve como ejemplo y primera aproximación a la definición de grados de interoperabilidad entre PSS, una idea importante si pretendemos ofrecer interoperabilidad en un ambiente heterogéneo donde deben coexistir sistemas con capacidades muy diferentes.

Considerando que la integración de varios sistemas de soporte de procesos se realiza mediante la ejecución de un proceso global, el lugar o la forma de expresarlo puede llevarnos a una primera clasificación. La existencia de un proceso compartido implica la ejecución de subprocesos en cada una de las empresas que queremos interconectar. Dependiendo de cómo se exprese el proceso o estado común tendremos un estilo arquitectónico u otro. [Estublier et al., 1998] clasifican las arquitecturas en:

- **Basadas en control:** los servicios se definen en un formalismo común y existe un modelo de proceso global. El proceso global posiblemente se ejecute en un sistema de soporte de procesos supervisor, encargado de controlar la ejecución distribuida, mientras que cada subsistema es autónomo y responsable de llevar a cabo únicamente las tareas que le han sido asignadas por el supervisor.
Algunos inconvenientes de esta arquitectura pueden ser: la centralización o la necesidad de expresar todos los servicios ofertados por las organizaciones en un mismo formalismo.
- **Basadas en estado:** las organizaciones colaboran observando el estado común, y actualizando su estado local en consecuencia. También puede cambiar el estado global como respuesta a cambios en el estado local debido a la ejecución del proceso local. Permite colaborar a sistemas con representaciones muy diferentes, sin necesidad de conocer a los demás participantes en el proceso, facilitando así la evolución. La centralización del estado también favorece la monitorización.
Aquí, no se hace explícito el proceso común, con lo que la ejecución correcta de los procesos es responsabilidad de cada una de las organizaciones que componen la federación.

- **Mixtas:** al menos una parte del proceso o mecanismo de comunicación es global, mientras que puede seguir existiendo una zona de estado común habilitada para facilitar la comunicación donde el proceso global no normaliza la colaboración. Generalmente el proceso global definirá el modelo de interoperabilidad, mientras que las colaboraciones puntuales se realizan por medio de un estado global, aprovechando así lo mejor de cada una de las arquitecturas.

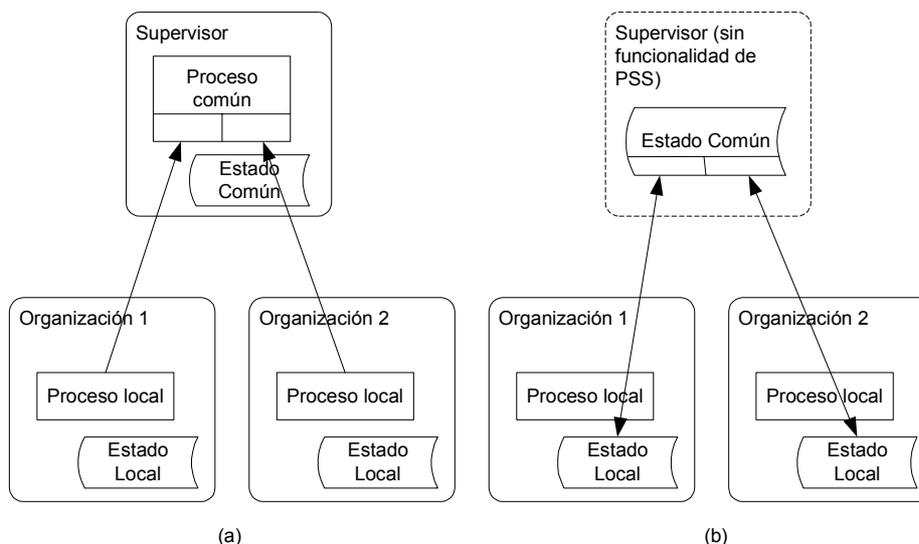


Figura 12. Estilos arquitecturas dependiendo del elemento utilizado para la interconexión: (a) control (b) estado.

Respecto al grado de diversidad entre los sistemas que deben hacerse interoperar puede consultarse la clasificación propuesta por [Ben-Shaul y Kaiser, 1998]. En este trabajo se especifican los diferentes grados de heterogeneidad y acoplamiento que deben tratarse en PSS (véase la Figura 13). Los niveles de acoplamiento (grado de unión entre los participantes en la colaboración) se dividen en:

- **Elección independiente.** Mínimo nivel de acoplamiento, en el que cada entidad selecciona el entorno más apropiado a sus necesidades.
- **Mismo entorno o similar sin conexión.** Los participantes en la colaboración pueden seleccionar el entorno común más apropiado para su propósito (entorno homogéneo) o pueden elegir entornos distintos pero con un acuerdo sobre formatos de intercambio. Existe un ente externo que gestiona la coordinación y colaboración.
- **Entorno multi-sitio o multi-entorno.** Por entorno multi-sitio se entiende aquella agrupación que ofrece la capacidad de distinguir entre subentornos. Proporciona los mecanismos de colaboración entre los subentornos, por lo que tanto los subentornos como el mecanismo de coordinación están definidos por una entidad central, es lo que podríamos definir como un entorno cuyos componentes están distribuidos. Un multi-entorno representa una versión más heterogénea del entorno multi-sitio, ya que permitiría la existencia de diferentes entornos, pero con un estándar de control (como notificación de eventos) junto con el estándar de intercambio de datos. Es lo que podríamos definir como un entorno con componentes independientes pero federados.

