Sociedad Venezolana de Computación

Revista Venezolana de Computación

REVECOM

# Revista Venezolana de Computación

## Sociedad Venezolana de Computación

ISSN: 2244-7040

Vol. 3, No. 1
Junio 2016

REVECOM

# Revista Venezolana de Computación

Sociedad Venezolana de Computación

Editores:
Eric Gamess, Wilmer Pereira, Yudith Cardinale

ISSN: 2244-7040

Vol. 3, No. 1
Junio 2016

# Editorial

## Acceso Abierto y Seguro en las Revistas Digitales

Entre los numerosos cambios que ha generado Internet en nuestra sociedad está la manera en cómo nos relacionamos con el acceso a la información. La privacidad y los derechos de autor parecen tener un significado diferente en el mundo digital y esos valores aparentan ser menos restrictivos cuando se refiere a la propiedad de los productos intelectuales disponibles desde Internet. Más aún, la proliferación de las técnicas de violación del *copyrigth* e infiltración de sistemas ha sido el punto de partida de una corriente social, ligada a aspectos muy técnicos, conocida como el *hacking* oscuro o "*hacking* de sombrero negro".

Inicialmente el término *hacking* nació a comienzos de la década de los sesenta en el MIT y se asociaba a personas con un alto nivel de experticia técnica, sobre todo en el área de informática. Sus conocimientos eran muy valorados en centros de investigación y empresas en búsqueda de la innovación tecnológica. Pero con la aparición de la interconectividad de los sistemas, los *hackers* centraron su atención en el potencial que aportaban las conexiones remotas, y el panorama cambió. Desafortunadamente, se había abierto la posibilidad de que algunos *hackers* inescrupulosos podían acceder a información remota que no necesariamente era pública ni de contenido abierto, la cual podían alterar o eliminar. En muchos casos, el periodismo tecnológico contribuyó a darle importancia a esta concepción del término que está al borde de la legalidad. Esta cobertura mediática se centraba en el trabajo fraudulento de los *hackers* y hacía olvidar al gran público los aspectos benéficos del "*hacking* de sombrero blanco". Más aún, en muchos países con bajo poder adquisitivo, la apropiación de productos, sin respetar los derechos de autor, le daba peso a esta corriente no ética del *hacking* y se ha convertido en un problema legal que traspasa las fronteras.

Las revistas no están exentas de este problema, tanto para los autores como para las grandes editoriales que editan actas de conferencias o preparan *journals* de temas de investigación muy específicos. Por ejemplo, las editoriales sufren un número creciente de ataques a sus plataformas, en parte debido a sus modelos de distribución de información no gratuita, que generan inconformidad en los defensores de la circulación del conocimiento libre. Adicionalmente a los problemas de seguridad, hay otras situaciones que incomodan en el manejo de muchas revistas. Una de estas situaciones aparece bajo el siguiente escenario: los editores solicitan a los autores renunciar al *copyrigth* para que la editorial pueda comercializar la publicación sin cancelar ningún beneficio por el trabajo intelectual de los investigadores. Los autores aceptan este trato desfavorable por el prestigio que representa tener un trabajo publicado en la revista. Otra vertiente del problema es que las bibliotecas de las universidades de países en vías de desarrollo no cuentan con suficientes divisas para abonarse a todas las revistas que podrían ser de interés para los docentes/investigadores. Así que los estudiantes y colegas de los investigadores no pueden descargar legalmente la publicación por lo que se valen de copias ilegales para poder acceder a la información. Muchos docentes permiten esta situación para expandir el conocimiento que construyen, en sus entornos sociales más cercanos, a expensas de infringir acuerdos a los que se pliegan al ceder sus derechos de *copyrigth*.

Desde el punto de vista más técnico, *Open Journal System*, una infraestructura muy popular para la administración y visualización de revistas digitales así como el manejo de artículos para conferencias, sufre de ataques de modificación de la información almacenada. Específicamente no verifica el tipo de información que se sube y permite montar archivos ejecutables que pueden cambiar la información almacenada como borrar o alterar el contenido de un artículo. En particular, no verifica si la información subida es un programa en un lenguaje de programación de páginas Web como lo es PHP. Tampoco verifica que las claves de acceso sean lo suficientemente robustas para resistir ataques de predicción y obtención de claves. Si esta barrera es franqueada, los autores o hasta los propios editores del portal, pueden perder el control de la información almacenada.

Con el fin de mitigar estos problemas, hay esfuerzos más colectivos como el de *Public Library of Science* (PLOS). Este es un proyecto sin fines de lucro que tiene como objetivo crear una biblioteca de revistas científicas bajo una licencia de contenido abierto. Sin embargo, en muchos casos son los autores los que pagan el costo de la publicación y del mantenimiento de la plataforma y, aún cuando puede ser un pago pequeño, resulta prohibitivo para algunos investigadores.

La solución sin duda está en definir medios de difusión abiertos y de contenido gratuito, desligados de las grandes editoriales, robustos desde el punto de vista de la seguridad y que tengan el suficiente prestigio para que sean atractivos a los investigadores. En esa vía se inscriben ReVeCom y otras publicaciones digitales que brindan a los investigadores una alternativa de publicación sin renunciar al *copyrigth* y que les permitan ofrecer, de manera segura, sus productos intelectuales a estudiantes y colegas, sin costo alguno.

Dr. Wílmer Pereira
Profesor de la Escuela de Ingeniería Informática
Universidad Católica Andrés Bello

# Revista Venezolana de Computación

ReVeCom (Revista Venezolana de Computación) es la primera revista venezolana arbitrada, periódica, digital, orienta a la publicación de resultados de investigación en el campo de la computación. ReVeCom fue creada por la SVC (Sociedad Venezolana de Computación) y tiene entre sus objetivos hacer conocer los trabajos de alta calidad investigativa que se realizan a nivel nacional, latinoamericano e internacional. La revista permite la divulgación de artículos con aporte original en castellano o inglés.

En mayo de 2016, se celebró el Cuarto Simposio Científico y Tecnológico en Computación (SCTC 2016), en la Universidad Central de Venezuela, Caracas, Venezuela. El SCTC es un evento bienal cuya finalidad es consolidar el intercambio sobre experiencias investigativas, académicas y tecnológicas, para crear lazos estrechos de cooperación a nivel nacional e internacional. Este evento se llevó a cabo en el marco de las Jornadas de Investigación y Extensión 2016, de la Facultad de Ciencias de la misma Universidad.

La edición de este cuarto número de ReVeCom está dedicada a los mejores trabajos presentados en el SCTC 2016. Esta edición consolida un esfuerzo grande que se ha venido haciendo en el seno de la SVC, para promover la investigación en el campo de la computación a nivel nacional, e impulsar una nueva generación académica y profesional en nuestra área de saber para el desarrollo del país.

ReVeCom es una revista abierta para una mayor difusión de los resultados de investigación. Cuenta con una página web (http://www.svc.net.ve/revecom), donde se encuentran los trabajos publicados e información sobre la revista. La revista promueve la pluralidad de intereses, dando cabida a la divulgación de trabajos de todos los campos del conocimiento inherentes a la computación.

Además de selecciones de los mejores artículos de conferencias, ReVeCom también publica artículos de investigación en el campo de la computación, a través de un arbitraje por expertos del área. Por ende, se hace una invitación amplia a la comunidad informática nacional, latinoamericana e internacional, a someter sus propios trabajos para los números de ReVeCom por venir.

# Directorio de la Sociedad Venezolana de Computación

**Presidente:**
Dr. Wilmer Pereira (Universidad Católica Andrés Bello)

**Secretario:**
Dr. Carlos Acosta (Universidad Central de Venezuela)

**Tesorero:**
Profesora Rosseline Rodríguez (Universidad Simón Bolívar)

**Vicepresidente:**
Dra. Yudith Cardinale (Universidad Simón Bolívar)

**Coordinador de Eventos:**
Dr. Leonid Tineo (Universidad Simón Bolívar)

**Coordinador de Publicaciones:**
Dr. Eric Gamess (Universidad Central de Venezuela)

**Coordinadora de Educación e Investigación:**
Dra. Judith Barrios (Universidad de Los Andes)

# Edición

## Comité Editorial



## Miembros del Comité Editorial

# Tabla de Contenido

**Revista Venezolana de Computación**
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 3, No. 1, pp. 1-12, Junio 2016
Selección de los Mejores Artículos del SCTC 2016

# Ontología para Apoyar al Sector Turismo en Venezuela

Roxydel Dulcey [1], Esmeralda Ramos [2]
roxydeld@gmail.com, esmeralda.ramos@ciens.ucv.ve

[1] Departamento de Informática, IUT - Región Capital Dr. Federico Rivero Palacio, Caracas, Venezuela
[2] Centro ISYS, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Venezuela

**Resumen:** En este trabajo se presenta la construcción de una ontología de turismo y el desarrollo de una aplicación Web que permite acceder a ella, para visualizarla, consultarla y navegarla. Esta ontología apoya al sector turismo en Venezuela y por lo tanto al turista potencial, específicamente en la búsqueda de información al momento de planificar un viaje, ya que desde la aplicación Web se puede acceder a una gran cantidad de información necesaria para realizar esta actividad. La ontología almacena de manera organizada y estandarizada información concerniente a: sitios turísticos, sitios no turísticos así como servicios turísticos de una localidad: estado, ciudad y zona. La aplicación desarrollada posibilita la búsqueda de información por preferencias (característica deseada, palabra clave o por el nombre de un sitio en particular) y por hospedaje (servicios de hospedaje y/o número de estrellas, en el caso de hoteles). Las pruebas realizadas a la ontología y a la aplicación muestran que éstas están en capacidad de proporcionar información oportuna al usuario, según sus requerimientos. Para construir la ontología se utilizó la metodología Methontology y con el editor Protégé se codificó en OWL (Ontology Web Languaje). Para el desarrollo de la aplicación se utilizaron las tecnologías Web (PHP, JQuery, CSS, HTML).

**Palabras Clave:** Ontologías; Turismo en Venezuela; Methontology; OWL; Protégé.

**Abstract:** This paper exposes the building of a tourism ontology and the development of a Web application, which allows the user the possibility to access in order to view, consult and navigate it. Ontology supports the tourism sector in Venezuela and the potential tourist, specifically in search of information to planning a trip, with the application, the tourist will be able to access a large amount of necessary information to develop this activity. The ontology stores organized and standardized information about: tourist sites, non-tourist sites, and tourist services of a locality: state, city or location. The developed application allows the search information by preferences (desired characteristic, keyword or the name of a particular site) and hosting (hosting services and/or number of stars, in the case of hotels). The tests carried out to the ontology and the application disclose that these are able to provide timely information to the user, depending on their requirements. To built the ontology Methontology methodology was used and was codified with the editor Protégé in OWL. For the application development the Web technologies were used (PHP, JQuery, CSS, HTML).

**Keywords:** Ontologies; Tourism in Venezuela; Methontology; OWL; Protégé.

## I. Introducción

El turismo es una actividad que tiene un alto potencial y contribuye al desarrollo socioeconómico y cultural de un país, debido a que es una fuente generadora de empleos y de divisas, que fomenta el intercambio cultural entre regiones, la protección y el cuidado del medio ambiente y el desarrollo de las actividades locales económicas y culturales.

El desarrollo del sector turístico en Venezuela es de gran importancia, ya que se presenta como una fuente de ingresos alterna a la industria petrolera.

El turismo en Venezuela se ha desarrollado considerablemente, sin embargo, podría tener un mayor auge ya que nuestro país es un destino turístico por excelencia, dadas sus riquezas naturales, su clima tropical y los distintos hábitats que posee: montañas, playas, desiertos, selvas, sabanas, llanos, ríos, lagunas, lagos, ciudades, etc.

En la actualidad el sector turismo es apoyado por las tecnologías emergentes, principalmente las TICs (Tecnologías de Información y Comunicaciones) y la IA (Inteligencia Artificial), a través de aplicaciones Web, sistemas multiagentes, aplicaciones basadas en ontologías, sistemas de razonamiento basado en casos, sistemas de recomendación, combinaciones de éstos, entre otros.

Estos sistemas apoyan a los turistas en la planificación de una posible estancia en una localidad determinada, con el objetivo de facilitar su visita a distintos lugares de interés. Para tal fin, estas aplicaciones colocan a disposición del usuario información sobre los lugares, y adicionalmente le permiten realizar búsquedas basándose en sus intereses y preferencias.

Este trabajo se fundamenta en la aplicación de una de estas tecnologías, específicamente el uso de las ontologías como mecanismo para estandarizar, organizar y hacer accesible de manera organizada y estandarizada a información relativa al turismo. En esta investigación se presenta la construcción de una ontología para apoyar al sector turismo en Venezuela, de manera de dar a conocer nuestras maravillas naturales tanto nacional como internacionalmente y así aumentar la actividad en este sector y obtener el sin fin de beneficios económicos, sociales y culturales que provee su desarrollo y en consecuencia el desarrollo del país.

En la Sección 2 de este documento, se plantean algunas carencias de la actividad turística en Venezuela, en cuanto a las fuentes de información centralizadas de asistencia a los viajeros. Seguidamente, en la Sección 3, se presentan los antecedentes a esta investigación, donde se exponen desarrollos de ontologías y desarrollos Web que emplean ontologías, ambos en el dominio turístico. Por último, se muestra el objetivo general y los objetivos específicos de esta investigación. En la Sección 4, se presentan algunas nociones básicas sobre las ontologías y se describe brevemente la metodología utilizada para construir la ontología. La construcción de la ontología de turismo y su evaluación se describen en la Sección 5. En la Sección 6, se expone la arquitectura de la solución propuesta. En la Sección 7, se presenta una breve descripción de la aplicación Web y los resultados obtenidos de las pruebas de recorrido y acceso realizadas a ésta. En la Sección 8, se muestran las conclusiones del trabajo realizado; y finalmente, en la Sección 9, algunas recomendaciones para la continuidad de esta investigación.

## II. Planteamiento de la Situación Turismo en Venezuela

Como se comentó, Venezuela es un destino turístico por excelencia, su mayor potencial es que cuenta con distintos ambientes: montañas, playas, desiertos, selvas, sabanas, llanos, ríos, lagunas, lagos, ciudades, etc. No obstante, actualmente los turistas nacionales e internacionales no disponen de fuentes de información que puedan ser accedidas vía Web, que consideren la información más relevante y pertinente que necesita una persona para viajar a un destino determinado de nuestro país.

En este trabajo se considera que la información más relevante para los turistas es aquella relacionada con: a) Sitios turísticos: museos, parques, plazas, sitios naturales (playas, ríos, cascadas, etc.), sitios históricos, sitios religiosos, entre otros. b) Sitios no turísticos: restaurantes, cines, teatros, centros comerciales, ubicación de embajadas y consulados, entre otros. c) Servicios turísticos: transporte (taxis, metro, buses, lanchas, etc.), hospedaje (hoteles, posadas y campamentos), oficinas de turismo, agencias de viaje, líneas aéreas, entre otros.

Son muchas y variadas las fuentes de información disponibles en este contexto, como por ejemplo: periódicos, revistas, libros, televisión y portales Web; entre estos últimos destacan: venezuelatuya.com, despegar.com.ve, felizviaje.com, hoteles.ve y venetur.gob.ve.

Por ejemplo, en venezuelatuya.com, se puede encontrar información sobre hospedaje, algunos sitios turísticos, paquetes turísticos, entre otros; despegar.com.ve ofrece información de hoteles, vuelos y autos en alquiler; felizviaje.com proporciona información de hoteles y paquetes turísticos de algunos destinos de Venezuela; hoteles.com.ve ofrece información sobre hoteles y paquetes turísticos; y en el portal venetur.gob.ve se puede encontrar información sobre los hoteles y transporte Venetur, y paquetes turísticos nacionales e internacionales.

Como resultado de la revisión de estos portales, se pudo concluir que la búsqueda de información de interés se ve restringida, dado que las fuentes de información se especializan en aspectos muy particulares; lo que obliga al turista a invertir mucho tiempo buscando en diferentes sitios al momento de planificar un viaje.

Sería de gran utilidad que los turistas tengan a su disposición fuentes, donde puedan encontrar información de interés a nivel global, como por ejemplo, un sitio Web que incluya: hospedaje, transporte, sitios turísticos, lugares para comer, entre otros. Adicionalmente, sería interesante que los turistas tuvieran la posibilidad de realizar búsquedas sobre esta información basándose en sus intereses y preferencias. Una herramienta con estas características, le permitirá al viajero ahorrar tiempo y dinero a la hora de buscar la información necesaria para planificar un viaje.

La problemática planteada anteriormente da origen a las siguientes interrogantes:

¿Qué mecanismo sería idóneo para que la información de interés, necesaria para planificar un viaje esté a disposición del viajero de una manera organizada, estandarizada y centralizada?

¿Cómo apoyar al viajero en la búsqueda de información vía Web al momento de planificar un viaje determinado?

## III. Antecedentes de la Investigación

El uso de la inteligencia artificial en la industria del turismo es cada vez mayor, debido al amplio y eficiente servicio que proporciona a los viajeros a través de sistemas inteligentes como: sistemas multiagentes, sistemas de razonamiento basado en casos, ontologías, entre otros. Estos sistemas permiten al usuario consultar información turística de una manera eficiente, realizar búsquedas avanzadas tomando en cuenta sus intereses y preferencias, planificar su estadía en un lugar determinado, entre otras facilidades; por tanto, el usuario ahorra tiempo, costos y consigue de forma eficaz y eficiente la información turística de interés, alcanzando así su satisfacción.

En esta Sección se presenta una revisión bibliográfica relacionada con el uso de ontologías en el dominio turístico.

### *A. Desarrollo de Ontologías*

En [1], se muestran las principales ontologías, taxonomías y glosarios turísticos utilizados hasta el momento. También se crea una ontología turística para representar rutas turísticas en las Valls d'Aneu (subcomarca situada al noroeste del Pallars Sobirá, comarca de la provincia de Lérida en Cataluña) y se muestra como enlazar dicha ontología con otras ontologías existentes para hacerla más genérica y reusable.

Un sistema que genera rutas turísticas adaptadas a las preferencias y necesidades de cada usuario en cada situación, es presentada en [2]. Las preferencias y restricciones del usuario se almacenan en una ontología, a partir de la cual un dispositivo podrá generar rutas turísticas personalizadas.

En [3], se presenta la ontología *OntPersonal*, una ontología de personalización para la aplicación *ITINER@* [1], un sistema generador de rutas turísticas basado en información semántica. La ontología *OntPersonal* modela un conjunto de preferencias turísticas y restricciones de contexto asociadas al usuario final (turista), lo que se denomina su *perfil*. A partir de un conjunto de reglas SWRL (Semantic Web Rule Language) se infieren los puntos de interés más relevantes para cada perfil, estos se obtienen de una ontología externa.

En [4], se desarrolló una ontología sobre rutas Turísticas (a pie o en bicicleta) por espacios naturales, con el fin de aconsejar y apoyar al usuario en sus recorridos; conduciéndolo a través de una ruta en función de sus preferencias, la posición y el momento del día en el que se encuentre.

### *B. Desarrollos Web que Emplean Ontologías*

Una aplicación Web Turística orientada al sector hotelero del puerto de Acapulco en México, se muestra en [5]. Utilizaron una ontología geográfica que describe servicios turísticos: hospedaje (únicamente hoteles) y entretenimiento. Esta ontología está escrita en el lenguaje OWL y permite realizar consultas avanzadas como por ejemplo, ubicación de hoteles con restaurante y bar en Acapulco.

Una Aplicación Web para la Generación Automática de Rutas Turísticas en Zaragoza, en base al perfil del turista se presenta en [6]. En este proyecto se construyó una ontología para la descripción formal de los recursos turísticos del Ayuntamiento de Zaragoza. Esta ontología reutiliza e integra la información que está almacenada en las bases de datos y en los sistemas de información de la Web Municipal. La aplicación genera una ruta propuesta según las preferencias y algunos datos del usuario (como por ejemplo: acompañantes, motivo de su estancia, días de la visita, entre otros). La aplicación distingue dos tipos de recursos turísticos: puntos de interés (monumentos, parques, hoteles, entre otros) y eventos (conciertos, espectáculos, entre otros).

Un *Framework* que utiliza la Tecnología de Web Semántica para mejorar la búsqueda y clasificación de hoteles de los clientes de negocios con el fin de reducir tiempo y costo en esta búsqueda fue propuesto en [7]. Como parte de este desarrollo, se construyeron dos ontologías principales: una de personas, la cual describe a los viajeros y sus preferencias, una ontología de hotel, que describe: datos de contacto (teléfono, dirección, etc.), información general del hotel (número de habitaciones, pisos, tipos de pago, etc.), entre otros y una sub - ontología que describe: características generales del hotel, tal como: servicios de habitación (acceso a Internet, acceso para discapacitados, etc), puntos de interés y medios de transporte.

Como resultado de la revisión de antecedentes realizada, se puede concluir que son variados los sistemas de apoyo al turismo que se fundamentan en una ontología, ya que éstas posibilitan incrementar su potencial semántico (significado, claridad y consistencia de los términos) y permiten organizar, estandarizar, almacenar y consultar la información de manera eficiente. Adicionalmente, se puede observar que la mayoría de estas aplicaciones tienen como objetivo principal apoyar al turista en la búsqueda de información de interés para planificar un viaje determinado.

Tomando en consideración la conclusión anterior, sería oportuno apoyar la actividad turística en Venezuela y solventar la situación planteada en la Sección 2, haciendo uso de algunas tecnologías emergentes, específicamente mediante el uso de las ontologías. Esta última, además, permite formalizar, conceptualizar y compartir el conocimiento de un dominio específico. Consecuentemente, a través del uso de esta tecnología, la información y el conocimiento se almacenan de una manera formal, estándar y organizada.

Con base a esta reflexión se propone como objetivo de esta investigación:

### *C. Objetivo General*

Construir una Ontología para el dominio del turismo en Venezuela, que pueda ser accedida a través de una aplicación Web, para apoyar al turista en la búsqueda de información.

### *D. Objetivos Específicos*

- Conceptualizar el dominio del turismo en Venezuela.
- Formalizar el conocimiento en una estructura ontológica.
- Adquirir el conocimiento relacionado con el sector turismo en una zona específica de Venezuela, para poblar la Ontología.
- Validar y verificar la Ontología construida.
- Desarrollar una aplicación Web, que permita acceder a la Ontología de turismo.

## IV. ONTOLOGÍAS

Las ontologías permiten expresar el conocimiento de un dominio de manera general, de forma tal que pueda ser utilizado y manipulado por diversas técnicas o algoritmos de aprendizaje. Por otro lado, las ontologías pueden funcionar como un marco para la unificación de diferentes puntos de vista del conocimiento y servir como base para [8]: a) La comunicación entre personas con diferentes necesidades, pero en un área común de conocimiento; b) Facilitar la interoperabilidad entre sistemas, la cual se alcanza por la traducción entre diferentes modelos, métodos, paradigmas, lenguajes y herramientas de software; c) La reutilización del software (base para la codificación de entidades, atributos, procesos, entre otros), realizar chequeos de consistencia (fiabilidad del software), adquirir conocimiento (punto de partida en la construcción de sistemas basados en conocimiento) y para la especificación de requerimientos.

Con las ontologías se intenta expresar un esquema conceptual exhaustivo y riguroso de un dominio en particular para facilitar la comunicación, reutilizar y compartir información entre organizaciones, computadores y humanos. Una ontología define un vocabulario común que además incluye la interpretación de los conceptos básicos del dominio y sus relaciones.

Según [9], una ontología es una especificación formal y explícita de una conceptualización compartida. Esta conceptualización de la información permite que los sistemas

tengan un alto nivel de comprensión de la Web desde el punto de vista semántico, con el fin de localizar, procesar e integrar la información disponible y dispersa en la Web, para luego clasificarla y usarla con un fin determinado.

El proceso de construir una ontología no difiere mucho, en líneas generales, del usado para construir software. Las ontologías son productos de software y por lo tanto su desarrollo deberá seguir los estándares establecidos, por supuesto, adaptados a las características de las ontologías [10].

La metodología *Methontology* [11], permite la construcción de ontologías a nivel de conocimiento e incluye la identificación del proceso de desarrollo, un ciclo de vida basado en el desarrollo de prototipos y técnicas particulares para realizar cada actividad. Tiene sus raíces en las actividades identificadas por la IEEE para el proceso de desarrollo de software y ha sido propuesta para la construcción de ontologías por la FIPA (Foundation for Intelligent Physical Agents) [12].

*Methontology* propone un ciclo de vida basado en la evolución de prototipos que permite añadir, cambiar y eliminar términos en cada nueva versión (prototipo). Las actividades de desarrollo identificadas para *Methontology* son: **a) Especificación:** realizar un documento donde se señale el alcance, objetivos, propósito, nivel de formalidad y usuarios finales de la ontología; **b) Conceptualización:** consiste en organizar y convertir una percepción informal de un dominio en una especificación semi-formal usando un conjunto de representaciones intermedias (tablas, diagramas); **c) Formalización:** realizar la transformación del modelo conceptual en un modelo formal o semi-computable; **d) Implementación:** codificar la ontología utilizando un lenguaje formal y **e) Mantenimiento:** permite la actualización y corrección de la ontología.

## V. Construcción de la Ontología de Turismo

Para el desarrollo de la ontología de turismo, se utilizó la metodología Methontology [11]. Entre las razones más relevantes para la escogencia de esta metodología destacan:

- Es un modelo basado en la evolución de prototipos, lo que permite hacer modificaciones o actualizaciones en cualquier etapa de desarrollo.
- El esquema de plantillas, diagramas y tablas planteadas en las tareas de conceptualización, lo cual facilita la integración y cooperación de desarrolladores y expertos del dominio, dado que permiten la fácil comprensión de los mismos [13][14].
- Esta metodología ha sido aplicada con éxito en otros desarrollos ontológicos [15][16][17].

**Construcción de la Ontología:** A continuación, se describen brevemente las actividades sugeridas por Methontology:

### *A. Actividad de Especificación*

Se construyó el documento de especificación de la ontología, en el cual se define el dominio, nombre de la ontología, metas, propósito, alcance, nivel de formalidad, tipo de ontología, usuarios finales, fuentes de conocimiento, escenarios y preguntas de competencia.

### *B. Actividad de Conceptualización del Dominio*

El conocimiento contenido en la ontología fue adquirido de la siguiente manera:

- En primer lugar con conocimiento propio de las autoras acerca de los lugares existentes en Venezuela y la clasificación de éstos.
- Revisión de documentación especializada en el dominio, tales como: a) artículos de investigación; b) desarrollos previos de ontologías de Turismo; c) Sitios Web de turismo; d) catálogo del patrimonio cultural venezolano 2004–2005, suministrado por el IPC (Instituto de Patrimonio Cultural) en formato digital; e) guía gastronómica de Caracas, editor Miro Popic, 2012; f) Gaceta aérea Vol. 40 Nº V, Junio 2012.
- Adquisición del conocimiento mediante la aplicación de técnicas como cuestionarios, encuestas, visita a entes tales como: IPC, IMVITRACV (Instituto Municipal de Vialidad, Tránsito y Transporte Colectivo de Vargas), oficinas de turismo de la gobernación de Vargas, ubicada en la casa Guipuzcoana, y la de la alcaldía de Vargas. También se realizaron recorridos por las parroquias Naiguatá y Caruao del estado Vargas, visitando lugares como playas, ríos, cascadas, hoteles, posadas, campamentos, plazas, iglesias, restaurantes, paradas de transporte (autobuses, taxis, moto-taxis), entre otros.

Los resultados obtenidos al realizar las tareas de conceptualización permitieron representar el conocimiento del dominio de una manera organizada y estructurada. Se identificaron los términos relevantes del dominio (conceptos y atributos) y las relaciones entre los conceptos. Obteniéndose un total de 105 conceptos, 132 atributos y 174 instancias.

En el contexto de este trabajo, los conceptos se organizaron tomando en cuenta dos puntos de vista: el primero, referente a los posibles lugares que un turista puede visitar en un determinado viaje, y el segundo, los servicios que se le pueden brindar al turista durante su estadía en un lugar determinado. Por lo tanto, se organizó el conocimiento en dos taxonomías:

- Sitios de interés, que abarca sitios turísticos (sitios naturales, parques, museos, sitios religiosos, sitios históricos, entre otros) y sitios no turísticos (restaurantes, cines, teatros, centros comerciales, entre otros). En la Figura 1, se muestra un fragmento de esta taxonomía.
- Servicios turísticos, tales como: hospedaje, transporte, agencias de viaje, oficinas de turismo, líneas aéreas, entre otros. En la Figura 2, se muestra un fragmento de esta taxonomía.

### *C. Actividades de Formalización e Implementación*

Luego de construir el modelo conceptual del dominio de turismo en la actividad de conceptualización, la siguiente actividad de la metodología consiste en formalizar e implantar el modelo utilizando un lenguaje formal. Para ello, se seleccionó el sub-lenguaje OWL-DL de OWL, estándar recomendado por W3C [18], y el editor de ontologías Protégé [19].

*D. Actividad de Mantenimiento*

Esta actividad fue realizada durante todas y cada una de las fases del ciclo de vida de desarrollo, lo cual permitió actualizar y corregir (evaluar) la ontología construida.

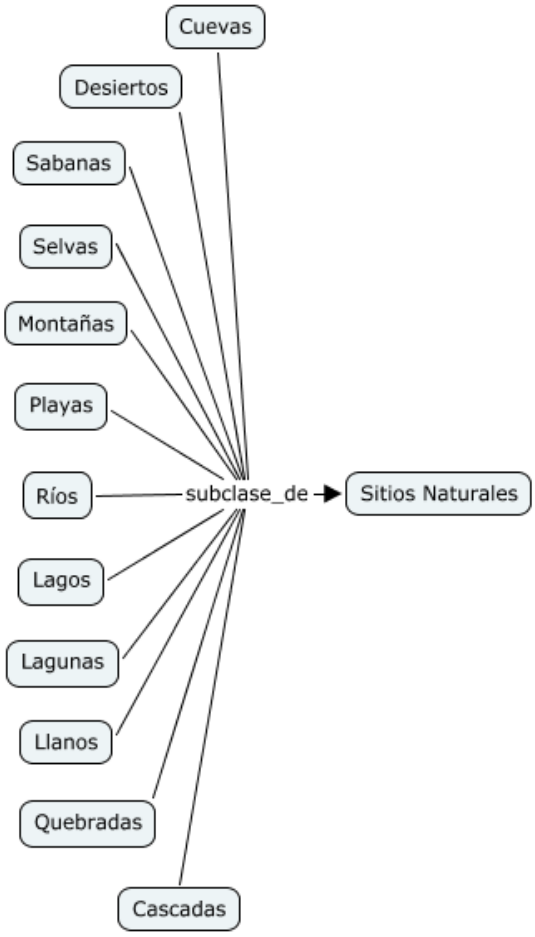


**Figura 1:** Fragmento de la Taxonomía de Sitios de Interés

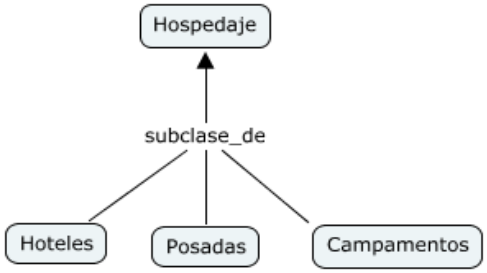


**Figura 2:** Fragmento de la Taxonomía de Servicios Turísticos

Aunque la ontología de turismo no es única en su dominio, para evaluarla se decidió utilizar el esquema propuesto en [20] como parte fundamental de la evaluación, ya que examina los criterios utilizados en la mayoría de los esquemas propuestos para evaluar ontologías.

Entre los aspectos evaluados destacan:

*1) El Uso Correcto del Lenguaje:* Para codificar la ontología se seleccionó el sublenguaje -OWL-DL- de OWL, el cual cumple con los estándares para desarrollos ontológicos. Adicionalmente, se utilizó el analizador sintáctico de archivos OWL-DL, de la Universidad de Manchester [21], este analizador chequea la sintaxis del archivo OWL, el cual arrojo como resultado una sintaxis correcta.

*2) Exactitud de la Estructura Taxonómica:* Se revisó la taxonomía de conceptos identificando y corrigiendo de manera oportuna algunas inconsistencias entre conceptos, se evalúo la completitud de los términos de la taxonomía identificando ausencia de algunos términos, los cuales se agregaron oportunamente, no se encontraron redundancias entre los términos de la taxonomía.
Adicionalmente, se utilizó el razonador Pellet 1.5.2 provisto por la herramienta Protégé, el cual hace un chequeo de inconsistencia de conceptos, clasificación de la taxonomía y tipos de inferencias. Los resultados obtenidos al aplicar este razonador fueron satisfactorios.

*3) Validez del Vocabulario:* Se evaluaron los conceptos codificados en la ontología (105 términos), utilizando para ello el corpus del dominio (se refiere al conocimiento experto, textos y otras fuentes), el cual consta de 154 términos, se identificaron los términos coincidentes entre la ontología y el corpus, obteniéndose 51 términos coincidentes.
En resumen se tiene:
**CCorp** (Cantidad de términos del corpus) = 154
**COnto** (Cantidad de términos de la ontología) = 105
**CO_C** (Cantidad de términos que se solapan entre la ontología y el corpus) = 51
Seguidamente, se evaluó el vocabulario de la ontología utilizando medidas de calidad de resultados usadas en escenarios de recuperación de información (búsqueda de documentos), estas son:

- Precisión: porcentaje de los términos de la ontología que aparecen en el corpus con relación a la cantidad total de términos de la ontología. Ver (1).

$$Precisión = CO\_C / COnto \quad (1)$$

- Recall: porcentaje de términos del corpus que aparecen en la ontología con relación al total de términos en el corpus. Ver (2).

$$Recall = CO\_C / CCorp \quad (2)$$

Obteniéndose los siguientes resultados: Precisión = 0,49 y Recall = 0,33. Estos resultados indican que 49% de los conceptos de la ontología se encuentran en el corpus y 33% de los términos del corpus, están en la ontología.
Existen diversas razones que pueden explicar los valores obtenidos, en primer lugar con respecto a los conceptos codificados en la ontología cabe destacar que existen términos que son utilizados mayoritariamente en Venezuela, como por ejemplo: areperas, metrobús, metrocable, entre otros.
Adicionalmente, la ontología presenta una clasificación bastante completa acerca de los tipos de restaurantes, la cual contiene 21 términos que representan aproximadamente el 20% de los conceptos de la ontología, de los cuales sólo 3 existen en el corpus. Por otra parte, con respecto a los

términos del corpus es importante resaltar que existen palabras que se refieren a organizaciones especificas (éstos representan instancias), términos que son características y términos específicos de otros países. Una de las razones que influye en gran parte en los resultados obtenidos, es el alcance de la ontología, ya que ésta modela sólo una parte del dominio, enfocándose en los lugares físicos que puede visitar un turista.

*4) Adecuación a Requerimientos:* En cada fase del ciclo de vida del desarrollo de la ontología, se verificó el cumplimiento de las especificaciones del documento de requerimientos, haciendo énfasis en las actividades necesarias que permitieran alcanzar las metas y el propósito planteado, así como también en el cumplimiento de los formalismos de representación del conocimiento y la consecución de respuestas correctas para las preguntas de competencia.
Por tanto, se alcanzaron los objetivos planteados, estos son:

- Formalizar, estandarizar, compartir y representar el conocimiento del dominio, para que esté a la disponibilidad de la comunidad Web que lo requiera.
- Garantizar la organización, integración de la información, confiabilidad y precisión de resultados de consultas.

Se realizaron recorridos sobre la ontología para verificar que el conocimiento representado permitiera dar respuestas correctas y pertinentes a las preguntas de competencia. De esta manera, se verificó que los recorridos sobre la estructura taxonómica, dieron respuestas correctas a las preguntas de competencias.

Como resultado de aplicar este esquema de evaluación en cada fase del ciclo de desarrollo de la ontología de turismo, se obtuvo un desarrollo ontológico de calidad para este dominio.

