Arquitectura de Software para Dominio Específico (DSSA) : Su Aplicación en la Reutilización de Software

Sandra Oviedo, Daniel Díaz Araya, Aldo Cáceres, Sergio Zapata
Departamento e Instituto de Informática
Universidad Nacional de San Juan,
Cereceto y Meglioli (5400), San Juan, Argentina
e-mail: {sandrao, ddiaz, acaceres, szapata}@iinfo.unsj.edu.ar

1. RESUMEN

Un análisis comparativo de sistemas de software muestra que en general hay un 60 a 70 porciento de similitudes entre sistemas de software. Esto incluye similitudes en el código, diseño, funcionalidades y arquitecturas [McClure94]. Estas similitudes, por un lado, fundamentan la práctica de la reutilización, para desarrollar nuevas aplicaciones mediante el ensamble de componentes reutilizables predefinidos. Por otro lado, mediante el estudio de las aplicaciones existente de software es posible determinar, antes de desarrollar un nuevo software en el mismo dominio, qué módulo de construcción tendrá.

En la práctica de la reutilización surgen dos problemas fundamentales: la ausencia de artefactos de software, y la existencia de artefactos dificultosos de integrar. Una forma de resolver estos problemas es mediante la construcción de una arquitectura de software para un dominio especifico ó DSSA (domain-specific software architecture).

En el presente trabajo se presentan a las DSSA como un medio para administrar la reutilización, como así también una metodología para obtener una DSSA exponiendo los resultados obtenidos, a la fecha, de la aplicación de la misma en un dominio de Bibliotecas.

2. INTRODUCCION

Reutilización de software es el proceso de construir nuevos sistemas de software a partir de componentes que fueron creados para ser reutilizados [McClure94].
Para la formalización de la reutilización es esencial el análisis de dominios [McClure94].
En un contexto amplio, un dominio es "una esfera de actividad o interés, también denominado campo" [PD85]. Un dominio específico es un área de aplicación, un campo para el cual los sistemas de software fueron creados.
El análisis de dominios es el proceso mediante el cual la información utilizada en .el desarrollo de sistemas es identificada, capturada y organizada con el propósito de hacerla reutilizable en la creación de nuevos sistemas de software [PD90] para tal dominio.

Utilizando el conocimiento adquirido durante el análisis de dominios, los ingenieros de dominio desarrollan una arquitectura genérica, que les permitirá obtener componentes reutilizables catalogados en una biblioteca de reutilización para uso de los ingenieros de aplicación. Esta actividad es regida por la arquitectura ya que el desarrollo de los componentes está basado en ella. El objetivo principal de un proceso conducido por una arquitectura es lograr la reutilización de "cajas negras". Si se logra visualizar un sistema o subsistema como una caja negra que reúne ciertos requerimientos y si estos sistemas son reutilizados en la construcción de otros sistemas, entonces se ha reducido potencialmente el costo total de desarrollar un sistema/subsistema que satisfaga los requerimientos de la reutilización[1] [CARDS94].

Una de las maneras de obtener una Arquitectura Genérica, es mediante la construcción de una Arquitectura de Software para Dominios Específicos ó DSSA (Domain-Specific Software Architecture), las cuales a la vez ofrecen un medio para administrar la reutilización.

Los procesos que soportan la reutilización en un dominio específico, es decir, aquellos procesos que permiten la obtención de una arquitectura genérica, constituyen la ingeniería de dominios, comienzan con el análisis de dominios, desarrollan un modelo de dicho dominio y luego crean la arquitectura genérica en base al dominio modelado [CARDS94].

Las actividades de Ingeniería de Dominios pueden describirse como análogas con las actividades de la Ingeniería de Aplicación. Aunque es necesario recalcar que son dos procesos bastante distintos. Los productos (resultados) de la Ingeniería de dominios pueden ser (y en realidad son pensados para ser) utilizados en alguna actividad de Ingeniería de Aplicación la cual puede proporcionar feedback que podría influir en futuras actividades de Ingeniería de Dominios. El centro de la Ingeniería de Aplicación es un sistema único, en cambio la Ingeniería de Dominios centraliza su acción sobre múltiples sistemas relacionados dentro de un dominio.

3. LA METODOLOGÍA

Una DSSA es un contexto para patrones de elementos problema, elementos solución y situaciones que definen mapping entre ellos. Un elemento problema puede estar asociado a varios elementos solución.

Como se verá luego, al crear una DSSA se determinan los elementos problema que constituyen el espacio problema, y también se obtiene un conjunto de elementos solución que después de la aplicación de un conjunto de restricciones de implementación, constituyen el espacio solución, figura 1.

[1] Este concepto de Megaprogramación ha sido fuertemente adoptado por el departamento de defensa de los EE.UU. (DoD)