- **Entorno distribuido geográficamente.** Es el caso de mayor acoplamiento, donde no se distingues subentornos sino que los usuarios comparten un entorno y unos recursos comunes. La definición y ejecución de procesos se haría sin tener en cuenta problemas de distribución, ofreciendo la visión de una organización virtual.

Independientemente del nivel de acoplamiento, y basándose en el tipo de entorno que utilice cada uno de los participantes, podemos dividir el problema entre interoperabilidad homogénea y heterogénea. La primera se da cuando los sistemas que han de interoperar no están restringidos pero están sujetos a un mismo modelo. El caso heterogéneo se da cuando los sistemas son diferentes e incluso modelos conceptualmente equivalentes pueden aparecer diferenciados debido a los formalismos particulares utilizados por cada sistema también pueden ser diferentes.

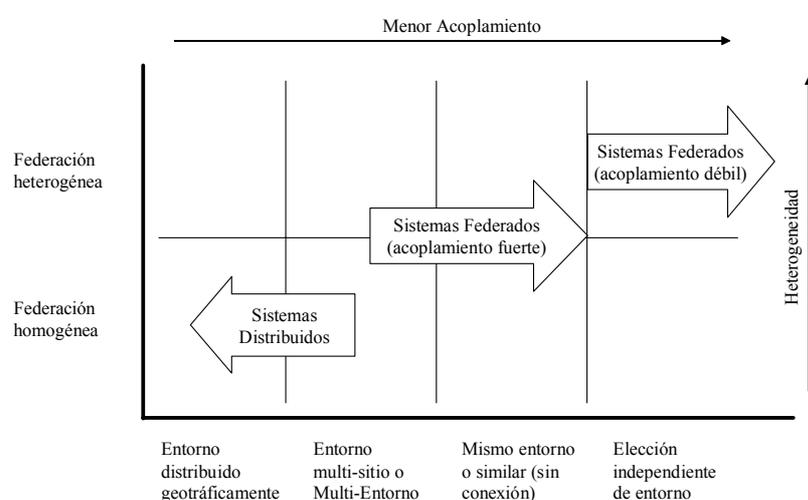


Figura 13. Clasificación propuesta por [Ben-Shaul y Kaiser, 1998] para los entornos de desarrollo de software. Verticalmente se representa el grado de heterogeneidad y horizontalmente el grado de acoplamiento entre los componentes del grupo de colaboradores. Se han incluido flechas para mostrar la equivalencia con la definida en bases de datos.

De nuevo podríamos establecer una analogía con las bases de datos, donde los entornos más acoplados son las bases de datos distribuidas y las bases de datos federadas estarían en un nivel de acoplamiento intermedio. El problema que estamos analizando; proporcionar interoperabilidad entre organizaciones autónomas, hace necesario contemplar el mínimo rango de acoplamiento, es decir, solucionar los problemas de interoperabilidad situados a la derecha en la Figura 13. Por simplicidad y analogía con bases de datos, haremos referencia a este tipo de sistemas como federaciones de sistemas de procesos, sin especificar exactamente el nivel de acoplamiento.

4.2 Principales sistemas y arquitecturas propuestas

4.2.1 Marco de interoperabilidad propuesto por la WFMC

En el apartado 3.1.1 se vio el modelo general propuesto por la WfMC. Derivado de la división en conjuntos de funcionalidad que se describe en ese apartado la coalición define un modelo de referencia (Workflow Reference Model [WFMC, 1999]) que identifica las diferentes áreas funcionales y escenarios típicos de uso:

- **Definición de procesos:** especificación de los procesos, sus datos y el intercambio con el entorno de ejecución.
- **Interoperabilidad:** interfaces para soportar interoperabilidad entre diferentes sistemas de flujo de trabajo.
- **Aplicaciones Invocadas:** interfaz que soporta la interacción con una variedad de tipos de aplicaciones.
- **Aplicaciones clientes de flujos de trabajo:** interfaz para soportar interacción con las funciones de interfaz de usuario y escritorio.
- **Administración y monitorización:** interfaces para proporcionar monitorización y métricas para facilitar la gestión de procesos complejos.