Adicionalmente, se evaluó la ontología en base a métricas, esta evaluación ofrece una perspectiva cuantitativa de la calidad de la ontología, para ello se utilizó el método ONTOQA (Metric-Based Ontology Quality Analysis).

Este método presenta dos tipos de métricas (métricas de esquemas y métricas de base de conocimiento). Las métricas de esquemas, evalúan el diseño de la ontología y su potencial para representar conocimiento. Y las métricas de base de conocimiento, evalúan la ubicación de las instancias en la ontología y la utilización eficaz de los conocimientos modelados en el esquema [22].

En las Tablas I, II y III, se presentan los resultados obtenidos de la evaluación de la ontología en base a métricas:

*1) Métricas de Esquemas:* Evalúan riqueza, amplitud, profundidad y herencia del esquema ontológico.

- Riqueza de Relaciones (RR): Esta medida indica la diversidad de tipos de relaciones presentes en la ontología. Una ontología que sólo contiene relaciones de subclase, transmite menos que una que contiene diversos tipos de relaciones.
- Riqueza de Herencia (RH): Describe la distribución de la información a través de los diferentes niveles de la ontología. Indica qué tan bien se agrupan los conocimientos en las diferentes categorías y subcategorías.
- Riqueza de Atributos (RA): La cantidad de atributos que se definen para las clases, pueden calificar el diseño y la cantidad de información relativa a datos de las instancias.

**Tabla I:** Resultados Obtenidos en las Métricas de Esquemas

| Métricas de esquemas | Valor |
|---|---|
| RR (Riqueza de Relaciones) | 1 |
| RH (Riqueza de Herencia) | 0,98 |
| RA (Riqueza de Atributos) | 1,26 |

En la Tabla I, se observa que el valor de RR es 1, lo cual refiere que no existen relaciones definidas como subclase_de, esto quiere decir que existen diversos tipos de relaciones expresadas en lenguaje natural, tales como: a) *son* y su inversa *tipo de*. b) *se considera* y su inversa *es considerado*. c) *está constituido por* y su inversa *forma parte de*, lo cual implica que la ontología tiene una gran contenido semántico, por tanto transmite más conocimiento.

El valor de RH es cercano a 1, lo cual indica que existe riqueza en amplitud, es decir, la ontología representa un amplio rango de conocimiento general con un bajo nivel de detalle.

El valor de RA es mayor que 1, este resultado indica que las clases de la ontología se describen con al menos un atributo.

En base a los resultados obtenidos en las métricas de esquemas, se puede afirmar que la ontología presenta un diseño de calidad y tiene un alto potencial para representar conocimiento.

*2) Métricas de Base de Conocimiento:* Evalúan la efectividad del diseño ontológico y la cantidad de conocimiento del mundo real representado en la ontología (instancias).

- Riqueza de Clases (RC): Describe cómo las instancias están distribuidas a través de las clases, comparando la cantidad de clases instanciadas contra la cantidad total de clases.
- Importancia de Clases ($IC_i$): Define el porcentaje de instancias que pertenecen a un subárbol de la base de conocimiento con raíz en la clase i, en comparación al número total de instancias de clase en la base de conocimiento.

En la Tabla II, se presentan los resultados obtenidos al calcular el valor de la $IC_i$, esta medida es un indicador de cuáles clases de la ontología se destacan (representan el mundo real). A partir de los resultados obtenidos en la Tabla II, se puede afirmar que las clases más relevantes son: posadas, buses, restaurantes de pescados y mariscos, plazas, playas y ríos.

Esto quiere decir que en las parroquias donde se pobló la ontología (Naiguatá y Caruao del estado Vargas), estas clases se destacan, dado que tienen más instancias, las cuales representan la cantidad de conocimiento del mundo real.

En la Tabla III, se observa que el valor de RC es 0,23, dado este resultado se puede afirmar que la ontología representa en un 23% el conocimiento del dominio.

El valor de RC es bajo, lo cual indica que hay pocas clases instanciadas con respecto al total de clases.

Con respecto a los resultados obtenidos en la métricas de base de conocimiento, se observa que el valor de RC y la mayoría de los valores de $IC_i$, son cercanos a 0, ambos representan porcentajes bajos (menores al 50%), esto se debe a que la ontología está poblada sólo en dos parroquias del estado Vargas, dado el alcance de este trabajo.

**Tabla II:** Cálculo de la Importancia de Clases ($IC_i$)

| Clase | Instancias | $IC_i$ | $IC_i$ (%) |
|---|---|---|---|
| Campamentos | 4 | 0,17 | 17 |
| Hoteles | 5 | 0,21 | 21 |
| Posadas | 36 | 1,50 | 150 |
| Oficinas de turismo | 3 | 0,13 | 13 |
| Lanchas | 1 | 0,04 | 4 |
| Buses | 13 | 0,54 | 54 |
| Moto Taxi | 5 | 0,21 | 21 |
| Taxi | 2 | 0,08 | 8 |
| Club | 2 | 0,08 | 8 |
| Cadenas de comida | 2 | 0,08 | 8 |
| Pizzería | 3 | 0,13 | 13 |
| Restaurante Chino | 1 | 0,04 | 4 |
| Restaurante de carnes | 3 | 0,13 | 13 |
| Restaurante de pescados y mariscos | 24 | 1,00 | 100 |
| Restaurante de pollos | 1 | 0,04 | 4 |
| Restaurante español | 4 | 0,17 | 17 |
| Paseos | 2 | 0,08 | 8 |
| Plazas | 14 | 0,58 | 58 |
| Cascadas | 3 | 0,13 | 13 |
| Cuevas | 1 | 0,04 | 4 |
| Playas | 22 | 0,92 | 92 |
| Ríos | 13 | 0,54 | 54 |
| Sitios religiosos | 9 | 0,38 | 38 |
| Zona Colonial | 1 | 0,04 | 4 |

**Tabla III:** Resultados Obtenidos en las Métricas de Base de Conocimiento

| Métricas de base de conocimiento | Valor |
|---|---|
| RC (Riqueza de Clases) | 0,23 |
| $IC_i$ (Importancia de Clases) | (Ver Tabla II) |

## VI. Arquitectura de la Solución Propuesta

En esta Sección se presenta el esquema de la arquitectura de la solución propuesta (ver Figura 3) y la descripción de sus componentes.

**Descripción de los componentes de la solución propuesta:** tal como se muestra en la Figura 3, la solución propuesta se fundamenta en una arquitectura de tres capas, las cuales se describen a continuación:

### *A. Capa de Presentación*

A través de esta capa se interactúa con el usuario, el cual envía su solicitud a la capa de lógica del negocio para su procesamiento y recibe la respuesta a dicha solicitud (resultados del procesamiento). La capa de presentación está compuesta por la interfaz de la aplicación Web, la cual permite la interacción con el usuario. Estos elementos se describen a continuación:

*1) Interfaz de la Aplicación Web:* Permitirá la interacción del usuario con la ontología, con el fin de que el usuario realice consultas de la información necesaria para la planificación de un viaje. Además, ofrecerá al usuario una opción de búsqueda por preferencia, a través de la cual el viajero podrá ingresar una palabra o frase clave (característica deseada o nombre de un sitio en particular), de manera de agilizar la búsqueda. Adicionalmente, el usuario tendrá la posibilidad de navegar, explorar y visualizar la ontología, es decir, podrá ver de forma organizada la información contenida en la misma.

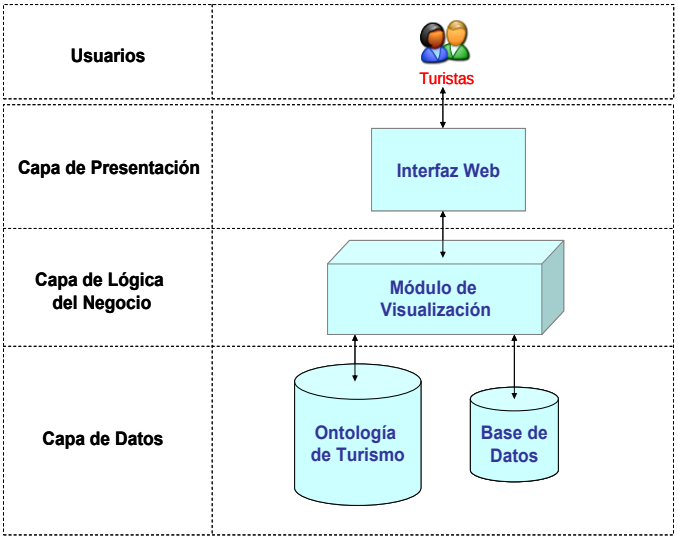


**Figura 3:** Arquitectura de la Solución Propuesta (Fuente: Elaboración Propia)

*2) Usuarios (Turistas):* Se refiere principalmente a cualquier turista nacional o internacional que desee buscar información de interés en el sitio Web para la planificación de un viaje. Sin embargo, la aplicación estará disponible para cualquier persona que esté interesada en consultar dicha información. El usuario accede a la aplicación Web, desde una máquina cliente, colocando en el navegador el URL http://lia.ciens.ucv.ve/onto_turismo/turismo/inicio.html.

### *B. Capa de Lógica del Negocio*

Recibe la solicitud de la capa de presentación e interactúa con la capa de datos para llevar a cabo las operaciones de negocio y proporcionar los resultados a la capa de presentación. Esta capa está compuesta por:

Módulo de Visualización de la Aplicación Web: Se encargará del procesamiento de las consultas.

### *C. Capa de Datos*

Esta capa se encarga de recuperar los datos de la ontología que dan respuesta a la solicitud de los usuarios; está compuesta por:

*1) Ontología de Turismo (Repositorio de Datos):* Es el componente principal de esta investigación, contendrá información de interés para el turista de un área específica de Venezuela. Además, estandarizará e integrará parte de la información que se encuentra disponible en la Web en diversos sitios.

*2) Base de Datos:* Almacena los datos para la localidad (estados, ciudades y zonas). Estos datos permitirán al usuario elegir la localidad donde desea consultar información turística.

## VII. Descripción de la Aplicación Web para Acceder a la Ontología

La aplicación Web desarrollada permite visualizar, consultar y navegar la ontología de turismo (ver Figura 4). Esta aplicación tiene como requerimientos funcionales, los siguientes:

- Consultar sitios turísticos de Venezuela.
- Consultar sitios no turísticos de Venezuela.
- Consultar servicios turísticos de Venezuela.
- Consultar detalles de sitios turísticos.
- Consultar detalles de sitios no turísticos.
- Consultar detalles de servicios turísticos.
- Consultar preferencia de sitios turísticos.
- Consultar preferencia de sitios no turísticos.
- Consultar preferencia de servicios turísticos.
- Consultar preferencia de hospedaje.

La aplicación Web fue implementada bajo el patrón de diseño arquitectónico MVC (Modelo Vista Controlador o Model View Controller), desarrollado por el ambiente Smalltalk utilizable para cualquier aplicación interactiva.

MVC es una arquitectura de tres capas que desacopla la interfaz del usuario (vista) de la funcionalidad (controlador) de la aplicación Web y de los datos (modelo).

Las solicitudes de usuarios se manejan mediante el controlador, el cual transmite esta solicitud al modelo y éste implementa la funcionalidad para recuperar desde la ontología, los datos que dan respuesta a dicha solicitud.

Las tecnologías Web utilizadas para el desarrollo de la aplicación son las siguientes: Dreamweaver (herramienta de desarrollo), PHP, JQuery, CSS, HTML, ARC RDF Store **(**ARC2): librería de PHP 5.3 que permite la serialización, análisis, almacenamiento y consulta de archivos XML/RDF [23], MySQL (manejador de base de datos).

En la Figura 5, se presenta un ejemplo de una consulta a la aplicación (Consulta de Cascadas en el Estado Vargas, Parroquia Caruao y Zona Caruao), y en la Figura 6, se muestra el resultado de esta consulta.

En la Figura 7, se presenta el esquema de implementación de la aplicación Web de turismo, donde se observa cómo están distribuidos los componentes de dicha aplicación, bajo el patrón MVC.

En la Vista se encuentra la interfaz Web de la aplicación, la cual está implementada con HTML, CSS y JQuery.

En el Controlador está el módulo de turismo y el módulo de localidad, los cuales están implementados en PHP. En el controlador se reciben los datos que cumplen con la petición del usuario, estos datos son: los sitios turísticos, no turísticos o servicios turísticos que desea consultar, la localización de su interés (estado, ciudad y/o zona) y alguna característica específica (preferencia). Estos datos se reciben en los módulos de turismo y de localidad, y se llaman a las clases del modelo para realizar la consulta a la ontología (módulo de turismo) y a la base de datos de localidad (módulo de localidad), de esta manera se cargan los datos en la página Web correspondientes a la consulta realizada por el usuario.

En el Modelo se encuentran los siguientes módulos: módulo de turismo, módulo de estado, módulo de ciudad, módulo de zona, módulo de conexión con la ontología de turismo y módulo de conexión con la base de datos de localidad denominada "Venezuela".

El módulo de turismo del modelo recibe los datos enviados por el controlador y se encarga de armar los *querys*, en el lenguaje SPARQL [24], seguidamente llama a la clase que se conecta con la ontología y ejecuta los *querys*, la cual se encuentra en el módulo de conexión con la ontología, en esta clase se hace uso de la librería ARC2 de PHP, la cual permite ejecutar las consultas en SPARQL y de esta manera consultar los datos de la ontología, los cuales están almacenados de manera persistente en una base de datos.

De manera similar, los módulos de estado, ciudad y zona, se encargan de armar los *querys* para consultar la base de datos de localidad y luego llama a la clase que se conecta con la base de datos y ejecuta los *querys*, esta clase se encuentra en el módulo de conexión con la base de datos de localidad.

Existen dos base de datos, una para almacenar la ontología de manera persistente y la otra denominada "Venezuela" que almacena los datos para la localidad (estados, ciudades y zonas).

Todos los módulos del modelo están implementados en PHP.

Finalmente, se realizaron las pruebas de recorrido por la aplicación Web con el fin de validar que la aplicación respondiera de manera correcta las preguntas de competencia definidas en el documento de especificaciones de la ontología. Para ello, se realizaron distintas consultas, con el fin de abarcar las principales funcionalidades de la aplicación. Al realizar estos recorridos, se obtuvo como resultado que la aplicación respondió de manera correcta las preguntas de competencias planteadas.

## VIII. Conclusiones

El uso de la ontología de turismo, parte fundamental de este trabajo, permitió:

- Garantizar la organización, integración de la información, confiabilidad y precisión de resultados de consultas.
- Formalizar, estandarizar, compartir y representar el conocimiento del dominio, para que esté a la disponibilidad de la comunidad Web que lo requiera.

De esta manera, se logró concentrar en un sitio Web la mayor cantidad de información relevante para planificar un viaje. En este trabajo se considera que esta información es:

- Sitios turísticos: museos, parques, plazas, sitios naturales (playas, ríos, cascadas, etc), sitios históricos, sitios religiosos, entre otros.
- Sitios no turísticos: restaurantes, cines, teatros, centros comerciales, entre otros.

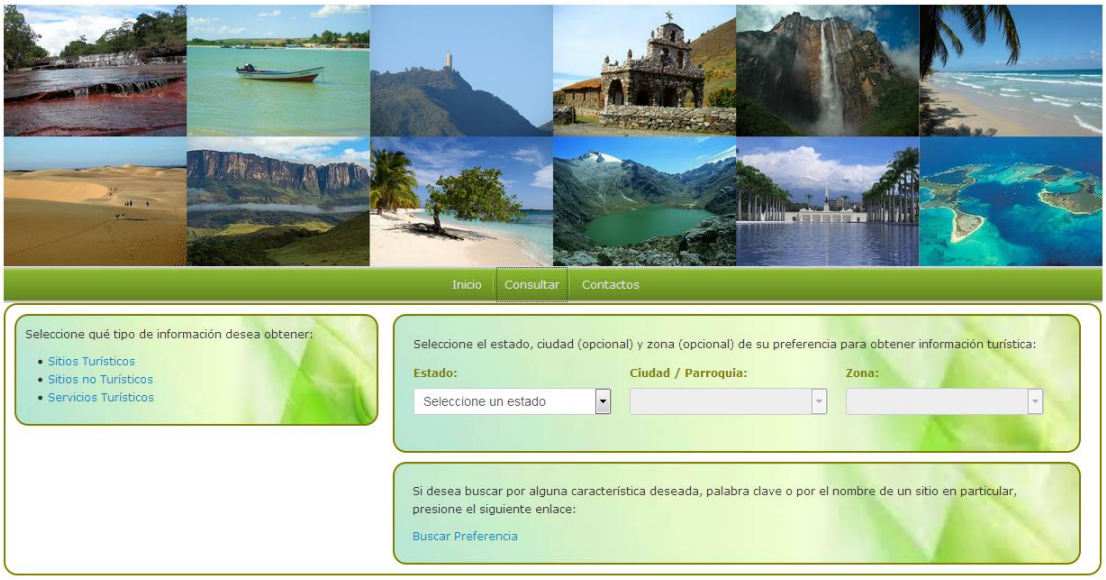


**Figura 4:** Pantalla de Consultas de la Aplicación Web de Turismo

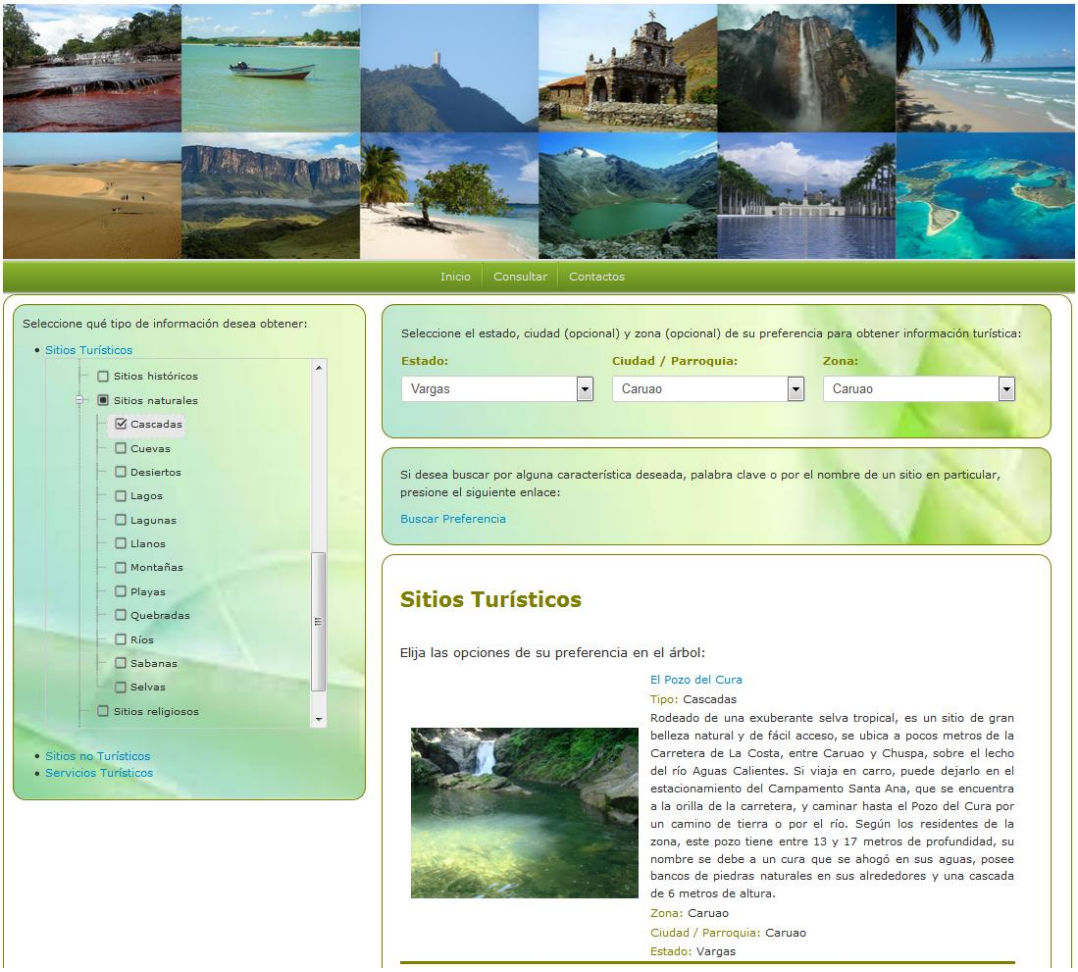


**Figura 5:** Consulta de Cascadas en el Estado Vargas, Parroquia Caruao y Zona Caruao


**Figura 6:** Resultado de la Consulta de Cascadas en el Estado Vargas, Parroquia Caruao y Zona Caruao

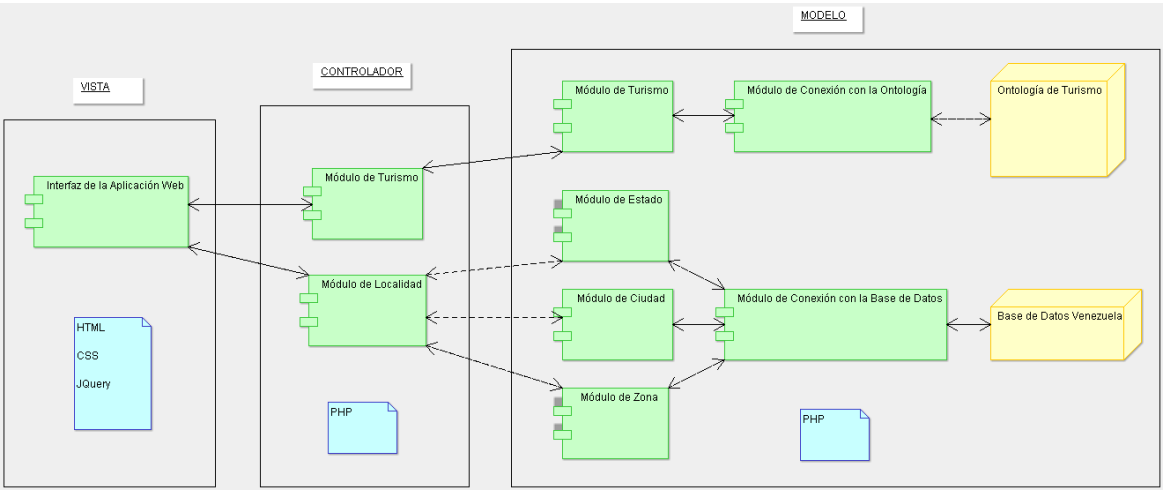


**Figura 7:** Esquema de Implementación de la Aplicación Web de Turismo

- Servicios turísticos: transporte (taxis, metro, buses, lanchas, etc), hospedaje (hoteles, posadas y campamentos), oficinas de turismo, agencias de viaje, líneas aéreas, entre otros.

Para la construcción de la ontología, se utilizó la metodología Methontology, realizando las actividades que propone, se conceptualizó el dominio del turismo en Venezuela, elaborando la taxonomía de conceptos, seguidamente utilizando varias técnicas se adquirió el conocimiento del sector turismo en Venezuela, se formalizó el conocimiento en una estructura ontológica, luego para poblar la ontología se adquirió el conocimiento de las instancias de la ontología en una zona específica de Venezuela.

Para este trabajo, se pobló la ontología en las parroquias Naiguatá y Caruao del estado Vargas. Por último, se realizó la evaluación de la ontología, aplicando el esquema de evaluación propuesto en [20], como resultado de esta evaluación, se verificó el cumplimiento de las especificaciones del documento de requerimientos y se obtuvo un desarrollo ontológico de calidad para el dominio.

Adicionalmente, se evaluó la ontología en base a métricas utilizando el método ONTOQA, esta evaluación dio como resultado que la ontología presenta un diseño de calidad y tiene un alto potencial para representar conocimiento.

Finalmente, se realizaron un conjunto de pruebas a la aplicación Web, para validar que la aplicación diera respuesta a las principales preguntas de competencias definidas en el documento de especificaciones de la ontología. Obteniéndose como resultado, que la aplicación respondió de manera correcta a las preguntas de competencias planteadas, las cuales abarcan las principales funcionalidades de la aplicación.

Por otra parte, se realizó una publicación de este trabajo en una revista digital de tecnología no arbitrada, Revista Digital de la DTIC (Dirección de Tecnología de Información y Comunicaciones) de la Universidad Central de Venezuela, con el fin de dar a conocer a la comunidad interesada los avances de esta investigación [25].

## IX. Recomendaciones

Una investigación como la realizada en este trabajo, no se agota con la solución dada al problema planteado, sino que propicia la exploración de otras opciones que permitan obtener aportes que enriquezcan aún más el área objeto de estudio.

A continuación se presentan algunas recomendaciones, que permitirán la continuación y ampliación de esta investigación:

### *A. Ontología de Turismo*

Sería de gran interés para la comunidad de turistas potenciales contar con otro tipo de información, no contemplada en el alcance de este trabajo, tal como: turismo de aventura, tradiciones, gastronomía, entre otros.

Por tal motivo, se propone extender el vocabulario de la ontología, con el fin de abarcar este conocimiento. De esta manera, se obtendría un incremento en la medida Recall calculada en la evaluación de la ontología, aumentando así la confiabilidad de ésta.

Por otra parte, sería un gran aporte al sector turismo de Venezuela, poblar la ontología de turismo en las parroquias restantes del estado Vargas y posteriormente en el resto de los estados de Venezuela. Esto aumentaría los valores de RC e $IC_i$ obtenidos en la evaluación en base a métricas de la ontología. Al obtener un valor de RC cercano a 1, se alcanzaría una ontología con un alto porcentaje de representación del conocimiento del dominio.

De igual manera, al aumentar la cantidad de instancias de la ontología, el valor de $IC_i$ se incrementa, obteniéndose como resultado que la mayor cantidad de clases de la ontología tengan importancia y así representen en gran medida el conocimiento del mundo real.

Adicionalmente, se propone realizar la traducción de la ontología al idioma inglés, con el fin de facilitar el uso y acceso a turistas internacionales.

### *B. Aplicación Web*

Con la finalidad de facilitar la incorporación y mantenimiento del conocimiento de la ontología, se propone desarrollar un módulo de actualización, que permita a un usuario administrador de la aplicación: agregar, modificar y eliminar instancias de la ontología. Este módulo sería de gran utilidad para poblar la ontología.

## Referencias

[1] L. Descamps-Vila, J. Casas, J. Conesa, A. Pérez-Navarro, y I. Gutiérrez, *Hacia la Mejora de la Creación de Rutas Turísticas a Partir de Información Semántica*, V Jornadas de SIG Libre, SIGTE, Universidad de Girona, Girona, España, URI: http://hdl.handle.net/10256/3384, Marzo 2011.

[2] L. Descamps-Vila, J. Casas, A. Pérez-Navarro, y J. Conesa, *Personalización de Servicios Basados en Localización: un Caso Práctico*, V Jornadas de SIG Libre, SIGTE, Universidad de Girona, Girona, España, Marzo 2011.

[3] V. Ocegueda-Hernández y J. Conesa-Caralt, *OntPersonal: Ontología de Personalización para ITINER@, un Sistema Generador de Rutas Turísticas Basado en Información Semántica*, Trabajo final de Master, Universidad Oberta de Catalunya, Barcelona, España, http://hdl.handle.net/10609/11640, 2012.

[4] I. Gutiérrez, J. Conesa, y F. Geva, *Ontologías Turísticas Geográficas: Creación de una Ontología sobre Rutas Turísticas (a Pie o en Bicicleta) por Espacios Naturales*, Trabajo final de carrera, Universidad Oberta de Catalunya, Barcelona, España, http://hdl.handle.net/10609/2284, 2010.

[5] R. Zagal, M. Torres, T. Ramírez, y M. Moreno, *Diseño de una Aplicación Web Hibrida Aplicada a Servicios Turísticos Descritos Semánticamente*, 5to Congreso Internacional de Ingeniería Electromecánica y de Sistemas, ISBN: 978-607-414-049-1, ESIME, IPN, D.F., México, Noviembre 2008.

[6] J. Fernández, *Tu Ruta por la Ciudad de Zaragoza: Aplicación de la Web Semántica en la Web del Ayuntamiento de Zaragoza*, Día W3C en España: Standars for Business, Madrid, España, Mayo 2008.

[7] M. Niemann, M. Mochol, y R. Tolksdorf, *Enhancing Hotel Search with Semantic Web Technologies*, Journal of Theoretical and Applied Electronic Commerce Research, vol. 3, no. 2, pp. 82-96, ISSN 0718–1876 Electronic Version, Universidad de Talca, Chile, Agosto 2008.

[8] M. Uschold y M. Gruninger, *Ontologies: Principles, Methods and Applications*, AIAI-TR-191, Knowledge Engineering Review, vol. 11, no. 2, 1996.

[9] R. Studer, V. Benjamins, y D. Fensel, *Knowledge Engineering: Principles and Methods*, Data and Knowledge Engineering (DKE), vol. 25, no. 1-2, pp. 161-197, 1998.

[10] FIPA, *Ontology Service Specification*, Foundation for Intelligent Physical Agents, Número de documento: XC00086C, 2000.

[11] A. Gómez Pérez, M. Fernández López, y M. Corcho, *Ontological Engineering*, London: Springer-Verlag, 2004.

[12] O. Corcho, M. Fernández-López, A. Gómez-Pérez, y A. López, *Building Legal Ontologies with Methontology and WebODE*, Law and the

Semantic Web, Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications, Springer-Verlag, LNAI 3369, 2005.

[13] M. Fernández-López, A. Gómez-Pérez, A. Pazos, y J. Pazos, *Building a Chemical Ontology Using Methontology and the Ontology Desing Environment*, IEEE Intelligent Systems & their Applications, vol. 4, no. 1, pp. 37-46, ISSN: 1094-7167, 1999.

[14] L. Vilches, M. Bernabé, M. Suárez-Figueroa, A. Gómez-Pérez, y A. Rodríguez, *Towntology & Hydrontology: Relationship between Urban and Hydrographic Features in the Geographic Information Domain*, 1st Workshop of COST Action C21 Ontologies for Urban Development: Interfacing Urban Information Systems, University of Geneva, Geneva, Switzerland, November 2006.

[15] E. Ramos, *Sistema Basado en Agentes para Apoyar el Diagnóstico de la Calidad del Semen Humano*, Tesis Doctoral, Laboratorio de Inteligencia Artificial, Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Venezuela, Abril 2009.

[16] A. Ghanem, *Sistema de Gestión de Conocimiento para Apoyar las Actividades de Soporte Técnico en los Infocentros del País*, Trabajo de Grado de Maestría, Laboratorio de Inteligencia Artificial, Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Venezuela, Septiembre 2013.

[17] V. Fernández, *Sistema Basado en Ontologías para Apoyar la Identificación de Macroinvertebrados Acuáticos del Orden Plecóptera*, Trabajo de Grado de Maestría, Laboratorio de Inteligencia Artificial, Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Venezuela, Diciembre 2014.

[18] *W3C*, http://www.w3.org.

[19] *Protégé*, http://protege.stanford.edu.

[20] E. Ramos, H. Núñez y R. Casañas, *Esquema para Evaluar Ontologías Únicas para un Dominio de Conocimiento*, Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento, vol. 6, no. 1, pp. 57-71, ISSN: 1690-7515, 2009.

[21] *OWL Validator*, http://owl.cs.manchester.ac.uk/validator.

[22] S. Tartir, B. Arpinar, y A. Sheth, *Ontological Evaluation and Validation*, In R. Poli (Editor): Theory and Applications of Ontology (TAO), volume II: Ontology: The Information-science Stance, Springer, June-2010, 2007.

[23] *GitHub ARC2: Easy RDF and SPARQL for LAMP Systems*, https://github.com/semsol/arc2.

[24] *W3C SPARQL*, http://www.w3.org/TR/rdf-sparql-query.

[25] R. Dulcey, H. Núñez, y E. Ramos, *Ontología El Turismo a un Click*, #VirtualDTIC Revista Digital de la Dirección de Tecnología de Información y Comunicaciones (DTIC) de la Universidad Central de Venezuela, vol. 3, no. 003, pp. 14-15, URL: http://issuu.com/dticucv, Junio 2014.

**Revista Venezolana de Computación**
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 3, No. 1, pp. 13-22, Junio 2016
Selección de los Mejores Artículos del SCTC 2016

# El Proceso de Desarrollo RUP-GDIS

Christiane Metzner[1], Norelva Niño[1]
christiane.metzner@ciens.ucv.ve, norelva.nino@ciens.ucv.ve

[1] Escuela de Computación, Universidad Central de Venezuela, Caracas, Venezuela

**Resumen:** En este trabajo se presenta y se describe el proceso de desarrollo de software RUP-GDIS y sus conceptos, utilizado actualmente en la asignatura de pregrado Ingeniería de Software de la Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela. RUP-GDIS, es una configuración de RUP centrada en las primeras cinco disciplinas correspondientes al núcleo de RUP, y adaptado para un curso de pregrado en el cual equipos de trabajo de estudiantes deben desarrollar por primera vez un sistema de software. El proceso guía a los estudiantes en "qué", "por qué", "cuándo" y "cómo" realizar exitosamente las diferentes actividades del proceso de desarrollo de software.

**Palabras Clave:** Ingeniería de Software; Proceso de Desarrollo de Software RUP-GDIS; Artefactos de Software.

**Abstract:** This paper describes the software development process RUP-GDIS and its concepts that are currently taught in the Software Engineering course at Universidad Central de Venezuela (Ciencias-UCV), Faculty of Sciences, School of Computer Science (http://www.ciens.ucv.ve/ciens and http://computacion.ciens.ucv.ve/escueladecomputacion). RUP-GDIS is a configuration of RUP centered on the five core disciplines of RUP, and tailored for an undergraduate course in which student project teams have to develop for the first time a software system. The process guides students in "what", "why", "when" and "how" to perform successfully the different activities defined in the development process.

**Keywords:** Software Engineering; Software development process RUP-GDIS; Software Artifacts.

## I. Introducción

*Rational Unified Process* (RUP) se define como un meta-proceso que permite configurar procesos iterativos e incrementales y se estructura en dos dimensiones: fases y disciplinas [1]. Las fases son: Incepción, Elaboración, Construcción y Transición. Las disciplinas se categorizan en dos grupos: disciplinas del núcleo de RUP y las disciplinas de soporte al núcleo. Las disciplinas del núcleo de RUP son: Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, *Deployment*; y las disciplinas de soporte al núcleo son: Gerencia de Configuración y Cambio, Gerencia de Proyecto, Entorno. Cada fase tiene un propósito específico y en cada disciplina se realizan actividades que producen un resultado observable de valor en cada fase. Un proceso configurado a partir de RUP se organiza en términos de iteraciones; cada iteración cubre las disciplinas a lo largo de cada fase y el resultado de cada iteración es un producto ejecutable que se prueba, integra, entrega y se transformará en un sistema final. En la Figura 1, se ilustran las dos dimensiones de RUP donde el área bajo las curvas representa un estimado del esfuerzo de trabajo en cada disciplina cuando se itera a lo largo de las cuatro fases. Se destaca que en este trabajo se mantienen algunos nombres en el idioma inglés, y no su posible traducción al idioma castellano dado que estos son los nombres que se utilizan en el dictado de la asignatura Ingeniería de Software.

Admiraal [2] resume los modelos que se especifican en RUP centrando la atención en las disciplinas de Modelado del Negocio, Requerimientos, Análisis / Diseño e Implementación. En la Figura 2 se presenta un diagrama de paquete que muestra la visión general de los modelos de RUP y las dependencias entre los modelos y las disciplinas en las que Admiraal se centra [2]. Nótese que Admiraal aun cuando no considera la disciplina de *Deployment*, recomienda realizar el Modelo de *Deployment* en la disciplina de Análisis y Diseño.

Como se mencionó previamente, en un proyecto de software que se desarrolla bajo una configuración de RUP, el trabajo se organiza en iteraciones en dos dimensiones: a lo largo de cada fase y a lo largo de cada disciplina.