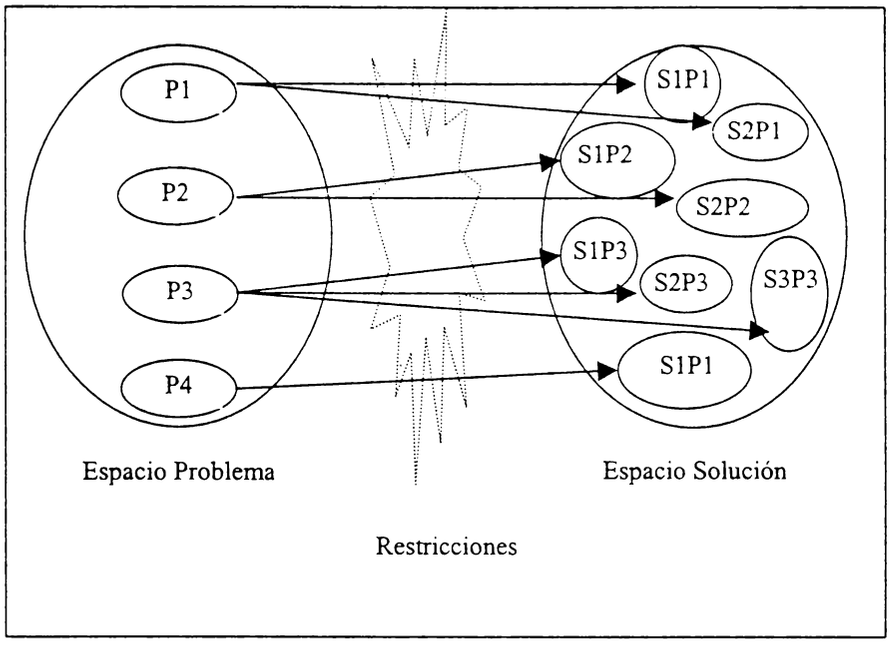

Figura 1 – Esquema de una DSSA

Una de las diferencias entre estos procesos de ingeniería de dominios y otros procesos de análisis de dominios, es que en una DSSA las necesidades del usuario se clasifican en requerimientos al sistema y restricciones de implementación, mientras que las otras metodologías solo consideran "requerimientos" [Tracz95].

Otra diferencia, es que el razonamiento basado en casos y la ingeniería reversa no son el mecanismo central para la identificación de recursos reutilizables [PD87] en su lugar existen aplicaciones que son usadas como vehículo para validar la arquitectura que es derivada mediante un diseño top-down desde la generalización de los requerimientos del usuario.

Cuando se hace uso de la arquitectura deben clasificarse las necesidades del cliente en requerimientos funcionales y restricciones de implementación. Los requerimientos funcionales establecerán la correspondencia entre necesidades del usuario y los elementos problema, por transitividad se vinculan las necesidades del usuario con los elementos solución, de esta manera queda definido un espacio solución. Luego mediante la aplicación de las restricciones de implementación este espacio solución es reducido, figura 2.

Como se observa en la figura 1, los elementos del espacio del problema son P1, P2, P3 Y P4, para los cuales la arquitectura ofrece las siguientes soluciones:

Para: P1 -> S1P1 y S2P1
P2 -> S1P2 y S2P2
P3 -> S1P3, S2P3 y S3P3
P4 -> S1P4

La arquitectura debe ser lo bastante genérica para poder aplicarse en resolución de problemas similares de diferentes contextos, al usarse, figura 2, como ya se dijo, las necesidades del cliente deberán clasificarse en: requerimientos al sistema (por ejemplo, imprimir un reporte), y en restricciones de implementación (por ejemplo, solo se puede imprimir en impresoras con matriz de punto). Los requerimientos se identifican con los elementos problema de la arquitectura y las restricciones de implementación se aplicaran sobre el enlace de cada elemento problema con el/los elementos solución propuestos, así estos enlaces algunos persistirán y otros se eliminarán. Luego quedaran relacionados los