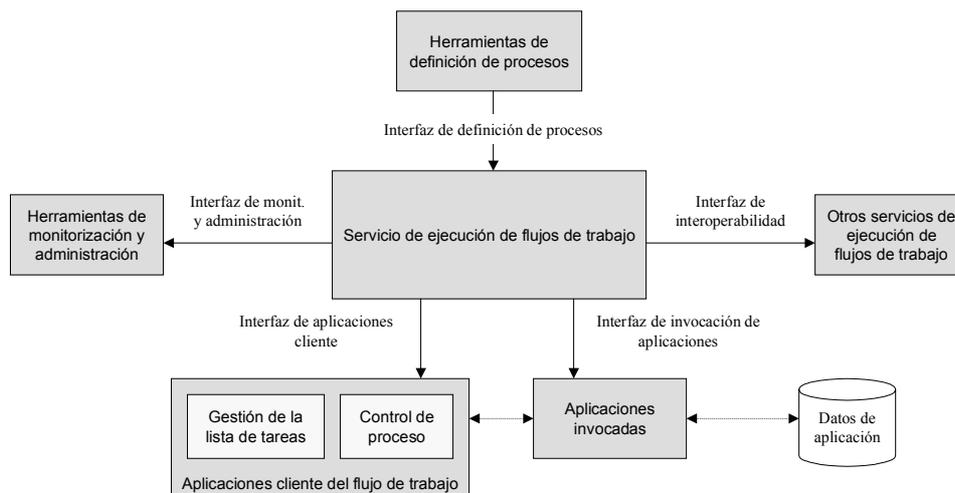


Figura 14. Modelo de referencia de la coalición [WFMC, 1999].

Puesto que nuestro estudio se centra en la interoperabilidad entre PSS, nuestro interés se centra principalmente en el interfaz de interoperabilidad. El problema de interoperabilidad al que se enfrenta el modelo parece ser el de interconectar únicamente los motores de ejecución, relegando problemas como la definición compartida de procesos (interoperabilidad a nivel de herramientas de definición) o el de la monitorización de procesos distribuidos (cómo interacciona una herramienta de monitorización con varios servicios de ejecución). Estas restricciones, por otra parte, restringen la complejidad del problema y hacen abordable la definición de estándares para la interoperabilidad en un campo con unas características muy concretas [WFMC, 1999b].

4.2.2 Interacción basada en eventos

Casati et al. [Casati et al., 1996] identifican el problema de la interoperabilidad y proponen una solución a la comunicación de sistemas de soporte de flujos de trabajo basado en la posibilidad de enviar eventos previamente definidos. La alternativa al uso de un lenguaje de definición común es el uso de administradores de eventos para coordinar la comunicación. El único requisito es que los modelos de proceso puedan orientarse a eventos, y que durante su ejecución puedan provocarlos o solicitarlos a una nueva entidad, el administrador de eventos. Los eventos se producen o consumen en “actividades” denominadas “nodos de evento”, que pasan a forma parte de los flujos de trabajo (véase la Figura 15). Esto lleva a la necesidad de modificar los flujos de trabajo insertando nodos nuevos, que pueden bloquear o no la ejecución, pero que permiten coordinar diferentes flujos de trabajo sin necesidad de realizar ninguna modificación al motor que los soporta. En la Figura 15 puede observarse un ejemplo donde se necesita coordinar un proceso consistente en las actividades A y B. Cada una de ellas debe ejecutarse en una organización, por lo que se define un proceso con la actividad A en la organización 1 y un nodo de evento que se encargará de notificar a la organización 2 (evento E) la terminación de A. Mientras, la organización 2 quedará bloqueada en su nodo de evento. Cuando A termina se envía un evento al administrador de eventos, en ese momento se hará que el nodo de evento en la organización 2 se desbloquee y comience la ejecución de B.

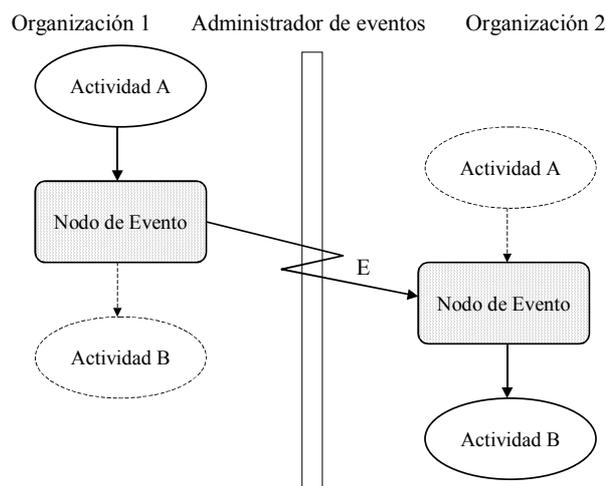


Figura 15. Esquema de interacción mediante nodos de evento. Ejemplo de ejecución de una secuencia de actividades insertando un nodo de envío (org. 1) y recepción (org. 2) para la interacción mediante el evento E.

Una extensión de esta aproximación se presenta en [Hagen y Alonso, 1999], donde se discute con mayor profundidad la interacción entre procesos basada en eventos. Aquí se utiliza un mecanismo de suscripción a los eventos producidos por otro proceso, lo que permite crear colas de eventos importados utilizables dentro del proceso cliente. También contempla aspectos de manejo de excepciones y recuperación ante fallos, proponiendo algoritmos para operaciones de rollback o revocación de eventos. Es una aproximación basada en el lenguaje OCR y el modelo asociado al kernel de soporte de procesos OPERA, por lo que no contempla el tratamiento de la interacción entre procesos definidos en ambientes heterogéneos.