Es importante entender que no solo las iteraciones de una configuración de RUP, se realizan recorriendo las disciplinas a lo largo de las fases, sino que además, las fases se recorren a lo largo de cada disciplina. El propósito de cada una de las fases se resume a continuación [3]:

Incepción: establecer una visión general para el negocio, alcance, esfuerzo en horas-hombre y costo del proyecto.

Elaboración: refinar la visión, definir la arquitectura, identificar los requerimientos principales, el alcance y los riesgos de la solución.

Construcción: implementar de manera iterativa los requerimientos en el orden de prioridades establecidas, preparar la instalación.

Transición: finalizar, instalar y entregar la versión del *release*. Realizar las pruebas de aceptación. Examinar el *release* y evaluar desde la perspectiva del negocio qué partes satisfacen la visión de acuerdo con el documento de visión.

En cada iteración se realizan las actividades correspondientes a la mayoría o a todas las disciplinas. Iteraciones a lo largo de las fases de Elaboración, Construcción y Transición deberían producir código operativo. Mientras que las iteraciones a lo largo de Incepción, generalmente no producen código. El propósito de las disciplinas en el núcleo de RUP se resume a continuación [3]:

Modelado del Negocio: comprender las necesidades del negocio, describir su funcionamiento y los servicios que ofrece.

Requerimientos: trasladar las necesidades del negocio en comportamientos de un producto de software con el fin de describir lo que el producto debe hacer.

Análisis y Diseño: trasladar los requerimientos a una arquitectura de software con el fin de guiar la implementación.

Implementación: transformar el diseño en código fuente utilizando los mecanismos lingüísticos de un lenguaje de programación, establecer y seguir un estándar de codificación, definir la organización del código en términos de implementación. Implementar clases y objetos en términos de componentes.

Prueba: realizar una evaluación objetiva del producto [1]. Esto incluye encontrar y corregir errores, validar que el producto opere tal como fue diseñado y verificar que los requerimientos hayan sido implementados.

*Deployment*: producir un *release* del producto y entregar el software a los usuarios finales.

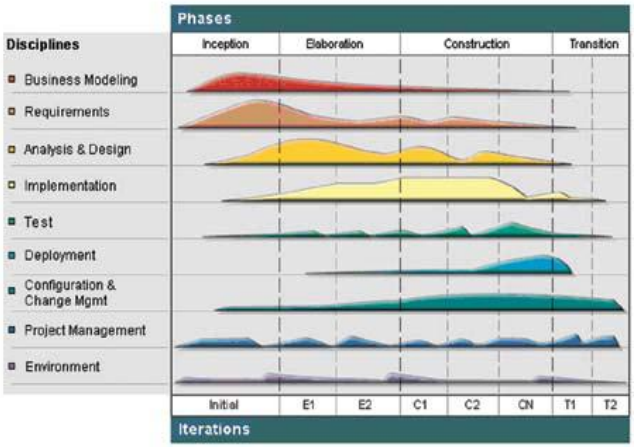


**Figura 1:** Disciplinas y Fases en RUP [4]

La idea central de las dos dimensiones en RUP es que el desarrollo consiste en realizar una serie de *release* incrementales o incrementos progresivamente más completos. Un *release* puede ser interno o externo. Los internos los utiliza el equipo de desarrollo para demostrar alguna característica o para realizar una presentación. Los externos se le entregan al Negocio. Cada incremento es el resultado de la iteración a lo largo de cada disciplina. Cada *release* es un producto operativo.

En la enseñanza y aprendizaje de los fundamentos de RUP si se comienza por las fases, usualmente los estudiantes caen en el error de interpretar las fases tal como se definen en el modelo de cascada, solo con nombres diferentes. Si se comienza por las iteraciones mostrando como un producto es el resultado de una serie de iteraciones y como el software y sus artefactos evolucionan a lo largo de esta serie de iteraciones hasta lograr un *release* que los estudiantes entregan para la evaluación del docente, esta idea es más fácil de comprender. En consecuencia y a nivel de la asignatura, se enfatiza en las iteraciones y la estructura de un producto.

Las iteraciones iniciales naturalmente tienden a centrarse en las disciplinas de Modelado del Negocio y Requerimientos, mientras que las iteraciones subsiguientes se concentran en la adaptación y retroalimentación. Por otra parte, en la disciplina de *Deployment* se incluyen las actividades necesarias para que las componentes del *release* en *Deployment* se preparen y se entreguen para su instalación. El proceso como tal de distribuir e instalar las componentes en *Deployment* no forma parte de RUP, siendo usualmente del dominio de un Departamento de Operaciones y no del grupo de desarrollo. Lo importante para los estudiantes es definir el proceso a seguir por su equipo de desarrollo para generar los artefactos de instalación y producir un documento que describa la versión.

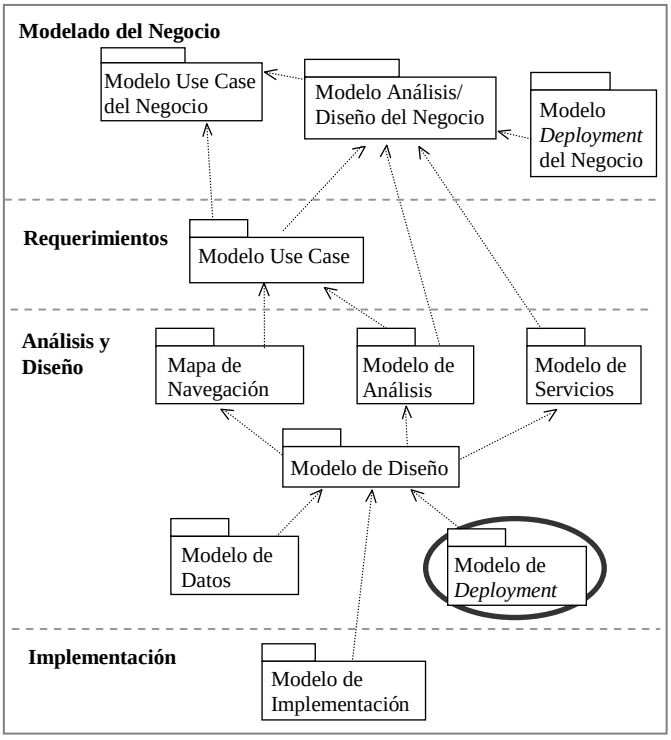


**Figura 2:** Modelos de RUP y Dependencias entre Modelos [2]

Este artículo está organizado como se describe a continuación: en la segunda sección se especifica el proceso de desarrollo, fundamentado tanto en RUP [3] como en la publicación de Admiraal [2], y utilizado a partir del semestre I-2011 en la asignatura de pregrado Ingeniería de Software para el

desarrollo de proyectos. En la tercera sección se presentan las perspectivas de este trabajo y se indican los resultados alcanzados.

## II. Definición del Proceso de Desarrollo

Un proceso puede contener diversos métodos y un método diversas técnicas. Exactamente cuando una técnica se transforma en método es difícil de precisar. El punto importante es que estos conceptos son necesarios. Un método describe como realizar algo. Un proceso es la simulación o ejecución de un método o colección de métodos ("haciendo algo"); refiere a una serie de acciones, cambios de estado o funciones que obtienen un resultado. Es útil recordar que un proceso tiene lugar en el "mundo real" mientras que la descripción de un proceso es lo que se documenta por ejemplo, con RUP.

Una metodología es una colección de métodos relacionados. Refiere al conjunto de prácticas (una práctica es una manera sistemática de realizar y lograr algo), procedimientos y reglas utilizados por quienes trabajan en una disciplina (por ejemplo, Tecnologías de Información) así como el estudio o análisis teórico de métodos. Un método refiere solo a una de las prácticas. Una técnica es un procedimiento sistemático con el cual realizar una tarea.

La importancia de utilizar un proceso de desarrollo para la construcción de software radica en la necesidad de reducir riesgos, como por ejemplo: retraso en las fechas pautadas para las entregas.

Un proceso de desarrollo guía las actividades a realizar durante el desarrollo de un software. Sin embargo, también es cierto que el hecho de utilizar un proceso de desarrollo no garantiza el éxito de una aplicación, entendiendo por éxito que el software produzca los resultados esperados, el Negocio esté satisfecho con el software producido y los riesgos mencionados anteriormente no hayan impactado el desarrollo.

Actualmente, existen diversos procesos de desarrollo; entre los más conocidos y difundidos se pueden mencionar las configuraciones de RUP [3], UP [5], FDD [6], XP [7], SCRUM [8]. La tendencia en empresas que desarrollan aplicaciones Web o buscan ser competitivas en el mercado, es utilizar procesos livianos o ligeros, es decir se generan solo aquellos artefactos de software necesarios.

En nuestro contexto académico, se tiene la limitación del tiempo y del escaso nivel de conocimiento que tienen los estudiantes acerca de procesos de desarrollo cuando cursan la asignatura Ingeniería de Software, siendo esta la primera asignatura en la cual los estudiantes adquieren conocimientos en el área. En consecuencia es recomendable utilizar un proceso definido que guíe a los estudiantes en el desarrollo de su proyecto en la asignatura. Adicionalmente, es necesario que el proceso de desarrollo utilice los conceptos y técnicas del temario de la asignatura.

El proceso de desarrollo utilizado que se describe a continuación se centra en las disciplinas del núcleo de RUP indicándose los artefactos que se generan para cada modelo, considerando las recomendaciones de Admiraal y en base a la experiencia adquirida en el uso de este proceso. Un modelo es una representación descriptiva grafica o textual, ejecutable o estática, de un subconjunto de propiedades de un sistema. Un artefacto puede ser un modelo, un elemento de un modelo o un documento. Un documento puede contener a su vez otros documentos. Es un producto usado o producido durante un proceso de desarrollo de software y es el resultado de una actividad. Ejemplos de artefactos incluyen modelos, código fuente, código ejecutable.

### *A. Disciplina de Modelado del Negocio*

Los estudiantes deben utilizar esta disciplina para desarrollar su proyecto. En la disciplina se incorpora el artefacto denominado "Tabla de Eventos del Negocio", que no está definido en RUP, con el fin de extraer de una descripción textual los eventos de interés para un actor del negocio. Un evento del negocio es algo que un actor del negocio puede solicitarle al negocio. Es importante destacar que un evento es instantáneo, mientras que el Negocio tiene que realizar una serie de acciones o actividades para atender un evento. La Tabla de Eventos del Negocio les facilita a los estudiantes la tarea de identificar los use case del negocio desde la perspectiva de un actor del negocio. Los eventos identificados se registran en una tabla de eventos conformada por dos columnas: Identificador de Evento y Descripción del Evento del Negocio. En base a la tabla de eventos se generan los siguientes modelos de RUP:

**Modelo Use Case del Negocio:** un use case del negocio presenta lo que el Negocio ofrece a los actores. El Negocio decidirá cómo realizarlos, bien sea manualmente, o bien sea automatizarlos parcial o totalmente.

Los documentos y productos que intervienen en un flujo de trabajo se denominan entidades del negocio. Las entidades del negocio pueden representarse como clases del negocio.

El modelo describe en términos de los use case del negocio las interacciones externas de la organización, lo que deben realizar tanto el Negocio como los actores del negocio para llevar a buen término un flujo de trabajo. Los elementos de este modelo son estables y facilitan el desarrollo de modelos subsecuentes que pueden no ser estables. Permite adquirir conocimientos acerca del dominio e identificar posibles soluciones a problemas del negocio. Estos modelos evolucionan sobre periodos extendidos de tiempo, si el Negocio modifica la manera en que opera y/o sus productos. Los diagramas a utilizar son:

***Diagrama Use Case:*** identifica los use case del negocio, los actores del negocio y sus relaciones. Un use case del negocio modela los servicios que el Negocio ofrece a los actores del negocio, y es independiente de las tecnologías. Una estrategia que permite evitar la consideración de funcionalidades de sistema en los use case del negocio consiste en considerar que el negocio implementa sus servicios de manera no computarizada.

Un trabajador del negocio no es un actor del negocio, porque es parte del negocio. Es un representante del negocio con el cual un actor del negocio interactúa y que se transformará en un actor en los use case de sistema, donde utiliza el sistema como herramienta para prestarle un servicio a los actores del negocio. Es importante tener en cuenta que en los use case del negocio, no se diferencia entre actores primarios y secundarios. Es decir, solo hay actores del negocio. Para identificar los use case del negocio, se consideran únicamente los eventos registrados en la tabla de eventos. En la especificación de cada

uno de los use case del negocio se presenta tanto el flujo básico como los flujos alternativos de trabajo del negocio, es decir, qué hace el negocio para ofrecerle algo de valor al actor dado que éste ha generado uno o más eventos en la interacción con el negocio. Los pasos de un proceso de negocio se realizan en uno o más use case de sistema.

El diagrama use case no describe los procesos del negocio; para esto se utilizan diagramas de actividad. Para especificar los use case del negocio y establecer los pasos en el flujo de eventos, se puede utilizar alguno de los siguientes artefactos:

***Diagrama de Actividad:*** describe las acciones y los resultados asociados a un flujo de eventos de un use case del negocio.

***Plantilla de Especificación*** de los use case: describe de manera rigurosa que se hace cuando un actor interactúa con el use case (ver Apéndice A1).

**Modelo de Análisis/Diseño del Negocio:** modela el funcionamiento interno de la organización para realizar los use case del negocio. Se modela también la estructura organizacional y el flujo de la información en caso de que se considere relevante. Los diagramas a utilizar son:

***Diagrama de Clase:*** identifica las clases del negocio y las relaciones entre ellas. Se corresponde con la estructura de la organización y de la información. Es de destacar que el modelado orientado a objetos (OO) específicamente modela el comportamiento. Los objetos existen en un sistema computacional y están sujetos a restricciones y acciones del sistema, poseen un ciclo de vida: se crean y se destruyen, pueden ser asignados a otros objetos y tienen un comportamiento. Al definir un objeto/clase asegúrese de considerar que existe en el sistema y que tiene comportamientos para su existencia. Los trabajadores del negocio en principio no son clases del negocio. Pueden llegar a serlo dependiendo de las necesidades de persistencia de la información. Algunos criterios que pueden ayudar a decir si se representan como clases se presentan a continuación:

¿El trabajador del negocio tiene un comportamiento identificable en el dominio del problema? Esto es, ¿se pueden nombrar servicios o funciones necesarias en el dominio del problema que son propias del trabajador y que este provee? Indíquelos.

¿El trabajador del negocio tiene relaciones con otros trabajadores del negocio? Analícelas e incorpórelas.

¿El trabajador del negocio actúa dentro de los límites del sistema? Si no lo hace, puede ser un actor del sistema.

¿El trabajador del negocio posee una estructura identificable? Esto es, ¿es posible identificar algún conjunto de atributos que el trabajador posee y que deben administrarse? Agréguelos.

***Diagrama de Actividad:*** se usa principalmente para modelar el flujo de trabajo y es útil para analizar los use case describiendo las acciones que necesitan realizarse, cuándo se realizan y quién es el responsable de realizarlas; no da el detalle de cómo cooperan o colaboran los objetos, por eso no substituye un diagrama de secuencia. Los trabajadores del negocio y los actores del negocio se representan en las particiones del diagrama y éstas contienen las acciones o actividades. Las entidades del negocio son las entradas y / o salidas de las acciones o actividades. Las notas se utilizan para indicar que un objeto del negocio se genera en una acción o actividad para un actor del negocio. Una acción o actividad puede corresponderse con un use case del negocio. Se diferencian los objetos físicos de los objetos de información [9]. El estereotipo *<<Information>>* se utiliza para indicar que un objeto es de tipo información y el estereotipo *<<Physical>>* se utiliza para indicar que un objeto representa un objeto físico real. No deben existir transiciones entre objetos de información que representen un flujo de control.

***Diagrama de Secuencia:*** representa el flujo de trabajo centrado en el intercambio de mensajes entre entidades del negocio. Las entidades del negocio representan las instancias de las clases del negocio identificadas en el diagrama de clase del negocio. Los trabajadores del negocio y actores del negocio se representan con el icono de actor y el estereotipo *<<user>>*. Los trabajadores del negocio intercambian mensajes con las entidades del negocio.

*B. Disciplina de Requerimientos*

Se utilizan los modelos generados en la disciplina de Modelado del Negocio para elaborar los requerimientos del sistema para el Negocio. En la práctica, el Negocio establece cuáles de los use case del negocio que han sido identificados y especificados van a ser automatizados actualmente; a futuro se considerará y planificará la automatización de otros use case del negocio. Los use case del negocio se identifican durante la disciplina de Modelado del Negocio y describen un proceso del negocio. Los use case (de sistema) se identifican durante la disciplina de Requerimientos y describen un requerimiento funcional desde la perspectiva de un actor; muestran las funcionalidades del sistema para que los actores logren sus objetivos.

La experiencia en la realización del proyecto del semestre II-2010, motivó la incorporación del artefacto denominado "Tabla de Eventos del Sistema" que no está definido en RUP. Este artefacto facilita la tarea de identificar los requerimientos funcionales (lo que el producto debe hacer [10]), no funcionales (ciertas características de calidad que el producto debe poseer [10]) y las restricciones (del negocio o las del uso de herramientas que apoyan la generación de artefactos de software). Los eventos del sistema se registran en una tabla de eventos.

Para la identificación de los requerimientos funcionales del sistema se consideran las especificaciones de los use case del negocio. Se utiliza la tabla de eventos para registrar los eventos del sistema.

Una estrategia para comenzar a identificar eventos del sistema es trasladar cada una de las acciones o actividades que están en los pasos del flujo básico en la respuesta del negocio de los use case del negocio y colocarlos en cada fila de la tabla. Luego se examinan y se transforman utilizando terminología de sistema. Además se analiza si existen eventos que puedan ser generalizados o si existen eventos que forman parte de otros eventos. Por otra parte, se identifica y analiza cuáles otros eventos pueden ocurrir a nivel de sistema, eventos que no se identificaron y/o se consideraron en el Modelo del Negocio por ser internos al negocio.

La tabla de eventos del sistema está conformada por siete columnas: Identificador de Evento, Descripción del Evento del Sistema, Restricciones, Actor, Prioridad, Identificador del Use

Case del Negocio con el cual se relaciona, e Identificador del Use Case del Sistema (se indica una vez generado el Modelo Use Case del Sistema). La prioridad indica el orden en que los eventos del sistema van a ser diseñados e implementados; su rango de valores va de 1 al número máximo de eventos. El valor 1 se asocia al evento de mayor importancia para el Negocio (o para el Desarrollo si el evento debe considerase antes de diseñar o implementar otro evento) y el valor máximo de prioridad se le asigna al evento que se considere de menor importancia. Un evento del sistema que ya se implementó no tiene prioridad y se indica con el identificador "IMP".

Una vez identificados los eventos del sistema, se genera el siguiente modelo de RUP:

**Modelo Use Case:** describe las interacciones entre los actores y el sistema, y la meta de los actores al usar el sistema (use case). Para identificar los use case del sistema se utiliza la tabla de eventos y generalmente la correspondencia no es uno a uno. Se utilizan los siguientes diagramas:

***Diagrama Use Case del Sistema:*** describe lo que debe hacer el sistema para automatizar uno o más pasos de la realización del use case de negocio. Se representan los use case del sistema, los actores del sistema y las relaciones entre los use case y sus actores. Un actor puede corresponderse con un actor del negocio, en caso de que el actor del negocio acceda al sistema. Un actor primario es aquel que inicia un use case y obtiene un beneficio cuando se obtiene el propósito del use case, un actor secundario participa en obtener el propósito. Si se identifica un use case sin actor, esto probablemente es el resultado de una descomposición funcional. No debería asociarse más de un actor con un use case debido a que la especificación se redacta desde la perspectiva de un solo actor que tiene un propósito específico. Tratar de describir un flujo de eventos desde más de una perspectiva es confuso y lleva a descripciones sobrecargadas y difíciles de comprender. Por convención, los actores primarios se colocan del lado izquierdo y los actores secundarios del lado derecho en el diagrama use case [11].

Los diagramas use case son una herramienta que comunica el alcance de un negocio o sistema; la información importante está en la especificación de estos. Es de destacar que los use case deben ser cajas negras, describiéndose solamente el comportamiento que es visible para los actores. Para especificar los use case y establecer los pasos en el flujo de eventos, se puede utilizar alguno de los siguientes artefactos:

***Diagrama de Actividad:*** documenta flujos de trabajo del negocio ante las solicitudes del actor. Puede ser: (1) detallado cuando es necesario comprender un proceso complejo del negocio (en la disciplina de Modelado del Negocio) o (2) simplificado para el actor y el sistema. El caso 2 documenta los detalles del use case y describe las acciones y los resultados asociados a un flujo de eventos de un use case del sistema.

***Plantilla de Especificación*** de use case del sistema: la descripción se presentó en la subsección *A*. A nivel de sistema se distingue entre actores primarios y secundarios.

Un diagrama de actividad describe las actividades de un actor o conjunto de actores, mientras que un use case describe las interacciones con un sistema que permiten realizar las actividades.

### *C. Disciplina de Análisis y Diseño*

El Modelo de Análisis presenta un "diseño preliminar" de un conjunto de requerimientos; el Modelo de Diseño muestra como las tecnologías seleccionadas realizan el Modelo de Análisis. Admiraal [2] recomienda no realizar un Modelo de Análisis, argumentando que el Modelo de Análisis del Negocio y el Modelo Use Case proveen suficiente información que permiten hacer un primer esbozo de la arquitectura de componentes en el Modelo de Diseño y para comenzar a hacer las realizaciones de los use case en términos de las componentes que interactúan. Esta sugerencia fue considerada en los semestres I-2011 y II-2011, sin embargo se detectó dificultad por parte de los estudiantes al momento de comprender y trabajar con el diagrama de clase de diseño. Por supuesto que la experiencia de Admiraal es distinta a la de los estudiantes de Ingeniería de Software, pero consideramos conveniente desarrollar tanto el Modelo de Análisis como el Modelo de Diseño que comprenden los siguientes modelos.

**Modelo de Análisis:** se analizan y refinan los requerimientos del modelo use case para obtener una visión detallada de los requerimientos del sistema. El modelo de análisis se describe en un lenguaje para desarrolladores y proporciona una visión general y conceptual del sistema respecto a lo que se tiene qué hacer y no cómo se va a hacer [12]. Por este motivo es que es un modelo útil y conveniente ya que facilita comprender el sistema sin mostrar detalles de alternativas de diseño que pueden variar y están atadas al entorno de implementación. El Modelo de Análisis se considera como una versión inicial del Modelo de Diseño. Los diagramas a utilizar son:

***Diagrama de Clase:*** representa la estructura estática del sistema con las clases, atributos, operaciones y relaciones que van a ser diseñadas e implementadas. Se incorporan en las clases del Modelo de Análisis / Diseño del Negocio atributos y responsabilidades u operaciones a un nivel alto de abstracción, y se etiquetan las clases con un estereotipo para categorizar las clases como interfaz, entidad o control. Puede ocurrir que ciertas clases entidad del negocio se conviertan en atributos de otras clases o no sean consideradas como clases entidad. El uso de estereotipos se omite en el caso de las clases entidad para no sobrecargar visualmente al diagrama. Las clases entidad se utilizan en el Modelo de Análisis para modelar la persistencia de información.

Las clases interfaz se indican con el estereotipo *<<boundary>>* o *<<b>>* y se identifican a partir de las interacciones entre los actores del sistema y los use case del sistema. Los atributos de las clases interfaz se identifican a partir del flujo básico y/o alternativo de las especificaciones de los use case en los que un actor interactúa con objetos del sistema y el sistema realiza acciones sobre objetos como consecuencia de las interacciones. Las clases interfaz se utilizan en el Modelo de Análisis para modelar las interacciones entre el sistema y sus actores.

Las clases control se definen para evitar que las clases interfaz tengan relación de asociación con las clases entidad. Decisiones respecto al orden en que se instancian las clases interfaz, o acciones a realizar cuando un elemento gráfico se presiona, por ejemplo un botón, deben implementarse en clases control y no en las clases interfaz ya que por lo general esas acciones involucran la consulta, recuperación o almacenamiento de data en las clases entidad. Sin embargo, las

validaciones de datos capturados por un objeto interfaz pueden definirse en su clase interfaz. Las clases control se indican con el estereotipo <<*control*>> o <<*ctrl*>>. Es útil y recomendable realizar un pseudocódigo para identificar las operaciones en las clases control así como operaciones adicionales en las clases entidad e interfaz.

***Diagrama de Secuencia:*** representa el orden de envío de mensajes entre instancias de clases que sean de interés, y para identificar nuevas operaciones de las clases. En análisis el diagrama documenta solo las interacciones que son entradas y resultados. Pueden evolucionar a lo largo de un proyecto cuando se agregan instancias que representan decisiones de diseño. Se muestra el actor y los objetos del sistema así como los mensajes de interacción. El diagrama de secuencia permite describir un comportamiento que es más complejo de lo que se ve a simple vista, así como identificar las asociaciones y las operaciones que se requieren. Es importante tener en cuenta que un actor del sistema puede enviar mensajes solo a objetos interfaz y no a objetos control ni a objetos entidad.

**Modelo de Mapa de Navegación:** describe la secuencia de navegación que puede recorrer un actor del sistema. Una relación entre un use case y un actor implica la existencia de una interfaz. Si el actor es humano, es una interfaz usuario; si es un sistema es una interfaz de sistema. La interfaz de sistema se especifica en el diagrama de clase de diseño donde se definen los métodos de la clase con el estereotipo <<sistema>>. Se utilizan los siguientes artefactos:

***Prototipos:*** muestra los elementos de la interfaz gráfica de usuario. Los prototipos pueden generarse con una herramienta o manualmente. Por lo general, cada prototipo de interfaz usuario se corresponde con una clase interfaz.

***Diagrama de Estado:*** representa la secuencia de navegación entre las instancias de las clases interfaz del sistema. Las instancias de las clases interfaz del sistema se representan con estados en el diagrama y las transiciones representan los posibles caminos de navegación, resultado de interacciones.

**Modelo de Diseño:** El lenguaje de programación utilizado en la asignatura Ingeniería de Software es *Java*™ [13] con el entorno de desarrollo *NetBeans* [14]. Estas herramientas se seleccionaron dado que por una parte *Java* es un lenguaje orientado a objetos, por lo tanto los conceptos cubiertos en el programa de la asignatura corresponden directamente a propiedades que se implementan en *Java*, por ejemplo, clase/objeto. Por otra parte, es un lenguaje ampliamente documentado, utilizado generalmente en las organizaciones. El uso de *NetBeans* facilita el diseño e implementación de los prototipos de interfaz y se cumple con uno de los objetivos de la asignatura respecto a utilizar entornos de desarrollo.

El Modelo de Diseño especifica el diseño detallado de las clases. Los diagramas a generar son:

***Diagrama de Clase:*** se especifican propiedades de las clases entidad, interfaz y control: tipo de datos y visibilidad de los atributos y operaciones especificados en las clases del Modelo de Análisis. Una operación especificada en una clase del Modelo de Análisis se convierte en uno o más métodos en el Modelo de Diseño. Se analiza si es posible generalizar operaciones de las clases de análisis. Atributos especificados en una clase de análisis se pueden convertir en clases de diseño. Las clases de diseño pueden etiquetarse con estereotipos para reflejar decisiones de implementación en un lenguaje de programación, por ejemplo, una clase interfaz que va a ser implementada bajo el entorno de desarrollo *NetBeans* se puede etiquetar con el estereotipo <<*form*>>. Si se utilizan patrones de diseño [15], generalmente se agregan atributos, métodos, relaciones y eventualmente clases para dar solución a un problema específico de diseño. Puede ser de utilidad incorporar notas con pseudocódigo en los métodos de clases.

Es útil y recomendable generar el pseudocódigo para identificar si existen los métodos en las clases de control que permiten realizar las funcionalidades o si es necesario definir otros métodos en las clases entidad e interfaz. Este pseudocódigo facilita determinar las funcionalidades de los métodos, así como definir y comprender el flujo de control de la aplicación. Se especifican estructuras de datos para implementar las relaciones 1:N.

***Diagrama de Secuencia:*** se muestran los intercambios de mensajes entre los objetos de diseño y son más detallados que los diagramas de secuencia que se generan en el Modelo de Análisis.

### *D. Disciplina de Implementación*

Se establece el estándar de codificación en cuanto a nombramiento de clases, métodos y atributos. En la asignatura se utiliza el estándar que se presenta en el Apéndice A2.

Se realiza el siguiente modelo:

**Modelo de Implementación:** describe la implementación del diseño del sistema y se utilizan los siguientes artefactos de software:

***Código Fuente Documentado*** en el lenguaje de programación *Java*™.

***Diagrama de Paquete:*** se utiliza para organizar clases en los subdirectorios de un proyecto bajo *NetBeans* de acuerdo con un criterio.

### *E. Disciplina de Prueba*

En RUP se distinguen cuatro tipos de prueba [3]: unitaria, de integración, de sistema y de aceptación. Las pruebas que realizan los estudiantes de la asignatura en el proceso RUP-GDIS son pruebas unitarias y de integración. Las pruebas de sistema y de aceptación se realizan en el contexto académico con la entrega que hacen los estudiantes a los miembros del grupo docente. Se realiza el siguiente modelo:

**Modelo de Prueba:** describe las pruebas. Debe indicarse el identificador de clase, el identificador del caso de prueba, su descripción, y un reporte del resultado de la prueba.

***Especificación de Casos de Prueba:*** describe cuáles son los datos con los que se ejecuta el caso de prueba (ver Apéndice A3).

En la Tabla I se resumen, por cada una de las disciplinas, los modelos que se consideran en el proceso RUP-GDIS y los artefactos de software que se pueden utilizar, y en la Figura 3 se resumen los modelos considerados en el proceso RUP-GDIS y las dependencias entre los modelos y las disciplinas.

**Tabla I:** Artefactos de RUP-GDIS

| **Disciplina** | **Modelos a generar en la disciplina** | **Artefactos de software utilizados** | **Observación** |
|---|---|---|---|
| Modelado del Negocio | | - Tabla de Eventos del Negocio | |
| | Modelo Use Case del Negocio | - Diagrama use case<br>- Plantilla para especificar use case, o<br>- Diagrama de actividad | La especificación de cada use case del negocio se realiza utilizando la plantilla o un diagrama de actividad |
| | Modelo de Análisis/Diseño del Negocio | - Diagrama de clase<br>- Diagrama de actividad<br>- Diagrama de secuencia | |
| Requerimientos | | - Tabla de Eventos del Sistema | |
| | Modelo Use Case | - Diagrama use case<br>- Plantilla para especificar use case, o<br>- Diagrama de actividad | |
| Análisis y Diseño | Modelo de Análisis | - Diagrama de clase<br>- Diagrama de secuencia | |
| | Modelo de Mapa de Navegación | - Prototipos<br>- Diagrama de estado | |
| | Modelo de Diseño | - Diagrama de clase<br>- Diagrama de secuencia | |
| Implementación | Modelo de Implementación | - Código fuente operativo y documentado<br>- Diagrama de paquete | |
| Prueba | Modelo de Prueba | - Especificación de casos de prueba | Identificador de clase, identificador de caso de prueba y su descripción, reporte del resultado de la prueba |

No se considera a la disciplina de *Deployment* dado que un producto de software desarrollado como proyecto en la asignatura no se entrega a usuarios finales. La entrega se realiza al grupo docente de la asignatura para su evaluación. Los modelos especificados en el proceso RUP-GDIS y sus dependencias se basan en Admiraal [2] y están adaptados para ser generados y utilizados por estudiantes del 3er semestre.

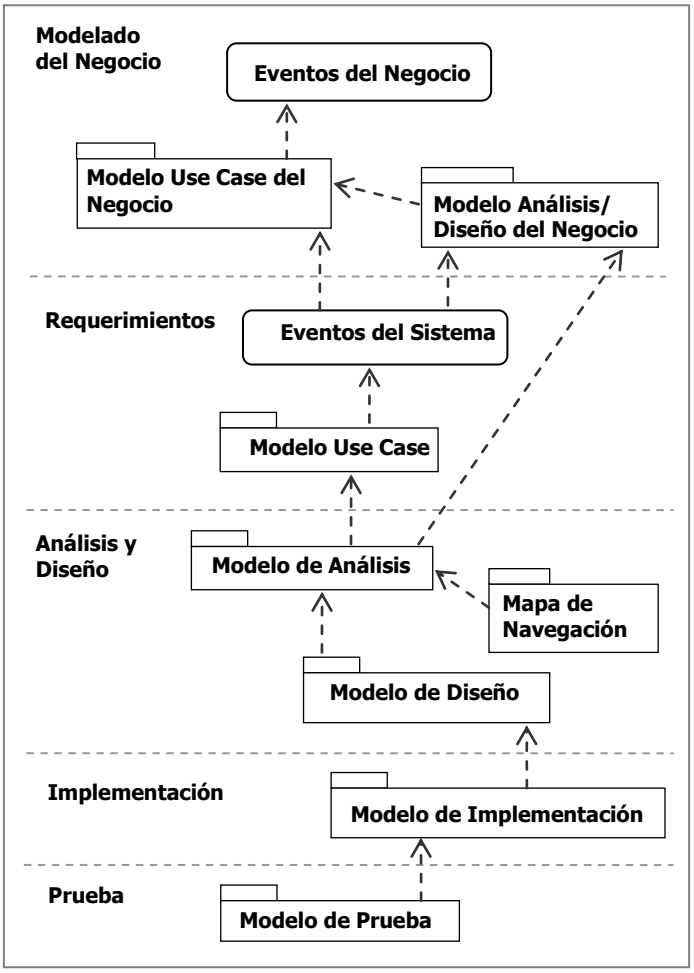


**Figura 3:** Modelos en RUP-GDIS y sus Dependencias

*F. Síntesis de Recomendaciones*

A continuación en la Tabla II se resumen las indicaciones que se recomiendan seguir al utilizar el proceso RUP-GDIS.