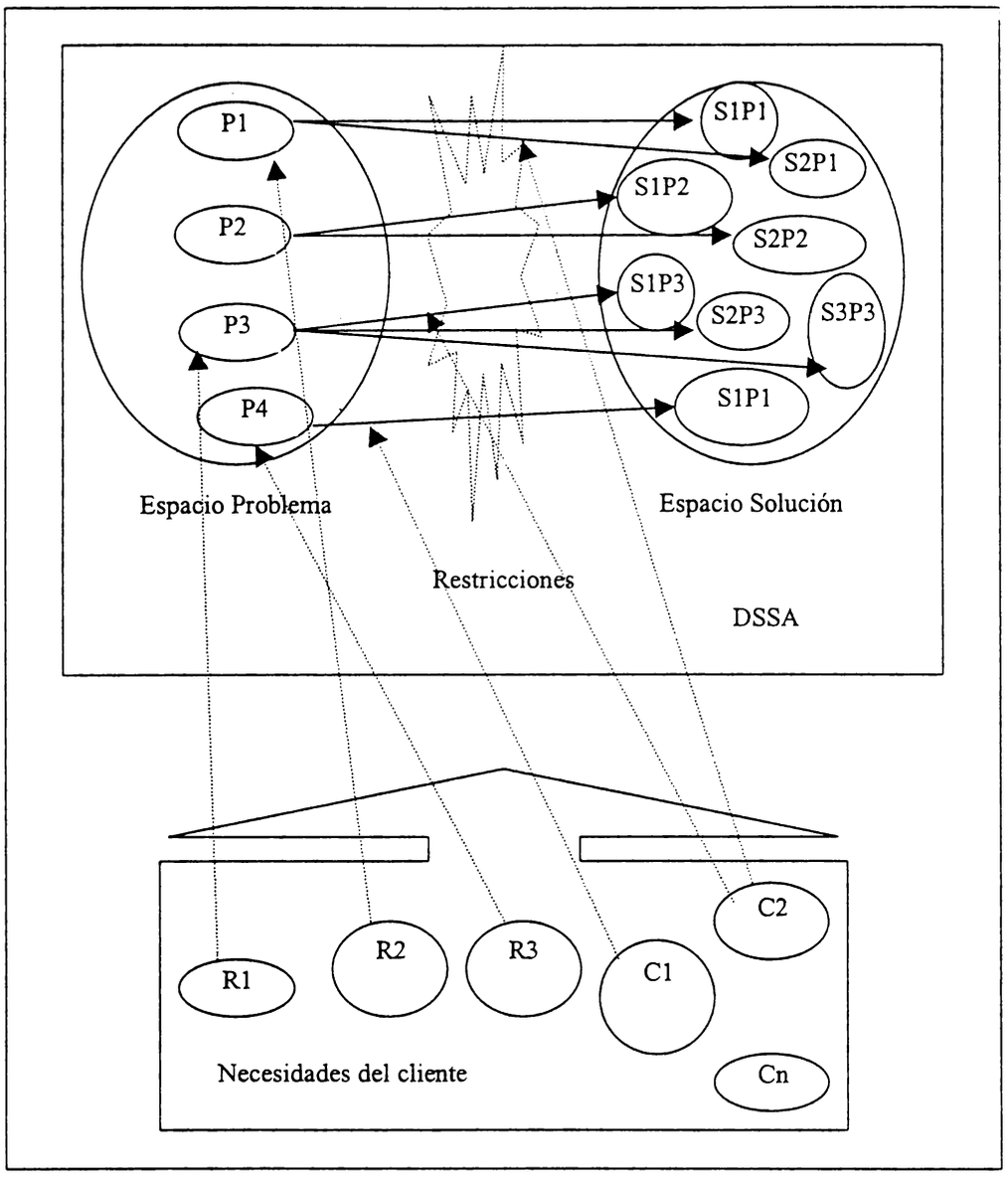

Figura 2- Uso de una DSSA.

requerimientos del cliente con soluciones 'aplicables' a su problema. En la figura 2 se observa lo siguiente:
Para el requerimiento R1 (P3) se proponen tres soluciones S1P3, S2P3 y S3P3 al aplicar las restricciones de implementación: C2 queda restringido el uso de la solución S1P3, por lo tanto el requerimiento R1 puede satisfacerse mediante S2P3 ó S3P3, el usuario de la arquitectura podrá elegir entre ambas soluciones propuestas, y así con los demás requerimientos.

De esta manera, una Arquitectura de Software para un Dominio Específico proporciona una familia de soluciones a un conjunto de requerimientos de una aplicación específica, los cuales definen el dominio del problema.

Las actividades que definen una DSSA se ordenan en 5 etapas[Tracz95], y son los siguientes:

1- Definir el alcance del dominio (definir lo que puede ser consumado, poniendo enfasis en las necesidades del usuario).

2- Definir/Refinar les elementos del dominio especifico (de modo similar al analisis de requerimientos, poniendo enfásis en el espacio del problema).
3- Definir/Refinar requerimientos y restricciones de diseño e implementación del dominio especifico.
4- Desarrollar modelos/arquitecturas del dominio, similar a un diseño de alto nivel con enfasis en la definición de interfaces modulo/modelo y semánticas.
5- Producir/Reunir productos reutilizables: Implementación/colección de artefactos reutilizables (ej. código, documentación, etc.)

Cada una de estas etapas generan feedback para las anteriores, es un proceso iterativo y recursivo.

4- LA APLICACION

4.1 Etapa 1: Identificación del Dominio

El propósito del modelado del dominio es proveer a quien desarrolla o mantiene aplicaciones en un dominio un entendimiento claro y preciso de varios aspectos del dominio.
Como ya se dijo, en las actividades de ingeniería de dominios para obtener una DSSA se considera una aplicación como vehículo para validar la arquitectura, en la presente aplicación se consideró el ámbito de una biblioteca.
Comenzamos con el relevamiento de los requerimientos del usuario mediante una entrevista con un experto del dominio. De la información obtenida se redactan los escenarios[2], por cuestiones de espacio solo se muestra una porción de ellos en la figura 3. A esta altura si se considera necesario, se puede comenzar la realización de un diccionario de términos del dominio, donde constaran aquellas palabras o

Escenario Circulación de libros.
Solicitud de un libro. Préstamo.

- *El Lector consulta al bibliotecario si un determinado libro está disponible.*
- *El agente consulta al sistema sobre la existencia del libro, pueden presentarse las siguientes situaciones:*
- *El libro está disponible: Se realiza el préstamo de la siguiente manera: Se registra el número de socio, la fecha de préstamo, la fecha de vencimiento y las condiciones del préstamo. Si el socio solicita el mismo libro inmediatamente después de haberlo devuelto, se considera una renovación del préstamo, en tal caso se verifica la cantidad de renovaciones que haya realizado sobre el mismo libro, si no supera las tres se presta y se registra una renovación más, en caso de haber cumplido las tres renovaciones no se puede prestar al mismo socio durante las próximas 24 hs..*
- *El libro no está disponible y no existe en el inventario, en tal caso se informa al solicitante la- no existencia de tal libro y se ingresa al Catálogo de Desiderata.*
- *El libro existe y no está disponible, en ese caso se puede reservar, se identifica el número de inventario de la obra solicitada cuya fecha de devolución sea más próxima y se marca como reservado, informándose tal fecha al solicitante del libro...*

Figura 3- Escenarios

frases usadas en

los escenarios y que se considere necesario explicarlos o hacer aclaraciones al respecto, figura 4.

[2] Un escenario es una secuencia de eventos que se producen durante la ejecución concreta de un sistema. Un escenario puede incluir a todos los eventos o solamente aquellos eventos que afecten ciertos objetos del sistema.[Rumbaugh96].

Queda determinado el alcance y los límites del dominio de interés lo que se refleja en un diagrama bloque de alto nivel ó diagrama de contexto.

Las salida de la primera etapa de estos procesos de Ingeniería de Dominios son (además de los escenarios y el diccionario de datos del dominio, que a esta altura es optativo):

- Una lista de las necesidades a ser satisfechas por las aplicaciones en este dominio, figura 5, (por cuestiones de espacio se presenta solo una porción de dicha lista):

- Un diagrama bloque del dominio de interés que incluye entradas y salidas al dominio y unidades funcionales/elementos de alto nivel y la relación entre ellos, figura 6.