4.2.3 Sistemas basados en metáforas

En [Ben-Shaul y Kaiser, 1998] se proponen arquitecturas para la federación de entornos basados en procesos, tanto en el caso homogéneo como en el heterogéneo, donde se trata de abordar los casos en los que existe mínimo grado de acoplamiento entre los sistemas, sin llegar a la necesidad de modelos idénticos. Su propuesta se basa en la metáfora de alianza internacional, donde cada grupo gestiona sus propios procesos al igual que un país gestiona sus propias leyes. Se utilizan los tratados para establecer los acuerdos que, extendiendo procesos propios permitan crear procesos globales, al igual que un conjunto de países necesita hacer pequeños cambios a sus leyes internas para formar parte de un tratado. La ejecución de procesos interorganizacionales se definen como cumbres. Finalmente se proponen dos arquitecturas, una basada en la inclusión de módulos de comunicación y conexión en los sistemas existentes para soportar federaciones homogéneas y otra con menor capacidad de comunicación para el caso heterogéneo. Esta aproximación presenta el problema principal de asumir la posibilidad de modificar sistemas existentes para agregar los módulos que permitan la interacción. La elección de esta arquitectura hace que se tenga una gran dependencia de los sistemas existentes. Es apropiada para aquellos casos en que la federación trata entornos preferentemente homogéneos.

Como hemos visto, [Estublier et al., 1998] clasifican las federaciones de PSS en basadas en control o basadas en estado. Desde la primera alternativa, que adolece de una excesiva centralización, hasta la segunda, que no ofrece un control adecuado del trabajo interorganizacional, pueden definirse arquitecturas mixtas con las ventajas de éstas dos. Esto es lo que se intenta realizar en este mismo trabajo con la arquitectura APEL, donde se combina la existencia de un repositorio que contiene el estado común con la agregación un PSS que se encarga de almacenar el modelo de interoperabilidad. Este modelo de interoperabilidad es particularmente interesante desde el punto de vista de una federación heterogénea, ya que permite especificar el modelo de interacción, constituyendo un modo de flexibilizar la arquitectura, habiendo una separación clara entre aspecto funcional y operacional. El inconveniente a esta arquitectura es el derivado de la centralización, sigue siendo necesaria la existencia de un PSS común con un modelo común. En esta metáfora el sistema basado en control corresponde a una sociedad con gobierno centralizado, el basado en estado corresponde a una sociedad liberal mientras que el mixto corresponde a una sociedad con gobierno que planifica el desarrollo pero da libertad para que los componentes de la sociedad reaccionen “libremente” a las órdenes del gobierno

4.3 *Discusión*

Respecto a la clasificación y visión del problema de la interoperabilidad propuesto por la WfMC, podemos identificar varios problemas. Por un lado los objetivos de interoperabilidad no tienen por qué coincidir con el nivel máximo de interoperabilidad (apariencia común). Generalmente la apariencia común no es necesaria, pues como ya hemos visto cada ámbito de aplicación puede necesitar un tipo de herramienta o modelo diferente [Conradi y Liu, 1995]. Otro problema es la especificación de formatos de intercambio compartidos, lo que conlleva restringir el modelo de proceso global. Este problema puede atacarse mediante la creación de un formato genérico, que con toda seguridad se vería expandido por diferentes competidores. El otro punto de vista radicaría en definir un formato extensible, lo que provocaría la aparición de diferentes formatos, sobre los cuales sería imposible crear protocolos de comunicación compatibles. De todos modos la solución de la WfMC, un formato genérico, es razonable si tenemos en cuenta que se aplicará a un dominio restringido, pero no es una solución abierta.

Deducciones similares pueden hacerse respecto del conjunto de APIS. Si aceptamos que los PSS deben tener libertad para definir sus propios modelos y estándares, entonces no podremos basar la interoperabilidad en su diseño arquitectónico, sino en la agregación de una capa de interacción semántica.

Respecto a la posibilidad de utilizar eventos para la coordinación, también podemos encontrar desventajas importantes en su contra:

- La necesidad de modificar los modelos de flujo de trabajo original, con lo que no se cumplen el requisito de transparencia.
- No permite la potencialidad de un modelo común, no cumple con el requisito de potencia expresiva. Por ejemplo no permite la creación de nuevos procesos o provocar su evolución ya que no se tiene información sobre la estructura de los procesos en el destino. Esto mismo, sin embargo, puede ayudar al requisito de coexistencia entre múltiples paradigmas.

En general todas las aproximaciones que hemos visto representan buenas solución, pero parciales. No nos podemos conformar con comunicación orientada a mensajes o eventos, o con el establecimiento de una metáfora concreta. Si queremos alcanzar un alto grado de automatización de procesos interorganizacionales y definir marcos de interoperabilidad global como el perseguido por la iniciativa UDDI [UDDI], debemos ser capaces de comunicar conocimiento. En términos de procesos este conocimiento se concreta en la posibilidad de conocer e incluso explorar los procesos ofertados por una organización. Así, por ejemplo, una organización podría ofrecer un servicio de suministro de productos, la información del producto y el servicio de evolución asociado, que definiría los cambios que el cliente puede realizar en el modo de envío, pago o características del producto asociado, sin estar previstas en el modelo inicial. Para conseguir este requisito se necesita el uso de niveles meta (descripciones ontológicas) y subprocesos asociados al servicio que serán explorados e invocados por entidades externas.