**Tabla II:** Recomendaciones

| **Modelos** | **Artefactos** | **¿Qué hacer?** |
|---|---|---|
| *Modelo Use Case del Negocio* | Tabla de Eventos del Negocio | Identificar eventos de interés para los actores del negocio. |
| | Diagrama UCN | UCN se identifican únicamente a partir de los eventos del negocio.<br>Evitar considerar funcionalidades de sistema –> pensar que todo se realiza manualmente.<br>Un trabajador del negocio no es un actor del negocio.<br>Un UCN no describe un proceso del negocio. |
| | Plantilla para especificar use case | Eventos del negocio son entrada del actor.<br>Indicar acciones/actividades que realiza el Negocio para atender un evento.<br>No usar términos que refieran a un sistema automatizado.<br>Usar predicados y lógica de primer orden.<br>No colocar elementos que informa un actor.<br>Considerar propiedades de entidades del negocio. |

| Modelos | Artefactos | ¿Qué hacer? |
|---|---|---|
| | Diagrama de Clase | Identificar clases/objetos involucrados en las especificaciones de los use case.<br>Actores del negocio y trabajadores del negocio en principio no se modelan como clases del negocio.<br>Asegúrese que un objeto/clase existe en el sistema y tiene comportamiento.<br>¿El trabajador del negocio tiene un comportamiento identificable en el dominio del problema? Esto es, ¿se pueden nombrar servicios o funciones necesarias en el dominio del problema que son propias del trabajador y que este provee? Indíquelos.<br>¿El trabajador del negocio tiene relaciones con otros trabajadores del negocio? Analícelas e incorpórelas.<br>¿El trabajador del negocio actúa dentro de los límites del sistema? Si no lo hace, puede ser un actor del sistema.<br>¿El trabajador del negocio posee una estructura identificable? Esto es, ¿es posible identificar algún conjunto de atributos que el trabajador posee y que deben administrarse? Agréguelos. |
| *Modelo de Análisis / Diseño del Negocio* | Diagrama de Actividad | Solo para los UCN donde el diagrama representa algo de valor.<br>Usar estereotipos.<br>No deben existir transiciones entre objetos de información. |
| | Diagrama de Secuencia | Un diagrama de secuencia por cada UCN.<br>No se genera si no existen interacciones entre dos o más objetos en el UCN. |
| | Tabla de Eventos del Sistema | Utilizar especificaciones de UCN.<br>Trasladar acciones y/o actividades del flujo básico del negocio a la tabla.<br>Examinar y transformar usando terminología de sistema.<br>Generalizar o combinar eventos.<br>Identificar y analizar nuevos eventos a nivel de sistema.<br>Eventos se consideran operaciones en pseudo-lenguaje, cercanos al lenguaje de programación. |
| *Modelo Use Case* | Diagrama Use Case | UC se identifican a partir de los eventos del sistema.<br>Actores del sistema pueden ser actores del negocio o trabajadores del negocio.<br>¿Quién inicia un UC? ¿Quién participa en el UC?<br>No usar relaciones entre UC sino después de varias iteraciones.<br>No debería asociarse más de un actor con un use case que tiene un propósito específico. |
| | Plantilla para especificar use case | Utilizar como base las especificaciones de los UCN.<br>Incluir funcionalidades de sistema. |
| *Modelo de Análisis* | Diagrama de Clase | Clases entidad del negocio pueden representarse como atributos de otras clases.<br>Qué tiene que hacer el sistema, no cómo se hace.<br>Atributos de clases interfaz se identifican a partir del flujo básico y/o alternativo de las especificaciones.<br>Clases control se definen para evitar asociar clases interfaz con clases entidad.<br>Decisiones respecto al orden en que se instancian las clases interfaz, o acciones a realizar cuando un elemento gráfico se presiona deben implementarse en clases control.<br>Validaciones de datos capturados por un objeto interfaz pueden definirse en su clase interfaz. |
| | Diagrama de Secuencia | Un actor del sistema puede enviar mensajes solo a objetos interfaz.<br>Identificar nuevas asociaciones y operaciones. |
| *Modelo Mapa de Navegación* | Prototipos | No agregar botones para capturar data de *text field*. |
| | Diagrama de Estado | Nombrar estados según acción.<br>No referir a elementos gráficos en el nombre de eventos en transiciones.<br>Usar transiciones internas en los estados para el ingreso de valores. |
| *Modelo de Diseño* | Diagrama de Clase | Agregar propiedades a clases entidad, interfaz y control.<br>Incorporar métodos.<br>Especificar tipo, estructuras de datos y visibilidad.<br>Usar estereotipos.<br>Usar Patrones de diseño.<br>Incorporar notas con pseudocódigo en los métodos de clases. |
| | Diagrama de Secuencia | Solo si se especifican intercambios de mensajes que no se especifican en el UC correspondiente. |
| *Modelo de Implementación* | Código fuente | Respetar el estándar de codificación. |
| | Diagrama de Paquete | Definir un criterio para agrupar clases. |
| *Modelo de Prueba* | Especifica-ción de casos de prueba | Considerar pre y post condiciones especificadas en los UC.<br>Utilizar pruebas caja blanca y caja negra. |

## III. Perspectivas

El grupo docente de la asignatura Ingeniería de Software de la Licenciatura en Computación debe enfrentar el reto de plantear de manera consciente el "qué", "por qué", "cuándo" y "cómo" realizar las diferentes actividades de un proceso de desarrollo de software. En este trabajo se describió el proceso de desarrollo RUP-GDIS que ha sido y está siendo utilizado para el desarrollo del proyecto de la asignatura desde el semestre I-2011. RUP-GDIS es una configuración de RUP que se centra en sus primeras cinco disciplinas.

A través de la utilización de RUP-GDIS durante un semestre se estudian seis diagramas UML a saber: diagrama use case, diagrama de actividad, diagrama de secuencia, diagrama de clase, diagrama de estado y diagrama de paquete, que se correlacionan con los objetivos de la asignatura, siendo la premisa subyacente que el desarrollo del proyecto provee a los estudiantes de una base para el desarrollo de futuros proyectos. La evolución del proyecto a lo largo de varios semestres fomenta en los estudiantes la habilidad de desarrollar

componentes que deben incorporar en un software existente, promoviendo asimismo técnicas de resolución de problemas que podrán aplicar en situaciones y problemas similares.

La utilización de RUP-GDIS se evalúa al final de cada semestre con un cuestionario anónimo para identificar los elementos en los cuales los estudiantes presentan dificultades. Es importante destacar que a medida que transcurren los semestres el proceso está sujeto a modificaciones en base al análisis de las respuestas que indican los estudiantes en el cuestionario.

Por ejemplo, para identificar las dificultades al elaborar diagramas de clase, se presentan las siguientes alternativas: a) Identificar clases; b) Identificar asociaciones; c) Uso de Generalización/Especialización; d) Evitar asociaciones redundantes; e) Especificación de multiplicidades en asociaciones; f) Identificar atributos en las clases; g) Identificar operaciones en las clases; h) Otras razones, especifique; i) NO tuvo dificultades.

La evaluación de los resultados del cuestionario permite determinar las dificultades que enfrentan los estudiantes al utilizarlo y guía las posibles modificaciones que pueden ser pertinentes de realizar tanto al proceso como a las actividades de docencia. Las evaluaciones realizadas en los semestres I-2012 y II-2012 pueden ser consultadas en Niño [16] y en los informes de docencia correspondientes a los semestres II-2014 y I-2015 que se encuentran en el Departamento de la Escuela de Computación. En el Apéndice A4 se presentan las modificaciones más significativas que se han realizado al proceso. Por ejemplo, se amplió la explicación del concepto "evento del negocio" en la disciplina Modelado del Negocio y se incluyó una estrategia para identificar eventos del sistema en la disciplina de Requerimientos. Las modificaciones al proceso definido originalmente en el año 2011 se realizaron con la finalidad de resolver las dificultades que enfrentan los estudiantes con el concepto de evento, en particular evento del negocio, así como aportar estrategias, lineamientos y recomendaciones para la elaboración de ciertos artefactos de software.

El resultado de este trabajo es un proceso de desarrollo de software definido y documentado, a disposición en la Escuela de Computación, proceso que actualmente se utiliza sistemáticamente en la asignatura Ingeniería de Software. Esperamos que esta experiencia aporte a los estudiantes conocimientos y habilidades esenciales para su desempeño en sus estudios y en su vida profesional.

## REFERENCIAS

[1] S. Ambler, *A Manager's Introduction to The Rational Unified Process (RUP)*, http://www.ambysoft.com/unifiedprocess/rupIntroduction.html.

[2] H. Admiraal, *Pitfalls using UML in RUP*, https://www.scribd.com/document/74841882/Pitfalls-Using-UML-in-RUP-part-1 & http://www.sparxsystems.com/downloads/whitepapers/Pitfalls%20using%20UML%20in%20RUP%20_part%202_.pdf.

[3] Rational Unified Process, *Rational Unified Process. Best Practices for Software Development Teams*, Rational Software Corporation White Paper, TP026B, https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf.

[4] http://www.ibm.com/developerworks/library/ws-soa-term2/index.html.

[5] I. Jacobson, G. Booch and J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*, Addison-Wesley, 2000.

[6] http://www.featuredrivendevelopment.com.

[7] D. Wells, *Extreme Programming: A Gentle Introduction*, http://www.extremeprogramming.org.

[8] K. Schwaber and J. Sutherland, *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*, Scrum.org., http://www.scrum.org.

[9] H. Erickson and M. Penker, *UML Toolkit*, John Wiley & Sons, Inc. 1998.

[10] Essi-Scope, *Quality Characteristics*, http://www.cse.dcu.ie/essiscope/index.html.

[11] C. Larman, *Chapter 6: Use-Case Model: Writing Requirements in Context* en *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Designs and The Unified Process*, Second Edition, Prentice Hall, 2001.

[12] http://people.cs.uchicago.edu/~matei/CSPP523/lect4.ppt.

[13] Java™, http://www.java.com/en.

[14] NetBeans, http://netbeans.org.

[15] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Dessign Patterns. Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.

[16] N. Niño, *Evaluación Cuantitativa de la Enseñanza y el Aprendizaje de un Proceso de Desarrollo de Software en el Pregrado de la Licenciatura en Computación, UCV*, Universidad Central de Venezuela, Trabajo de Ascenso para optar a la categoria de Asociado, Caracas, Venezuela, Enero 2014.

[17] C. Metzner and N. Niño, *El Proceso de Desarrollo RUP-GDIS*, Lecturas en Ciencias de la Computación, Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, ISSN 1316-6239, ND 2012- 03, pp. 1–14, Caracas, Venezuela, Septiembre 2012.

## APÉNDICES

En esta sección se incluyen en el Apéndice A1 la plantilla para especificar tanto los use case del negocio como los use case del sistema. En el Apéndice A2 se presentan algunos elementos del estándar de codificación a usar en la disciplina *Implementación*. En el Apéndice A3 se presenta la plantilla utilizada para especificar casos de prueba y en el Apéndice A4 se presentan las modificaciones realizadas a la publicación ND-2012-03 [17].

*Apéndice A1: Plantilla para Especificar Use Case*

1. **Identificador y Nombre:** UC[N]# - Nombre del Use Case
   1.1. **Breve Descripción:** descripción y propósito
   1.2. **Actores:** lista de actores que participan. A nivel de sistema se distingue
   entre actores primarios y actores secundarios, a nivel de negocio no se hace esta distinción
   1.3. **Flujo de Eventos:**
      1.3.1. **Flujo Básico:** describe el proceso normal

| | Entrada del actor | Respuesta del Negocio / Sistema |
|---|---|---|
| 1.- | Eventos que realiza un actor cuando interactúa con el Negocio | Secuencia de acciones o actividades que realiza el Negocio para atender un evento que genera el actor |
| 2.- | … | …<br>El use case finaliza |

      1.3.2. **Flujos Alternativos:** describe procesos alternativos
      **Alternativa 1:** nombre de la alternativa. Nota: las pre y post condiciones se satisfacen solo para el flujo básico

| | Entrada del actor | Respuesta del Negocio / Sistema |
|---|---|---|
| 1.- | | Secuencia de acciones o actividades que realiza el Negocio cuando ocurre la condición bajo la cual se realiza el flujo alternativo |

   1.4. **Requerimientos Especiales:** identifique requerimientos adicionales. Incorpore en esta sección requerimientos no funcionales
   1.5. **Pre-condición:** predicados que deben satisfacerse antes de realizar el use case
   1.6. **Post-condición:** predicados que se satisfacen cuando el use case finaliza exitosamente (flujo básico)
   1.7. **Puntos de Extensión:**
      1.7.1. **Include:** lista de identificador y nombre de los use case con los cuales mantiene relación de inclusión
      1.7.2. **Extend:** lista de identificador y nombre de los use case con los cuales mantiene relación de extensión

*Apéndice A2: Estándar de Codificación*

El estándar de codificación utilizado en cuanto a nombramiento de clases, métodos y atributos es:

- clases se colocan tipo título, por ejemplo: AgendaDeCitas
- clases control se inician con la palabra Ctrl, por ejemplo: CtrlCentral
- la primera letra de los atributos y métodos se coloca en minúscula
- el nombre de la asociación entre clases se coloca con la inicial de las tres o cuatro primeras letras de la clase y en minúscula.

*Apéndice A3: Plantilla para Especificar Casos de Prueba*

**Id Prueba**: #
**Tipo de Prueba:** se indica el tipo de prueba a ser considerada
**Descripción**: breve descripción del propósito de la prueba
**Clase:** nombre de la clase a ser probada
**Método:** signatura del método a ser probado
**Tipo de retorno:** tipo de retorno del método
**Pre-condición:** predicados que deben satisfacerse antes de realizar la prueba
**Post-condición:** predicados que deben satisfacerse cuando se realiza la prueba
**Casos de prueba:** conjunto de datos con los que se va a ejecutar el software
**Valor esperado:** valor(es) que se debe(n) obtener después de la ejecución
**Resultado:** valor(es) que se obtienen como salida una vez realizada la prueba (experimentación)

*Apéndice A4: Modificaciones*

| Sección | Descripción |
|---|---|
| 2.1 | Se amplió la explicación del concepto "evento del negocio" |
| 2.1 | Se modificó el orden en que se generan los diagramas del Modelo de Análisis / Diseño del Negocio |
| 2.2 | Se incluyó una estrategia para identificar eventos del sistema |
| 2.3 | Se modificó el orden en que se generan los modelos en la disciplina de Análisis y Diseño |
| 2.3 | Se destaca la utilidad de generar pseudocódigo |
| 2.3 | Se incluyó un lineamiento respecto al envío de mensajes de un actor |
| 2.3 | Se incluyó un lineamiento para identificar métodos en la disciplina Análisis y Diseño |
| 2.5 | Se indican los tipos de pruebas que utilizan los estudiantes de la asignatura |
| Tabla I | Se modificó el orden en que se generan los artefactos de software de los modelos en la disciplina Modelado del Negocio y en la disciplina Análisis y Diseño |
| Figura 3 | Se modificaron etiquetas y dependencias entre los modelos Mapa de Navegación y Modelo de Análisis |

**Revista Venezolana de Computación**
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 3, No. 1, pp. 23-37, Junio 2016
Selección de los Mejores Artículos del SCTC 2016

# Block-based Migration from HTML4 Standard to HTML5 Standard in the Context of Web Archives

Andrés Sanoja[1], Stéphane Gançarski[2]
andres.sanoja@ciens.ucv.ve, stephane.gancarski@lip6.fr

[1] Escuela de Computación, Universidad Central de Venezuela, Caracas, Venezuela
[2] Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie, Paris, France

**Abstract:** Web archives are not exempt of format obsolescence. In the near future Web pages written in HTML4 format, could be obsolete. We will have to choose between two preservation strategies: emulation or migration. The first option is the most evident, however due to the size of the Web and the amount of information that Web archives handle it is not practical. In the other hand migration to HTML5 format seems plausible. This is a challenge because we need to modify a page (in HTML4 format) and include elements that not even exists in this format (as the HTML5 semantic elements). Using the Web page segmentation we show that, with the appropriate granularity, blocks look alike these semantic elements. We present the use our segmentation tool, BoM (Block-o-Matic), for helping achieve the migration of Web pages from HTML4 format to HTML5 format in the context of Web archives. We also present an evaluation framework for Web page segmentation, that helps to produce metrics needed to compare the original and migrated version. If both versions are similar the migration has been successful. We show the experiments and results obtained on a sample of 40 pages. We made the manual segmentations for each page using our MoB tool. Results shows that in the migration process there is no data loss but in the migrated version (after adding the semantic elements) the margin is changed. This is, it adds whitespace that change the elements position, shifting elements slightly on the page. While this is imperceptible to the human eye, for systems it is difficult to handle without previous knowledge of this situation.

**Keywords:** Migration; Web; Segmentation; Blocks; HTML5; Web Archive; Format Obsolescence.

## I. Introduction

Obsolescence, adjustment, and renewal are necessary parts of the development cycle. Improvements usually require changes. That includes technologies, products, processes, and people, as well. In July 2012, the WWW Consortium introduced a recommendation for HTML5[1]. It represents an important change regarding the preceding version of HTML and the XHTML specification. For instance it introduces the semantic tags allowing browsers to easily access contents, audio and video among others. The first question raised by HTML5 is: why to use it? Laws [1] discusses this from the competition point of view and he concludes that organizations and publishers need to be ready for this technological change if they want to outperform their competitors and stay in the technological race. This raises another question: once publishers switch to HTML5, what happens with the current HTML4 content? The W3C and the WHAT group are figuring out how Web browsers can be compatible with older versions of the specification. They say that is necessary to evolve HTML incrementally into XML[2]. The strategy is to process this pages differently. So far, Web browsers have been very permissive with malformed documents. In general, Web archives store pages along with all their dependencies. We agree with Rosenthal [2] that eventually, modern browsers will no longer be able to render document in HTML4 or XHTML formats in a proper way (i.e they will not be very permissive). Thus, a strategy for their preservation must be taken. Archivists must decide to perform either a emulation or migration.

In the context of digital preservation the emulation is "the replicating of functionality of an obsolete system, but on the hard- and software environment in which the object is rendered" [3]. In other words emulation consists in recreating the environment in which a Web page was originally created. This implies keeping old versions of tools or old tools. Migration refers to transferring data to newer system environments [4]. This includes converting a Web page file from one file format

[1] The proposed recommendation is out September 2014

[2] http://diveintohtml5.info

to that another so the resource including its functionalities remains fully accessible.

Rosenthal also describes the difficulties of using only emulation. Its cost is very high in terms of storage and operation. Conversely, migration of Web content from an obsolete format to a current one seems to be a good strategy to minimize emulation, but it increases data duplication and there is the risk of loosing document information in the process. The obsolescence of Web content is usually associated with its presentation, that is, its rendering and visual aesthetic. However, the document semantic should be also taken into account also. The main goal of HTML5 is to improve the language, keeping it readable by humans and by computers and useful, and able to enrich the semantic content of documents.

In this article we present how we use Web page segmentation to perform the migration of HTML4 pages to HTML5 format. We think that a block-based solution is more effective than a tag-by-tag approach, since we must differentiate between "regular" tags and "semantic" tags.

Semantic tags (in theory) have no impact in the rendering of the page, but they help to organize the content into coherent regions. Thus, using segmentation seems relevant for the migration, which can be performed by segmenting HTML4 pages and incorporating semantic tags to the result.

To measure the correctness of a migration we perform an Web page segmentation evaluation with a set of predefined manual segmentations and the corresponding migrated versions.

To this end, we present Block-o-Matic (BoM), our segmentation approach, and the model for evaluating segmentation algorithms. We apply both in this work to measure the correctness of the migration. The manual segmentation is made using the Manual-design-Of-Blocks tool (MoB) and a computed one, made with BoM. In this process we give a score based on the geometry of both segmentations. In addition to this the labels of each block are also compared.

The document is organized as follow. In Section IV we present the Web page segmentation concepts and notation. In Section V is presented BoM our approach to Web page segmentation. In Section VI we present our evaluation framework. In Section VII our solution, while in Section VIII the experiments and in Section IX the results. We conclude in Section X with the perspectives and outlook.

## II. Related Work

Several efforts have taken place in order to make uniform the migration from one format to another [5]. Existing methods usually perform a tag-by-tag migration, in other words they translate tags. However, it is difficult to define an appropriated translation of HTML5 semantic tags (which defines the layout of the Web page) from HTML4 pages where such tags do not exist.

There is a lot of online references to perform the tag-by-tag migration[3] however, as far as our knowledge goes, there are very few systematic and automatic approaches to solve the problem described above.

As an example, Park [6], present their experience in the migration of ETD (Electronic Theses and Dissertations) from the PDF format to HTML5 format. Most of ETD have linked multimedia documents and connected by hyperlinks (in PDF format). Storing them in this format, requires to have the corresponding multimedia readers, libraries and plug-in, as well. HTML5 is a convenient migration format because in this way it is possible to have one single file that has all of the content linked together, including all of the multimedia information in the ETD and metadata available for Web search indexing and other general tasks.

Roshental [2] present and describe the design and implementation of a transparent, on-access format migration capability for the LOCKSS system for preserving Web content. Their implementation is capable of transparently presenting content collected in one Web format to readers in another Web format, with no changes needed to browsers. They present an user case of this type of migration on GIF image format migrated to PNG format. They identify the practical difficulties that face any implementation of emulation; they led them to choose the migration strategy

Conversely, Jackson [7] describe a method to identify how HTML and PDF formats changes in Web archives though time. They conclude that software obsolescence is rare on the Web and uncover evidence indicating that network effects act to stabilise formats against obsolescence.

However, we think that obsolescence can occurs in Web environments. We observe this behaviour with old *plugins* (*e.g.* Macromedia Shockwave content) in old Web pages. We agree with Rosenthal that any format is susceptible of been obsolete, and the HTML4, and earlier formats, are not the exception.

In the following sections we present our approach to Web page segmentation and its evaluation as a preliminary to describe our migration approach.

## III. Overview of the Migration Process

In this section we present an overview of the migration process. The idea is to take a Web page in HTML4 format and produce a version of the same page according to the HTML5 format. The main goal of the process described in this paper is to measure a what extent this process is correct, and how reliable it is.

The process is illustrated in Figure 1 and can be divided in five steps, describe as follows:

1) **Segmentation of the input page**: a Web page in HTML4 format is segmented using the BoM segmenter (*c.f.* Section V).

[3]Googling the term 'translating html 4 tag to html5' will give these references

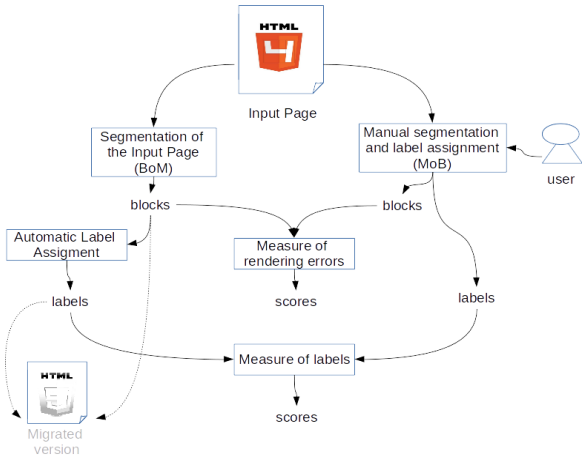


**Figure 1:** Migration Overview

2) **Automatic label assignment**: Based on the properties and characteristics of the blocks found in the segmentation we assign a label (*i.e.* HTML5 semantic elements tags) to each block (*c.f.* Section VIII-D).
3) **Manual segmentation and label assignment**: Using the MoB tool (*c.f.* Section VIII-B) we produce a *ideal* segmentation of the input page. In the same process the user assign a label to each block.
4) **Measure of labels**: from both segmentations (*i.e.* the manual and automatic one) we apply some measures to determine how different both assignments are. The metrics are described in detail in Section VIII-E.
5) **Measure of rendering errors**: Using the Web page segmentation evaluation framework (*c.f.* Section VI) we measure the difference on the rendering both of the automatic and manual segmentation

From the automatic segmentation outcome it is possible to produce the HTML representation, that is, the migrated Web page. This detail is not included in this paper, but technically is the transformation of a XML document into a HTML Web page.

## IV. Web Page Segmentation

Web page segmentation refers to the process of dividing a Web page into visually and semantically coherent segments called blocks. For determining the coherence of each segment we relies on the content categories classifications made by the W3C for the HTML 5 specification (e.g. sectioning content). Detecting these different blocks is a crucial step for many applications, such as mobile devices [8], information retrieval [9], Web archiving [10], Web accessibility [11], evaluating visual quality (aesthetics) [12], among others. In the context of Web archiving, segmentation can be used to extract interesting parts to be stored. By giving relative weights to blocks according to their importance, it also allows for detecting important changes (changes in important blocks) between pages versions [13]. This is useful for crawling optimization, as it permits tuning crawlers so that they will revisit pages with important changes more often [10]. It also helps for controlling preservation actions, by comparing the page version before and after the action.

It is crucial for Web page segmentation to know which elements of the page are considered. For a Web page we extract visual and structural aspects found in the rendered DOM ($W$) of a Web page. From its structure we extract the elements in the form of a hierarchy (DOM tree), the root element ($W.root$). We obtain the text of the page ($W.text$) by recursively concatenating the the text of all elements. Each element corresponds to a HTML 5 content category. From its visual we get the *visual cues* (lines, blank areas, colors, pictures, fonts, etc) and the boxes of each element (rectangles). We have a special box called viewport representing the *body* element.

### A. Concepts

Inspired by the concepts presented by Tang [14] and Nie [15], we describe the Web page segmentation with the following abstractions:

- *Page* is a special block that represents the whole Web page and covers the whole Viewport.
- *Simple block* is an element or a group of elements. It is also denoted simply as *Block*. It is represented as a rectangular area resulting of merging the boxes of elements. Each block has a label related with those of the underlying elements. It is also associated with the text of those elements.
- *Composite block* is a special block that can contain other blocks. Usually such blocks correspond to template elements.
- *Block graph* is a connected planar graph representing the blocks and their relationships (*e.g.* parent/child). It can be an edge-weighted graph (each edge has a weight), or a vertex-weighted graph (each vertex has been assigned a weight). A weight associated with a vertex usually represents how coherent a blocks is, while a weight associated with an edge usually represents the cost of merging two blocks, distance or similarity between blocks.
- *Geometric model* represents the set of blocks as a set of rectangles in a plane. They are obtained from the scheme of the Web page. All rectangles are modelled as quadruples ($x$,$y$,$w$,$h$), where $x$ and $y$ are the coordinates of the origin point and $w$ and $h$ are the width and height of the rectangle. Blocks can be represented in the plane as a hierarchy or a set of non-overlapping rectangles, called Manhattan layout [14]. It can be hierarchical [9] or non-hierarchical [16], [17]. The latter can be obtained from the former by only considering the leaves.
- *Stop condition* is a predefined value (real number) used by algorithms that indicates when a segmentation is achieved. It its based on the edge/vertex weights of the block graph.

An algorithm may have one or more stop conditions.
- *Label* is the role that a block plays in the Web page such as navigation, content, header, footer, etc.

### B. Notation

We present in this section several definitions, in order to have an uniform presentation of Web page segmentations algorithms.

*1) The Segmentation Function:* The segmentation function $\Phi$ is described as follows:

$$\Phi_A\,(W,\,SC) \longrightarrow (W'_A, GM_A) \tag{1}$$

where $A$ is a Web page segmentation algorithm, $W$ is the rendered DOM of a Web page, *SC* is a set of stop conditions. $W'_A$ is the block graph defined just below and $GM_A$ is a set of rectangles representing the geometric model of the segmentation.

*2) The Block Graph:* The block graph is defined as a planar graph $W'_A = (Blocks, Edges)$. Each vertex $B$ in *Blocks* corresponds to a rectangle in $GM_A$ (denoted $B.rect$) and a label (denoted $B.label$). It is associated with a function *weight* on the edges and vertices, and two subset of vertices: $SimpleBlocks \subset Blocks$ (also called terminals), $CompositeBlocks \subset Blocks$, which includes a special vertex $Page$, labeled as the root of the graph.

The rectangle of the vertex *Page* covers the whole viewport of the Web page $W$ and all the blocks fit in. Thus,

$$\forall\, B \in Blocks,\, B.rect \subseteq Page.rect$$

The weight of a vertex $B$ is noted as $B.weight$. The weight of an edge $E$ is noted as $E.weight$

Usually the block graph is a tree. However, some algorithms such as Homory-HuPS [18] and GraphBased [16] define it as a general planar graph.

## V. Block-o-Matic (BoM): a New Web Page Segmenter

In this section we present BoM, our Web page segmentation approach. One of the main features of BoM is that we segment a Web page without having previous knowledge of its content and using only the heuristic rules defined by the W3C Web standards. For instance, we detect blocks using HTML5 content categories instead of using the tag names or text features. That gives genericity to BoM and allow it (in theory) segmenting all types of Web pages.

Another feature of our approach is the introduction of methods and techniques of document processing systems. We leverage existing techniques from the field of computer vision for segmenting scanned documents, in order to adapt them to Web pages. This produces more interesting results for the applications that depends on the segmentation, such as blocks labels.

Let $W$ be the rendered DOM of a Web page. A segmentation $\Phi_{BoM}$ of $W$ is defined as follows :

$$\Phi_{BoM}(W,\, pA,\, pD,\, pND) = (W'_{BoM}, GM_{BoM})$$

where $W'_{BoM}$ is the block graph (a tree) of the segmentation, $GM_{BoM}$ is the geometric model and *pA*, the stop condition. In BoM, the stop condition is the normalized area parameter which is the proportional size of a block respect to the page. We include other parameters used in the algorithm: *pD* is the Distance parameter used for merging blocks. *pND* which is used to compute the normalized area and the weights of blocks. The $pA$ and $pD$ parameters are described on detail in section V-C and V-D. The *pND* is described at the end of this section for computing the weight of a block.

Each block $B$ is associated with its rectangle ($B.rect$), its label ($B.label$), its weight ($B.weight$) as defined in Section IV-A, and a set of DOM elements ($B.elements$).

Consider $W'_{BoM}$ as a rooted, planar and vertex-weighted tree. The root vertex is the *Page* block, inner vertices are the composite blocks, terminal vertices are the simple blocks.

The edges between blocks represent a hierarchical relationship of geometric containment. In other words, consider $Page$, $B_c$ and $B_p \in Blocks$, the following constraints apply:

1) For every pair of blocks ($B_c$, $B_p$), where $B_p$ is the *parent* of $B_c$ in the $W'_{BoM}$ tree, we write $B_c$ *child* of $B_p$ and $B_p$ *parent* of $B_c$.
2) For every block $B_c$, child of $B_p$, $B_c.rect$ is contained in $B_p.rect$

$$\forall\, B_c, B_p,\, B_c \text{ child of } B_p \Rightarrow\, B_c.rect \subset B_p.rect$$

3) The *Page* rectangle cover the whole page and all blocks fit inside it.

$$\forall\, b \in Blocks,\, b.rect \subseteq Page.rect$$

Only simple blocks are associated to DOM elements, thus for the page and composite blocks the $B.elements$ is an empty set.

The weight of a block is the normalized area of its rectangle. It is used to check the stop condition (*cf.* section 2). Thus, the weight of a block $B$ is:

$$B.weight = 0.1 \times \frac{B.rect.w \times B.rect.h \times pND}{Page.rect.w \times Page.rect.h}$$

where *pND* is the predefined constant. In this work we fix this value to $pND$=100, so that both $B.weight$ and $pA$ belongs to the interval [0,10].

### A. Model

In this section we present the Web page segmentation model. It is an hybrid approach, and it follows the bottom-up strategy [19].

First, we describe the segmentation as a black box indicating its input and output. A more detailed explanation follows, describing the three sub-processes that achieve the final segmentation.

We define the Web page segmentation as the process of finding coherent regions of content (blocks) into the rendered DOM ($W$) of a Web page. As a result, the block graph $W'_{BoM}$ and the geometric model $GM_{BoM}$ are produced. The block graph is a tree structure as defined in section IV-A.

Figure 2 shows how a rendered Web page $W$ is segmented. The output is the block graph $W'_{BoM}$ shown on the right side of the figure and the geometric model in the center of the figure.

The sub-processes of the segmentation are:

1) **Fine-grained segmentation construction**. Builds the fine-grained segmentation of $W$ producing $W'_{BoM}$ and $GM_{BoM}$.
2) **Composite block**. Detects the composite blocks. This sub-process updates $W'_{BoM}$ and $GM_{BoM}$
3) **Merging blocks**. Merges blocks according to their area, distance, alignment, labels and content categories. This sub-process produces the final version of $W'_{BoM}$ and $GM_{BoM}$.

### B. Fine-grained Segmentation Construction

The idea of the fine-grained segmentation is to find coherent blocks as small as possible. It serves as a starting point for the whole process by creating a first version of the block graph $W'_{BoM}$ and the geometric model $GM_{BoM}$. The condition $C$ that a DOM element must satisfy to be considered as a block is that it does not belongs to the following content categories: text, phrasing, embedded, interactive or form-associated elements. The value to the label ($B.label$) is the most inclusive content category of its elements ($B.elements$). For instance, if the block has one element which content category is *flow* the label of the block is the same. If the block is associated with two elements, one element in the *embedded* category and the other in the *heading* category, the most inclusive category is *flow*. Figure 4 shows which content category includes other content categories.

The process begins from the leaves of the DOM tree, towards the $W.root$. If an element is found that meets the condition $C$ above defined, the process stops for this branch. Figure 3 shows how an element is selected as a block. Element *li* is the first element that does not belong to the categories above listed, then it is marked as a block and the label *flow* is assigned. From the information obtained during this sub-process a geometric model (cf. section IV-A) and a first version of the block graph are built (cf. section IV-A).

Algorithm 1 shows the steps to build the fine-grained segmentation. First, the rendered DOM tree $W$ is traversed and leaves elements are selected (line 5). If a selected element does not match the condition $C$ its parent become the current element (line 7-8).

The same process continues until either the $W.root$ element (*i.e.:* the *body* element) is reached or the current element meet the condition $C$. If the condition $C$ is met a new block is created (lines 10-11). The element becomes the block's element (line 12), the block label is the element category (line 13), a new rectangle is created (line 14), the geometric model is updated (line 15) and the weight is computed (line 17). The rectangle is based on the box of the element and it is associated to the block (line 16). The block graph is updated with the new block $b$, adding an edge between the Page block and block $b$ (lines 18-19)

```
Data: Rendered DOM : W
Result: block graph W'_BoM, geometric model GM_BoM
Blocks = {Page};
E = {};
W'_BoM = (Blocks, E);
GM_BoM = {};
Terminal ← getTerminalElements(W);
foreach element ∈ Terminal do
    while element ≠ W.root and ¬C(element) do
        element ← element.parentElement;
    end
    if element ≠ W.root then
        create block b;
        b.elements ← element;
        b.label = element.category;
        rect = createRectangleFromElement(element);
        add rectangle rect to GM_BoM;
        b.rect = rect;
        b.weight = normalized_area(b);
        add vertex b to W'_BoM;
        add edge (Page, b) to E;
    end
end
```

**Algorithm 1:** Fine-grained Segmentation Construction

The fine-grained segmentation form a flat segmentation, that is $height(Page) = 1$.

### C. Composite Block

Composite blocks usually are Web page regions that lie along separation lines. A separation line is the space that goes from one limit of the page to another without crossing any block. A horizontal separation line *S* in a block is represented by the line formed by the points $(x_1, y_1)$ and $(x_2, y_2)$, where $y_1 = y_2$ if it is horizontal, $x_1 = x_2$ if it is vertical. The spaces found either at the beginning or at the end of the document are omitted.

Algorithm 2 shows the *CompositeBlockDetection* function in order to find the composite blocks and the flow of a segmentation. It accepts a composite block as input and outputs the $W'_{BoM}$ graph and the geometric model *GM* updated with new blocks (if any) and including the computed order.

We start finding the composite blocks in the Page block itself, considered as a composite. Two composite blocks are

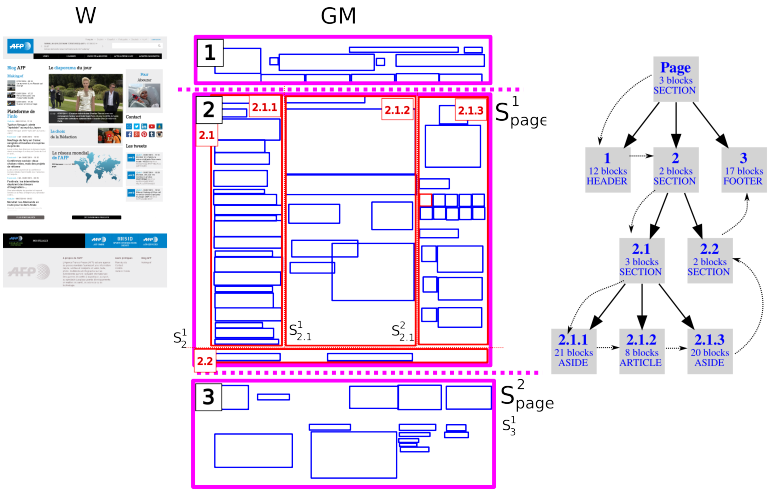


**Figure 2:** Segmentation Model Example


**Figure 3:** Block Detection Based on Content Categories

formed on both sides of the separation line (line 12). All simple blocks that are covered by these new blocks are aggregated accordingly and become their children blocks (line 22). The process stops if it is met one of two conditions: their weights are below the predefined stop condition parameter (*pA*) or the horizontal or vertical limits of the block are not those of the Page (line 1), *i.e.* if $B.rect.x > Page.rect.x$ and $B.rect.w < Page.rect.w$ (respectively $B.rect.y > Page.rect.y$ and $B.rect.h < Page.rect.h$).

