

Línea de Producto de Software para Aplicaciones Interactivas de TV Digital

Mirtha Fabiana Miranda, Sandra Casas

Instituto de Tecnología Aplicada - Universidad Nacional de la Patagonia Austral

Rio Gallegos, Argentina

mirfamir@gmail.com.ar, scasas@unpa.edu.ar

Abstract— La Televisión Digital interactiva (TVDi) es una tecnología de transmisión digital de contenidos televisivos que, implica grandes cambios; mejoras en la calidad de audio, video, software, datos y nuevas formas de acceso y de interacción con el televidente. El avance de la implementación de la TVDi refleja una notable ausencia de software y, necesidad de enfoques para el proceso de desarrollo en este dominio a nivel industrial. Por otro lado, se presenta el problema de usabilidad, para ello se estudian patrones de diseño de interacción centrados en el usuario. La implementación de una Línea de Productos de Software que permita generar producto para este dominio es una solución que este trabajo aborda. Tomando como estrategia un conjunto de patrones de diseño de interacción, se diseña el modelo de características, el cual es soportado por la herramienta SPL-TVDi, que ha sido construida como parte de este trabajo. Además se presenta un caso de estudio.

Palabras Claves: Interactividad, TV Digital, Línea de Producto de Software, Patrones de Diseño de Interacción, Modelo de Características.

1 INTRODUCTION

La Televisión Digital es una tecnología de transmisión digital de contenidos televisivos, que permite una óptima calidad del video, sonido y envío de software que puede ser ejecutado en el aparato receptor.

Una aplicación de TV Digital interactiva (TVDi) es un software multimedia a través de la cual el televidente puede interactuar vía control remoto [8]. Para la ejecución de una aplicación de TVDi se necesita un decodificador, denominado set top box (STB). Diversos países latinoamericanos han adoptado el estándar que usa el middleware Ginga [4] para la ejecución de las aplicaciones TVDi, codificadas en lenguaje de programación NCL (Nexted Context Lenguaje) y Lua [3]. El lenguaje NCL es básicamente un documento XML, que define claramente cómo los objetos media (elementos de contenido multimedial: vídeos, imágenes, sonidos, etc.) son estructurados y relacionados, en tiempo y en espacio.

Dos problemas que presenta el desarrollo de las aplicaciones de TVDi en Latinoamérica son, (i) los procesos de desarrollo aún son inmaduros para afrontar la construcción de aplicaciones TVDi a escala industrial, el desarrollo es esencialmente manual y artesanal [8]; (ii) falta de métodos eficaces para incorporar criterios de usabilidad en las aplicaciones TVDi desarrolladas, de manera generalizada [10]. Superar estas falencias reducirá costos, esfuerzos, tiempos en el desarrollo y mejorará la calidad de los productos finales.

Considerando el dominio de la TVDi y los desafíos planteados, se pretende aportar soluciones combinando

diversas estrategias y enfoques de reutilización del software. La integración de los conceptos Líneas de Productos de Software (SPL) [15] y patrones de diseño de interacción para TVDi [5] resultan ser una alternativa que este trabajo profundiza.

Se presenta una SPL para generar aplicaciones TVDi ejecutables en middleware Ginga, puesto que produce automáticamente código NCL. La SPL-TVDi se construyó a partir de un modelo de características genérico, basado en patrones [5] y lenguajes de diseño para la TVDi; considerando a los patrones de diseño como características, dado que aportan variabilidad y reusabilidad. Previamente se han expuesto resultados preliminares [8].

La organización del resto de este trabajo es como sigue. En la Sección 2 se describe brevemente las Líneas de Producto de Software, los Modelos de Características y patrones de diseño de interacción para aplicaciones de TVDi. En la Sección 3 se presenta el Modelo de Característica en base a los patrones de diseño. En la Sección 4 se explica la herramienta de software que implementa el Modelo de Características. En la Sección 5 se expone un caso de estudio sobre la generación de un producto desde la herramienta desarrollada. En la Sección 6 se discuten los trabajos relacionados. Finalmente, en la Sección 7 se presentan las conclusiones y trabajos futuros.