5 Conclusiones y trabajos futuros

Dada la estructura del trabajo la mayoría de conclusiones se han expuesto durante su desarrollo. Se ha analizado el problema fundamental al que se enfrentan las agrupaciones de organizaciones actuales, definiendo los requisitos fundamentales de la colaboración mediante procesos electrónicos. El trabajo se ha dividido por un lado en el estudio de los modelos de proceso y por otro en el estudio de las arquitecturas y tecnologías actuales que permiten la interoperabilidad. Esto nos ha permitido ofrecer una visión global del tratamiento actual del problema, identificando las debilidades y capacidades de las aproximaciones y estándares a la representación de procesos en una amplia variedad de campos.

La conclusión principal, que resume todo lo discutido durante el trabajo, es la existencia de diferentes niveles/planos de interoperabilidad: semántica (de datos), semántica (de control), tecnológica, intra/inter dominio, etc. Cada sistema de soporte de procesos interorganizacionales suele aportar soluciones a uno o varios de estos niveles, pues es complejo encontrar soluciones universales. Por ello, la aproximación más aconsejable para tratar el problema consiste en aportar soluciones para cada uno de los niveles por separado, haciendo abordable el problema y consiguiendo interoperabilidad a un nivel determinado.

La gran diversidad de modelos y estándares mostrados debe hacer reflexionar sobre la imposibilidad de definir métodos de interconexión basados en la definición de modelos generalistas, debiendo aplicarse la investigación futura a la definición de métodos de interacción integradores. Como se explicitó en la relación de requisitos iniciales, el objetivo más importante de la investigación debe ser que los procesos sean ejecutados utilizando la tecnología particular de cada organización. No debe orientarse a soportar procesos, sino a construir un marco donde se facilite la comunicación entre ellos, y por tanto la comunicación entre las organizaciones que colaboran para realizarlos. Otro frente de investigación lo constituye la automatización de los procesos de interconexión o colaboración. Cada vez más se hace necesario automatizar procesos como el establecimiento de contratos para obtener un cierto servicio o producto de una empresa (virtual o no). Automatizar las tareas de comercio electrónico B2B o la inclusión en cadenas de suministro conlleva la automatización de los procesos que definen estas actividades, siempre y cuando éstas estén orientadas a procesos. Otro punto de interés lo representa la introducción formal de la orientación a objetos en el modelado de procesos, algo que también se ha identificado como tratado de manera insuficientemente.

6 Referencias

- [Alho, 1998] Kari Alho, Supporting multiple software process modeling formalisms through common denominator representation. LICENTIATE THESIS. HELSINKI UNIVERSITY OF TECHNOLOGY, Department of Computer Science and Engineering. 1998.
- [Allen y Garlan, 1997] R. Allen and D. Garlan. A formal basis for architectural connection. ACM Transactions on Software Engineering and Methodology, 1997
- [Alonso y Hagen, 1997] G. Alonso and C. Hagen. Geo-Opera: Workflow concepts for spatial processes. In Proc. 5th Intl. Symposium on Spatial Databases (SSD '97), Berlin, Germany, 1997
- [Alonso et al., 1999] G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, N. Weiler, "WISE: Business to Business E-Commerce", In Proceedings of the IEEE 9th International Workshop on Research Issues on Data Engineering. INFORMATION TECHNOLOGY FOR VIRTUAL ENTERPRISES (RIDE-VE'99). Sydney, Australia, March 1999.
- [Bandinelli y Fugetta, 1993] Bandinelli, S.; Fugetta, A., "Computational Reflection in Software process Modeling the SLANG Approach", in Proc. of the 15th Int. Conf. on Software Engineering, IEEE Computer Society Press, 1993; pages 144-154
- [Ben-Shaul y Kaiser, 1995] I. Z. Ben-Shaul, G. E. Kaiser, "An Interoperability Model for Process-Centered Software Engineering Environments and its Implementation in Oz", Technical report CUCS-034-95. Columbia University. 1995.
- [Ben-Shaul y Kaiser, 1998] I. Z. Ben-Shaul, G. E. Kaiser, "Federating Process-Centered Environments: the Oz Experience", Automated Software Engineering, Kluwer Academic Publishers, vol. 5-1, January 1998, pp. 97-132.
- [Bichler et al, 1997] P. Bichler, G. Preuner, M. Schrefl. Workflow Transparency. In Proc. of the 9th Int. Conf. on Advanced Information Systems Engineering (CaiSE '97). Barcelona, Spain. 1997.
- [Box et al., 2000] Don Box, et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note 08 May 2000, <http://www.w3.org/TR/SOAP>.
- [BPMI] Business Process Management Initiative, <http://www.bpmi.org>.
- [BPMI, 2001] BPMI Business Process Modeling Language (BPML), Working Draft 0.4, 3/8/2001
- [BPMI, 2001b] <http://www.bpmi.org/bpql.esp>, Working Draft previsto para finales del 2001.
- [Campderrich et al., 1996] B. Campderrich, M. Oliva, E. Rodriguez, L.C. Rodriguez, F. Saltor, J. Samos & J. Sistac. "El proyecto BLOOM de investigación en Sistemas de Gestión de Bases de Datos Federadas". Actas de las Primeras Jornadas en Investigación y Docencia en Bases de Datos, A Coruña, pp 272-282. 1996.
- [Carlsen, 1997] S. Carlsen, "Conceptual Modeling and Composition of Flexible Workflow Models". PhD thesis, Norwegian University of Science and Technology, Diciembre 1997.
- [Casati et al., 1996] F. Casati, S. Ceri, B. Pernici, G. Pozzi, "Semantic Workflow Interoperability", EDBT, France 1996.
- [Conradi et al., 1992] R. Conradi, M. L. Jaccheri, C. Mazzi, A. Aarsten, and M. N. Nguyen, "Design, use, and implementation of SPELL, a language for software process modeling and evolution," in J.-C. Derniame (ed.): Proc. from EWSPT'92, Sept. 7--8, Trondheim, Norway, Springer Verlag LNCS 635, pp. 167--177, Sept. 1992.