Figure 2 shows the separation lines, $S^1_{page}$ and $S^2_{page}$, found in the Page block, generating blocks 1, 2 and 3. On the same figure, block 1 and 3 are not processed because their weights are higher than *pA*, but the same process is applied to block 2. First the horizontal separator $S^1_2$ is discovered, generating the composite blocks 2.1 and 2.2. We assume that the weight of block 2.2 is below the predefined stop condition parameter, thus no further processing is needed. However, in block 2.1, two vertical separators $S^1_{2.1}$ and $S^2_{2.1}$ are found.

**Data**: block $b$
**Result**: $W'_{BoM}$ and $GM_{BoM}$ updated
**if** *$b$ limits equals to $Page$ and $b.weight > pA$* **then**
  Separators ← findSeparatorsIn(b);
  **foreach** *$s \in Separators$* **do**
    **if** *$s$ is horizontal* **then**
      $rect_1 = \{b.rect.x, b.rect.y, b.rect.w, s.y_1\}$;
      $rect_2 = \{b.rect.x, s.y_1, b.rect.w, b.rect.h\}$;
    **else**
      $rect_1 = \{b.rect.x, b.rect.y, s.x_1, b.rect.h\}$;
      $rect_2 = \{s.x_1, b.rect.y, b.rect.w, b.rect.h\}$;
    **end**
    add rectangles $rect_1$, $rect_2$ to $GM_{BoM}$;
    create blocks $b_1$, $b_2$;
    $b_1.rect = rect_1$;
    $b_2.rect = rect_2$;
    add vertices $b_1$, $b_2$ to $W'_{BoM}$;
    add edge $(b, b_1)$ to $E$;
    $CompositeBlockDetection(b_1)$;
    add edge $(b, b_2)$ to $E$;
    $CompositeBlockDetection(b_2)$;
  **end**
**else**
  update $W'_{BoM}$ and *GM* to associate blocks covered by $b$
**end**

**Algorithm 2:** Composite Blocks Detection

### *D. Merging Blocks*

Once composite blocks are created, the merging process starts. This process allows obtaining simple blocks the weight of which is greater than the predefined stop condition parameter (*pA*). Two blocks are merged if the following heuristic rules are all satisfied:

1) Their weights are less than the the predefined stop condition parameter.
2) The distance between them is below a predefined distance parameter $pD$.
3) Both blocks are horizontal or vertical aligned with a tolerance than no more that $pD$ pixels.
4) They are not aligned but one's rectangle covers completely the other's one.
5) Their label is not *sectioning*.

The rules are checked in the given order for efficiency purpose:

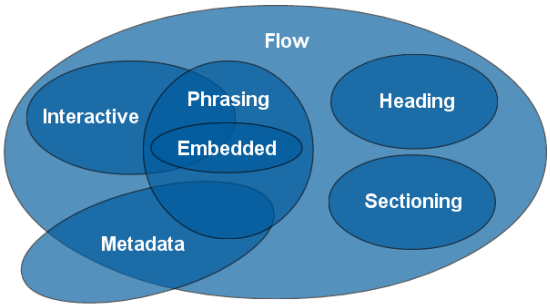


**Figure 4:** HTML5 Content Models. Source: http://www.w3.org

the first rules are most discriminant.

This process is repeated until no more merges are possible. Then we check if the proportion of blocks with a weight less than $pA$ is greater than a constant (for instance 75%). If it is the case, all the children of the composite blocks are removed. If the composite block has only one child, this latter is removed.

To illustrate the merging process, let $pA = 4$, $pD = 50$ and $pND = 100$. Figure 5 shows the merging process for the block 2.1.2 of an example page. Each blocks has its weight and its label. In Figure 5a blocks *a*, *b* and *c* are merged because they are aligned and the distance between them is less than $pD$. The label *flow* is assigned. The same applies for blocks *e* and *h*. However blocks *d* and *f* are too far. Blocks *f* and *g* are not aligned. Figure 5b shows the result of merging those blocks and in a second round the blocks *d* and *e* are merged because their distance is below the parameter $pD$ and they are aligned using the tolerance. Figure 5c show the merged blocks. Block *f* is contained into block *d*, so they are merged and the label *flow* is assigned. Figure 5c shows the final merging, the process stops because the weight of both blocks *a* and *d* is greater than the predefined stop condition $pA = 4$.

Algorithm 3 presents details about the algorithm for merging blocks. We only consider the composite blocks that have simple blocks as children and the weight of which is greater than the predefined stop condition parameter (*pA*). If it is the case we try to merge the children.

```
Data: composite block b
Result: W'_BoM, GM_BoM updated
if b.weight > pA then
    Children ← getChildren(b);
    foreach child ∈ Children do
        if child.weight < pA then
            Siblings ← getSiblings(child);
            foreach sibling ∈ Siblings do
                if child and sibling are aligned then
                    if distance between child and sibling less
                    than pD then
                        if labels of child and sibling are not
                        sectioning then
                            merge sibling with child as child;
                            label child from both labels;
                        end
                    end
                else
                    if child covers sibling then
                        merge sibling with child as child;
                    end
                end
            end
        end
    end
    if |getChildren(b)| = 1 then
        remove child of b;
    end
    if proportion of non merged small children is superior to
    75% then
        remove children of b;
    end
else
    remove children of b;
end
```

**Algorithm 3:** Merging

## VI. Segmentation Evaluation Model

In this section we present our approach to Web page segmentation. We aim segmenting a Web page without previous knowledge about its content. This allows segmenting different type of Web pages. The heuristic rules are based solely on rules defined in the Web standards, such as content categories.

We do not do any assumption about the text. However, this can be a weakness because in some cases analyzing the text can be relevant. For instance, two consecutive blocks that talk about different subjects should not be merged. Solving this issue would imply studying the semantics of the block content and is out of the scope of this work.

There are three different implementations of the BoM algorithm. One version is developed as a Ruby application, the second as a Java application and the third as a JavaScript library. The Ruby version is intended as functional prototype, the Java version to production environments for the European project SCAPE[4] and the JavaScript version for the open source community[5].

Introducing concept and techniques from the computer vision field of scanned document image segmentation allow having a more complete segmentation, as it contains more useful information for applications than most of the other segmenters.

Evaluating web page segmentation algorithms is not an easy task. Usually, each algorithm proposes its own *adhoc* validation mechanism that can not be really applied to other approaches.

[4]http://www.openplanetsfoundation.org/blogs/2014-02-12-scape-qa-tool-technologies-behind-pagelyzer-ii-web-page-segmentation

[5]https://github.com/openplanets/pagelyzer/tree/master/SettingsFiles/js

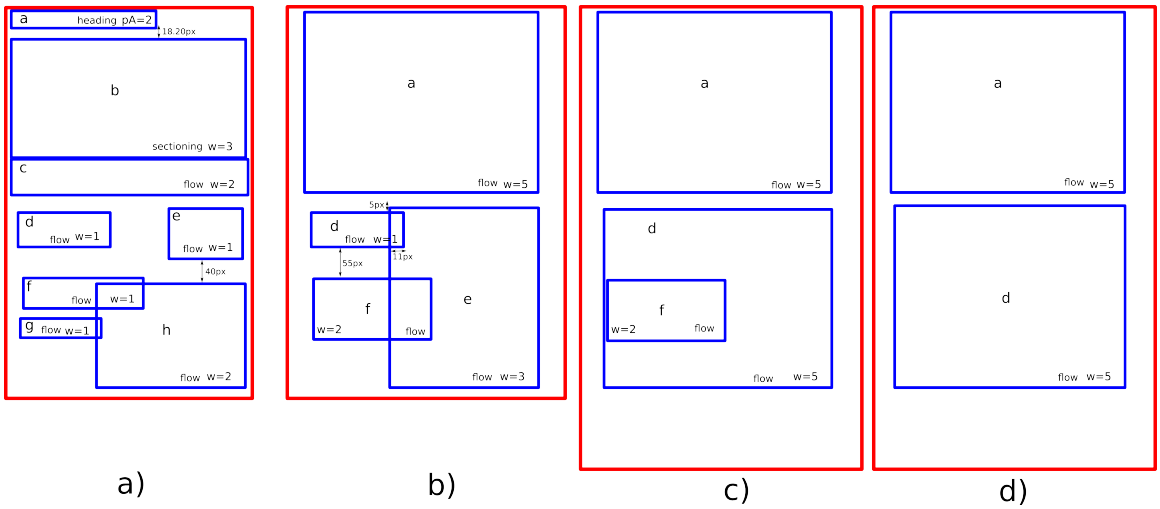


**Figure 5:** Merging Blocks and Labeling

This section attempts to close this gap by proposing a number of evaluation metrics that essentially measure how well the generated segmentation maps to a ground truth segmentation. This can be formulated as a graph matching problem, and we propose a number of metrics based on the generated matching to assess the quality of the generated blocks.

In this section, we present our evaluation model in order to measure the quality of a segmentation according to a discrepancy parameter (*i.e.:* determine how far the two segmentation are one from the others). The goal of the evaluation model is to compare an automated segmentation of a web page $W$ with the corresponding ground truth, in order to determine its quality. Both segmentations are organized as non-hierarchical Manhattan layout, in other words, they are flat segmentations. Our evaluation model is an adaptation to web pages of the model presented by [20] for scanned page segmentation evaluation (see Section VI-A). The quality of a segmentation is evaluated by using the block correspondence. The block correspondence measures allows knowing to what extent the generated blocks match those of the ground truth.

We present the evaluation model adaptation (VI-A), the representation of a segmentation (VI-B) and the representation of the evaluation (VI-D).

### A. Model Adaptation

In order to adapt to web pages the model presented by Shafait et al. [20] for scanned page segmentation evaluation we need to identify the different aspects of both type of documents. Shafait represent a segmentation of scanned documents images using a pixel-based representation. Each foreground pixel belongs to a zone or region. The evaluated documents (and the ground truth) must have the same dimension.

Their evaluation model defines several performance metrics to evaluate different aspects of the behaviour of a scanned page segmentation in image form. These metrics allow measuring the correspondence of each pair of rectangles the segmentation and the ground truth. A region (or block) is significant if it the amount of foreground pixels associated with it is greater than a parameter.

By analogy, web pages consist of elements and text. In our adaptation, a block is significant if the amount of elements and text is greater than a parameter. Other features of our model are intrinsic to web pages, such as the block importance.

### B. Representation of Segmentation

In this section we model a segmentation in order to describe its evaluation. We describe the absolute and normalized representation of a segmentation (VI-B1 and VI-B2), as well as the importance of blocks and how it is computed (VI-C).

We present the concepts used along the section. We use the notation described in Section IV-B. We use the concepts of page, block and block graph based on the concepts described in the same section.

*1) Absolute Representation of a Segmentation:* Each block $B$ is associated with its rectangle ($B.rect$), its label ($B.label$) and its weight ($B.weight$). To each $B$ we add three values: the amount of elements it covers ($B.ec$), the text associated to the block ($B.text$) in the original page $W$ and the importance ($B.importance$). Note that $B.ec = |B.elements|$.

The importance of a block depends on the area covered by its rectangle. Section VI-C explain how it is computed.

An absolute segmentation for the rendered DOM *W*, using the algorithm *A* and *SC* a set of stop conditions, is defined by the following function $\Phi$:

$$\Phi_A(W, SC) \longrightarrow (W'_A, GM_A)$$

where $W'_A$ is the block graph and $GM_A$ is a set of rectangles representing the geometric model of the segmentation.

Consider $W'_A$ as a rooted, planar and vertex-weighted tree. The root vertex is the *Page* block and the terminal vertices are the simple blocks. We consider the segmentation as flat, that is the $height(Page) \leq 1$. $GM_A$ is the geometric model of the segmentation consisting of a set of rectangles.

*2) Normalized Segmentation Representation:* In order to compare two segmentations, we need to normalize the rectangles.

Given an absolute segmentation $\Phi_A$, the geometric model of its normalized version $N\Phi_A$ fits in a *ND* $\times$ *ND* square, where *ND* is a fixed value, called Normalized Document Size. In our experimentation, we fixed this value to 100. Thus if $N\Phi_A$ is the normalized segmentation of $\Phi_A$:

$$N\Phi_A(W, SC) \longrightarrow (NW'_A, NGM_A) \tag{2}$$

where $NW'_A$ is the block graph of the normalized segmentation, $NGM_A$ is the normalized geometric model. All the segmentation rectangles are normalized. Thus, the *Page* block rectangle is normalized as:

$$NW'_A.Page.rect = \{0, 0, ND, ND\}$$

Each block rectangle is then normalized according to the stretch ratio of the page, *i.e.*

$$\forall\, b \in NW'_A,\ b.rect.x = \frac{ND \times W'_A.Page.rect.x}{W'_A.Page.rect.w}$$

The other values of the block rectangle ($y$, $w$ and $h$) are normalized in the same way.

### C. Block Importance

The regions in a web page are not all equally important. A block is more important than another block if it contains more important information. Usually, important blocks are located in the most visible part of the page. A good segmentation algorithm must mostly find important blocks.

The block importance is obtained from the geometric model of the segmentation, that is the spatial features. A segmentation is mapped to a grid of *NP* $\times$ *NP*, where *NP* is the Normalized Partition Size. This grid be represented as a matrix *IM(NP,NP)*. Each cell of the matrix ($im_{ij}$) is assigned with a value representing the importance that a block has if it lies within this area. For instance, with the *window spatial features* defined by Song et al. [21], a highest importance is assigned to blocks found in the middle of the visible part of a web page, and a lower importance to blocks found outside of this area.

The computed importance of a block is the sum of the cell values obtained by mapping the block rectangle over the grid. The rectangle coordinates are divided by the constant *NP*. This defines two intervals, one for each dimension. If *i* and *j* respectively belong to those intervals, then the cell value $im_{ij}$ is taken into account. Thus the computed importance of a block $B \in W'_A.Blocks$ is:

$$computed\_importance(B) = \sum_{ij} im_{i,j} \tag{3}$$

where

- $i \in \left[round(\frac{B.rect.x}{NP}), round(\frac{B.rect.w}{NP})\right]$ and,
- $j \in \left[round(\frac{B.rect.y}{NP}), round(\frac{B.rect.h}{NP})\right]$

In order to uniformize the importance we define $B.importance$ as the average importance of a blocks in a segmentation. The computed importance of each block is divided by the sum of all the computed blocks importance in a segmentation. Thus the importance of a block $B \in W'_A.Blocks$ is:

$$B.importance = \frac{computed\_importance(B)}{\sum_{b \in W'_A.Blocks} computed\_importance(b)} \tag{4}$$

### D. Representation of the Evaluation

In this section we model the evaluation itself, described in terms of input and output. We describe also the metrics used in for measuring the block correspondence (VI-E).

The evaluation is described as a function that takes two segmentations and four constants as parameters. The two segmentations $\Phi_G$ and $\Phi_P$ are absolutes segmentations as described in section VI-B producing the block graphs $W'_G$ and $W'_P$. The four parameters are the relative tolerance ($t_r$), the importance tolerance ($t_i$), the Normalized Document size (*ND*) and the Normalized Partition size (*NP*) as defined in section VI-B1 and VI-B2. These parameters are described in detail in the following sections. The evaluation function returns a vector of metrics representing the quality of $\Phi_P$ with respect to $\Phi_G$. Equation 5 shows the function.

The quality of a segmentation is measured by block correspondence. It measures how well the blocks of $W'_P$ match with the ones of $W'_G$.

The block correspondence takes into account the location and geometry of block. It allows for detecting which blocks were correctly discovered and which ones raised issues.

### E. Measuring Block Correspondence

The block correspondence indicates whether the blocks rectangles of a segmentation match those of the ground truth.

Consider two normalized segmentations for a page $W$: a computed one $N\Phi_P$ and the ground truth $N\Phi_G$. The associated normalized block graphs are $NW'_P$ (denoted $P$ in the rest of the section) and $NW'_G$ (denoted $G$). Figures 6(a) and (b) give respectively an example for $G$ and $P$.

To compute the block correspondence, we build a weighted bipartite graph called *block correspondence graph* ($BCG$). We start with an example and then give the algorithm.

$$evaluate(\Phi_G,\, \Phi_P,\, t_r,\, t_i,\, \text{ND},\, \text{NP}) = (\text{text coverage metric}, \text{correspondence metrics}) \quad (5)$$

As seen on Figure 6(c), nodes of the $BCG$ are the blocks of $P$ and of $G$. An edge is added between each couple of nodes $n_i$ and $n_j$ such that the weight $w(n_i, n_j)$ of the edge is equal to the number of underlying HTML elements and text in the intersection of the regions covered by the rectangle of each of the blocks corresponding to the two nodes. If the blocks rectangles do not overlap in $P$ and $G$, no edge is added.

Algorithm 4 shows how is built the $BCG$. If the blocks

**Data**: nodes $n_i \in G$,$n_j \in P$
**Result**: vertex ($n_i$,$n_j$) and its weight (if apply)
**if** $n_i.rect$ *is contained in* $n_j.rect$ **then**
    create vertex ($n_i$,$n_j$);
    $w(n_i, n_j) = n_i.htmlcover + n_i.textcover$;
**else if** $n_i.rect$ *contains* $n_j.rect$ **then**
    create vertex ($n_j$,$n_i$);
    $w(n_i, n_j) = n_j.htmlcover + n_j.textcover$;
**else**
    /* no vertex is created */
    $w(n_i, n_j) = 0$;
**end**

**Algorithm 4:** Algorithm for Building the BCG Graph

in $P$ fits perfectly with the ground-truth blocks $G$, then the $BCG$ will be a perfect matching. That is, each node in the two component of the graph has exactly one incident edge. If there are differences between the two segmentations, nodes of $P$ or $G$ may have multiples edges. If there is more than one edge incident to a node $n$ in $P$ (resp. in $G$), $n$ is considered oversegmented (resp. undersegmented). Using these definitions, we can introduce several measures for evaluating the correspondence of a web page segmentation algorithm.

Intuitively, if all blocks in $G$ are in $P$, this means that the algorithm has a good quality. If one set of blocks in $G$ are grouped into one block in $P$ or if one block in $G$ is divided in several blocks in $P$ then there is an issue with respect to the granularity but no error. We determine a segmentation error if one block in the ground truth is not found in the computed segmentation or if there are blocks that were "invented" by the algorithm.

The metrics for block correspondence are defined as follows:

1) **Correct segmentation** $C_c(\Phi_A)$, $C_c$ for short. The number of one-to-one matches between $P$ and $G$. A one-to-one match is defined by a couple of nodes $(n_i, n_j)$, $n_i$ in $P$, $n_j$ in $G$, such that $w(n_i, n_j) \geq t_r$, where $t_r$ is a threshold that defines how well a detected block must match to be considered as correct. For instance, in Fig. 6, there is an edge between node 2 and node B and another one between node 2 and node C. However, as the weight $w(2, C)$ is less than $t_r$, and the weight $w(2, B)$ is greater than $t_r$, B is considered as a correct block. The metric value for the example is $C_c = 2$ . $C_c$ is the main metric for measuring the quality of a segmentation.
2) **Oversegmented blocks** $C_o(\Phi_A)$, $C_o$ for short. The number of G nodes having more than one edge. This metric measures how much a segmentation produced too small blocks. However, those small blocks fit inside a block of the ground truth. In the example of Fig. 6, node 6 of the ground truth is oversegmented in the proposed segmentation. In the example, the metric value is $C_o = 2$ because nodes 6 and 2 are both over-segmented.
3) **Undersegmented blocks** $C_u(\Phi_A)$, $C_u$ for short. The number of P nodes having more than one edge. The same as above, but for big blocks, where blocks of the ground truth fit in. For instance, on Fig. 6, node D of the proposed segmentation is undersegmented with respect to the ground truth, and the value for the metric is $C_u = 1$.
4) **Missed blocks** $C_m(\Phi_A)$, $C_m$ for short. The number of G nodes that have no match with any in P. This metric measures how many blocks of the ground truth are not detected by the segmentation. One example is node 3 shown in the Fig. 6 and the value of the metric is $C_m = 1$.
5) **False alarms** $C_f(\Phi_A)$, $C_f$ for short. The number of P nodes that have no match with any in G. This metric measures how many blocks are "invented" by the segmentation. For instance, in Fig. 6 node I has no correspondent in the ground truth making the metric value as $C_f = 1$.

Each metric $C_x$ has a version, noted $IC_x$, that takes the importance of the blocks into account. In other words, $C_x$ can be seen as the metric when all the blocks have the same importance. $C_c$ is a positive measure, $C_m$ and $C_f$ are negative measures. $C_o$ and $C_u$ are "something in the middle", as they count "not too serious" errors : found blocks could match with the ground truth if they were aggregated or split. Note that the defined measures cover all the possible cases when considering the matching between $G$ and $P$.

Thus, the evaluate function returns a vector made of all the computed metrics, *i.e.*

$$evaluate(\Phi_G,\, \Phi_P,\, t_r,\, t_i,\, \text{ND},\, \text{NP}) = (TC,\, C_x,\, IC_x\,) \quad (6)$$

To evaluate the quality of the segmentation we define a score $C_q$, as the total number of acceptable blocks discovered, *i.e.* $C_q = C_c + C_o + C_u$ and $IC_q = IC_c + IC_o + IC_u$. Note that $C_m$ is the complement of $C_q$ where $C_q + Cm = |G|$.

## VII. Proposed Solution for Migration

We propose to segment an HTML4 Web page, with the appropriate predefined stop condition parameter so that the resulting blocks will correspond to the semantic tags in the HTML5 format.

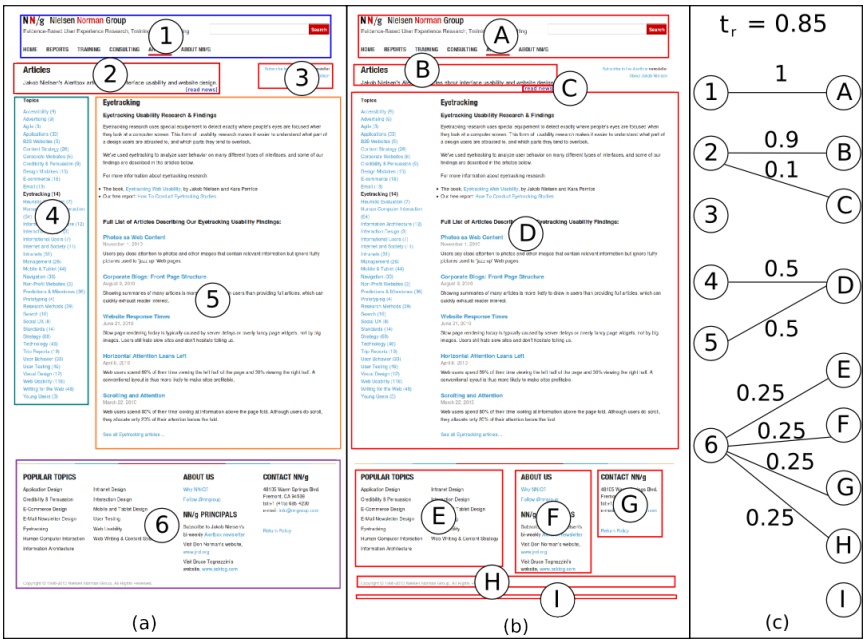


**Figure 6:** (a) Ground-truth Segmentation. (b) Computed Segmentation. (c) BCG

Then we compare the labels found by the segmentation with a manually labeled segmentation as ground truth. If both versions are similar the migration is achieved. If they are different we measure how discrepant they are in order to determine the causes and the possible actions to improve the migration method.

Finally, migration is evaluated in order to measure whether it has affected the rendering of the Web page. We use for this an adaptation of the framework of Section VI-D.

In the following section we describe the experiments to evaluate our migration approach.

## VIII. Experiments

In this section we present the setup of experiments, their design and the measures used.

### A. Experimentation Design

*1) GOSH Collection:* The dataset holds the offline version of Web pages, together with their segmentations obtained by the different algorithms (including the ground truth), organized in categories.

Within a collection, each page is rendered with different rendering engines with different predefined stop conditions values. To each quadruple (page, render engine, algorithm, predefined stop condition) corresponds a segmentation performed on that page, and rendered by that engine, using one algorithm with a predefined stop condition.

Web pages are taken from the GOSH (GOogle SearcH) collection that we built. It is described as follows.

Web pages in this collection are selected with respect to their category. This selection is based in the categorization made by Brian Solis [22], "The Conversation Prism". It depicts the social media landscape from ethnography point of view. In this work, we considered the five most common of these categories, namely Blog, Forum, Picture, Enterprise and Wiki. For each category, a set of 25 sites have been selected using Google search to find the pages with the highest *PageRank*. Within each of those sites, one page is crawled [6]. The GOSH collection contains 125 pages.

*2) The MIG5 Collection:* The MIG5 collection is a subset of the GOSH collection presented in previous section. It only contains Web pages in HTML4 format. We keep the same categories organization (blog, enterprise, forum, picture and wiki) in this collection.

*3) Experiments:* The first experiment is devoted to measure to what extent the labels found with the Block-o-Matic segmentation algorithm match to those in a ground truth of manually labeled blocks.

The second experiment aims of measuring if including the semantic elements affects the rendering of the page. The block correspondence method, as presented in Section VI-E, is used for evaluating the correctness of the migration. The segmentation of the original Web page is used as a ground truth, while the segmentation of the migrated Web page is the evaluated segmentation.

[6] https://github.com/asanoja/web-segmentation-evaluation/tree/master/dataset

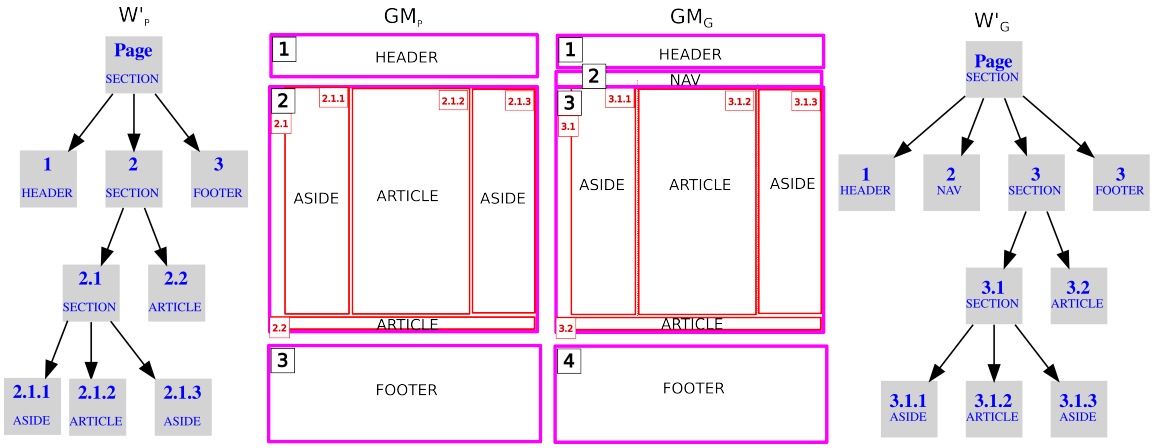


**Figure 7:** Labels for the Manual and Computed Segmentation

**Table I:** MIG5 Pages by Categories

| *Category* | *Pages* |
|---|---|
| blog | 5 |
| enterprise | 9 |
| forum | 14 |
| picture | 7 |
| wiki | 5 |
| total | 40 |

### B. Manual-design-Of-Blocks Tool

In order to build a Web page segmentation grountruth we develop the tool MoB (Manual-design-Of-Blocks). It is conceived as a browser extension and expose functionalities to expert users for creating a manual segmentations [7].

Users can create blocks based on Web page elements. They can merge blocks, navigate into the element hierarchy to produce a block graph [8] (*c.f* Section 2), or produce a flat segmentation (*i.e.* leaves in the block tree). These segmentations are stored in a repository [9] for the evaluation.

### C. Ground Truth Building

Table I shows the organization of the MIG5 collection. It is composed of 40 pages organized by category.

The MoB tool (*cf.* Section VIII-B) is used to annotate the blocks. Besides specifying the blocks, assessors assign a label to each block. Labels corresponds to a subset of the semantic elements defined in the HTML5 specification (header, footer, section, article, nav, aside). The stop condition for all the experiments is set to $pA = 6$. Indeed, through experiments, we noticed that this value generates blocks likely to correspond to template elements. The separation is set to $pD = 30$ because usually these regions can be very close one to each other.

[7] http://www-poleia.lip6.fr/ sanojaa/BOM
[8] Usually a tree
[9] http://www-poleia.lip6.fr/ sanojaa/BOM/inventory

### D. Assigning Labels

The BoM labeling method is modified to support the semantic elements as labels. Heuristics rules are defined in order to determine the label of each block. These rules assign labels depending on the position of a block and its relationship to the others blocks. A block is treated differently if it resides in the visible part of the page (*i.e.* the part of the page visible without using scrolling). For instance, a block is labeled as *header* if it is the first block found vertically (on top of the page), it resides in the visible part of the page, it is a simple block and it has siblings. A block with the same characteristics but outside of the visible area and at the bottom of the page is labeled as *footer*.

For the labels *section* and *nav*, two additional conditions are considered. If the proportion of elements a block covers is greater than a constant, it can be considered a *section*. If the proportion of hyperlinks (*i.e.* <A> elements) a block covers is greater than a constant, it can be considered a *nav*. Algorithm 5 describe the label assignment method for all possible cases.

### E. Measuring Labels

The manual segmentation $\Phi_G$ and the computed segmentation $\Phi_P$ are formal defined in Section 3. The manual segmentation, produced by assessors, takes the rendered DOM of a Web page ($W$) in HTML4 file format and produces the $W'_G$ block graph. The computed segmentation takes the same rendered DOM ($W$) and produces the $W'_P$ block graph.

We present the labels of a segmentation as a list of labels ($labels(W'_A)$).

Using the intersection of both list we get the amount of correct labels found by the segmentation with respect to the ground truth. The *correct_labels* measure is defined as:

$$correct\_labels(W'_G, W'_P) = labels(W'_G) \cap labels(W'_P)$$

Figure 7, shows the labels for the manual and computed segmentation. The list of labels from the manual segmentation

```
Data: Block: b
Result: B.label
if b.weight > pA then
    if b in the visible part of page then
        if b is the first block on top then
            if proportion of elements covered by b
            is greater than a constant then
                if b is composite then
                    B.label=SECTION;
                else if b has no siblings then
                    B.label=SECTION;
                else
                    B.label=HEADER;
                end
            else
                B.label=HEADER;
            end
        else if proportion of elements covered by b is greater
        than a constant then
            if b is composite then
                B.label=SECTION;
            else
                B.label=ARTICLE;
            end
        else if proportion of hyperlinks covered by b is
        greater than a constant then
            B.label=NAV;
        else if b is in the middle/center of the page then
            B.label=ARTICLE;
        else if b is the last block at bottom then
            B.label=FOOTER;
        else if b is at left/right of the page then
            B.label=ASIDE;
        else
            B.label=ARTICLE;
        end
    else if b is the last block at bottom then
        B.label=FOOTER;
    else
        B.label=ARTICLE;
    end
end
```

**Algorithm 5:** Label Assignment Algorithm

is: { header, nav, aside, article, aside, article, footer}. The list of labels for the computed segmentation is: { header, aside, article, aside, article, footer}. For simplicity, we denote the labels with one letter. Thus, the list of labels for both example segmentations are:

- $labels(W'_G) = \{H, N, D, A, D, A, F\}$
- $labels(W'_P) = \{H, D, A, D, A, F\}$

The migration of Figure 7 is not perfect since the segmentation did not find the block labeled as *nav*. Instead, it found the block labeled as *header* covering the corresponding region of the page. We measure this error with the Levenshtein distance [23].

$$error(W'_G, W'_P) = LD(labels(W'_G), labels(W'_P))$$

where $LD$ is the Levenshtein distance. For the example the error is 1: it is sufficient to insert 1 label (*N*) in the computed segmentation label list to produce the list of the ground truth. We represent also the results in terms of precision and recall:

$$precision = \frac{correct\_labels(W'_G, W'_P) + |labels(W'_G)|}{|labels(W'_G)|}$$

$$recall = \frac{correct\_labels(W'_G, W'_P) + |labels(W'_G)|}{correct\_labels(W'_G, W'_P)}$$

### F. Measuring Rendering Errors

In order to measure to what extent the migration affects the rendering of the migrated Web page, we use the correspondence measures defined in Section VI-E. We do not consider the metric version with importance.

We have two rendered DOM, *W* and *W5*, where *W* is the rendered DOM of a Web page in HTML4 format and *W5* is the rendered DOM of the migrated Web page. They respectively produce the blocks graphs $W'_P$ and $W5'_P$. Setting the parameters $t_r = 0$, $t_i = 0$, $ND = 100$ and $NP = 10$ we get the correspondence measures. We choose these parameters because we want to evaluate all blocks, so we consider all as significant and all are equally important.

If we find only correct blocks then the migration may be perfect, if both segmentations produce the same segmentation there is a high probability that their rendering is the same. If an oversegmentation or an undersegmentation occurs that means that the inclusion of semantic elements in $W5$ modified the size and position of the blocks, therefore segmentations are different. Blocks missed and false alarms are possible when the rendering changes, slightly displacing content in the migrated version.

## IX. Results

In this section we present the results of applying our approach to migrated Web pages from HTML4 format to HTML5 format. We present how we measure the labels found by the algorithm compared to the ground truth and the rendering errors using the evaluation model presented in Section VI-D.

*1) Measuring Labels:* Table II shows the average values of the metrics defined in Section VIII-E for the MIG5 collection separated by categories. Column *CL1* represents the correct label measure ($correct\_labels(W'_G, W'_P)$). The *CL2* column represents the amount of labels in a segmentation ($|labels(W'_G)|$). The *CL3* column represents the rendering error ($error(W'_G, W'_P)$). The last two columns represents de precision and recall measures.

**Table II:** Average Values for Correct, Expected Labels and Error for the MIG5 Collection

| *category* | *CL1* | *CL2* | *CL3* | *prec* | *rec* |
|---|---|---|---|---|---|
| blog | 2.50 | 3.50 | 2.00 | 0.28 | 0.4 |
| enterprise | 2.22 | 3.55 | 2.38 | 0.37 | 0.60 |
| forum | 3.00 | 3.53 | 1.44 | 0.14 | 0.17 |
| picture | 2.55 | 3.00 | 1.55 | 0.14 | 0.17 |
| wiki | 2.20 | 3.00 | 1.90 | 0.26 | 0.36 |

**Table III:** Correspondence Metrics for the MIG5 Collection with $t_r = 0.1$ and $t_t = 1$

| Algorithm | $C_c$ | $C_o$ | $C_u$ | $C_m$ | $C_f$ | $C_q$ | $GTB$ |
|---|---|---|---|---|---|---|---|
| blog | 6.50 | 0.50 | 0.00 | 0.00 | 0.50 | 7.00 | 7 |
| enterprise | 4.00 | 0.33 | 0.33 | 1.11 | 2.77 | 4.67 | 6.45 |
| forum | 3.41 | 0.59 | 0.41 | 2.11 | 1.29 | 4.41 | 6.59 |
| picture | 2.71 | 1.00 | 0.29 | 2.00 | 0.71 | 4.00 | 6.71 |
| wiki | 6.00 | 0.0 | 0.00 | 0.60 | 0.40 | 6.00 | 6.6 |

In general BoM produces a list of labels similar to the ground truth. In average it adds 1.85 unexpected labels. This is probably due to the introduction of semantic elements that affect the segmentation and the stop condition, producing smaller blocks than expected. For instance, for a blog post with two paragraphs, labeled as a whole in the ground truth, each paragraph become a block in the migrated page generating one additional unexpected label. It is interesting that both rendering looks equal but the segmentations differs.

Forum category presents the lowest error rate, because in general the question/response region of the page is detected in both segmentation, as one block labeled as *article*. The worst performance is for the enterprise category, because this type of pages are structured with complex navigation and main content, and the probability of mislabeling is high.

Table II shows the precision and recall metrics. Figure 8 shows these metrics graphically. The BoM algorithm has a high precision for the forum and picture categories. As we mention earlier both type of pages produces small and simple list of labels, while pages in the other categories their labeling is more complex, therefore less precision. However, all results present high recall values indicating that the algorithm find enough good labels but with a considerable error rate.