2 ESTRATEGIAS Y ENFOQUES

En esta Sección se presentan brevemente las principales es-

trategias y enfoques usados en la propuesta, Líneas de Producto de Software, Modelos de Característica y patrones de diseño de interacción para aplicaciones de TV Digital.

2.1 Línea de Producto de Software y Modelos de Características

Una SPL es un enfoque para desarrollar familias de sistemas basados en el uso de activos reutilizables como un medio para mejorar la calidad del software, reducir costos de producción y tiempo de salida al mercado [1]. Los productos de una SPL se especifican en términos de las diversas características (feature). Una característica se define como un incremento en la funcionalidad del producto, estas pueden ser características comunes y características variables. Un modelo de características (MC) es una representación compacta de todos los productos de una línea de productos en términos de características y relaciones entre ellos. Estos modelos especifican que elementos de la familia de productos son similares o variables a lo largo del ciclo de vida del desarrollo [6].

Por otra parte, se incorporan al MC un conjunto de reglas. Las reglas son expresiones lógicas formadas por características, conectivos lógicos y cuantificadores. La intención de usar una lógica de predicados para expresar reglas por defecto es de evitar las ambigüedades del lenguaje natural.

Varias herramientas [7] dan soporte al desarrollo de software orientado a características. Entre ellas, FeatureIDE; siendo éste, el framework seleccionado para el presente trabajo. La Figura 1 presenta un MC representado en la herramienta FeatureIDE. Los nodos del MC representan características y las líneas muestran las relaciones entre ellas. El nodo raíz denominado como A representa el concepto de dominio que se está modelando. Las características de un modelo se clasifican en obligatorias, opcionales y alternativas. Las características opcionales (Optional) como el nodo C representado con un círculo vacío, puede ser o no parte de un producto. Las características obligatorias (Mandatory), como B, D y, E, representados por un círculo relleno son parte de todos los productos que contienen la SPL. Las características alternativas pueden ser exclusiva (XOR) o no exclusiva (OR). XOR (Alternative) indica que sólo una sub-característica puede ser seleccionada F o G; OR (Or) permite la selección de más de una opción para un producto, H o I o los dos nodos H e I. Además, en el diagrama se representan a las características abstractas (Abstract) y concretas (Concrete). La característica A es abstracta, representa una interfaz; y las restantes como concretas, son las características que implementan las funcionalidades [8].

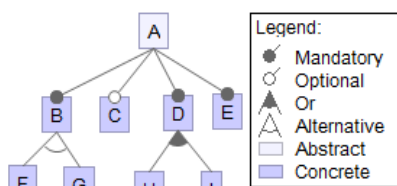


Fig. 1. Modelo de Características

2.2 Patrones de Diseño de interactividad para aplicaciones de TVDi

El concepto de patrón se aplica a estructuras de información que permiten resumir y comunicar la experiencia acumulada y la resolución de problemas, tanto en la práctica como en el diseño, en programas de enseñanza y aprendizaje. De esta forma un patrón puede entenderse como una plantilla, una guía, un conjunto de directrices o de normas de diseño. Un patrón presenta un problema y una solución [9].

El presente trabajo toma como punto de partida el concepto de patrones de diseño de interacción para las aplicaciones de TVDi propuestos por [5], que han sido ampliamente usados en aplicaciones TVDi que ofrece la Corporación Británica de Radiodifusión (BBC - British Broadcasting Corporation). Estos patrones de diseño se basan en el método de enfoque de grupo; este concepto permite el análisis de tareas y necesidades de los contenidos de los usuarios en aplicaciones de TVDi. Kunert [5] presenta 41 patrones de diseño, los cuales se agrupan en las categorías Page Layout, Navigation, Remote Control Keys, Basic Function, Content, Presentation y Help, Participation, Text Input, Accessibility & Personalisation y Specific User Groups. Este marco genérico de orientación integral para el diseño de interfaces de usuario de TVDi, proporciona por cada patrón una plantilla que indica su nombre, ejemplos de aplicación, contexto, problemas, soluciones, evidencias y patrones relacionados.