Diccionario de datos del dominio:

Bibliotecario: es la persona idónea que interactúa con el sistema para atender a los lectores.

Catálogo: Catálogo es el archivo que contiene todas las obras completamente descriptas, la diferencia entre Catálogo e Inventario es que el Inventario además de los libros tiene registrado todos los elementos propiedad de la institución, mientras que el Catálogo tiene solamente las obras, cada obra tiene un número de catálogo y todos los libros de la misma obra tendrán el mismo número de catálogo y un número único para cada unidad de Inventario.

Inventario: Es el archivo que contiene todos los elementos, no solo libros de la Universidad, cada elemento al ingresar se registra con una breve descripción y se le asigna un número de inventario.

Lector: Es la persona que tiene acceso a los elementos de la biblioteca, también se llama socio, cada socio se identifica con un número único, si es un alumno será el número de Registro como alumno en su facultad, sino se identificará por su número de documento.

Libro disponible: Es cualquier ejemplar que se encuentra en los estantes de la biblioteca. Puede ser prestado ya sea para consulta o a domicilio.

Socio: Ver Lector.

Figura 4- Diccionario de datos

- *Por cada libro se registra la siguiente información: ...información descriptora de libro (título, autor, editorial, año, nro. De catálogo, código Dewey, materia, etc).*
- *Los libro se pueden prestar a domicilio o en sala, los préstamos a domicilio se realizan por 7 días, los préstamos se pueden renovar hasta 3 veces.*
- *La sanción por demoras en la devolución consiste en una suspención que inhabilita a utilizar la biblioteca por una cantidad de días igual a la demora incurrida.*
- *Pasados los 15 días de demora en una devolución se envía un memorandum al lector en mora.*
- *Si un lector solicita un libro que está prestado a otro lector, puede reservarlo.*
- *El sistema debe ser lo suficientemente ágil como para asegurar un pronto despacho a cada solicitud.*

Figura 5- Lista de necesidades del cliente

- Otra información relevante como sería el nombre de personas conocedoras del dominio bajo estudio, documentación de sistemas existentes, etc.

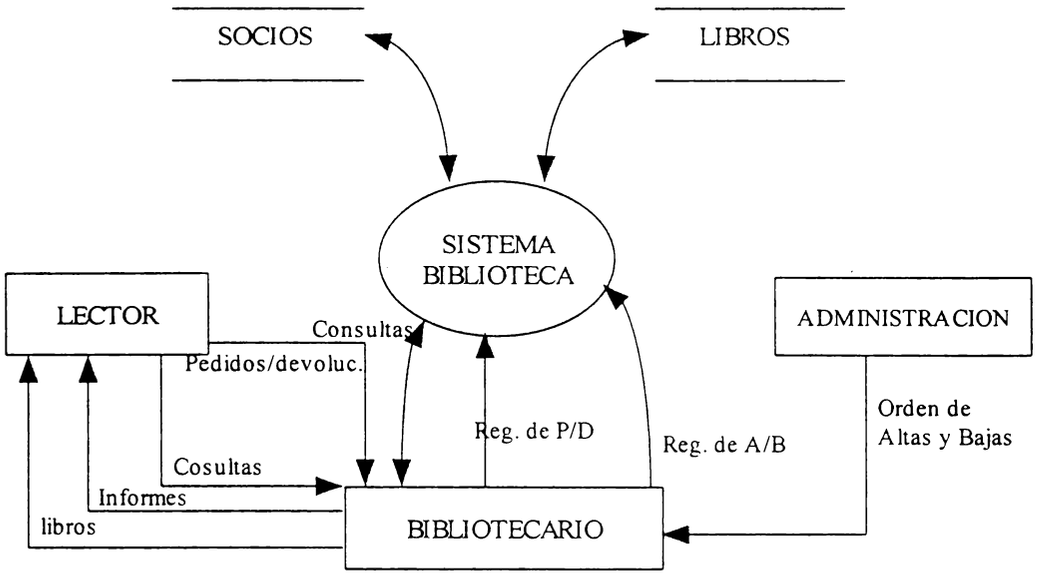

Figura 6 - Diagrama de contexto

4.2 Etapa 2: Definir y refinar elementos

En la segunda etapa definimos y agregamos detalle a los elementos determinados en la etapa anterior. Identificamos los elementos, sus atributos, flujos de datos y de control y relaciones entre los mismos.

Para lograr lo antes dicho realizamos:

- Diagrama Entidad Relación y para los elementos que consideremos necesario se realiza el análisis de dos tipos de relaciones de interés (relaciones de agregación y de generalización) figura 7. Partiendo del diagrama Entidad/Relación se llega al Modelo Objeto, éste constituye la primer fase del diseño de interfaces de componentes. Así, este provee una muy valiosa información para la arquitectura de referencia. El analista de dominios no debería ahondar en muchos detalles en el mismo, pues corre el riesgo de restringir demasiado la arquitectura de referencia [Tracz94] figura 8.