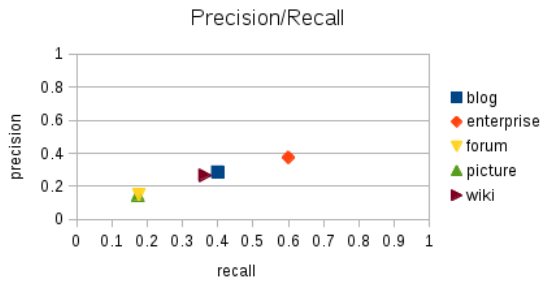


**Figure 8:** Precision and Recall for the MIG Collection

*2) Measuring Rendering Errors:* Table III shows the average correspondence metrics, by category, for the MIG5 collection. The values of the $C_q$ metric shows that the performance of the algorithm in both versions (original and migrated) is good. However, there are some missed blocks, particularly in the enterprise, forum and picture categories because of shifting of blocks due to rendering changes. But in both cases, the formatted content displayed is equal. Blog and wiki categories present the best performance. The regions in these type of pages are simple and the position and order of blocks are standard. The regions are well separated, making it easy to segmentation algorithms like BoM to detect correct labels. For instance, almost all pages in this categories start by a *header* followed by a *navigation*, then the *aside* at left, the main *article* and the *footer* at the bottom of the page.

## X. Perspectives and Outlook

In this section we presented our approach to block-based migration of Web pages from HTML4 format to HTML5 format. Using the segmentation, we produce a migrated version according to the HTML5 specification. We analyzed how the algorithm assigned labels to blocks in comparison to a ground truth of manually labeled segmentation. The rendering errors were measured using the block correspondence metrics defined in Section VI-E. The results show that, in the context of digital preservation, migrating Web pages from one format to another is possible using the BoM Web page segmentation algorithm, minimizing the emulation in Web archives. We show that there is no data loss in the process and no important changes in the rendering (few false alarms). However the segmentation is affected by the semantic tags. For instance, some browsers have no default style for these elements, and they are taken by the algorithms as invisible or not valid elements, therefore they are ignored. The evaluation model presented in Section VI-D is very helpful to measure the performance and detecting the rendering errors. The parameters and the stop conditions of the algorithm can be adjusted by category (using Machine Learning techniques) to have better performance depending on page category. This is left as future work.

This work focus on the migration of the rendered version of a Web page, however as a future work it is interesting to include into the analysis other components of the Web pages such as Javascripts, CSS1 and CSS2. We need to assure that all dependencies of the migrated version and its accessibility are according to the new format.

There are still challenges to overcome. Our approach gives insights of the upcoming issue raised by the migration of Web content in the context of Web preservation.

## References

[1] B. Laws. *Seriously, Another Format? You Must Be Kidding.* CSE NEWS, vol. 36, no. 2, pp. 41, 2013.

[2] D. S. H. Rosenthal, T. Lipkis, T. Robertson, and S. Morabito. *Transparent Format Migration of Preserved Web Content.* D-Lib Magazine, vol. 11, no. 1, 2005.

[3] J. Van der Hoeven. *Emulation for Digital Preservation in Practice: The Results.* The International Journal of Digital Curation, vol. 2, no. 2, pp. 123-132, Decembre 2007.

[4] J. Garret. *Preserving Digital Information.* Technical report, Commission on Preservation and Access and the Research Libraries Group, 1996.

[5] S. Pfeiffer. *The Definitive Guide to HTML5 Video*. Apress, Berkely, CA, USA, 1st edition, 2010.

[6] S. H. Park, N. Lynberg, J. Racer, P. McElmurray, and E. A. Fox. *HTML5 ETDs.* In Proceedings of International Symposium on Electronic Thesis and Dissertations, Austin, TX, USA, 2010.

[7] A. N. Jackson. *Formats Over Time: Exploring UK Web History.* CoRR, abs/1210.1714, 2012.

[8] Y. Xiao, Y. Tao, and Q. Li. *Web Page Adaptation for Mobile Device.* In proceedings of the The 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2008), pp. 1-5, Dailan, China, October 2008.

[9] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. *Extracting Content Structure for Web Pages Based on Visual Representation.* In proceedings of the 4th 2008 International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08), APWeb'03, pp. 406–417, Xian, China, 2003. Springer-Verlag.

[10] M. B. Saad and S. Gançarski. *Using Visual Pages Analysis for Optimizing Web Archiving.* In Proceedings of the 2010 EDBT/ICDT Workshops, EDBT '10, vol. 7, no. 43, pp. 1-43, New York, NY, USA, 2010.

[11] J. U. Mahmud, Y. Borodin, and I. V. Ramakrishnan. *Csurf: A Context-Driven Non-Visual Web-Browser.* In Proceedings of the 16th International Conference on World Wide Web, WWW '07, pp. 31–40, New York, NY, USA, 2007. ACM.

[12] O. Wu, Y. Chen, B. Li, and W. Hu. *Evaluating the Visual Quality of Web Pages Using a Computational Aesthetic Approach.* In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM'11, pp. 337-346, Hong Kong, China, 2011.

[13] Z. Pehlivan, M. Ben-Saad, and S. Gançarski. *Vi-diff: Understanding Web Pages Changes.* In Proceedings of the 21st International Conference on Database and Expert Systems Applications: Part I, DEXA'10, pp. 1–15, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] Y. Y. Tang and C. Y. Suen. *Document Structures: A Survey.* International Journal of Pattern Recognition and Artificial Intelligence, vol. 8, no. 5, pp. 1081-1111, 1994.

[15] Z. Nie, J.-R. Wen, and W.-Y. Ma. *Webpage Understanding: Beyond Page-Level Search.* SIGMOD Rec., vol. 37, no. 4, pp. 48-54, March 2009.

[16] D. Chakrabarti, R. Kumar, and K. Punera. *A Graph-Theoretic Approach to Webpage Segmentation.* In Proceedings of the 17th ACM International Conference on World Wide Web, pp. 377–386, Beijing, China, 2008.

[17] C. Kohlschütter and W. Nejdl. *A Densitometric Approach to Web Page Segmentation.* In Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 1173–1182, New York, NY, USA, 2008.

[18] X. Liu, H. Lin, and Y. Tian. *Segmenting Webpage with Gomory-Hu Tree Based Clustering.* Journal of Software, vol. 6, no. 12, pp. 2421–2425, Decembre 2011.

[19] A. S. Vargas. *Web Page Segmentation, Evaluation and Applications*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2015.

[20] F. Shafait, D. Keysers, and T. Breuel. *Performance Evaluation and Benchmarking of Six-Page Segmentation Algorithms.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 30, pp. 941-954, 2008.

[21] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma. *Learning Block Importance Models for Web Pages.* In Proceedings of the 13th ACM International Conference on World Wide Web, WWW '04, pp. 203–211, New York, NY, USA, 2004.

[22] B. Solis. *The Conversation Prism*. https://conversationprism.com.

[23] G. Navarro. *A Guided Tour to Approximate String Matching.* ACM Computing Surveys, vol. 33, no. 1, pp 31-88, March 2001.

**Revista Venezolana de Computación**
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 3, No. 1, pp. 38-50, Junio 2016
Selección de los Mejores Artículos del SCTC 2016

# Product Line Scoping for Healthcare Information Systems Using the ISO/IEC 26550 Reference Model

Juan Carlos Herrera[1], Francisca Losavio[2], Oscar Ordaz[2,3]
jchr1982@gmail.com, francislosavio@gmail.com, oscarordaz55@gmail.com

[1]PFG Informática para la Gestión Social, Universidad Bolivariana de Venezuela, Caracas, Venezuela
[2]Escuela de Computación, Laboratorio MoST, Universidad Central de Venezuela, Caracas, Venezuela
[3]Escuela de Matemáticas, Universidad Central de Venezuela, Caracas, Venezuela

**Abstract:** The main goal of this work is to present a process for Software Product Lines (SPL) scoping focused on early consideration of software product quality. Product Line Scoping is the first phase of SPL Engineering (SPLE), in the Domain Engineering (DE) lifecycle, where the SPL long-term feasibility must be determined. The PLScoP process proposed here for SPL scoping, is an adaptation of the general PL Scoping phase defined in the new ISO/IEC 26550 standard describing a reference model or framework for SPLE, and it concerns three main stages, Portfolio Scoping, Domain Scoping and Asset Scoping. In this work, the complete PLScoP is outlined, but only the Domain Scoping phase will be detailed and applied. General guidelines on "What to do", as the majority of standards offer, are defined in ISO/IEC 26550. Our PLScoP complements this framework by presenting the "How to do", offering precise techniques and artefacts to be applied and constructed, and by considering early and systematically quality issues; this will allow reduction of the development effort in the subsequent DE phases, Domain Requirements Engineering and Domain Design, where the major effort is concentrated SPL development workload. The Domain Scoping step of PLScoP will be applied to the Healthcare Information Systems domain.

**Keywords:** Software Product Lines; Product Line Scoping, PLScoP; Domain Scoping; ISO/IEC 26550; Software Quality; ISO/IEC 25010; Healthcare Information Systems.

## I. INTRODUCTION

Software Product Lines (SPL), or simply Product Lines (PL), is an approach that provides a way of massive personalization of individual solutions from a repository of reusable software assets, in a particular domain; it is inspired in the Fordism technique used to increase production while lowering costs in early 20th century automotive industry. The term domain is often used in reference to a particular knowledge area; an application domain denotes any aspect where computing can be applied [1]; a *domain* is defined by Bérard as the minimal set of properties describing precisely a family of problems in which a computational application or system is involved for their solution [2], and it is the definition adopted in this context. The problem of software development based on reusing components or a core of software elements, favouring efficient and reliable development is not new [3][4][5][6] and it is a complex problem not yet completely solved in academic or industrial practices; moreover, there is still a huge gap between research results and their industrial application [7][8]. The SPL development, also called SPL Engineering (SPLE) [9], aims to promote maximal reuse exploiting common elements in similar products of the SPL family in a particular domain. The main idea, but not an easy task, is to capture the essential common elements and possible variable issues to construct an evolutionary SPL, since it must manage changes and last over time to provide an economic payback. Instead of describing a single system, the SPL model describes a set or family of software systems, products or applications. Complex domain analysis must be achieved in order to specify and delimit the family that can be developed from the core of assets, with their commonality and variability [3][4][10].

This work is framed within the Domain Engineering (DE), first lifecycle of SPLE, where the major development effort is concentrated in constructing the SPL Reference Architecture (RA) and Core Asset Repository (see Figure 1). The main goal of this paper is to present and apply a PL Scoping Process (PLScoP), with precise techniques and artefacts, adapting the PL Scoping phase or initial DE phase, defined in the Reference Model of the new standard ISO/IEC 26550 [7][8][11]. Notice, in general that standards or general frameworks specify the "What to do", however the details of the "How to do" have always to be properly defined; even if in [11] a list of available techniques and methods are provided, how to combine or adapt them to the SPL context to achieve a particular activity is not specified. Our work complements the SPL standard framework, offering explicit techniques to be applied to

perform PL Scoping activities. In particular, for the study of domain existing products in the Portfolio Scoping step, a bottom-up process [12] can be considered, providing as output an initial candidate architecture; moreover, the ISO/IEC 25010 quality model [13] is used to specify quality properties related to functional (FR) and non functional requirements (NFR), and BPMN [1][14] is proposed to specify the domain model. PLScoP emphasizes software product quality assurance at early stages of SPLE; quality properties and their traceability w.r.t. FR and NFR requirements has in general been poorly considered in SPL development, playing however a major role in the SPL variability and evolutionary capacity. In this work, the Domain Scoping phase will be applied to the Healthcare Information Systems domain to obtain a first draft of a domain model.

Usual approaches focus more on products' features directly perceived by users [15], which is not the case of quality properties that appear as "implicit functionality" often later on during the SPL development; nevertheless, quality requirements are responsible of most of the SPL variability at the moment of deriving concrete SPL products during the Application Engineering (AE) or second SPLE lifecycle [16][17]. Figure 1 shows the ISO/IEC 26550 SPL Reference Model [11]. If quality properties are not considered early in the DE lifecycle of the SPL development, the global quality of the products derived from RA cannot be guaranteed, compromising organizational goals and the whole SPL ROI (Return of Inversion).

Besides this introduction and the conclusion, the structure of this paper is the following: some related works are discussed in Section 2; Section 3 describes the SPL Domain Engineering guidelines of the ISO/IEC 26550 standard; Section 4 presents PLScoP, as an adaption of the Pl Scoping phase [11]. Finally, Section 5 is dedicated to the application of the Domain Scoping step of PLScoP to a case study in the Healthcare Information Systems domain.

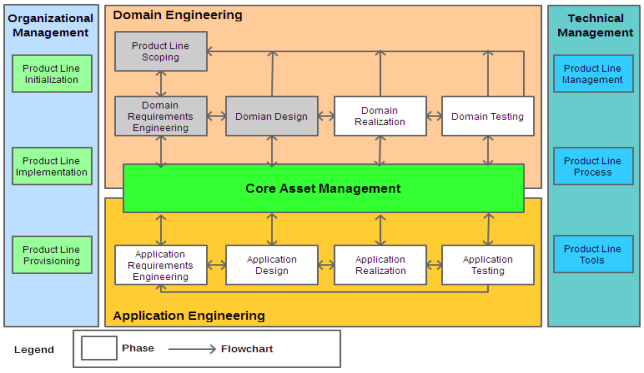


**Figure 1:** ISO/IEC 26550 Reference Model for Software and Systems Product Lines

## II. Related Works

In what follows, some related works relevant to the subject of SPL scoping, context of this research, are discussed:

An SDR (Systematic Documental Review) [18] is presented in [19] to identify best practices, challenges and limitations of the SPL scoping phase; this study points out that scoping is important and even essential for the achievement of the product line. Its main goal is to determine the feasibility of the SPL, identifying crucial aspects such as products that will conform the SPL, risks, potential reuse and costs to implement main assets. An important question related with our work is the following: *How is the SPL development affected by the scoping phase?* It is clear that the artefacts produced during this phase are input to the DRE phase and should reduce the effort also in subsequent phases. The study claims that the majority of the approaches reviewed do not have a clear understanding of the relation between scoping and DRE phases. In this sense, our proposition to use Bjørner's domain modelling [20] in PL Scoping will fill this gap.

A characterization of the benefits and weakness perceived in the Domain Scoping and DRE phases using the agile method is presented in [21]. The observed variables were the stakeholders' motivation, effort, communication and collaboration, iterative and adaptability aspects of the process, requirements and technological volatility. A question arisen from this study is: *How the effort to perform SPL scoping is characterized by the stakeholders?* The effort is measured in man-hours, and the answer was "great", due to the huge amount of domain documentation (often incomplete and inconsistent) that has to be analysed to capture enough domain knowledge to provide acceptable asset scoping and product portfolio, and the lack of domain experts. Activities identified for the scoping phase were: *Pre-scoping*, *Domain scoping*, *Product scoping*, and *Assets scoping*. Bottlenecks found were: the absence of domain and product experts to capture products' main functionalities, and the clear identification of features and their granularity. Moreover, variables stakeholders' communication and collaboration, iterative and adaptability aspects of the process, were also found to affect the effort. Our research do not use an agile method, nevertheless we claim to reduce the scoping effort by introducing the Bjørner's business process-centric technique of domain modelling [20], modified by introducing the quality intrinsics descriptor, to specify quality properties related to business processes functions at this very early stage.

A *framework* is described in [22] for SPL developments including the SPL scoping phase, to specify what the SPL can or cannot do, by defining those behaviours or aspects that will be incorporated or eliminated from the SPL. Scoping defines the long-term feasibility of the SPL; it starts with a broad document, which is being refined in the measure that more domain knowledge is captured. The goal is to establish a limit for the SPL to achieve business and market goals. Our research aims to establish a "structure" for this document (input/output artefacts and their structure, techniques used, etc.), since a precise definition was not found in the literature; it will be used as a valid input to reduce the effort in the DRE and DD phases. Hence the scoping document will result an important asset.

The proposed framework suggests the following activities for SPL scoping:

- Workshops and interviews with the stakeholders
- Examine existing products (bottom-up approach [23])

[1] Business process Modeling Notation

- Context diagrams
- Develop a matrix of attributes/products
- Develop SPL scenarios

where a context diagram represents relevant entities related with the SPL w.r.t the users of the products; an attributes/products matrix is used to define the variability of the SPL; scenarios are used to identify interactions that are common to all products and those that are variants.

The Bjørner's domain modelling [20] used in our Domain Scoping step is compliant with most of the above requirements, excepting for - Examine existing products, which is part of the Portfolio Scoping step, where a bottom-up approach will be used [12][23]; however, this point will not be treated here, being the object of an on-going work.

In [24], a specific method based on PuLSE, Pulse-Eco, is presented. This process uses a DE bottom-up development strategy; we are actually focusing on a DE top-down approach [23] combined with a bottom-up approach to study existing products [12], which is recommended in SPL scoping by the new standard [11]; this study however, can be performed on the basis of an "agile" bottom-up process [25].

In [26], three domain engineering approaches were compared, two of them related to SPL, namely, Pohl et al. [9] and the ISO/IEC standard [11], the other one, the classic Bjørner's DE approach [20] strongly based on business processes modelling. From this comparison, [20] was found suitable to integrates the Domain Scoping step of the ISO/IEC 26550 SPL Reference Model [11], and quality was included as a new facet; however, since quality is involved in all facets used in [20] to specify the domain model from different stakeholders' viewpoints; in our present work, quality is specified as an intrinsics facet descriptor and the whole approach is illustrated with a complete case study.

## III. SPL Domain Engineering Guidelines with the New Standard ISO/IEC 26550

The complete DE lifecycle will be briefly described in what follows, according to the ISO/IEC 26550 guidelines [11]; however in this work, only the Domain Scoping step of the PL Scoping phase will be treated in details.

According to the SEI[2] definition [22][27][28], PL Scoping is itself a core asset. In SPL development, scoping is a fundamental activity that will determine the long-term viability of the SPL. Like scoping in general, PL Scoping determines what's "in" and what's "out" of the SPL. The scope definition identifies those entities with which products in the SPL will interact (that is, the product line context), and it also establishes the commonality and sets limits on the variability of the products in the SPL. The scope definition usually begins as a broad, general draft document that is refined as more knowledge is captured and more analysis is performed. For example, for an SPL for a Web-based system, browsers would definitely be "in". Aircraft flight simulators instead, would definitely be "out"; the PL scope may not come into sharp focus all at once. The goal of the scope definition is to draw the boundary between "in" and "out" in such a way that the SPL satisfies its business and market goals.

Five phases are considered in ISO/IEC 26550; notice that phases and sub-phases can be performed in the order established by the company or organization requiring the SPL [11].

### A. Product Line Scoping (PL Scoping)

It defines SPL boundaries for DE, envisioning major common and variable features to all products within the SPL. Economical aspects are analysed and the commercialization of the SPL product family is planned; PL Scoping is responsible of the whole SPL management and consequent evolution: it involves 3 sub-phases:

*1) Product Portfolio Scoping:* 1. Identify products that the SPL should be developing, producing, marketing and selling (product "roadmap"); 2. The study of common and variable feature of existing products should provide guidance to meet business objectives and face SPL evolution; 3. Schedule for introducing products to the market. It is input to Domain Scoping.

*2) Domain Scoping:*identify and bound functional or organizational areas that SPL will impact to provide sufficient reuse potential to justify the SPL creation.

*3) Assets Scoping:*identify the boundaries of core assets, providing a first glance at common and variable assets. It Identifies reusable assets and calculate the cost/benefit estimated from each asset in order to determine whether an organization should launch an SPL.

Major outputs of PL Scoping are: the asset proposal; it includes major assets (functional areas and high-level common and variable features of all SPL products) that will be included in an SPL with their quantified costs and benefits estimation results. The features defined in the asset proposal directly affect Domain Requirements Engineering (DRE) and Application Requirements Engineering (ARE) shown in Figure 1. More than one asset proposal can be made to find out an optimal set of products and assets. The asset proposal defines also a schedule for delivering specific products to customers and for bringing them to market.

### B. Domain Requirements Engineering (DRE)

It has to adhere to the specification of the SPL's high-level features provided by PL Scoping. Based on these features, it creates detailed common and variable requirements sufficient to guide subsequent Domain Design (where the SPL RA is designed), realization and testing phases. It involves 5 sub-phases: - Domain requirements elicitation, - Domain requirements analysis, - Domain requirements specification, - Domain requirements validation, and - Domain Requirements Management (to handle changes in requirements).

### C. Domain Design (DD)

It draws upon the specifications to develop an SPL architecture that enables the realization of the planned commonality and variability within the SPL. The main goal is to produce the RA, defining general SPL structure and

[2] Software Engineering Institute, MIT, Carnegie Mellon

textures. RA reflects additional internal variability introduced by technical solutions besides the external variability, i.e., commonality and variability in the user's perceived requirements. It involves the sub-phases: - RA design, - RA evaluation (quality assurance technique), and - Domain Design Management (to handle changes in the RA design).

### *D. Domain Realization/Implementation*

Design and implementation of reusable loosely coupled software components and configurable interfaces, implementing common and variable artefacts offered by RA. Domain realization includes configuration mechanisms to realize variability domain implementation, such as building and buying components supporting the RA infrastructure. They are not yet executable applications.

### *E. Doman Testing/Validation*

It validates the domain artefacts created in previous phases and generates domain test artefacts that can be reused later on in Application Testing. Testing in this context means review, validation and verification of artefacts as well as eventually testing some available implementations.

In this work, only the Domain Scoping sub-phase, within the PL Scoping phase will be applied to a case study, to illustrate our approach.

## IV. THE PL SCOPING PROCESS: PLSCOP

The adaptation of the ISO/IEC 26550 PL Scoping phase is constituted by the PLScoP process that is outlined in what follows:

### *A. PLScoP Context*

The PL Scoping guidelines of the ISO/IEC 26550 standard were completed by integrating to the Domain Scoping sub-phase, the stages described by Bjørner [20] for classic DE for single software systems, not considered for an SPL context. However, they provide a nice technique to specify domain knowledge, based on facets and stakeholders' viewpoints [26].

A *facet* is defined in [20] as a finite set of generic forms of describing a domain from different stakeholders' perspectives or viewpoints, namely, business processes, support technology, management & organization, rules & regulations, human behaviour, and intrinsics; complete definitions of these terms were presented in [26].

The facet notion is not new in Software Engineering [29]; according to [20], each facet represents a view of the domain from different perspectives; the union of these views conforms the complete domain view, called Domain Model; moreover, the special *intrinsics facet* is a facet containing descriptors or attributes (entity, function, event, behaviour) necessary to describe all the other facets (see Table I); the intrinsics notion has allowed us to include software quality, specified by the ISO/IEC 25010 standard [13], as a new intrinsic descriptor. Software quality is then considered to specify all other facets.

**Table I:** Intrinsics to Describe all Domain Facets with the New Quality Descriptor

| Intrinsics descriptors | Description |
|---|---|
| Entities | Represent the phenomena and concepts of the domain |
| Functions | Operations (actions) performed on the entities |
| Events | Imply changes in entities by function invocations, i.e., actions in the domain |
| Behaviour | Sequences of actions and events affecting domain entities |
| Quality | Specified by the standard quality model ISO/IEC 25010 [13]. It is associated to the Business Processes facet as quality goals (obtained from NFR) required by FR (functions, activities or tasks), to facets Support Technology as quality supported by architectural styles, patterns or mechanisms involved, and Rules &Regulations as quality required by domain business rules [26]. Product quality must be specified to guarantee that SPL products will hold acceptable industrial quality levels. |

In particular, this work involves Business Processes, Support Technology, and Rules & Regulations facets, since the final aim of DE is to build an SPL reference architecture; Management & Organization and Human Behaviour facets can be also described in terms of quality, using models, such as CMMI[3]where organizational practices are deeply involved, but this topic is outside the scope of the present work.

Notice that in the Domain Scoping adaptation from [11] integrating Bjørner's domain development [20], a huge number of business processes can be derived from the so called Domain Description Units (DDU) specifications from the declarations of different stakeholders groups expressing their viewpoints. However, this complete specification is outside the PL Scoping spirit, which aims to offer a quick "glance" of the SPL feasibility and limitations, hence only the presentation of few basic modelling elements are considered sufficient to illustrate the "How to do" of our process.

### *B. PLScoP Context*

The adaptation of the ISO/IEC 26550 PL Scoping guidelines, is presented in what follows; notice that Asset Scoping and Portfolio Scoping are left as the last steps, since according to [11], the order in which they are executed depends on the organization building the SPL, and we have major interest here in Domain Scoping; however, if a study of existing market products for the domain has to be done, Product Portfolio Scoping should be executed first to perform this study, and its output should be input to the Domain Scoping step.

*PLScop process*

1. *Domain Scoping*:
   *Input*: Domain informal documentation provided by the organization requiring the SPL, Domain Quality Model (DQM) specified by ISO/IEC 25010 [13].
   a) *Stakeholders Identification*:

[3] Capability Maturity Model Integration

*Input:* visits, interviews, workshops, questionnaires (specific techniques for each one of these activities should be specified).

- Identify groups of stakeholders with similar interests in the organization requiring the SPL.

*Output:* list of stakeholders' groups type: text

b) *Domain Acquisition*:
*Input: List of* Stakeholders' groups

- Capture and gather information from stakeholders into *declarations* to build Domain Description Units (DDU) for each stakeholder viewpoint;

*Output:* DDU type: table

c) *Domain Analysis*:
*Input:* DDU
Analyse DDU, study possible inconsistencies, and business processes are extracted first from DDU [20], to specify the Business Processes facet from a stakeholder viewpoint relevant to the domain; it is represented by a table, UML [30] and BPMN diagrams, using intrinsics facet descriptors. Quality issues are included as a new intrinsic facet descriptor.
Then other facets, relevant to the domain, are also specified according to domain specific stakeholders' viewpoints; for example, in our case they will be *Support Technology* from the *Domain Engineers viewpoint*, and *Rules & Regulations* from the *Directors of healthcare governmental institutions viewpoint*.
The business processes extracted from the DDUs for these stakeholders' viewpoints are identified among the behaviours present in the facets, and specified as new Business Processes facets, considering the respective stakeholders' viewpoint.
*Output:* Facets specifications type: table; UML and BPMN diagrams for business processes

d) *Domain Modelling*:
*Input:* Facets specifications
A domain description is obtained from all the facets specifications by intrinsics; this document should be focused on commonality and variability of the entities involved. According to [20] a domain model is a meaningful domain description; it will be represented integrating the BPMN specifications for all business processes considered.
*Output*: Domain Model type: BPMN. To have a more general specification of the domain model, including all facets, an ontological approach could be considered.

2. *Asset Scoping*
*Input* Domain Model specification
Information on core assets will be extracted from domain model to conform the SPL Core Asset Repository, which will be informally described into the Asset Proposal document, analysing here also economical factors for the SPL feasibility.
*Output:* Asset Proposal document

3. *Product Portfolio Scoping*
*Input:* Available documentation on existing products
Existing products in the domain are assumed to exist; they will be studied to infer about the SPL products that can be developed, main capabilities and limitations; this study could be preformed applying an extractive or bottom-up process to construct automatically a draft *candidate architecture (CA)* [31] using reengineering techniques to handle similarity analysis of the products' components [12]. Notice that since in general widely used products on the market are considered, results of the existing products study are included into the market study on the SPL feasibility.
*Output:* Product Roadmap: CA: type: graph or UML diagram

Notice that the Product Roadmap artefact includes the candidate architecture artefact which is a first broad draft of the RA that can be built [12]. This possibility should be considered, to have also an additional input to the DRE phase, thus reducing the required effort there. This initial candidate architecture has imbedded the domain knowledge, extracted from existing products about main common and variant components, which can be enriched with the more general information obtained from the Domain Model. It will become also part of the Asset Proposal.

### C. PLScoP: Advantages and Limitations

If a first candidate architecture can be constructed or is available for a domain, our PLScoP, and in particular the Portfolio Scoping step, can be transformed into a process to perform the RA evolution, i.e., management of changes, in DRE and DD phases. The analysis of existing products is not an easy task, and it depends much on the available market products documentation, requiring a considerable effort to apply reengineering techniques [12]. If this draft architecture is not available, the Domain Scoping step of our PLScoP is still crucial to construct a detailed Domain Model that will help to delimit clearly the SPL scope and functional and non functional granularity, to reduce the effort in the subsequent DE phases where the RA is built.

The advantage of our process is to have combined top-down and bottom-up approaches to specify the domain: top-down is considered in the philosophical Bjørner's approach [20], which starts with the domain decomposition into organizational business processes specified by intrinsics descriptors, and it is generally used in SPLE [9][11]. The bottom-up approach is proposed to be used in the Portfolio Scoping step also in [9][11], to have a broad picture of the SPL present and future products, by studying domain existing products in the organization proposing the SPL or in different organizations with similar domains. In this sense, our proposition takes advantages from the combination of [9] and [11], reducing general weaknesses of DE lifecycle.

## V. Application to the Healthcare Information Systems Domain

In what follows, the Healthcare Information Systems (HIS) Domain for the SPL will be briefly discussed to be applied to illustrate PLScoP.

### A. SPL Domain: Healthcare Information Systems (HIS)

HIS [32] are software intensive systems, i.e., complex integrated information systems, generally located in different and distant institutions and with mandatory (priority) NFR, such as interoperability, availability and security. Interoperability (technical), the HIS crucial quality property for *Electronic Health Records (EHR)* management and sharing, is the ability of two or more systems or components to exchange information and to use the exchanged information; semantic interoperability refers to use a common terminology or

language to communicate systems; process interoperability incorporates business processes and healthcare professionals must standardise business rules to ensure that health information is properly recorded, such as the transfer of information between systems is consistent and complete [33]. The general architecture is a hybrid event-based style, SOA[4]/Layers [12][31], see Figure 2 (from Wikipedia). HIS must facilitate transparent sharing of different kinds of medical information such as EHR and laboratory&imaging results, offering also telemedicine services that can be performed online at remote locations, with wide support of information technology. The use of standards such as HL7, HL7 CDA, LOINC[5], and DICOM[6] are mandatory for interoperability of EHR and laboratory&imaging results [25][32][33][34].

Nevertheless, in actual medical practice, SPL for HIS have not yet been completely defined, developed and adopted; the lack of agreement on medical standards and psychosocial issues makes difficult the interoperability of EHR, and HIS general adoption is still difficult, even if specific laws and regulations towards these goals have been promulgated worldwide.

### *B. Software Quality Modeling in SPL Domain Scoping*

Quality has been defined in general as a level of excellence, conformance with specifications, requirements satisfaction, defect free, accomplishment of customer demands [35], and also related to human aspects such as usability and satisfaction [36]. The early specification in PL Scoping of software quality will facilitate to map this quality into all subsequent DE activities; this information on domain quality will be specified as a Quality Model (DQM), reflected into the Domain Model, and registered in the Asset Core Repository. As we have already pointed out, quality assurance is crucial in an industrial software production context to guarantee SPL evolution and the massive assets reuse, impacting on the quality of all products of the SPL family [37].

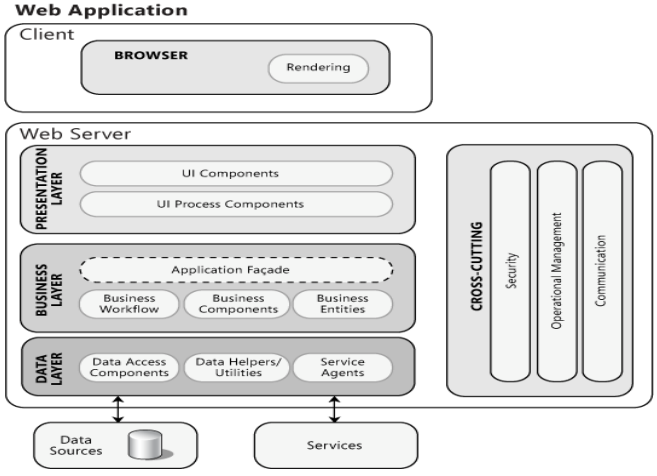


**Figure 2:** Hybrid Architecture SOA/Layers

The ISO/IEC 25010 Quality Model standard [13] to specify software product quality is part of the SQuaRE series of standards of the International Standards Organization (ISO). This series focuses on quality, requirements and evaluation of software products. It states compatibility with other ISO standards on quality measures and process quality [38]. The central document of the series is the known ISO/IEC 25010 Quality Model [13], describing a hierarchical framework where quality is decomposed into levels of characteristics, sub-characteristics, etc., until the attributes or measurable elements. The Product Quality Model will be used here, since we are in a software development context; Figure 3 [25][32], shows its adaptation to specify the HIS domain quality w.r.t. the SPL family of software products.

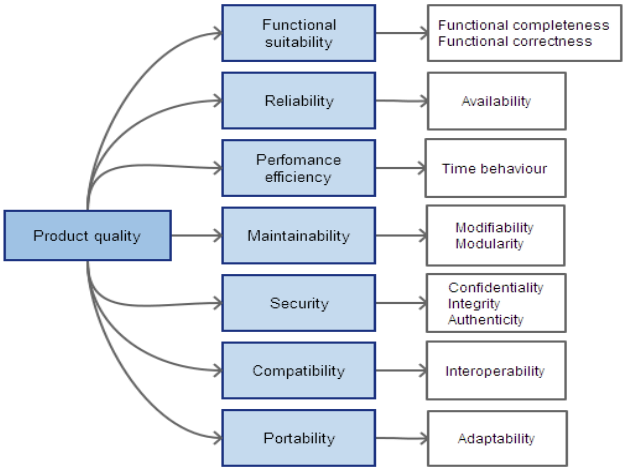


**Figure 3:** ISO/IEC 25010 HIS Domain Quality Model (HIS-DQM)

### *C. Case Study Scope*

For this work, the HIS domain for the SPL is restricted to its basic functionalities (EHR-HIS), namely EHR management, patient attention with patient appointment scheduling and capture of demographic data, emission of medical reports, basic administrative services for patient attention; imaging and laboratory services, hospital rooms management, nursing services, urgencies, general hospital administration, etc. will not be considered here [25][34]. Note that the example has been simplified, since only HIS elements at a high granularity level have been treated, avoiding low-level details to facilitate the illustration of our approach, following also the spirit of the PL Scoping phase.

On the other hand, the main facets that will be specified in this work are Business Processes, Support Technology, and Rules&Regulations. The viewpoints considered are: Doctors' group for Business Processes, Domain Engineers' group for Support Technology, and Directors of healthcare governmental institutions for Rules&Regulations. Facets, stakeholders' groups, and viewpoints were selected on the bases of the SPL DE development process, whose main goal is to design a Reference Architecture.

### *D. Application of PLScop to the HIS Case Study*

Following the basic steps defined in the *Domain Scoping* sub-phase, we have:

*1) Stakeholders Identification:*The technique used consits in doing the expertise of domain engineer extracted from visits,

[4] Service-Oriented Architecture
[5]Logical Observation Identifiers Names and Codes
[6]Digital Communication in Medicine

interviews, workshops or questionnaires. Different viewpoints are identified to conform the stakeholders' groups: Doctors, Domain Engineers and Directors of Healthcare governmental institutions.

*2) Domain Acquisition:* The technique used consists in statements or Declarations formulated by stakeholders to illustrate their viewpoints.

*DDU construction:*

The declarations are grouped into the *Domain Description Unit (DDU)* and they are represented textually as a table (see Tables II, III, and IV) for each one of the identified stakeholders' groups.