-
- [Conradi y Liu, 1995] R. Conradi, C. Liu, "Process Modelling Languages: One or Many?", EWSPT'95, Abril 1995, Leiden.
- [Covington, 1997] Covington, M. A., "Speech Acts in Electronic Communication With Special Reference to KQML and ANSI X12", Hawaii International Conference on System Sciences (HICSS-30), Maui, Hawaii, 1997.
- [Cunin, 2000] P.-Y. Cunin, "The PIE project: An Introduction.", PIE Workshop 1, EWSPT7 in Kaprun, near Salzburg, Austria, February 2000.
- [Davulcu et al., 1998] H. Davulcu, M. Kifer, C.R. Ramakrishnan, and I.V. Ramakrishnan. Logic based modeling and analysis of workflows. In ACM Symposium on Principles of Database Systems, June 1998.
- [De Michelis, 1995] De Michelis, G., "Work Processes. Organizational Structures and Cooperation Supports: Managing Complexity", 5th IFAC Symposium on Automated Systems Based on Human Skill –Joint Design of Technology and Organization, Berlin, 1995.
- [Eder et al., 1999] Johann Eder, E. Panagos, M. Rabinovich, "Time Constraints in Workflow Systems", Proc. of 11th Int. Conf. on Adv. Inf. Systems Engineering (CAiSE 99), Heidelberg, Germany, 1999
- [Estublier et al., 1998] J. Estublier, P.Y. Cunin, N. Belkhatir, "Architectures for Process Support System Interoperability", Proceedings of the Fifth International Conference on the Software Process, (Lisle, IL), Jun 1998
- [Georgakopoulos et al., 1995] D. Georgakopoulos, M. Hornick, A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*. Vol. 3. 1995.
- [Grefen et al., 2000] P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig; CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises; International Journal of Computer Systems Science & Engineering, Vol. 15, No. 5, 2000; pp. 277-290.
- [Hagen y Alonso, 1999] C. Hagen, G. Alonso, "Beyond the Black Box: Event-based Inter-Process Communication in Process Support Systems", International Conference on Distributed Computing Systems, 1999.
- [Harel et al., 1990] D. Harel, H. Lachover, A. Nammad, A. Pnueli, M. Politi, R. Sherman, A. Shtull-Trauring, and M. Trakhtenbrot. STATEMATE: A Working Environment for the Development of Complex Reactive Systems. IEEE Transactions on Software Engineering, 16, 1990, pp. 403-414.
- [Heineman, 1993] Heineman, George T. Automatic Translation of Process Modeling Formalisms. New York, NY, USA: Columbia University, Department of Computer Science, Technical report TR CUCS-036-93. 20p
- [Hull et al., 1999] R. Hull, F. Lirbat, E. Simon, J. Su, G. Dong, B. Kumar, and G. Zhou. Declarative workflows that support easy modification and dynamic browsing. In Proc. of Intl. Joint Conf. on Work Activities Coordination and Collaboration (WACC), pages 69--78, February 1999.
- [Hull et al., 2000] R. Hull, A. Kumar, J. Siméon, "Smart Supply Web: An Application of Web-based Data and Workflow Mediation", Workshop on Technologies for E-Services (VLDB-TES Workshop), Cairo, September 2000.
- [IDEF0, 1993] "Integration Definition For Function Modeling (IDEF0)", National Institute of Standards and Technology Draft Federal Information Processing Standards Publication 183, 1993. (<http://www.idef.com> para una visión completa de la familia de métodos IDEF)