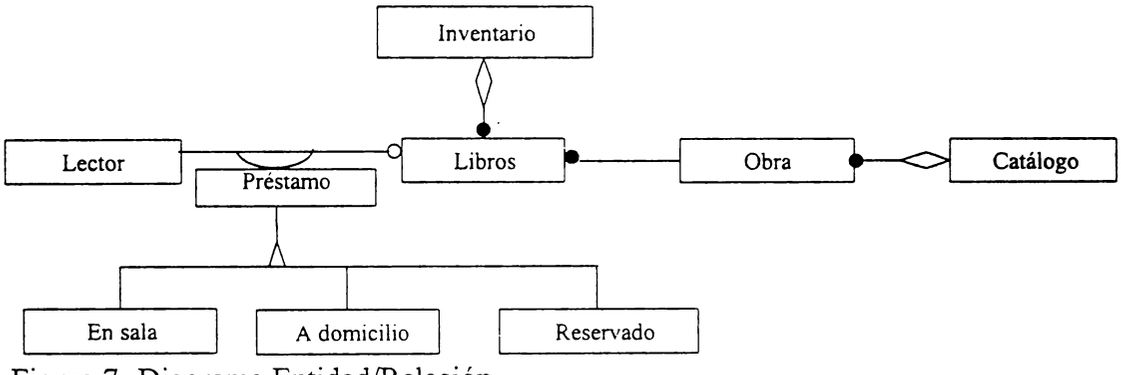

Figura 7- Diagrama Entidad/Relación

Tomando el diagrama objeto identificamos los siguientes elementos, de los que se detallan algunos:

1- Lector
2- Inventario
3- Catálogo
4- Libros
5- Préstamos

bjeto	Atributos	Operaciones
ector	Nombre Nro. de registro Nro. de documento Categoría (alumno, docente, etc.) Domicilio Estado (habilitado, moroso, etc) Antecedentes (suspensiones de los últimos 6 meses)	Registrar antecedentes Listar antecedentes Mostrar estado etc.
ibros	Nro. de Inventario Nro. de Catálogo Estado Estante nro.	Listar libros
réstamos	Nro. de Inventario Nro. de registro Tipo de préstamo (en sala, domicilio) Fecha de Préstamo Fecha de Vto. Reservado (si/no, por quien) Cant.de renovaciones	Listar reservas por día Listar vtos. por día Actualizar reservas (cada 24 hs. caducan)

gura 8- Objetos, atributos y operaciones.

nalizando el Modelo Objeto, advertimos que Reservas se pueden tratar como un préstamo por 24 ras, etc.

- Diagrama de Flujo de Datos, que brinda un Modelo Funcional del sistema., figura 9, cuyo objetivo es agregar detalle a los elementos identificados permitiendo definir más claramente sus atributos.

- Diagrama de Transición de Estados para obtener el Modelo Dinámico, la figura 10, muestra el diagrama de Transición de Estados para el elemento Lector. Este diagrama se hace a los efectos de aclarar detalles para mejor conocimiento de un objeto, su uso es opcional.

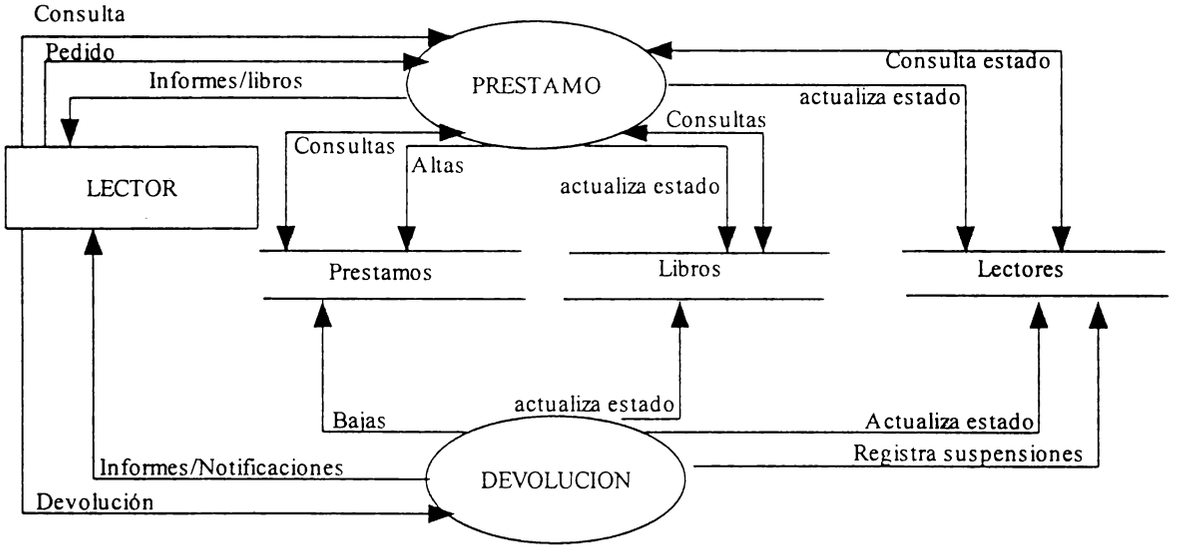

Figura 9- Diagrama de flujo de datos (porción).

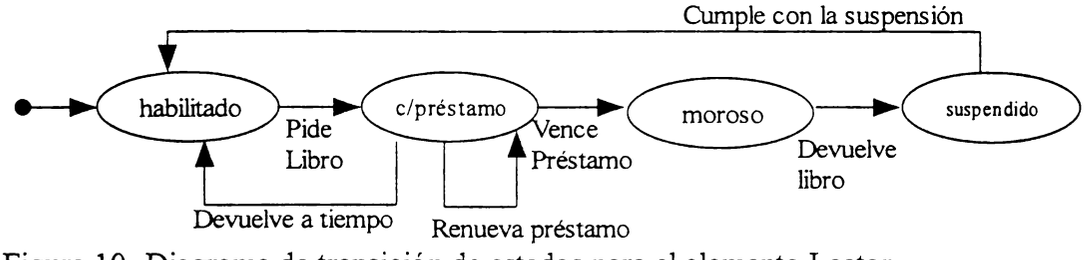

Figura 10- Diagrama de transición de estados para el elemento Lector.

- Diccionario del Dominio: A esta altura del análisis es muy importante su confección ó revisión del que ya existe. Conforme se avance en los procesos de ingeniería de dominios este diccionario va evolucionando, cada modificación que se realiza sobre el modelado del dominio en curso, se debe revisar para actualizar definiciones y mantener la consistencia con los diagramas.