**Table II:** DDU for EHR-HIS Domain from the Doctors' Viewpoint

| **EHR-HIS Domain Description Unit** | |
|---|---|
| **Viewpoint** | **Stakeholders' Group: Doctors** |
| ***Declarations*** | |
| Patient is attended in hospital under scheduled appointment | |
| If it is the patient first appointment, a new EHR must be created by nurse, else nurse retrieves patient existing EHR | |
| Patient EHR is accessed by doctor | |
| New exams and laboratory results can be added to patient EHR by doctor, if it is the case | |
| Diagnosis and medical orders for patient are produced by doctor to conclude medical attention | |
| A new appointment is scheduled by nurse if required by Doctor | |
| Medical equipment and material can be required by doctor to provide adequate medical attention | |

**Table III:**DDU for EHR-HIS Domain from the Domain Engineers' Viewpoint

| **EHR-HIS Domain Description Unit** | |
|---|---|
| **Viewpoint** | **Stakeholders' Group: Domain Engineers** |
| ***Declarations*** | |
| HIS is supported by an hybrid architecture SOA/Layers | |
| The HIS architectural style supports mainly modifiability, interoperability, performance (time-behaviour), and security services are provided by Internet protocols, crosscutting all layers; availability however depends on internet connection; | |
| Transmission Layer is managed by a Web Server that communicates all other layers | |
| Clients access HIS by a browser in Presentation Layer, which connects to the Transmission Layer via a Web Server | |
| Medical units should have wide range internet and intranet access | |
| Main EHR-HIS functionalities must be supported: patient appointment services, EHR management and emission of medical reports. | |

**Table IV:**DDU for EHR-HIS Domain from Directors of Healthcare Governmental Institutions' Viewpoint

| **EHR-HIS Domain Description Unit** | |
|---|---|
| **Viewpoint** | **Stakeholders' Group: Directors of Healthcare Governmental Institutions** |
| ***Declarations*** | |
| Digitalize EHR with standard format to achieve sharing among doctors and national and international healthcare institutions | |
| Have Database of national and international specialists | |
| Develop a Web platform to manage on-line appointment services | |

*3) Domain Analysis: The t*echnique used consists in the initial domain knowledge is captured from the DDUs (see Table II, III and IV, and described textually in Table III as business processes.

*Facets specification:*

In Table V, business processes specific to the Business Processes facet are derived first [20] from DDU, considering the Doctors' viewpoint. Tables VI and VII will specify business processes specific to facets Support Technology and Rules & Regulations, respectively; only one process will be specified for each stakeholder's group viewpoint to abridge this presentation.

**Table V:** Business Process Derived from the DDU of Doctors' Viewpoint

| **Viewpoint** | **Stakeholders' Group: Doctors** |
|---|---|
| ***Business Process*** | ***Description*** |
| Appointment Services | From the arrival of a patient to hospital to attend a scheduled appointment, check or create new patient EHR including capture of demographic data and general patient information by nurse; EHR management, medical consultation, diagnosis, emission of medical order |

**Table VI:** Business Process Derived from the DDU of Domain Engineers' Viewpoint

| **Viewpoint** | **Stakeholders' Group: Domain Engineers** |
|---|---|
| ***Business Process*** | ***Description*** |
| EHR Management | Consider in the User Interface (UI) component in Presentation Layer, the access to the EHR Management system in Process Layer; provide EHR access, modification, sharing and registering in database in Data Layer. A Transmission Layers should be present for network services, and it crosscuts all other layers. |

**Table VII:** Business Process Derived from the DD of Directors of Healthcare Governmental Institutions' Viewpoint

| **Viewpoint** | **Stakeholders' Group: Directors of Healthcare Governmental Institutions** |
|---|---|
| ***Business Process*** | ***Description*** |
| On-line appointment services | Provide precise appointment services with specific specialist; handle requests' volume, provide secure access to appointment services; have wide-range and reliable connection facility; have a friendly user interface |

Different notations can be used to specify facets from stakeholders' viewpoints with intrinsics descriptors, each one offering different specification granularity; from each specification, more details are extracted; in this work the following notations will be used:

- informal textual specification by tables (see Tables VIII, X, and XI),
- semi-formal UML [30] diagrams (see Figure 4, 6, and 7) for all facets,
- semi-formal BPMN [14] diagrams for business processes in the Business Process facet (see Figure 5 [39]).

In this context, a semi-formal notation language means that it has well-defined syntax and semantics; however it cannot be verified mathematically. Formal languages, such as VDM, Z, B, and RAISE/RSL, also mentioned in [20], are used to specify high assurance systems, to reduce errors in requirement definitions of safety-critical software systems. The HIS domain is basically constituted by non-safety-critical integrated enterprise systems; hence we chose UML and BPMN standard

notations, widely used by the software community in this domain.

**Table VIII:** Textual Specification of Doctors Viewpoint for Appointment Services Business Process Facet Specified by Intrinsics

| **Domain** | **EHR-HIS – Healthcare institution Requiring the SPL** | | | |
|---|---|---|---|---|
| **Viewpoint** | **Stakeholders' Group: Doctors** | | | |
| **Appointment Services Business Process Facet** | | | | |
| ***Entities*** | ***Functions*** | ***Events*** | ***Behavior*** | ***Quality*** |
| EHR | EHR Access | *EHR is found* or *created* | Nurse confirms the existence of patient EHR or creates an EHR for new patient; nurse provides EHR to doctor *(EHR is provided to doctor)* | *Security (confidentiality, authenticity, integrity)* to access EHR: confidentiality or access policy is different for doctors and nurses, authenticity is needed to identify user, and integrity is required for data consistency. |
| | EHR Register | *EHR is requested* by doctor; doctor attends patient medically and *EHR is modified* | Doctor reviews patient EHR, adds laboratory and/or examinations results, if any, to patient EHR; patient is attended medically; EHR is modified *(EHR is reviewed, modified registered)* | *Availability* of on-line connection to access EHR, *time behavior* to quick access to EHR, *modifiability* to change EHR, *availability-persistency* to retrieve always EHR and i*nteroperability* for EHR sharing. |
| Diagnosis | Perform diagnosis | *Medical hand-outs or catalogues are required on-line; on line consultation with other doctors* in different healthcare institutions or locally to perform diagnosis; *diagnosis is provided* and *registered on EHR* | Doctor requests on-line hand-outs or catalogues; he can consult on-line with other doctors in different healthcare institutions or local doctors, and patient's EHR must be shared by other doctors; review laboratory and/or examinations results; doctor produces diagnosis*(Request of on-line hand-outs)* | *Time behavior* and *availability* of on-line connection to access quickly hand-outs, catalogues and other doctors |
| Resources: Medical Materials (*) | Medical materials request | Doctor *requires medical material* for consultation | Doctor requests medical materials calling authorized staff (or using the EHR-HIS system if this facility is available) *(Request of medical material)* | Physical *availability* of materials, *time behavior* to receive materials from authorized staff (or *availability* of on-line connection to access the EHR-HIS system for materials request, if this facility is supported) |
| Resources: Medical Equipment (*) | Medical equipment request | Doctor *requires special equipment* for diagnosis | Doctor requests medical equipment calling authorized staff (or using the EHR-HIS system is this facility is available) *(Request of medical equipment)* | Physical *availability* of equipment, *time behavior* to receive equipment from authorized staff (or *availability* of on-line connection to access the HER-HIS system for equipment request if this facility is supported), |
| Medical Appointment | Emission of new appointment | *New appointment is required* for patient, if necessary; appointment is scheduled | Doctor registers patient for a new appointment; the appointment is scheduled. *(Request/schedule new appoint.)* | *Availability* of on-line connection to access EHR for Medical Appointment |
| Medical Order | Emission of medical order | *Medical order is elaborated* for present consultation | Doctor emits medical order *(Emit/register medical order)* | *Availability* of on-line connection to access EHR to emit and register Medical Order |
| (*) These services will not be considered for EHR-HIS in this work; the abridged behavior name is specified within ( ) in italics | | | | |

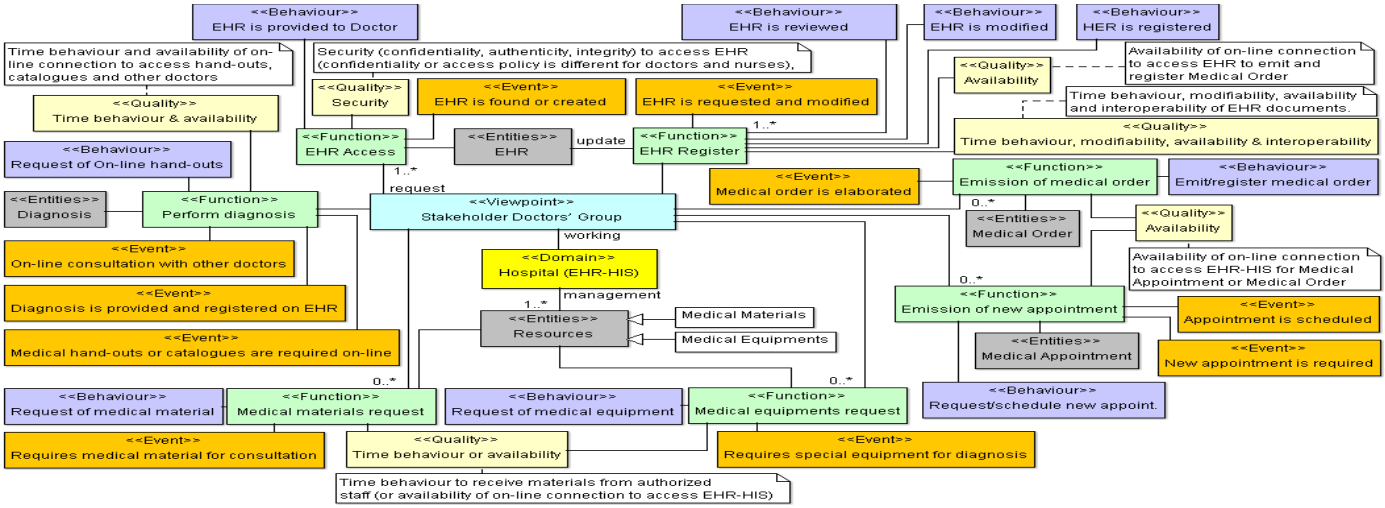


**Figure 4:** UML Specification for Doctors' Viewpoint of the Appointment Services Business Process Facet

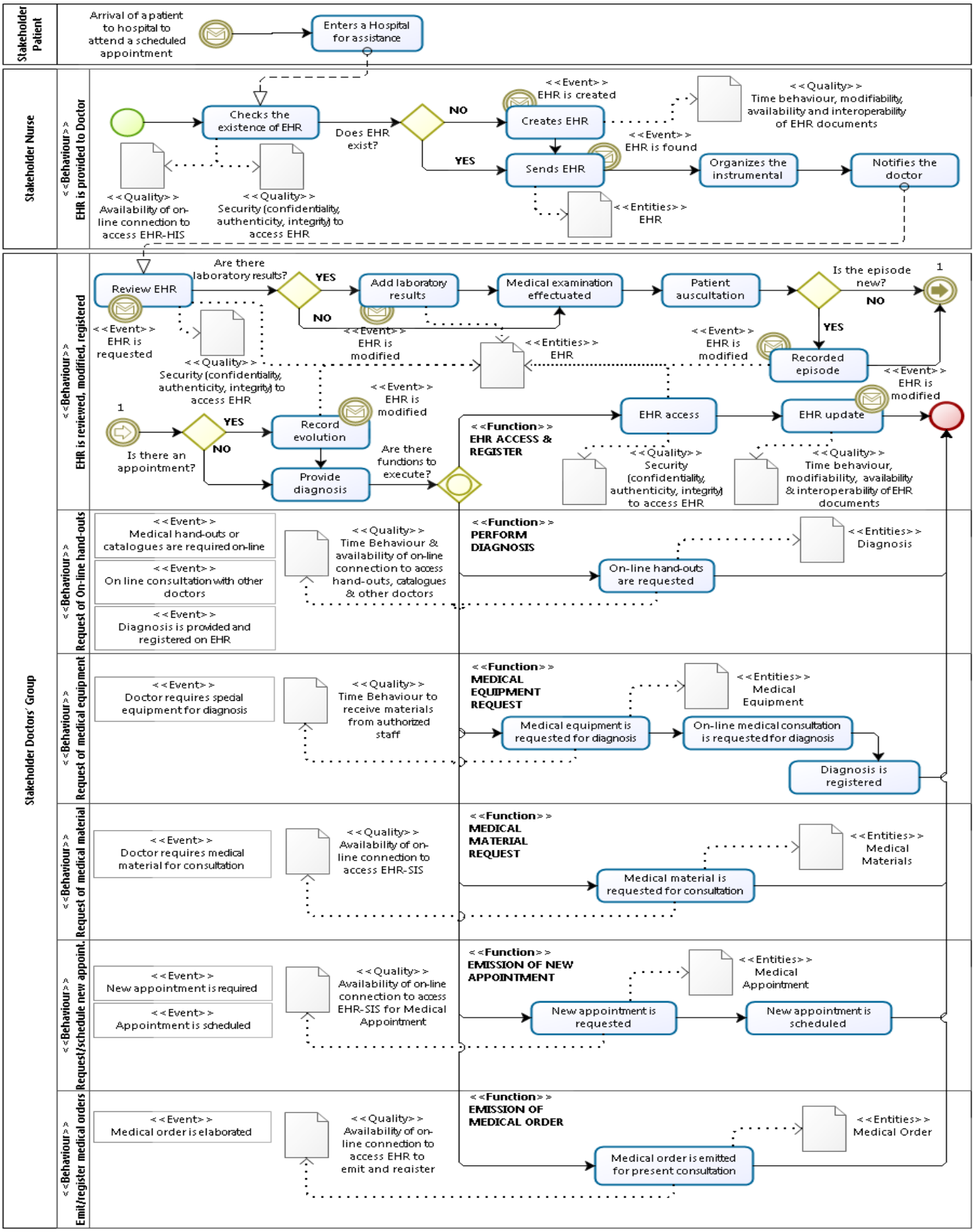


**Figure 5:** Appointment Services Process from Doctors' Viewpoint Specified from the Intrinsics Facet, Expressed in BPMN

**Table IX:** BPMN Notation Used in the Business Process Facet Specification

| BPMN Notation | Symbol | Description | Interpretation (for Figure 5) |
|---|---|---|---|
|  | Task | Simple or atomic activity representing the work in an organization. They consume resources; it is not detailed further. | They can be mapped into SPL RA architectural components or services |
|  | Exclusive gate | Control element of data workflow. A unique path is selected among several alternatives | It represents the SPL variability. |
|  | Inclusive gate | Control element of the data workflow. One or more path (s) can be selected from several alternatives | It represents the SPL variability. |
|  | Starting event | It initiates a process | It represents the starting or entry point of a business process. |
|  | Starting message event | A process initiates when a message is received | It represents the events arising from an active business process. |
|  | Ending event | It indicates the end of a workflow | It represents the end of some behavior in the business process. |
|  | Terminal ending event | It indicates that a process ends, even if there are active workflows | It represents the end of a business process. |
|  | Intermediate link event | It allows the connection of two process sections. | It allows a connection between two sections of the business process. |
|  | Swimlanes (Pool) | It is a process container; it represents participant, entity or role. | It represents the stakeholders' viewpoints. |
|  | Swimlanes (Lane) | They are pool's subdivisions; it represents participants within an organization. | It represents the different behaviors of the business process. |
|  | Connector object: Sequence | It represents the workflow and the sequence of activities. | It represents the execution flow of the activities in a business process. |
|  | Connector object: Message | It represents interactions among processes or pools. | It represents changes of stakeholders' viewpoints. |
|  | Associations | They are used to relate additional information on the process. | They relate entities, activities or functions with required quality properties. |
|  | Artefact: data object | They provide additional information on the process; they show the information required by an activity, such as input/output. | Document specifying the quality required by entities, activities or functions, in each behavior. |

Other business processes can be derived from the analysis of the other facets, such as the on-line appointment services process (see Table VI) from the DDU representing the viewpoint of Directors of healthcare governmental institutions, or the construction of EHR management system (see Table VII) process from the DDU representing the viewpoint of Domain Engineers; however, they will not be considered for this study to abridge the presentation.

Table IX describes the BPMN symbols used in the *Appointment Services Process from Doctors' Viewpoint,* specified from the intrinsics facet. A glimpse on SPL variability can be inferred from the inclusive and exclusive logic gates, since they reflect alternative workflows; they imply a sequence of actions to be performed to achieve a functionality, i.e. functional variability; however, since each activity has associated its quality property, this also can imply non functional variability. This point has to be signalled, because in the domain modelling by the intrinsic descriptors, which was represented in UML (see Figure 4), variability cannot be shown.

In consequence, the use of BPMN is advantageous for our approach, because it contributes to show variability at business process level, which will be mapped later-on into the SPL RA variability model.

*4) Domain Modelling:*The technique used consists in the facet specifications with the intrinsics, by tables and UML diagrams, obtained in step (c) (see Tables VIII, X and XI, and Figures 4, 6, 7). Notice that we have only the Appointment Services Business Process facet, which is specified in BPMN in Figure 5, and it is an example of a Domain Model. However, to have the complete Domain Model picture, the other business processes derived from DDUs in Tables VI and VII, EHR management, from the Domain Engineer viewpoint and On-line appointment services, from the Directors of healthcare governmental institutions viewpoint respectively, are found as behaviours in the corresponding facet specification (see Tables X and XI), and they can be specified by intrinsics as new Business Process facets, as it was done for Appointment Services in Table VIII. From the business process facet analysis, see Table VIII and Figures 4 and 5, and from the BPMN specifications, we obtain information on:

- The clear identification of the involved stakeholders.
- Possible SPL variants from a particular viewpoint, such as the security quality property (see Figure 5), which can be solved proposing later on different available technological mechanisms or services. Note that quality properties are reflected in all viewpoints specifications due to the intrinsic quality descriptor that has been introduced in the adaptation of the domain modeling approach of [20].
- Fine-grained functionalities present in the process as "functions" that can be mapped into large-grained architectural components or services with the BPMN "Aggregation" construct.

  Quality that must be satisfied by each functionality present in the process is specified as a comment, since BPMN has no notation for quality properties.
- The entities or objects produced or manipulated from/by functionalities.

**Table X:** Textual Specification by Intrinsics of Domain Engineers' Viewpoint for the Support Technology Facet

| **Domain** | **EHR-HIS – Healthcare institution requiring the SPL** | | | |
|---|---|---|---|---|
| **Viewpoint** | **Stakeholders' Group: Domain Engineers** | | | |
| **Support Technology Facet** | | | | |
| ***Entities*** | ***Functions*** | ***Events*** | ***Behaviour*** | ***Quality*** |
| Presentation Layer: User Interface (UI) | HIS on-line access | *HIS is accessed* through UI access buttons | Access to HIS main functionalities by authorized persons *(EHR access)* | *Security (confidentiality, authenticity, integrity)* to access EHR-HIS (confidentiality or access policy is different for doctors or nurses)*; availability-persistency* of connection to access EHR-HIS; authenticity is required for user identification, and integrity refers to maintain and assure data accuracy and consistency; *time behaviour* response time to retrieve HER; *adaptability-scalability* refers to add new medical standards |
| Process Layer: EHR-HIS functionalities | Patient System | Patient appointment services including demographic and general information data collection | Scheduling services for patient appointments *(Appoint. Mang.)* | *Correctness-Precision:* for computation algorithm, *security, availability-persistency* |
| | EHR Management System | EHR management: access, registering, modification, sharing | EHR access, recording, modification, sharing; provide access to medical catalogues for diagnosis, to on-line consultation for diagnosis, emission of diagnosis, consultation and/or addition to EHR of new laboratory & examination results *(EHR management)* | *Interoperability, adaptability-scalability, availability-persistency, security* |
| | Report Systems | Emission of medical reports and billing services | Medical reports and billing management: edition, access, modification, registering *(Reports/Billing Manag.)* | *Correctness-precision, availability-persistency, security* |
| Data Layer: Data Base | Data Base Management System | Provide all database services | Allows interoperability, adaptability, persistency, integrity for database services *(Provide quality DB services)* | *Interoperability, adaptability-scalability, availability-persistency, security (integrity)* |
| Transmission or communication) Layer: Network | Internet Communication protocols | Provide information exchange | Allows efficient, reliable, secure information exchange; information is exchanged independently from the platform *(Provide quality information exchange)* | *Time behaviour, availability-persistency, adaptability-scalability, security* |

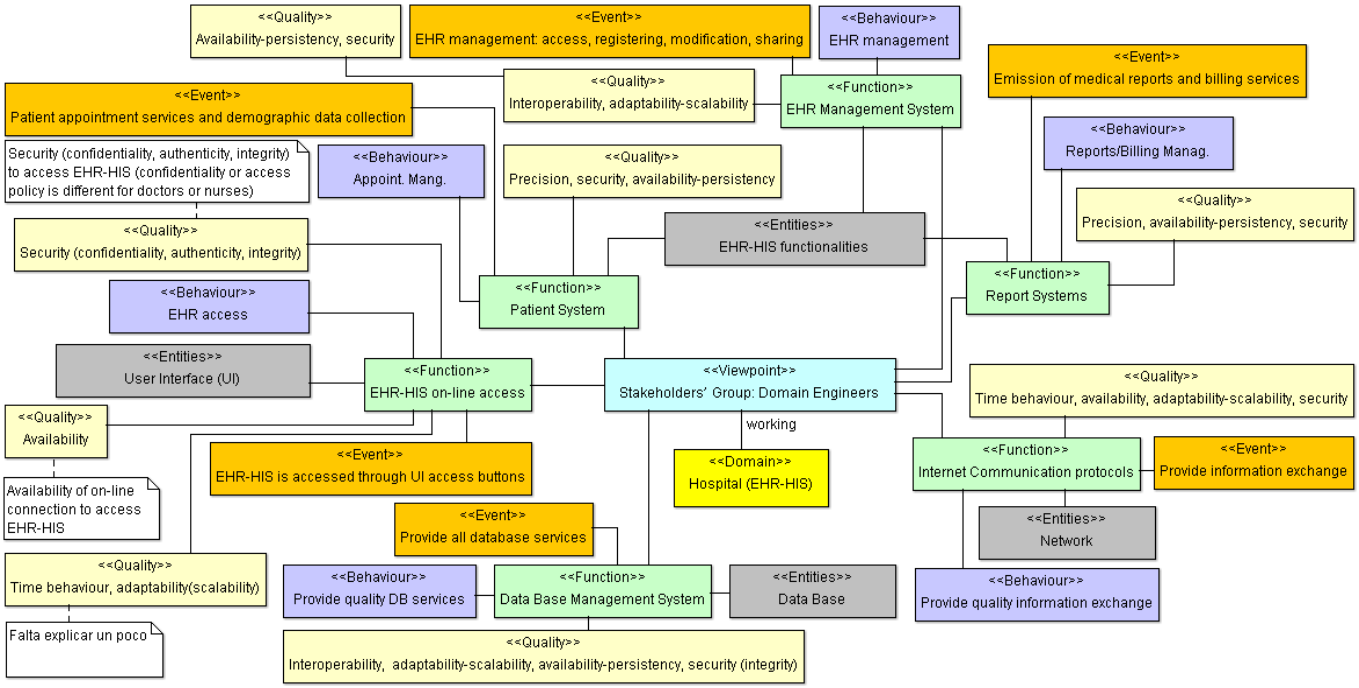


**Figure 6:** UML Specification by Intrinsics of Domain Engineers' Viewpoint for the Support Technology Facet

**Table XI:** Textual Specification by Intrinsics of Directors of Healthcare Governmental Institutions' Viewpoint for the Rules&Regulations Facet

| **Domain** | **EHR-HIS – Healthcare institution requiring the SPL** | | | |
|---|---|---|---|---|
| **Viewpoint** | **Stakeholders' Group: Directors of Healthcare Governmental Institutions** | | | |
| **Rules & Regulations Facet** | | | | |
| ***Entities*** | ***Functions*** | ***Events*** | ***Behaviour*** | ***Quality*** |
| EHR | EHR sharing | Healthcare records is digitalized using some standard format | Healthcare records are shared *(EHR sharing)* | *Interoperability* |
| Database of specialists | Data Base Management System | Provide all data base services; Retrieve, record, modify specialist data | Allows interoperability, adaptability, persistency, integrity for database services *(Provide database services)* | *Interoperability, adaptability-scalability, availability-persistency, security (integrity)* |
| Appointment | Appointment scheduling service | Appointment is assigned or kept in a waiting list; appointment is reported to patient; priority is managed | Provide precise appointment services with specialist; handle volume of requests, provide secure access to appointment services; have available connection facility; have a friendly user interface *(Provide on-line appoint. services)* | *Adaptability-scalability, availability-persistency, security, usability, precision* |

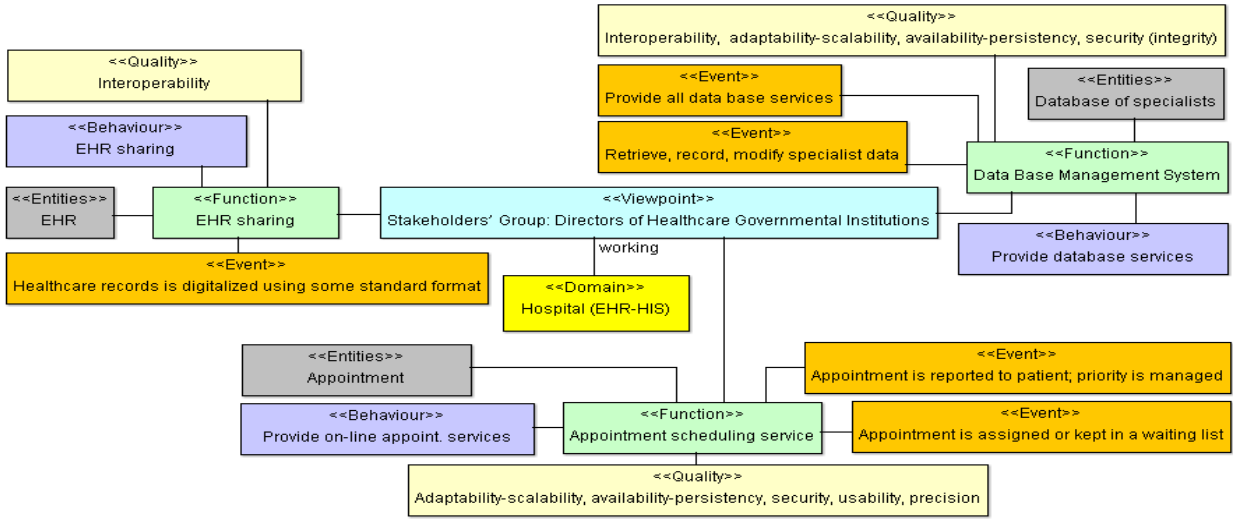


**Figure 7:** UML Specification by Intrinsics of Directors of Healthcare Governmental Institutions' Viewpoint for the Rules&Regulations Facet

## VI. Conclusion

It is known that general frameworks explain the "What to do" about things, but not the "How to do" to make things work. We presents the Domain Scoping sub-phase of the PL Scoping phase within the DE lifecycle, following the SPL Reference Model ISO/IEC 26550. The "How to do" is taken from Bjørner's domain modelling, which involves the facet notion and the stakeholders'viewpoint, focusing on business process modelling. Our main contribution is the specification of the Domain Scoping step as a systematic and repeatable process, centred on the early specification of quality properties as descriptors involved in all facets, considering their clear traceability, reflected into the BPMN representation; this issue will facilitate later on the reference architecture evolution. Notice also that the derivation of the business process specification in BPMN can be automatized from the UML representation of the facets; it is also a widely known and used notation, to bridge the gap between business processes and their implementation, for example as Web services [14]. It is claimed that the effort spent in PL Scoping will reduce the huge effort required in DRE and DD phases [21]. The Domain Scoping step of our PLScoP is crucial to construct a detailed Domain Model that will help to delimit clearly the SPL scope and functional and non functional granularity, to reduce the effort in the subsequent DRE and DD phases, where the RA is built. Our Domain Scoping process is applied to a complete case study in the Healthcare Information System domain to illustrate our approach. A more complete specification of PLScoP, and subsequent DRE and DD phases of the DE lifecycle to construct the SPL RA, are on-going works.

## Acknowledgment

We wish to thank the referees for their useful and pertinent comments. Financial support to this research has been provided by the DARGRAF No. 03-8730-2013-2 project of the CDCH (Consejo de Desarrollo Científico y Humanístico), and from the Postgraduate Studies in Computer Science, Faculty of Science, Universidad Central de Venezuela.

## References

[1] I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen, and J. Bettin, (Editors), *Domain Engineering. Product Lines, Languages and Conceptual Models*, Springer, 2013.

[2] E. Berard, *Essays in Object-Oriented Software Engineering*, New York: Prentice Hall, 1992.

[3] A. Helferich, G. Herzwurm, and S. Jesse, *Software Product Lines and Service-Oriented Architecture: A Systematic Comparison of Two Concepts*, in proceedings of the First Workshop on Service-Oriented Architectures and Software Product Lines (SOAPL 2007), Kyoto, Japan, May 2007.

[4] P. Istoan, G. Nain, G. Perrouin, and J. Jezequel, *Dynamic Software Product Lines for Service-Based Systems*, in proceedings of the 9th IEEE International Conference on Computer and Information Technology (CIT'09), INRIA, France, October 2009.

[5] P. Istoan, *Software Product Lines for Creating Service Oriented Applications*, Masters Internship at IRISA Rennes research institute, TRISKELL research team, Rennes, June 2009.

[6] F. Medeiros, E. de Almeida, and S. de Lemos, *Designing a Set of Service-oriented Systems as a Software Product Line*, in proceedings of the Software Components, Architectures and Reuse (SBCARS 2010),Fourth Brazilian Symposium, IEEE, Salvador, Bahia, Brazil, September 2010.

[7] A. Korff, *Implementing ISO 26550 Model-based*, in proceedings of the 6th edition of the International Academic-Industrial Conference in Complex Systems Design & Management (CSD&M 2015), Paris, France, November 2015.

[8] T. Käkölä, *Standards Initiatives for Software Product Line Engineering and Management within the International Organization for Standardization*, in proceedings of the 2010 43rd Hawaii International Conference on System Sciences (HICSS 2010), Koloa, Hawaii, USA, January 2010.

[9] K. Pohl, G. Bockle, and F. Van Der Linden, *Software Product Lines Engineering: Foundations, Principles, and Techniques*, Springer, 2005.

[10] Q. Munir and M. Shahid, *Software Product Line: Survey of Tools*, Master's thesis, Linköping University, Department of Computer and Information Science, 2010.

[11] ISO/IEC NP 26550, *Software and Systems Engineering – Reference Model for Software and Systems Product Lines*, ISO/IEC JTC1/SC7 WG4, 2013.

[12] F. Losavio, O. Ordaz, and V. Esteller, *Quality-based Bottom-up Design of Reference Architecture Applied to Healthcare Integrated Information Systems*,in proceedings of the 2015 IEEE 9th International Conference in Research Challenges in Information Science (RCIS 2015),Athens, Greece, May 2015.

[13] ISO/IEC 25010, *Systems and Software Engineering -- Systems and Software Quality Requirements and Evaluation (SQuaRE) -- System and Software Quality Models*, ISO/IEC JTC1/SC7/WG6, 2011.

[14] Object Management Group (OMG), *Business Process Model and Notation (BPMN)*, version 2.0, available in: http://www.omg.org/spec/ BPMN/2.0/, 2011.

[15] K. Lee, K. Kang, and J. Lee, *Concepts and Guidelines of Feature Modelingfor Product Line Software Engineering*,in proceedings of the 7thInternational Conference on Software Reuse (ICSR-7): Methods, Techniques, and Tools, Austin, TX, USA, April 2002.

[16] S. Apel, D. Batory, C. Kästner, and G. Saake, *Feature-Oriented Software Product Lines*, Springer, 2013.

[17] N. Siegmund, M. Rosenmuller, M. Kuhlemann, C. Kastner, S. Apel, and G. Saake, *SPL Conqueror: Towards Optimization of Non-functional Properties in Software Product Line*, Software Quality Journal, vol. 20, no. 3-4, pp. 487-517, September 2012.

[18] B. Kitchenham, *Guidelines for Performing Systematic Literature Reviews in Software Engineering*, Version 2.3, EBSE Technical Report, Software Engineering Group, School of Computer Science and Mathematics, Keele University, UK and Department of Computer Science, University of Durham, UK, 2007.

[19] M. de Moraes, E. de Almeida, and S. Romero, *Systematic Review on Software Product Lines Scoping*, in proceedings of 6th Experimental Software Engineering Latin American Workshop (ESELAW 2009), São Carlos, Brasil, November 2009.

[20] D. Bjørner, *Software Engineering 3 Domains, Requirements, and Software Design*, Texts in Theoretical Computer Science, EATCS Series, Editors: W. Brauer G. Rozenberg A. Salomaa, Springer-Verlag Berlin Heidelberg, 2006.

[21] I. Da Silva, P. Neto, P. O'Leary, E. De Almeida, and S. de Lemos Meira, *Software Product Line Scoping and Requirements Engineering in a Small and Medium-sized Enterprise: An Industrial Case Study*,Journal of Systems and Software, vol. 88, pp. 189-206, 2014.

[22] L. Northrop and P. Clements, *A Framework for Software Product Line Practice, Version 5.0*, Product Line Practice Initiative, SEI (Software Engineering Institute), 2012.

[23] J. Herrera, F. Losavio, and A. Matteo, *RDS de Enfoques y Técnicas para la Construcción de Arquitecturas en un Contexto de Líneas de Productos de Software*, Revista Venezolana de Computación (ReVeCom), vol. 1, no. 1, pp. 17-25, Junio 2014.

[24] J. Bayer, O. Flege, P. Knauber, R. Laqua, D. Muthig, K. Schmid, T. Widen, and J. M. DeBaud, *PuLSE: a Methodology to Develop Software Product Lines*, in proceedings of the 1999 Symposium on Software Reusability (SSR'99), ACM, Los Angeles, USA, May 1999.

[25] F. Losavio, O. Ordaz, and I. Santos, *Proceso de Análisis del Dominio Ágil de Sistemas Integrados de Salud en un Contexto Venezolano*, Revista Venezolana de Información, Tecnología y Conocimiento, ENL@CE, vol. 12, no. 1, pp.101-134 , Enero-Abril 2015.

[26] J. C. Herrera, F. Losavio, and O. Ordaz, *Ingeniería del Dominio con el Estándar ISO/IEC 26550 para LPS Considerando la Faceta Calidad*, in proceedings of the Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa 2015), Valencia, Venezuela, Octubre 2015.

[27] J. M. DeBaud, *A Systematic Approach to Derive the Scope of Software Product Lines*, in proceedings of the 21st International Conference on Software Engineering (ICSE'99), Los Angeles, CA, USA, May 1999.

[28] K. Schmid, *Customizing the PuLSE Product Line Approach to the Demands of an Organization*, in proceedings of the 7th European Workshop on Software Process Technology (EWSPT'2000), Kaprun, Austria, Febrery 2000.

[29] P. Kruchten, *Architectural Blueprints—The "4+1" View Model of Software Architecture*, in proceedings of the Conference on TRI-Ada'95,Anaheim, CA, USA, November 1995.

[30] Object Management Group (OMG), *Unified Modelling Language Superstructure*, version 2.0 (formal/05-07-04), http://www.omg.org/ spec/UML/2.0, August 2005.

[31] M. Shaw and D. Garlan, *Software Architecture. Perspectives of an Emerging Discipline*, Prentice-Hall, 1996.

[32] Institute of Medicine (US), *Key Capabilities of an Electronic Health Record System: Letter report*, Committee on Data Standards for Patient Safety, National Academies Press, 2003.

[33] HIQA (Health Information and Quality Authority), *Overview of Healthcare Interoperability Standards*, 2013.

[34] S. Samilovich, *OpenEMR – Historia Clínica Electrónica de Código Abierto y Distribución Gratuita, Apta para su Uso en el Sistema de Salud Argentina*, JAIIO CASI, 2010.

[35] M. Allauddin, F. Azam, and M. Zia, *A Survey of Quality Assurance Frameworks for Service Oriented Systems*, International Journal of Advancements in Technology, vol. 2, no. 2, pp. 188-198, 2011.

[36] W. Suryn, *Software Quality Engineering: a Practitioner's Approach*, IEEE Wiley & Sons, Inc., Hoboken, New Jersey, 2014.