Los resultados de los grupos sólo se refieren a tipos específicos de contenido, y se clasifican en tres clases: tareas de usuarios genéricas de TVDi, requisitos de contenido general y en requisitos generales de usabilidad. Considerando el concepto de patrones, y concluyendo que cumplen con los niveles de usabilidad requeridos para la TVDi, por lo cual pueden ser usados de forma efectiva, eficiente, segura y satisfactoria, estos pueden ser resignificados como características, aportando variabilidad y reusabilidad.

3 MODELO DE CARACTERÍSTICAS PARA APLICACIONES TVDI

En la fase ingeniería de dominio se analiza el área de aplicación en la que se desea crear la línea de productos, se define su alcance, se identifican los elementos comunes y variables que contendrá la línea de productos, y se diseñan y construyen los artefactos reutilizables que soportan la línea de productos. El resultado es el modelado, a partir del Modelo de Características [1].

La estrategia aplicada es definir la aplicación TVDi como raíz del diagrama, en el segundo nivel, los grupos de patrones constituyen características, siendo también abstractas. Luego a partir del tercer nivel, cada patrón se representa como una característica concreta. Para determinar los tipos de características y las relaciones entre las mismas, se utilizó la definición de los patrones, pero además se analizó manualmente 32 aplicaciones TVDi en las que se aplicaron los patrones de diseño de interacción. El primer MC se expuso en trabajos anteriores [8], dicho estudio detallado se omite en este artículo por cuestiones de extensión

y en consideración que ya ha sido presentado.

En esta primera etapa se pretende generar aplicaciones de interactividad local; se abarcan aquellos patrones relacionados con la presentación y selección de información, que engloba los grupos de PageLayout, Navigation, BasicFuntion, Content. Los patrones de PageLayout son Overlay, FullWithVideo, FullWhithoutVideo, entre otros. Los patrones de Navigation son Menu, MultiScreen, Tabs. Los patrones de BasicFuntions son InitialCall, Starting, Loading, Existing. Los patrones de Content son Presentation y TextDesing. En el MC existen los 12 patrones mencionados, que corresponden a 156 características, de los cuales 105 son características concretas. Para el grupo PageLayout se requieren de 33 características, Navigation 38 características, BasicFuntions 62 características y, para Content 21 características.

El conjunto de reglas aplicadas en el MC permite que se produzcan aplicaciones con usabilidad. Las aplicaciones resultantes presentan usabilidad por estar creadas en base a los patrones de kunert, por el cual demuestra que cumplen con esta propiedad. Para el diseño del MC se definen un conjunto de 42 reglas en total: a) reglas para controlar determinadas características del mismo grupo y, b) reglas para restringir características de distintos grupos de patrones.

A continuación se presentan reglas ya definidas:

R#1: TopVideo **and** LeftVideo **and** TopMenu **implies not** LeftMenu
 R#2: LeftVideo **and** LeftMenu **implies not** LeftCont
 R#3: StartRed **implies not** ExistRed

La R#1 es una restricción que permite ubicar el video en la parte superior izquierda, prohibiendo colocar al menú en la misma posición de la pantalla; esta regla está relacionada con los patrones FullWithVideo y Menu. La R#2 restringe la posición de elementos en pantalla, si el video y el menú estarán en la parte izquierda, el contenido no deberá estar a la izquierda; la misma está relacionada con los patrones FullWithVideo, Menu y Content. La R#3 evalúa que no se emplee la misma tecla color Rojo para comenzar y salir de la aplicación.

La R#1 y R#2 consisten en reglas que controlan la validación de distintos grupos de patrones, R#1 emplea patrones del grupo PageLayout y de Navigation, R#2 emplea patrones del grupo PageLayout, Navigation y Content. La R#3 emplea patrones del grupo Starting y Existing.