- Se lleva a cabo la creación de un documento de especificación de requerimientos a alto nivel, esta actividad es opcional, si el analista lo considera necesario debe realizarse un mapeo entre los requerimientos del usuario y los elementos identificados en las actividades previas a ésta, para advertir si quedó algún requerimiento insatisfecho. Nosotros en la presente aplicación consideramos que este control se puede realizar mejor mediante una exposición al experto del dominio, considerando las observaciones que el mismo realice.

De esta manera, se llega al **Modelo del Dominio.**

A continuación se realiza una generalización del Modelo del Dominio:

- Generalizando las necesidades inicialmente establecidas por el cliente, puede agregarse capacidades adicionales como por ejemplo:

- Podría considerarse la agregación de un módulo opcional que considere el cobro de un arancel mensual para el uso la biblioteca.
- La pena por el atraso en la devolución de un libro además de considerar la suspensión podría agregarse la opción de cobrar una multa por tal demora.
- Por lector se puede prestar hasta dos libros simultáneamente, podría agregarse la opción de que mediante una autorización se pueda alterar la cantidad de libro que el lector pueda pedir por vez..
- Aunque no lo haremos en el presente estudio, sería muy útil prever la posibilidad de que se pueda consultar títulos, realizar reservas, etc. vía Internet.

- Además, el analistas puede llegar a reconocer que la naturaleza del problema en este dominio es análogo a otros problemas en diferentes dominios, pudiéndose estructurar la arquitectura para potenciar sistemas existentes y extender su alcance a nuevos dominios y así explotar una base más grande de clientes, figura 11.

DOMINIO DE BIBLIOTECAS	DOMINIO DE ALQUILER DE PELICULAS (VIDEO CLUB)	DOMINIO DE ALQUILER DE AUTOS (RENT A CAR)	GENERALIZANDO EL DOMINIO
LIBROS	PELICULAS/CD	VEHICULO	ITEM DE ALQUILER
LECTOR	SOCIO	CLIENTE	USUARIO
PRESTAMO	ALQUILER	ALQUILER	ALQUILER
DIAS DE SUSPENSIÓN	MULTA (POR ATRASO)	MULTA (POR ATRASO)	SANCION POR ATRASOS
RESERVA	RESERVA	RESERVA	RESERVAS
MOROSOS	MOROSOS	MOROSOS	MOROSOS
BIBLIOTECARIO	EMPLEADO	EMPLEADO ENCARGADO	RESPONSABLE
NRO. DE INVENTARIO	NRO. DE INVENTARIO	PATENTE DE VEHICULO	IDENTIFICADOR ITEM
NRO. DE REGISTRO	NRO. DE SOCIO	NRO. DE CLIENTE	IDENTIFICADOR USUARIO
ESTANTE	ESTANTE	COCHERA	HUBICACION
AUTOR	DIRECTOR	MARCA	DESCRIPTOR ORIGEN
TITULO	TITULO	MODELO	DESCRIPTOR TIPO
COD. CATALOGO	COD. CATALOGO	COD. DE DESCRICCIÓN TECNICA	IDENTIFICADOR DE DESCRIPTOR

Figura 11- Comparación entre dominios de Bibliotecas, Alquiler de videos y de autos y generalización de dominios similares.

4.3 Etapa 3: Análisis de requerimientos

En la tercera etapa de los procesos de Ingeniería de Dominios se lleva a cabo el análisis de los Requerimientos de referencia.
El arquitecto de Dominios usa los Requerimientos de Referencia para conducir el diseño de la arquitectura de referencia. El modelo de dominio define el comportamiento de las aplicaciones en el sistema a través de escenarios, diagramas de flujo de dato y de transición de estados. El siguiente paso en el proceso de una DSSA es identificar la porción del espacio de solución que puede mapearse con el modelo del dominio (espacio problema).

Los requerimientos funcionales definen las características del espacio problema. Los requerimientos no funcionales, de diseño e implementación limitan las características (restringen) el espacio solución[3].

[3] Rubén Prieto-Diaz en Establishing Global Requirements de STAR Domain Analysis Activities [Tracz95] distinguió dos tipos de requerimientos:

- Requerimientos estables, los que no cambian de una aplicación a otra.
- Requerimientos variables, aquellos que sí cambian.

Siguiendo la separación tradicional de los requerimientos en los "Qué" y los "Cómo", se puede concluir que:
Los requerimientos funcionales se pueden identificar con los requerimientos "Qué" y
los requerimientos no funcionales lo hacen con los requerimientos "Cómo". Así, en nuestra aplicación encontramos los requerimientos que se listan en la figura 12, entre otros.

Requerimientos Funcionales:

- *Lectores:*
  - *Disponibilidad:*
    - *El sistema debe permitir que mediante la identificación de un lector éste pueda acceder al material disponible de la biblioteca.*
    - *El sistema podría permitir que se 'personalice' la cantidad de libros que se puede prestar por lector.*
  - *Reportes:*
    - *El sistema debe permitir imprimir informes de morosos, suspendidos y cantidad de suspenciones que tuvo un lector en los últimos seis meses.*
    - *El sistema al finalizar cada jornada deberá informar los morosos que se generen.*
- *Libros:*
  - *Consultas:*
    - *El sistema deberá permitir la consulta acerca de obras, autores, ediciones, etc.*
  - *Préstamos:*
    - *El sistema debe permitir que un libro disponible sea prestado a un lector autorizado.*
    - *Al finalizar la jornada, antes del cierre del sistema, deberá actualizarse automáticamente el registro de morosos.*
    - *Al registrarse un préstamo el sistema debe informar fecha de vencimiento del préstamo.*
    - *Al registrarse una devolución de un préstamo vencido el sistema debe informar la suspención ó multa a aplicar al lector.*
    - *El sistema debería permitir un registro de estadísticas de libros prestados, devueltos, reservados por día, así como de moras incurridas diariamente.*
  - *Reservas:*
    - *El sistema debe permitir que se pueda reservar un libro que se encuentre, en el momento de su solicitud, prestado a otro lector.*