[37] H. González, *Integration of Quality Attributes in Software Product Line Development*, Master Thesis en Ingeniería del Software, MétodosFormales y Sistemas de Información (ISMFSI), 2012.

[38] S. Wagner, *Software Product Quality Control*, Springer, 2013.

[39] K. I. Farroñay and A. J. Trujillo, *Sistema de Registro de Atención Médica para un Centro de Salud de Nivel I-3 de Complejidad*, Doctoral dissertation, Universidad Peruana de Ciencias Aplicadas (UPC), 2013.

**Revista Venezolana de Computación**
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 3, No. 1, pp. 51-56, Junio 2016
Selección de los Mejores Artículos del SCTC 2016

# Comparación Cualitativa del Desempeño de la Aplicación del Control Predictivo Basado en Modelo (CPBM) y el PID para el Control de Nivel en Pozos

Egner Aceros[1], Edgar Camargo[1], Osmer Parabavire[2]
acerose@pdvsa.com, camargoea@pdvsa.com, parabavireo@pdvsa.com

[1] LSAI, AIT, Maracaibo, Venezuela
[2] LSAI, AIT, Puerto La Cruz, Venezuela

**Resumen:** En este trabajo se realiza una comparación cualitativa general del desempeño de los controladores PID (Proporcional, Integral y Derivativo) y MPC (Model Predictive Control) en Procesos Industriales Petroleros. Tales técnicas de control son evaluadas para pozos de producción de agua. Así, se desarrolló un modelo matemático basado en las leyes físicas presentes en el proceso de extracción de agua, el cual es simulado con datos operacionales. El objetivo es controlar el nivel de fluido en el pozo a través del sistema de bombeo en el fondo del pozo, tomando en cuenta la completación mecánica y el potencial del reservorio, como también criterios relacionados al desempeño de los controladores.

**Palabras Clave:** Control Predictivo Basado en Modelo; Control de Nivel; Pozo.

**Abstract:** In this paper we made a qualitative comparison between the performance of PID controllers (Proportional, Integral and Derivative) and MPC (Model Predictive Control) in Industrial Oil Process. Such control techniques are evaluated for water production wells. Thus, a mathematical model based on physical laws presents in the water extraction process, which is simulated with operational data was developed. The aim is to control the level of fluid in the well through the downhole pump, taking into account the potential of the reservoir and mechanical completion, as well as criteria related to the performance of drivers.

**Keywords:** Model Predictive Control; Level Control; Well.

## I. Introducción

Actualmente, para el control de mayoría de los procesos en la industria son utilizados los clásicos algoritmos PID (Proporcional, Integral y Derivativo) y sus variantes, los cuales, por su simplicidad y estabilidad han sido la solución más implementada en los últimos 50 años [1]. En esta propuesta se presenta una alternativa al controlador PID, se utiliza el control predictivo basado en modelo (MPC por sus siglas en inglés – Model Predictive Control), para el control del nivel de un pozo. El criterio a optimizar, o función objetivo del control bajo estudio está relacionado con el comportamiento futuro del sistema, que se predice gracias a un modelo dinámico del mismo, denominado modelo de predicción (de ahí el término predictivo basado en modelo), en este caso específico, el modelo espacio-estado. Dicho modelo tiene su origen en las leyes de la física aplicadas a un pozo que posee en su completación mecánica un revestidor, una tubería interna para la extracción del líquido. El revestidor se encuentra completamente cerrado en la parte superior, el aire contenido en este espacio se mantiene constante y no es venteado hacia la atmósfera. La variable a controlar es el nivel de sumergencia de la bomba (salida "y"), la variable a manipular es el caudal de la bomba (entrada "u"), lo cual indica en este caso se implementó un modelo SISO y los parámetros de configuración del controlador son el horizonte de predicción (Np), el horizonte de control (Nc) y el peso de la variable manipulada (rw) la cual es inversamente proporcional a su magnitud, según el modelo espacio-estado propuesto por Wang [2].

## II. Bases Teóricas Control Predictivo Basado en Modelo

Control Predictivo Basado en Modelo o por sus siglas en inglés MPC (Model Predictive Control), su objetivo consiste en calcular valores futuros de la señal de control que minimicen valores futuros del error. La minimización se hace dentro de una ventana limitada de tiempo, a partir de información del sistema al inicio de dicha ventana. La estrategia de control se diseña en base a un modelo matemático de la planta [3]. En esta propuesta estaremos utilizando el modelo de espacio-

estado propuesto por Wang [2], para un proceso SISO, y su representación está dada por:

$$\boldsymbol{x}_m(k+1) = \boldsymbol{A}_m\boldsymbol{x}_m(k) + \boldsymbol{B}_m u(k) \quad (1)$$

$$y(k) = \boldsymbol{C}_m\boldsymbol{x}_m(k) + \boldsymbol{D}_m u(k) \quad (2)$$

Donde, el subíndice "m" hace referencia al modelo de la planta, u(k) es la entrada al modelo de la planta, y(k) es la salida de la planta y $x_m(k)$ es el vector de estado de la planta, todos en el instante k.

Para los modelo discretos, la entrada u(k) no afecta a la salida y(k), es decir $\boldsymbol{Dm} = \boldsymbol{0}$, por lo tanto (2) queda:

$$y(k) = \boldsymbol{C}_m\boldsymbol{x}_m(k) \quad (3)$$

El MPC incluye un integrador que corrige el error en estado estacionario, dicho el procedimiento está claramente explicado en [3]. Aquí sólo se mostrarán algunas ecuaciones y los procedimientos principales.

Restando (1) y su equivalente para $\mathbf{x_m}$ (k), se obtiene:

$$\Delta\boldsymbol{x}_m(k+1) = \boldsymbol{A}_m\Delta\boldsymbol{x}_m(k) + \boldsymbol{B}_m\Delta u(k) \quad (4)$$

Donde:

$$\Delta\boldsymbol{x}_m(k+1) = \boldsymbol{x}_m(k+1) - \boldsymbol{x}_m(k) \quad (5)$$

Para $\Delta\boldsymbol{x}_m(k)$, $\Delta u(k)$ se aplica el procedimiento similar al aplicado a $\Delta\boldsymbol{x}_m(k+1)$ en (5).

Similarmente a partir de (3) se tiene:

$$\begin{aligned} y(k+1) - y(k) &= \boldsymbol{C}_m[\boldsymbol{x}_m(k+1) - \boldsymbol{x}_m(k)\,] \\ &= \boldsymbol{C}_m\Delta\boldsymbol{x}_m(k+1) \end{aligned} \quad (6)$$

Sustituyendo (4) en (6) se obtiene:

$$y(k+1) = \boldsymbol{C}_m\boldsymbol{A}_m\Delta\boldsymbol{x}_m(k) + \boldsymbol{C}_m\boldsymbol{B}_m\Delta u(k) + y(k) \quad (7)$$

Definiendo ahora el estado aumentado, según se explica en [3]:

$$\boldsymbol{x}(k) = [\Delta\boldsymbol{x}_m^T(k) \quad y(k)]^T \quad (8)$$

A partir de (4) y (7) se obtiene el modelo espacio-estado aumentado:

$$\overbrace{\begin{bmatrix}\Delta\boldsymbol{x}_m(k+1)\\ y(k+1)\end{bmatrix}}^{\boldsymbol{x}(k+1)} = \overbrace{\begin{bmatrix}\boldsymbol{A}_m & \boldsymbol{0}_m^T\\ \boldsymbol{C}_m\boldsymbol{A}_m & 1\end{bmatrix}}^{\boldsymbol{A}}\overbrace{\begin{bmatrix}\Delta\boldsymbol{x}_m(k)\\ y(k)\end{bmatrix}}^{\boldsymbol{x}(k)} + \overbrace{\begin{bmatrix}B_m\\ C_mB_m\end{bmatrix}}^{\boldsymbol{B}}\Delta u(k) \quad (9)$$

$$y(k) = \overbrace{[\boldsymbol{0_m} \quad \boldsymbol{1}]}^{\boldsymbol{C}}\overbrace{\begin{bmatrix}\Delta\boldsymbol{x}_m(k)\\ y(k)\end{bmatrix}}^{\boldsymbol{x}(k)} \quad (10)$$

De forma matricial compacta:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\Delta u(k) \quad (11)$$

$$y(k) = \boldsymbol{C}\boldsymbol{x}(k) \quad (12)$$

En la estrategia MPC se calculan los $N_c$ valores futuros de la señal de control

$$\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+Nc-1) \quad (13)$$

Estos valores son los que minimizan la diferencia entre el valor de referencia y el valor pronosticado de la salida, a lo largo de un horizonte de predicción (ventana de optimización) de $N_p$ valores futuros.

Para resolver esto es necesario pronosticar $N_p$ estados futuros, a partir de (11)

$$\mathbf{x}(k+1|k), \mathbf{x}(k+2|k),\dots, \mathbf{x}(k+m|k),\dots, \mathbf{x}(k+Np|k) \quad (14)$$

donde x(k+m|k) denota el pronóstico del estado en k+m, dada la información de estado actual x(k).

En consecuencia los $N_p$ valores pronosticados de la salida, utilizando (12) corresponden a:

$$y(k+1|k), y(k+2|k),\dots, y(k+m|k),\dots, y(k+Np|k) \quad (15)$$

Desarrollando, agrupando y ordenando, llegamos a la ecuación en forma matricial compacta:

$$\boldsymbol{Y} = \boldsymbol{F}\boldsymbol{x}(k) + \emptyset\Delta\boldsymbol{U} \quad (16)$$

Donde para el sistema SISO:

$$\boldsymbol{Y} = \begin{bmatrix} y(k+1|k) & y(k+2|k) & \dots & y(k+N_p|k)\end{bmatrix}^T \quad (17)$$

$$\boldsymbol{F} = \begin{bmatrix}\boldsymbol{CA}\\ \boldsymbol{CA^2}\\ \vdots\\ \boldsymbol{CA^{N_p}}\end{bmatrix} \quad (18)$$

$$\emptyset = \begin{bmatrix}\boldsymbol{CB} & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & \boldsymbol{0}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{CA^{N_p-1}B} & \boldsymbol{CA^{N_p-2}B} & \cdots & \boldsymbol{CA^{N_p-N_c}B}\end{bmatrix} \quad (19)$$

$$\Delta\boldsymbol{U} = [\Delta u(k) \quad \Delta u(k+1) \quad \dots \quad \Delta u(k+\boldsymbol{N_c}-\boldsymbol{1})]^T \quad (20)$$

El valor de referencia r(k) se mantiene constante a lo largo de la ventana de optimización (de longitud $\mathbf{N_p}$), y la función objetivo a minimizar para calcular los $\mathbf{N_c}$ valores futuros de la señal de control es:

$$\boldsymbol{J} = (\boldsymbol{R_S} - \boldsymbol{Y})^{\boldsymbol{T}}(\boldsymbol{R_S} - \boldsymbol{Y}) + \Delta\boldsymbol{U}^{\boldsymbol{T}}\bar{\boldsymbol{R}}\Delta\boldsymbol{U} \quad (21)$$

Donde:

$$\boldsymbol{R_S^T} = \overbrace{[1 \quad 1 \quad \cdots \quad 1]}^{\boldsymbol{N_p^T}} r(k) = \bar{\boldsymbol{R}}_S r(k) \quad (22)$$

$$\bar{\boldsymbol{R}} = r_w \boldsymbol{I}_{\boldsymbol{N_c x N_c}} \quad (23)$$

$r_w$ es un parámetro de configuración del MPC, constante y es una medida inversamente proporcional a la magnitud de la variable manipulada.

De (21) la condición necesaria para que ocurra el mínimo de J con respecto a $\Delta\boldsymbol{U}$ es que

$$\frac{\partial J}{\partial \Delta\boldsymbol{U}} = 0 \quad (24)$$

Sustituyendo (16) y (22) en (21), derivando, reordenando, simplificando y despejando ΔU que minimiza a J, nos queda:

$$\Delta \boldsymbol{U} = (\emptyset^{\boldsymbol{T}}\emptyset + \bar{\boldsymbol{R}})^{-1}\emptyset^{\boldsymbol{T}}[\bar{\boldsymbol{R}}_S r(k) - \boldsymbol{F}\boldsymbol{x}(k)] \quad (25)$$

A pesar que $\Delta \boldsymbol{U}$ contiene las señales $\Delta u(k), \Delta u(k+1), \ldots, \Delta u(k+Nc-1)$, sólo se implementa el primer valor de esta secuencia, es decir, $\Delta u(k)$, y se ignoran los demás, por el principio del horizonte que se aleja.

Al siguiente instante de muestreo se "mide" **x**(k+1) y se repite el proceso para calcular la nueva secuencia de señales de control, de la cual se toma sólo el primer valor y así en cada nueva medición, según se explica en [3].

Entonces:

$$\Delta u(k) = \overbrace{[1 \quad 0 \quad \cdots \quad 0]}^{Nc}(\emptyset^{\boldsymbol{T}}\emptyset + \bar{\boldsymbol{R}})^{-1}\emptyset^{\boldsymbol{T}}[\bar{\boldsymbol{R}}_S r(k) - \boldsymbol{F}\boldsymbol{x}(k)] \quad (26)$$

Que puede reescribirse como:

$$\Delta u(k) = K_y r(k) - \boldsymbol{K}_{mpc}\,\boldsymbol{x}(k) \quad (27)$$

Donde $K_y$ es el primer elemento de $(\emptyset^{\boldsymbol{T}}\emptyset + \bar{\boldsymbol{R}})^{-1}\emptyset^{\boldsymbol{T}}\bar{\boldsymbol{R}}_S$ y $\boldsymbol{K}_{mpc}$ es la primera fila de $(\emptyset^{\boldsymbol{T}}\emptyset + \bar{\boldsymbol{R}})^{-1}\emptyset^{\boldsymbol{T}}\boldsymbol{F}$ . Observe que $\Delta u(k)$ tiene la forma estándar de una estrategia de control por retroalimentación de estado para un sistema LTI y la ganancia de retroalimentación de estado es $\boldsymbol{K}_{mpc}$.

El sistema a lazo cerrado se obtiene sustituyendo (27) en (11), y agrupando términos, se tiene

$$\boldsymbol{x}(k+1) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{mpc})\boldsymbol{x}(k) + \boldsymbol{B}K_y r(k) \quad (28)$$

Por lo tanto, los polos del sistema a lazo cerrado son las raíces de:

$$\left|z\boldsymbol{I} - (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{mpc})\right| = \boldsymbol{0} \quad (29)$$

Observando (8), (9), (10), (18) y (26), podemos expresar a $\boldsymbol{K}_{mpc}$ como:

$$\boldsymbol{K}_{mpc} = [\boldsymbol{K}_x \quad K_y] \quad (30)$$

Donde $\boldsymbol{K}_x$ es el vector ganancia de realimentación que multiplica a $\Delta \boldsymbol{x}_m(k)$ y $K_y$ es la ganancia de realimentación que multiplica a $y(k)$. Entonces (27), puede reescribirse como:

$$\Delta u(k) = K_y r(k) - [\boldsymbol{K}_x \quad K_y]\begin{bmatrix}\Delta \boldsymbol{x}_m(k) \\ y(k)\end{bmatrix} \quad (31)$$

Que es equivalente a:

$$\Delta u(k) = -\boldsymbol{K}_x\,[\boldsymbol{x}_m(k) - \boldsymbol{x}_m(k-1)] + K_y e(k) \quad (32)$$

Donde $e(k) = r(k) - y(k)$, corresponde al error. Luego del procedimiento similar aplicado en (5) a $\Delta u(k)$ y cambiando por el operador $q^{-1}$ de desplazamiento, se tiene:

$$u(k) = u(k-1) + \Delta u(k) \rightarrow u(k)(1 - q^{-1}) = \Delta u(k) \quad (33)$$

Expresando de la siguiente forma:

$$\frac{u(k)}{\Delta u(k)} = \frac{1}{(1-q^{-1})} \quad (34)$$

Representación en diagrama de bloques en la Figura 1 del modelo espacio-estado del control MPC; $q^{-1}$ denota el operador de desplazamiento hacia atrás, y $\frac{1}{(1-q^{-1})}$ es el integrador discreto, ver [3].

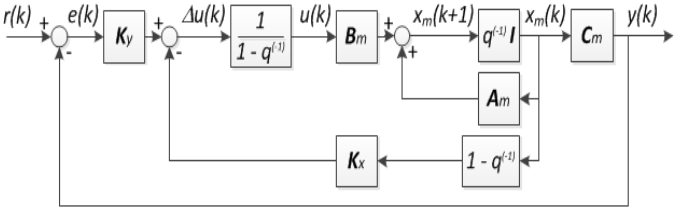


**Figura 1:** Diagrama de Bloques Modelo Espacio-Estado MPC

## III. Modelo del Pozo

El modelo utilizado corresponde a un pozo productor de agua con revestidor y tubo concéntrico interno de producción por el cual sale el líquido, la masa gaseosa contenida en el revestidor corresponde al aire que quedó presurizado dentro del espacio anular formado por ambas tuberías el cuál ingresó de la atmosfera antes del cañoneo en el fondo del pozo, tal y como se muestra en la Figura 2, el volumen de control (VC) bajo estudio se presenta en la Figura 3, al cual aplica el siguiente balance de masa:

$$\dot{M}_e(t) - \dot{M}_s(t) = \frac{d(M_T(t))}{dx} \quad (35)$$

Donde $\dot{M}_e(t)$ es el flujo másico de entrada, $\dot{M}_s(t)$ es el flujo másico de salida y $M_T(t)$ es la masa total acumulada dentro del VC.

$$M_T(t) = M_G(t) + M_L(t) \quad (36)$$

Con $M_G(t)$ la cantidad de masa de gas y $M_L(t)$ la cantidad de masa de líquido dentro del volumen de control (VC).

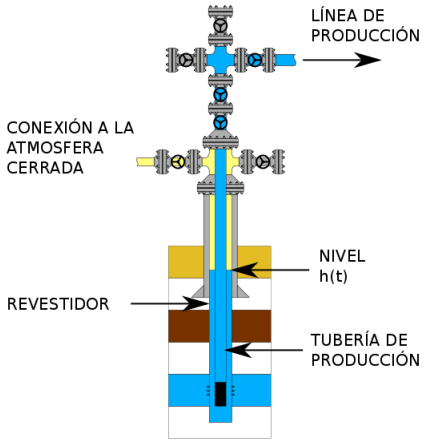


**Figura 2:** Completación del Pozo

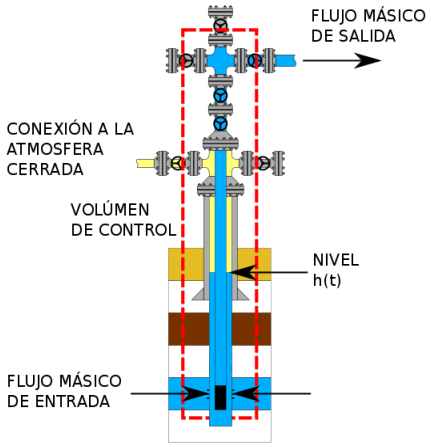


**Figura 3:** Volumen de Control (VC) Líneas Punteadas

Asumiendo que la entrada de líquido al pozo responde a un modelo de flujo en "tubería"

$$Q_e(t) = \frac{Pws - Pwf}{Re} \quad (37)$$

Donde, *Pws* es la presión del yacimiento, *Pwf* es la presión de fondo fluyente en la boca del pozo, *Re* es la resistencia hidráulica al paso del fluido por la "tubería" y $Q_e(t)$ es el flujo de entrada.

Utilizando la ecuación de estado de los gases ideales (aire) podemos definir la siguiente ecuación:

$$\dot{y}(t) + \frac{\rho_L . g}{A.Re} y(t) + \frac{Pg1.Vg1}{A^2.Re(ht - y(t))} = \frac{Pws}{A.Re} - \frac{u(t)}{A} \quad (38)$$

Donde *y(t)* (salida y variable a controlar) corresponde a la altura del líquido en el pozo, $\rho_L$ es la densidad del líquido (agua), $g$ es la aceleración de la gravedad, $A$ es al área de la sección transversal del espacio ocupado por el líquido en el revestidor, *Re* es la resistencia al paso del fluido por la "tubería", *Pg1* es la presión del gas (aire) en el estado inicial 1 y *Vg1* es el volumen ocupado por el gas (aire) en el estado inicial 1, *ht* es la altura total del pozo, desde el tope en superficie hasta el fondo donde están las perforaciones, *Pws* es la presión del yacimiento, *u(t)* (entrada y variable a manipular) corresponde al flujo de la bomba.

## IV. Diseño de la Estrategia de Control

### A. *Controlador PID*

Para el controlador PID, se utilizó el método de aproximación a un sistema de primer orden con tiempo muerto [4]:

$$\frac{y(t)}{u(t)} = \frac{K}{\tau s + 1} e^{-t_0 s} \quad (39)$$

Donde, $K$ es la ganancia, $\tau$ es la constante de tiempo $t_0$ es el retardo del sistema.

En base a las curvas mostradas en la Figura 4, correspondientes a la altura del líquido (salida "y") ante una entrada escalón en la tasa de crudo (entrada "u") se obtiene la curva de reacción de la Figura 4.

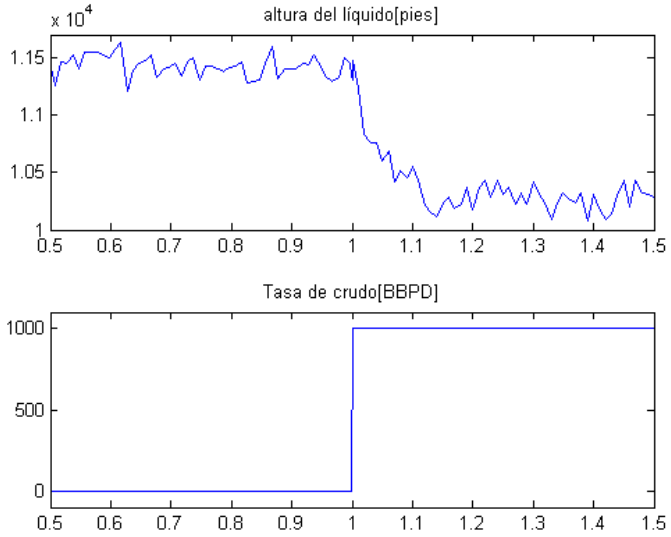


**Figura 4:** Curva de Reacción

Los valores de los parámetros anteriores corresponden a:

$K$ = -2,2242 [m/( ($m^3$)/día )]
$\tau$ = 0,043428 [día]
$t_0$ = 3,4722x$10^{-3}$ [día]

Utilizando un controlador PI (Proporcional Integrativo) por medio de la Integral del Tiempo Acumulada del Error, para el seguimiento al punto de ajuste o set point (ITAE-SP) según el procedimiento utilizado en [4], los valores del controlador PI corresponden a los siguientes:

Ti = 3.2922x$10^{-6}$ (día)
P = -8.5060x$10^{-6}$ (($m^3$)/día ))/m
I = -2.5837 $m^2$

### B. *Controlador MPC*

*1)* Identificación: Se utilizaron como datos de entrada el conjunto de valores generados con el modelo indicado en en la Ecuación (38) para distintos rangos de trabajo de las señales de entrada-salida, con el cual se obtiene un modelo paramétrico ARMAX, en nuestro caso se utilizaron dos (02) valores anteriores para la entrada, la salida y el error.

*2) Ciclos de Control:* En este punto ya se tiene el modelo espacio-estado no adaptativo, es decir, que no varía, con el cual se estará implementando el control MPC, seguidamente se crea el modelo espacio-estado aumentado según (9) y (10), para el control se ejecuta la variación de los 3 parámetros: "Horizonte de Predicción" ($N_p$ = 10, 15 y 20), "Horizonte de Control" ($N_c$ = 5, 10 y 20) y "Peso de la magnitud de la señal de control" (rw = 100, 300 y 900). En cada ciclo se varía un parámetro y los otros dos quedan fijos, para un total de nueve (09) ciclos.

## V. Experimentación con el Sistema de Control

A continuación se presentan nueve (09) ciclos de variación de los parámetros del MPC agrupados en tres conjuntos, uno por cada tres cambios en los parámetros $N_p$, $N_c$ y $r_w$, más el control PID. La secuencia de la entrada de control (Set-Point) y perturbaciones para la comparación del desempeño de los controladores es la siguiente: El transitorio inicial desde el nivel cero (0) en el tiempo igual cero (t=0), luego en el tiempo 2,5 días se agrega una perturbación, luego en el tiempo igual a 5 días (t=5) se cambia la entrada de control (Set-Point) desde 8000 a 4000 pies, y luego en el tiempo igual a 7,5 días se resta la perturbación incluida inicialmente.

Primer conjunto de variaciones, parámetro $N_p$ con valores de 10, 15 y 20, las salidas se muestran en la Figura 5 para los MPCs, PID y Set-Point; las entradas se muestran en la Figura 6 de los MPCs y PID. La salida "y" en el PID posee un pico en el transitorio inicial, y luego de 0,5 días logra alcanzar su punto de equilibrio, manteniendo un comportamiento similar al resto de controladores. La salida "y" en el MPC alcanza el valor de referencia o set-point de una forma rápida cuando el parámetro $N_p$ es igual a 10. Se logra observar en este caso que el controlador con $N_p$ igual a 10 es el que muestra el mejor desempeño de las cuatro versiones de los controladores. Las entradas "u" del MPC poseen un comportamiento suave y alcanza su punto de estabilidad a los 0,2 días para los valores $N_p$ 15 y 10, para el valor $N_p$ 20 presenta un pico en la entrada y el PID es el más lento de los cuatro ejemplos de controladores durante el transitorio, para el estado estacionario los cuatro controladores poseen desempeños similares.

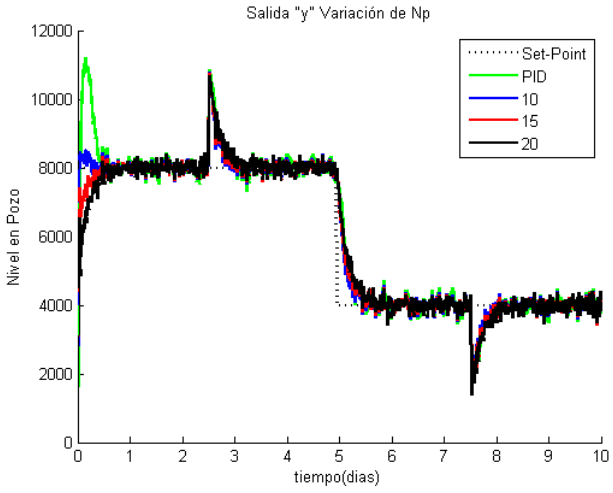


**Figura 5:** Salida "y" ante la Variación de $N_p$

Segundo conjunto de variaciones, parámetro $N_c$ con valores de 5, 10 y 20, las salidas se muestran en la Figura 7 para los MPCs, PID y Set-Point; las entradas se muestran en la Figura 8 de los MPCs y PID. La salida "y" en el PID posee un pico en el transitorio inicial, y luego de 0,5 días logra alcanzar su punto de equilibrio, manteniendo un comportamiento similar al resto de controladores. La salida "y" en el MPC alcanza el valor de referencia o Set-Point al mismo tiempo para cualquier valor del parámetro $N_c$. Se logra observar en este caso que el controlador MPC posee un comportamiento similar, sin embargo, mejor que el PID, porque no presenta un pico al inicio. La entrada "u" en el MPC posee sobre-picos en el transitorio, siendo los picos mayores para valores bajos del parámetro $N_c$ y alcanza su punto de estabilidad de forma similar al PID, es decir, a los 0,5 días, en el estado estacionario las cuatro entradas poseen comportamiento similar.

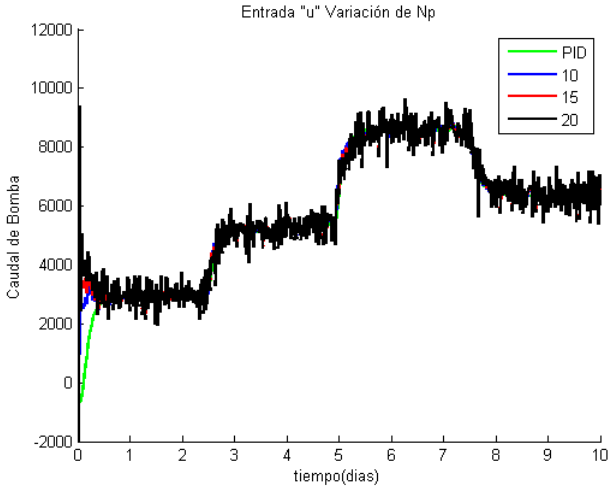


**Figura 6:** Entrada "u" ante la Variación de $N_p$

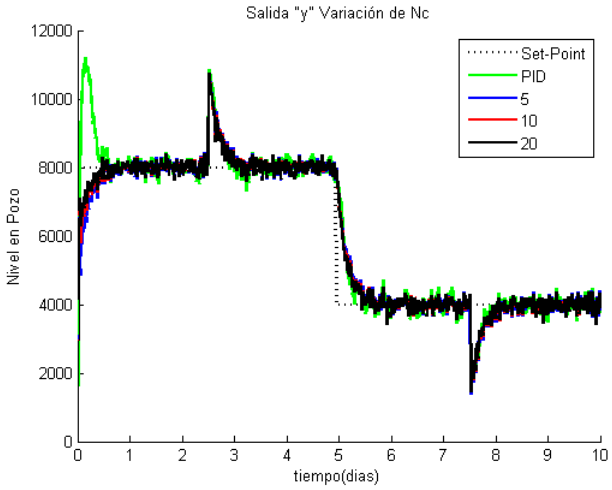


**Figura 7:** Salida "y" ante la Variación de $N_c$

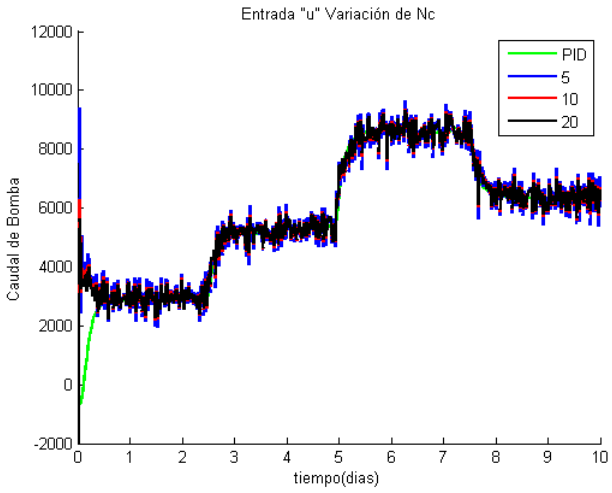


**Figura 8:** Entrada "u" ante la Variación de $N_c$

Tercer conjunto de variaciones, parámetro $r_w$ con valores de 100, 300 y 900, las salidas se muestran en la Figura 9 para los MPCs, PID y Set-Point; las entradas se muestran en la Figura 10 de los MPCs y PID. La salida "y" en el PID posee un pico en el transitorio inicial, y luego de 0,5 días logra alcanzar su punto de equilibrio, manteniendo un comportamiento similar al resto de controladores. La salida "y" en el MPC alcanza el valor de referencia o set-point al mismo tiempo para cualquier valor del parámetro rw. Se logra observar en este caso que el controlador MPC posee un comportamiento similar, sin embargo, mejor que el PID, porque no presenta un pico al inicio.

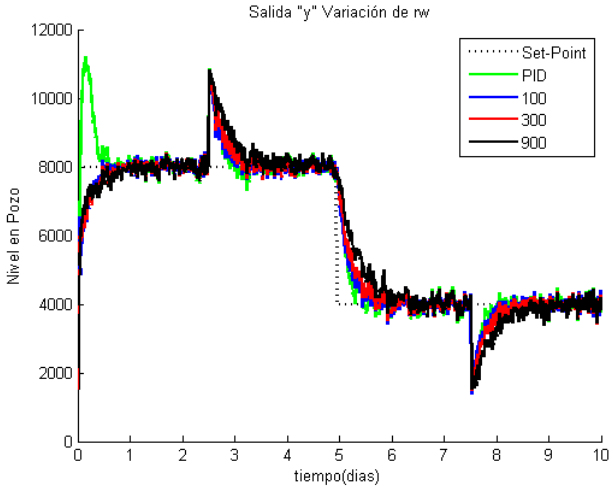


**Figura 9:** Salida "y" ante la Variación de $r_w$

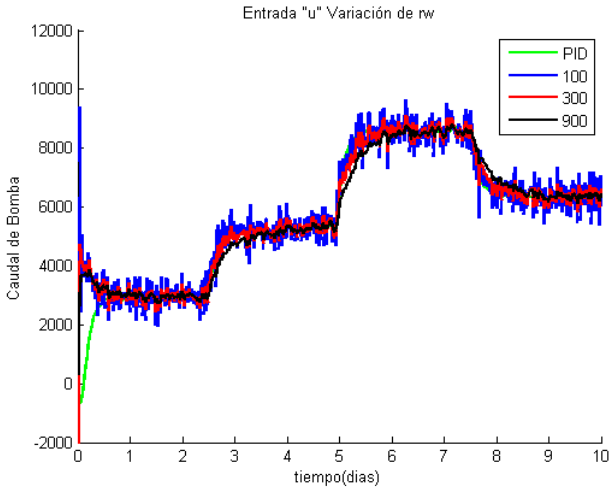


**Figura 10:** Entrada "u" ante la Variación de $r_w$

La entrada "u" en el MPC posee sobre-picos en el transitorio, siendo los picos mayores para valores bajos del parámetro rw y alcanza su punto de estabilidad de forma similar al PID, es decir, a los 0.5 días, en el estado estacionario el controlador PID muestra mejor desempeño, aunque no de forma muy determinante, por otro lado, se nota que se amplifica el ruido en la señal de entrada "u" y la rapidez para alcanzar su punto de equilibrio es mejor para valores bajos del parámetro $r_w$.

## VI. Conclusiones

Para el controlador MPC sobre el caso de la variación del parámetro $N_p$, el mejor desempeño fue con valores bajos, es decir, $N_p = 10$. Sobre el caso de la variación del parámetro $N_c$ el nivel (salida "y") posee un comportamiento similar, sin embargo el mejor desempeño fue con la manipulación del caudal (entrada "u") para valores altos, ya que ésta presenta menor sobre-pico, es decir, $N_c = 20$. Sobre el caso de la variación del parámetro $r_w$, el mejor desempeño fue con valores bajos, ya que ésta presenta menor tiempo para alcanzar el Set-Point, es decir, $r_w = 100$.

Del conjunto de variaciones de los parámetros del MPC y comparación del PID, se puede demostrar la razón por la cual el controlador PID aún sigue siendo el controlador más implementado en mayoría de los procesos en la industria, por su simplicidad y estabilidad.

## Referencias

[1] K. Astrom, T. Hagglund, *PID Controllers: Theory, Desing, and Tuning*; Instrument Society of America: Researd Triangle Park, 1995.

[2] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*, Springer-Verlag London Limited, 2009.

[3] J. Canelón, *Tópicos Avanzados en Identificación de Procesos*, Universidad del Zulia, Facultad de Ingeniería, División de Postgrado, 2014.

[4] E. Aceros, E. Camargo, J. Aguilar. *Intelligent Well Systems*, in proceedings of the Asia-Pacific Conference on Computer Aided System Engineering (APCASE), IEEE Computer Society, Quito, Ecuador, July 2015.

# Índice de Autores

# REVECOM

## Sociedad Venezolana de Computación

La Sociedad Venezolana de Computación está comprometida con el impulso de una nueva generación académica y profesional en nuestra área de saber para el desarrollo del país.

Los conceptos y puntos de vista expresados en los trabajos publicados en este libro representan las opiniones personales de los autores y no reflejan el juicio de los editores o de la Sociedad Venezolana de Computación.

ISSN: 2244-7040

9 772244 704006

www.svc.net.ve/revecom