-
- [IEEE 90] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.
- [IETF, 1998] "Simple Workflow Access Protocol (SWAP)", Keith Swenson, Internet-Draft, 7 Aug. 1998. <http://www.ics.uci.edu/~ietfswap>
- [isModeler, 2001] is/Modeler version 5.0, <http://www.ismodeler.com>.
- [Jaccheri et al., 1992] L. Jaccheri, J.-O. Larsen, R. Conradi, Software Process modeling and Evolution in EPOS. In Proc. 4 th Int. Conf. on Software Engineering and Knowledge Engineering, Capri, 1992
- [Joeris y Herzog, 1998] G. Joeris, O. Herzog, "Towards Object-Oriented Modeling and Enacting of Processes", Technical Report 07/98, University of Bremen, 1998.
- [Joeris y Herzog, 1999] Joeris G., Herzog O.: Towards Flexible and High-Level Modeling and Enacting of Processes. In Proc. of the 11 th Int. Conf. on Advanced Information Systems Engineering (CAiSE'99), Heidelberg, Germany (forthcoming) (1999).
- [Kammer, 2000] Peter J. Kammer: Supporting dynamic distributed work processes with a component and event based approach. ICSE 2000: 710-712
- [Keane et al., 1994] J.A.Keane, J.Sa and B.C.Warboys, Applying a Concurrent Formal Framework to Process Modelling. In Proceedings of FME'94, Barcelona, October, 1994. Pages 291-305, LNCS 873
- [Kifer, 1996] M. Kifer. "Transaction logic for the busy workflow profesional", Unpublished manuscript, <ftp://ftp.cs.sunysb.edu/pub/TechReports/kifer/tr-for-wf.ps.Z>. August 1996.
- [Kradolfer, 2000] M. Kradolfer, "A Workflow Metamodel Supporting Dynamic, Reuse-based Model Evolution", PhD Thesis, University of Zurich, May 2000.
- [Krogstie y Carlsen, 1996] Krogstie, J. and Carlsen, S., "Integrated process, role and speech act modelling for workflow support", 30th Annual Hawai'i International Conference on System Sciences (HICSS-30), 1996.
- [Lee et al., 1998] Jintae Lee, Michael Gruninger, Yan Jin, Thomas Malone, Austin Tate, Gregg Yost and other members of the PIF Working Group, "The PIF Process Interchange Format and Framework Version 1.2", The Knowledge Engineering Review, Vol. 13, No. 1, pp. 91-120, March 1998, Cambridge University Press.
- [Letelier et al., 1998] Letelier P., Sánchez P., Ramos I., Pastor O. OASIS 3.0: Un enfoque formal para el modelado conceptual orientado a objeto. Servicio de Publicaciones, Universidad Politécnica de Valencia, SPUPV -98.4011, ISBN 84-7721-663-0, 1998.
- [Martínez, 2000] A. Martínez, "Estado del arte en evolución en sistemas de gestión de flujos de trabajo" Trabajo de Investigación. Universidad de Castilla-La Mancha, 2000.
- [Metro] Metro Business Process Management Platform, Action Technologies, Inc., <http://www.actiontech.com>.
- [Meyer, 1998] BERTRAND MEYER, Construcción de Software Orientado a Objetos, ISE, Santa Barbara, Prentice Hall, 1998.
- [Microsoft, 2001] Microsoft BizTalk™ Server, <http://www.microsoft.com/biztalk/>, July, 2001.
- [Navision, 2001] Navision A/S, <http://www.navision.com/>, July, 2001.
- [Nguyen y Conradi, 1994] M.N. Nguyen, R. Conradi, "Classification of Meta-processes and their Models", In Proc. from the third International Conference on Software Process, Washington, USA, 10-11 October, pages 167-- 175, 1994.