En la Figura 2, se presentan algunos productos válidos en la herramienta FAMA Test Suite. Este framework realiza el proceso de análisis automatizado del MC que integran las representaciones lógicas [6]. El archivo fm contiene las características en la sección %Relationships y las reglas en la sección %Constraints. De esta forma se concluye que el modelo y las reglas están bien creadas, como resultado se obtienen el número posible de productos y su conformación.

En el Anexo del presente artículo se repone el modelo de características que corresponden a los patrones de diseño. En el menú, cada grupo con su conjunto de patrones, está bien identificados y ordenados. Un producto

ciones de TVDi y las reglas, una tabla que describe el modelo. El cual permite generar 106.241 productos válidos.

El MC presentado es un modelo genérico, por el cual se pueden emplear en otros middleware con el mínimo esfuerzo. Además permite generar productos en cualquier lenguaje.

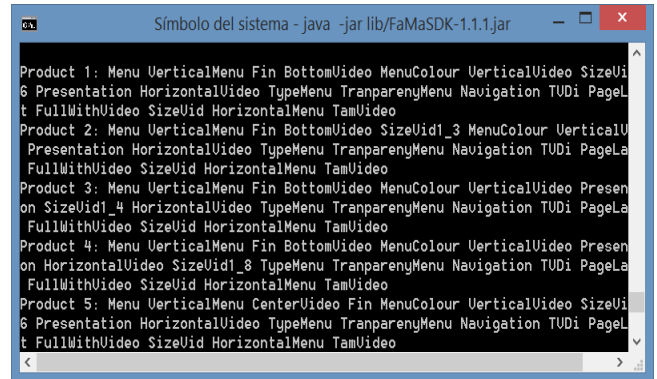


Fig. 2. Productos válidos

4 HERRAMIENTAS SPL-TVDi

La herramienta SPL-TVDi creada da soporte al modelo de características descrito para la producción de aplicaciones TVDi en lenguaje NCL, ejecutables en GINGA. SPL-TVDi está implementado en Java y para el modelado de características se empleó la herramienta FeatureIDE con el componente AHEAD. La Figura 3 presenta el proceso a seguir para el desarrollo de productos, el cual se estructura en tres pasos:

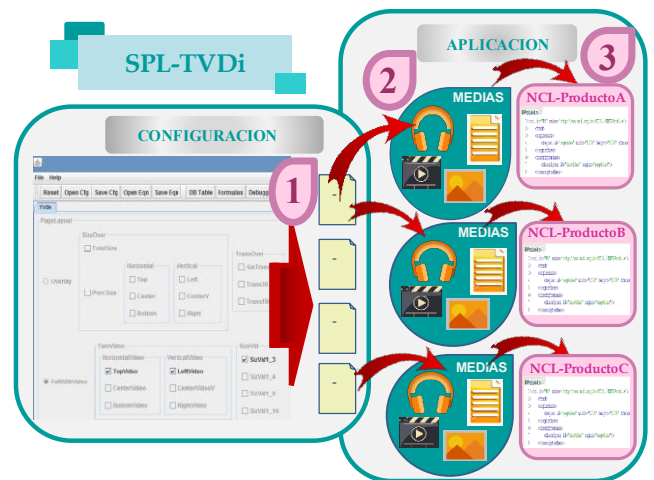


Fig. 3. Estructura para SPL-TVDi

1 - Configuración de Características: El primer paso para la generación de un producto consiste en especificar las características del mismo. Para ello, se deben especificar los elementos de interacción que la aplicación incorporará (imágenes, texto, menús, botones, etc.) a partir de los patrones de interacción. Para ello, la herramienta despliega en el menú, cada grupo con su conjunto de patrones, está bien identificados y ordenados. Un producto

de SPL se especifica de forma declarativa seleccionando o anulando la selección de características de acuerdo a la necesidad del usuario. Las decisiones tomadas deben respetar las limitaciones del MC (parámetros iniciales). El usuario selecciona las características deseadas tildando las opciones que muestra la Figura 4, y como resultado obtiene una configuración, que al guardar crea un archivo de formato config. Este archivo contendrá las características requeridas para crear posteriormente un código NCL.