Requerimientos no funcionales:

- *Libros*
  - *Préstamos:*
    - *mediante el registro del Identificador del Lector y el n° de Inventario del libro.*
    - *Cada préstamo puede renovarse hasta tres veces seguidas.*
  - *Reservas:*
  - *El sistemas deberá hacer que las reservas caduquen a las 24 horas.*
- *Lector*
  - *Renovaciones*
    - *El sistema debe permitir a cada lector renovar hasta tres veces seguidas un préstamo.*
- *Del Sistema:*
  - *Seguridad-Opt:*
    - *El bibliotecario deberá acceder al sistema mediante una clave conocida únicamente por él.*
- *Tolerancia a fallas:*
  - *Ante eventuales cortes de energía eléctrica el sistema deberá prever la no perdida de datos que se estén registrando.*

Figura 12- Clasificación de necesidades en requerimientos funcionales y no funcionales (restricciones de implementación).

4.3.1 Decisiones de diseño

El arquitecto del dominio se encuentra frente a una serie de decisiones de diseño que debe considerar. Las decisiones más significativas son en torno al estilo de arquitectura a adoptar (si será por capas, jerárquica, un flujo de datos, etc). El estilo de la arquitectura, además de afectar el rendimiento y costos de desarrollo, afectará el estilo de interface entre los componentes correspondientes [Tracz94].
Otras decisiones de diseño son relativas al estilo de interface de usuario (por ejemplo: menus pop up, teclas de función, líneas de comando, etc), esto dependerá del tipo de hardware y sistema operativo que se seleccione. [Tracz94].
Por ejemplo:
Interface de usuario (alt-1): El sistema puede proveer un entorno visual para su operación.
Interface de usuario (alt-2): el sistema puede proveer una interface de usuario con comandos en línea.

Obviamente , la arquitectura de referencia podría diseñarse para soportar numerosos estilos de interface de usuarios, sistemas operativos, etc.

4.3.2 Decisiones de Implementación

Los requerimientos[4] de implementación son similares a los requerimientos de diseño ya que el analista o arquitecto del dominio necesita determinar el subconjunto de varias opciones de implementación que podrán dirigir el diseño de la arquitectura de referencia y la implementación de los respectivos componentes.
Por ejemplo:
Lenguaje de programación: Lenguaje Visual
Sistema Operativo: Windows
...

En el diccionario del dominio debe haber una muy clara referencia a estos requerimientos, con una terminología que no dé lugar a ambigüedades y manteniendo la consistencia.
Además la especificación de los requerimientos de referencia debe capturar el razonamiento y la interdependencia entre los requerimientos, esta información es muy útil para el entendimiento por parte del ingeniero de aplicación que es quien deberá configurar los componentes.

4.4 Etapa 4: Arquitectura de referencia

Una arquitectura de referencia es un diseño parametrizado que satisface un subconjunto claramente distinguido de capacidades funcionales identificadas en los requerimientos de referencia dentro de los límites de ciertas restricciones de diseño e implementación también identificadas en los requerimientos de referencia [ARPA].

---

[4] Las decisiones de diseño e implementación en su mayoría serán extraídas de restricciones impuestas por el usuario de la aplicación bajo análisis, por lo cual puede tranquilamente llamárselas requerimientos (no funcionales), por su puesto, habrá otras decisiones totalmente sujetas a lo que el arquitecto disponga.

4.4.1 Modelos de arquitecturas de referencia

Todas las decisiones empiezan con una simple abstracción basada en algún estilo de arquitectura.
La figura 13 muestra una arquitectura de capas a muy alto nivel que lo único que permite observar es que las interfaces de usuario pueden estar separadas de la funcionalidad, esto implica que pueden existir una familias de componentes compatibles (plug-compatible) entre los que se puedan elegir componentes para las distintas capas.

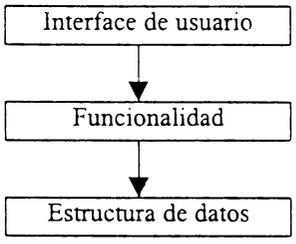

Figura 13-Modelo de arquitectura en capas, muy simple.

Si damos detalles a este modelo de referencia tan sencillo podemos obtener una arquitectura de referencia para el dominio de bibliotecas:

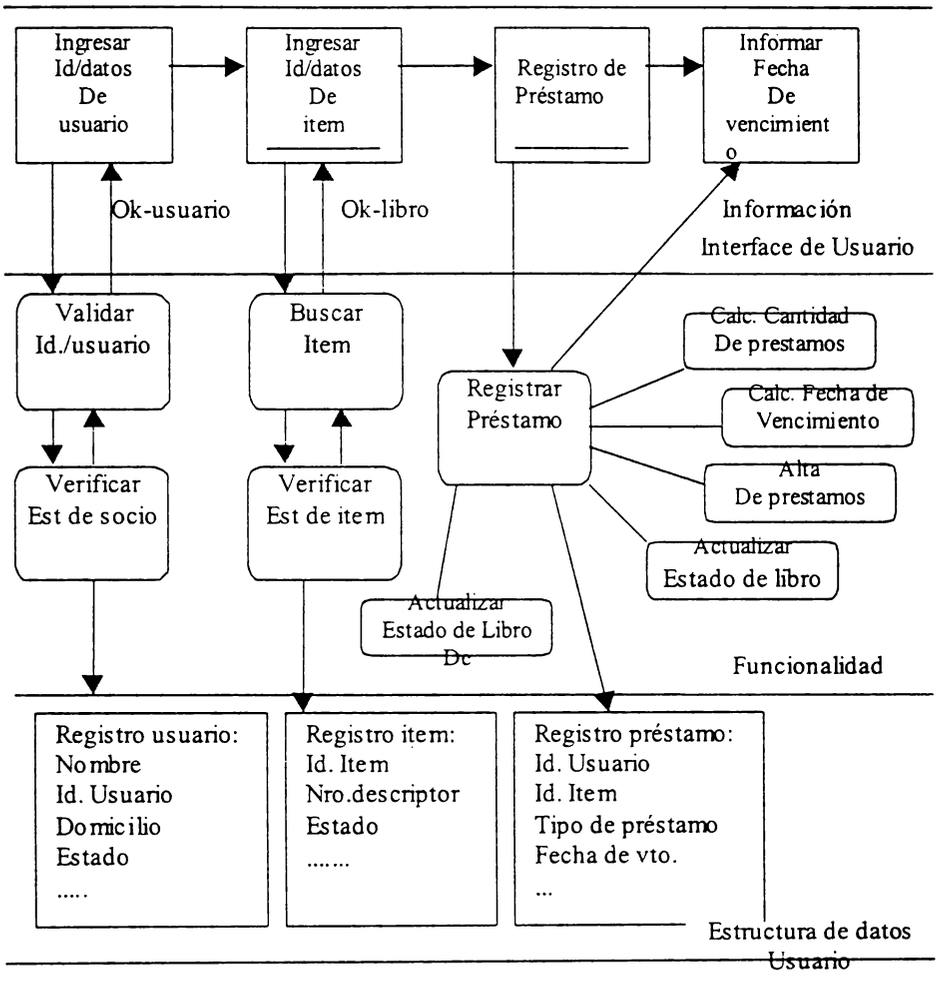

Figura 14- Primera versión de una arquitectura para una parte del dominio analizado.

La arquitectura de referencia debe complementarse con:

- **Esquema de la arquitectura o registro de diseño**: cuyo propósito es servir como un vehículo para el entendimiento del software . Este se presenta como una recopilación de conocimiento sobre los elementos que componen la DSSA.
- **Diagrama Arbol de desiciones**, si es necesario para aportar claridad a la arquitectura.
- **Descripción de interface de componentes**, mediante una notación estandarizada se describe la interface que tiene cada componente de la DSSA.
- **Restricciones y razonamiento**: el atributo fina de una arquitectura es la composición y las restricciones de configuración, además del razonamiento que debe emplear el ingeniero de aplicación en los procesos de generación de aplicaciones. Estas restricciones y razonamientos pueden tomar la forma tradicional de las reglas de los sistemas expertos o pueden ser un texto informal como parte del registro de diseño o esquema de la arquitectura. Las restricciones indican rango o parámetros de valores, relaciones entre parámetros y componentes, etc. El razonamiento puede ser la forma en que se derivaron las reglas, o lecciones de aprendizaje, etc.

4.5 Etapa 5: Poblar la arquitectura

Si el objetivo del analista es crear una base de conocimiento para soporte de una aplicación existente, este paso no es necesario, pero una DSSA se explota mejor si se emplea para identificar componentes de software reutilizable existentes o componentes que pueden servir como base para la creación de componentes reutilizables, así como en la importación de componentes reutilizables de otros dominios. Una DSSA es muy útil para regir el desarrollo de bibliotecas de componentes de un dominio específico.

5. CONCLUSIONES

Por lo expuesto en este trabajo se concluye que las DSSA distintas características que hacen de esta una herramienta muy favorable para administrar la reutilización, entre ellas se mencionan:

- Una DSSA no proporciona sólo un framework para componentes de software reutilizables (a nivel implementación) que puedan adaptarse a ella, sino que también captura el razonamiento del diseño y provee un cierto grado de adaptabilidad.

- Las DSSA pueden verse como elementos mediadores entre la generalidad y la especialización, a la vez que acotan los componentes a un dominio específico, los generaliza lo suficiente haciéndolos aplicables a cualquier contexto de dicho dominio.

- Por su concepción una DSSA rescata todas las similitudes de las distintas aplicaciones de un dominio, potenciando la producción de elementos reutilizables.

El trabajo aquí presentado se desarrolla dentro del marco del proyecto ISRI (Information Systems Reuse on Internet) en conjunto con la Universidad de Vigo (España)

Referencias

[McClure94] Carma MacClure. Reuse Ingineering: extending information engineering to enable software reuse. Extended Intelligence, Inc. 1994

[PD85] Rubén Prieto-Diaz. Domain Analysis: an introduction. The Contel Technology Center, 1985.

[CARDS94] Central Archive for Reusable Defense Software, STARS-VC-B017/001/00 25 March 1994.

[Tracz95] Will Tracz. Cofessions of a Used Program Salesman: Intitutionalizing Software Reuse. Addison Wesley, 1995

[Tracz94] Will Tracz. Lockheed Martin: DSSA (Domain-Specific Software Architecture) Pedagogical Example,1994.

[PD87] Rubén Prieto-Diaz. Domain Analysis for reusability. IEEE Computer Society Press, Los Alamitos, CA, 1987.

[Rumbaugh96] James Rumbaugh. Modelado y Diseño Orientado a Objetos. Prentice Hall, 1996.

[ARPA] Architecture – Based Adquisition and Development of Software Guidelines and Recomendations for the ARPA Domain –Specific Software Architecture (DSSA) Program. Technical Report, Tecknowledge.