-
- [OMG, 2000] Unified Modeling Language (UML), version 1.3 specification. OMG. 2000, <http://www.omg.org/technology/documents/formal/uml.htm>.
- [OMG, 2000b] Object Management Group, "Workflow Management Facility Specification, V1.2", http://www.omg.org/technology/documents/formal/workflow_management.htm, Abril 2000.
- [Papadopoulos y Arbab, 1998] George A. Papadopoulos and Farhad Arbab, "Coordination models and languages", Centrum voor Wiskunde en Informatica (CWI), 1998.
- [PIE, 2001] Process Instante Evolution, <http://www.cs.man.ac.uk/ipg/pie/pie-e.html>, June, 2001.
- [PSL] The Process Specification Language Project, <http://www.mel.nist.gov/psl/>.
- [Obeweis, 1996] A. Oberweis. *An Integrated Approach for the Specification of Processes and Related Complex Structured Objects in Business Applications*. Decision Support Systems, 17, pp. 31-53. 1996.
- [Ontology] [Ontology.org](http://ontology.org).
- [Oracle, 2001] Oracle Workflow, http://technet.oracle.com/products/integration/workflow/workflow_fov.html, July, 2001.
- [Pernici y Mecella, 2000] B. Pernici, M. Mecella, "Designing Components for e-Services". Proceedings of the VLDB. Workshop on Technologies for e-Services (VLDB-TES 2000), Cairo, Egypt, 2000.
- [Phios, 2001] The Process Repository, Phios Corporation, <http://repository.phios.com>. 2001.
- [Raposo et al., 2000] Raposo, A. B., Magalhes, L. P. and Ricarte, I. L. M. (2000) "Petri Nets Based Coordination Mechanisms for Multi-Workflow Environments", accepted for the Int.
- [Robertson, 1996] I. Robertson, "An Implementable Meta-process". Second World Conference on Integrated Design and Process Technology, 1996.
- [Sa y Warboys, 1995] J. Sa, B.C. Warboys, "A Reflexive Formal Software Process Model", . In Proceedings EWSPT'95, volume 913 of Lecture Notes in Computer Science, pages 241--254. Springer Verlag, 1995.
- [Saltor y García-Solaco, 1993] F. Sator and M. García-Solaco. "Diversity with Cooperation in database Schemata: semantic relativism". In Proc. Of the 14th International Conference on Information Systems (ICIS'93), pp. 247-254, 1993.
- [SAP, 2001] SAP AG, Workflow Module (WF), WebFlow, <http://www.mysap.com/>, July 2001.
- [Schlenoff et al., 1996] Craig Schlenoff, Amy Knutilla, and Steven Ray, Unified Process Specification Language: Requirements for Modeling Process, NISTIR 5910, National Institute of Standards and Technology, Gaithersburg, MD (1996).
- [Searle y Vanderveken, 1985] Searle, J. R. and Vanderveken, D., The Foundations of Illocutionary Logic, Cambridge University Press, 1985.
- [Shams-Aliee y Warboys, 1995] F. Shams-Aliee, B.C. Warboys, "Applying Object-Oriented Modelling to Support Process Technology", Proceedings of the 1st World Conference on Integrated Design & Process Technology, University of Texas, Austin, USA, December 1995
- [Sheth, 1996] A. Sheth, Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems: State of the Art and Future Directions. Athens, USA, 8-10 May 1996.
- [Sutton y Osterweil, 1997] S. M. Sutton, Jr. Ad L. J. Osterweil, The Design of a Next-Generation Process Language, Software Engineering, ESEC/FSE'97, LNCS 1301, Springer, 1997, pp. 142-158.
- [UDDI] Universal Description, Discovery and Integration of Business for the Web, www.uddi.org, Agosto, 2001.

-
- [Uschold y Gruninger, 1996] Uschold, M. and Gruninger M., *Ontologies: Principles, Methods, and Applications*, Knowledge Engineering Review, 1996, Vol. 11, pp. 96-137.
- [van der Aalst, 1998] W.M.P. van der Aalst. *The Application of Petri Nets to Workflow Management*. The Journal of Circuits, Systems and Computers, 8(1):21-66. 1998.
- [van der Aalst, 2000] W.M.P. van der Aalst. *Inheritance of Interorganizational Workflows: How to Agree to Disagree Without Loosing Control?* BETA Working Paper Series, WP 46, Eindhoven University of Technology, Eindhoven, 2000.
- [van der Aalst y ter Hofstede, 2000] W.M.P. van der Aalst and A.H.M. ter Hofstede. *Verification of Workflow Task Structures: A Petri-net-based Approach*. Information Systems, 25(1):43--69, 2000.
- [van der Aalst y Weske, 2001] WMP van der Aalst y Mahias Weske, *The P2P Approach to Interorganizational Workflows*, in proceedings of the CAISE'2001, pp. 140-156, 2001.
- [Wainer, 2000] Jacques Wainer, *Logic representation of processes in work activity coordination*, SAC'00, 2000.
- [Wang, 1998] A. I. Wang, "Use of Object Orientation in Process Modeling Languages", Technical report May 1998.
- [Warboys et al., 1999] *Collaboration and Composition: Issues for a Second Generation Process Language* B.C. Warboys, D. Balasubramaniam, R.M. Greenwood, G.N.C. Kirby, K. Mayes, R. Morrison and D. Munro In O. Nierstrasz and M. Lemoine (Eds.) *Proceedings ESEC'99, Lecture Notes in Computer Science, LNCS 1687*, Springer-Verlag, Toulouse, France, September 1999. pp. 75-91
- [Weske, 1999] M. Weske, "Workflow Management Systems: Formal Foundation, Conceptual Design, Implementation Aspects". PhD Thesis. Münster, 1999.
- [Winograd y Flores, 1986] Winograd, T. and Flores, F., *Understanding Computers and Cognition*, Addison-Wesley Publ. Co, 1986.
- [WFMC, 1999] *Workflow Management Coalition Terminology & Glossary, Issue 3.0*, Document Number WFMC-TC-1011 Feb 99.
- [WFMC, 1999b] *Workflow Management Coalition Workflow Standard-Interoperability Abstract Specification*. Document Numver WFMC-TC-1012. Version 2.0b (Draft) Noviembre 1999.
- [WFMC, 2000] *Workflow Management Coalition Workflow Standard-Interoperability Wf-XML Binding*, Document Number WFMC-TC-1023, Versión 1.0, Mayo 2000.