2 - Selección de Medias: Luego de crear el archivo de configuración, el usuario podrá obtener los productos deseados en la herramienta SPL-TVDi. Cada vez que genere un producto, el usuario deberá especificar las medias (imágenes, video, texto) requeridas para su propio diseño. Las medias solicitadas corresponderán a la configuración actual.

3 - Generación de un producto: Después de seleccionada una determinada configuración y el conjunto de medias particular, se genera el código NCL de un nuevo producto de la SPL-TVDi automáticamente, que implementa la funcionalidad de la aplicación.

La herramienta organiza la disposición de estos elementos en diversas carpetas. Por cada producto genera una nueva carpeta con un nombre especificado por el usuario. La misma contiene una carpeta denominada "Configuraciones", que contiene la configuración del producto y; otra carpeta denominada "Productos", que contiene el archivo NCL, el archivo de conectores (base de patrones de relación entre objetos y eventos), y además la carpeta "media", con las imágenes, videos y textos de la aplicación en particular.

5 GENERACION DE PRODUCTOS

Se presenta como caso de estudio la reproducción de la aplicación Anke Late Night ⁴ con la herramienta SPL-TVDi. En la Figura 5 se muestra la identificación de los patrones en la aplicación y su modelado en el MC correspondiente.

El usuario especifica el nombre para el producto, en este caso "Anke Late Night 4" y comienza con los pasos del proceso descrito anteriormente.

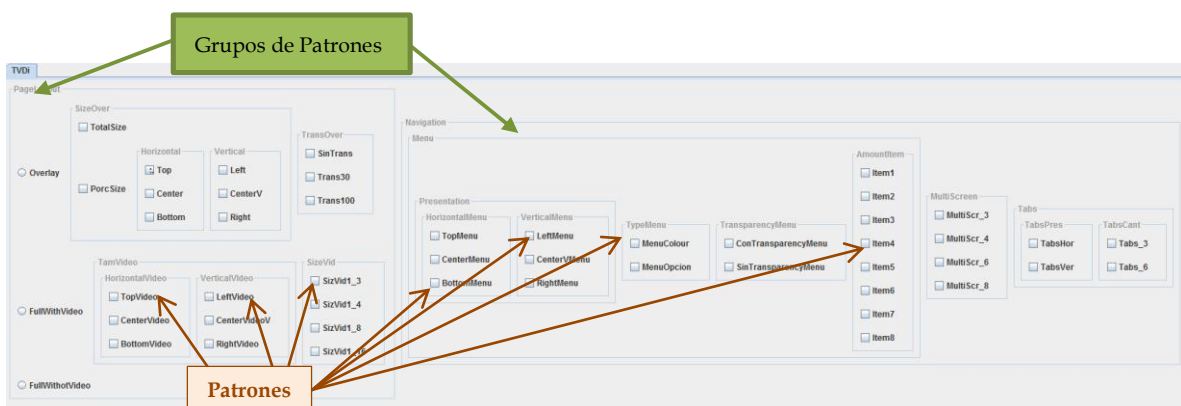


Fig. 4. Vista Parcial de la Configuración del Producto

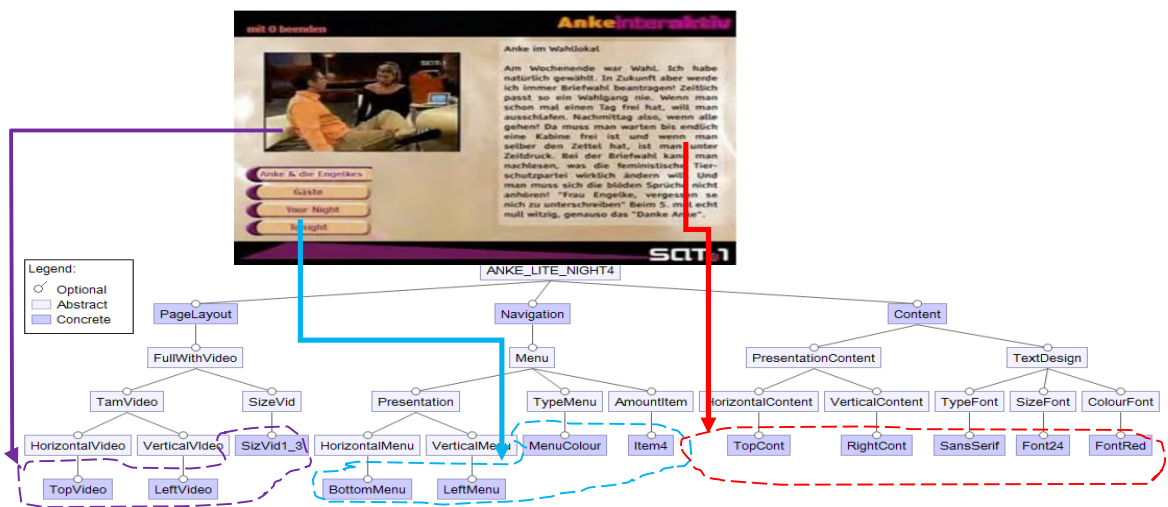


Fig. 5. Características de Aplicación Anke Late Night 4 [5]

¹ Anke Lite Night es una aplicación TVDi presentada en [5] y que ha sido

diseñada e implementada con patrones.

1- Configuración de características para un determinado producto:

Se seleccionaron las siguientes características: a) el video en la parte superior e izquierda de un tamaño de 1/3; b) el menú en la parte inferior, también a la izquierda, menú de colores y con 4 opciones; c) el texto en la parte derecha, tipo de fuente Sans Serif, tamaño 24 y de color rojo. Una vez seleccionadas todas las opciones indicadas se generó la configuración de la Figura 6.

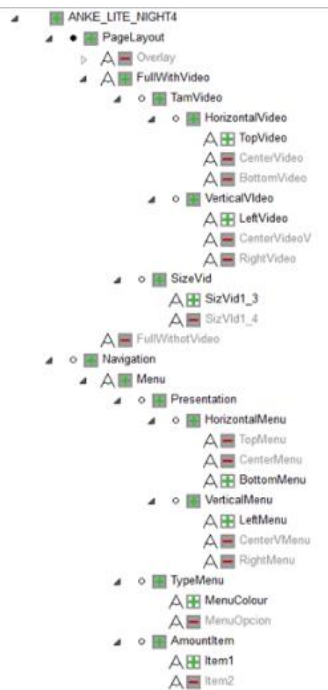


Fig. 6. Vista Parcial del archivo de configuraciones

2- Selección de medias: En este caso se ingresa el video, imágenes para el diseño del fondo de la aplicación de inicio y fin, cuatro imágenes, una para cada botón del menú, cuatro archivos de texto, que especifican el contenido de cada opción del menú.

3- Generación de un producto: Como resultado se obtiene el código NCL que usa las medias ingresadas en el paso 2, según los patrones seleccionados en el paso 1. En el anexo se muestra el código NCL generado por la herramienta automáticamente, en el mismo se ha identificado los segmentos que corresponden a la aplicación de los patrones seleccionados. La Figura 7 muestra la ejecución de aplicación generada con SPL-TVDi para el caso de estudio Anke Late Night 4, desde el emulador Gingga v0.13.3 de PUC-RIO/Laboratorio Telemendia.

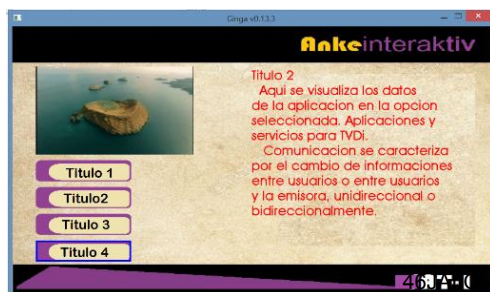


Fig. 7. Caso de estudio de la aplicación Anke Late Night 4

6 TRABAJOS RELACIONADOS

A continuación brevemente se analizan herramientas para el desarrollo de aplicaciones de TVDi y los dominios en los que se han aplicado SPL, para contextualizar este trabajo.

Las herramientas existentes para el desarrollo de aplicaciones TVDi proveen un marco de trabajo basado en metodologías ágiles, combinado con UML para la producción de programas unitarios de TV. En [10] emplean el método Scrum para identificar y definir roles: fases, secuencia de actividades y artefactos específicos en la creación de prototipos. En [11] recurren a un modelo denominado StoryTo-Code, el mismo permite especificar programas TVDi basados en la utilización de componentes de software. Se pueden observar como diferentes enfoques abordan la problemática de la variabilidad en diferentes momentos del ciclo de desarrollo de software. El Desarrollo de Software Basado en Componentes sostiene que la variabilidad debe ser representada y definida localmente en el componente variable, lo que favorece el desarrollo independiente del mismo. Además permite indicar el punto variante en donde se utilizará el componente o subcomponente variable. Las SPL permiten contar con una estructura jerárquica que posibilita tanto el reuso efectivo como la creación e integración de nuevos componentes. Los enfoques previamente presentados permiten implementar la variabilidad en artefactos de software separados a través de la utilización de componentes.

El problema surge en la implementación de estos componentes, como por ejemplo que la instanciación de una cierta variabilidad implique la inserción de código en diferentes unidades del código, o cambiar unas pocas líneas de código, etc. Los beneficios de las SPL posibilitan mantener el modelo jerárquico y generar adaptaciones a nivel de código, permitiendo una mayor flexibilidad y reduciendo el costo de derivación [17].

Las propuestas de SPL, si bien fueron inspiradas en las cadenas de producción automatizada [12], se destacan también en otros contextos como textil [13], robótica [14], telefonía móvil [15], ecología marina [16], geográficos [17], entre otros [18] y; algunos trabajos relacionados con la TVDi no recientes [19] [20]. No existiendo demasiados antecedentes en el dominio de TVDi, resulta novedoso crear una SPL basado en patrones de diseño para aplicaciones para la TVDi.

Sin embargo, existen diversas herramientas para el desarrollo de aplicaciones TVDi GINGA-NCL como, NCL Composer [21], Crea TV Digital [22], SGAi [23], ATHUS [24], ITV-Learning [25], API TVD [10], Template Generator [26], GinggaGame [27], NCL-Inspector, entre otros. Estas herramientas corresponden a editores de código, modelado, especificaciones, validación y/o pruebas. Si bien algunas de estas herramientas soportan niveles de reuso de activos de software existentes, recurren al desarrollo de una nueva aplicación desde el comienzo.

La principal diferencia de SPL-TVDi con los trabajos y herramientas mencionados es que permita la producción de aplicaciones para la TVDi en forma rápida y a bajo costo. Además este trabajo incorpora usabilidad, calidad y; por otro lado reuso de código.

7 CONCLUSION

Una de las principales motivaciones de este trabajo fue la falta de enfoques y herramientas de desarrollo de aplicaciones TVDi, a nivel industrial. Para ello, es necesario reducir el tiempo y costo de desarrollo y aplicar un enfoque de reutilización que permita generar rápidamente productos software.

Se ha presentado una SPL, en base a un modelo de características. El modelo de características es genérico, el cual contiene todas las características comunes y variantes que corresponden a patrones de diseño de interacción ya revisados, y garantizan la especificación de configuraciones con niveles de usabilidad aceptables.

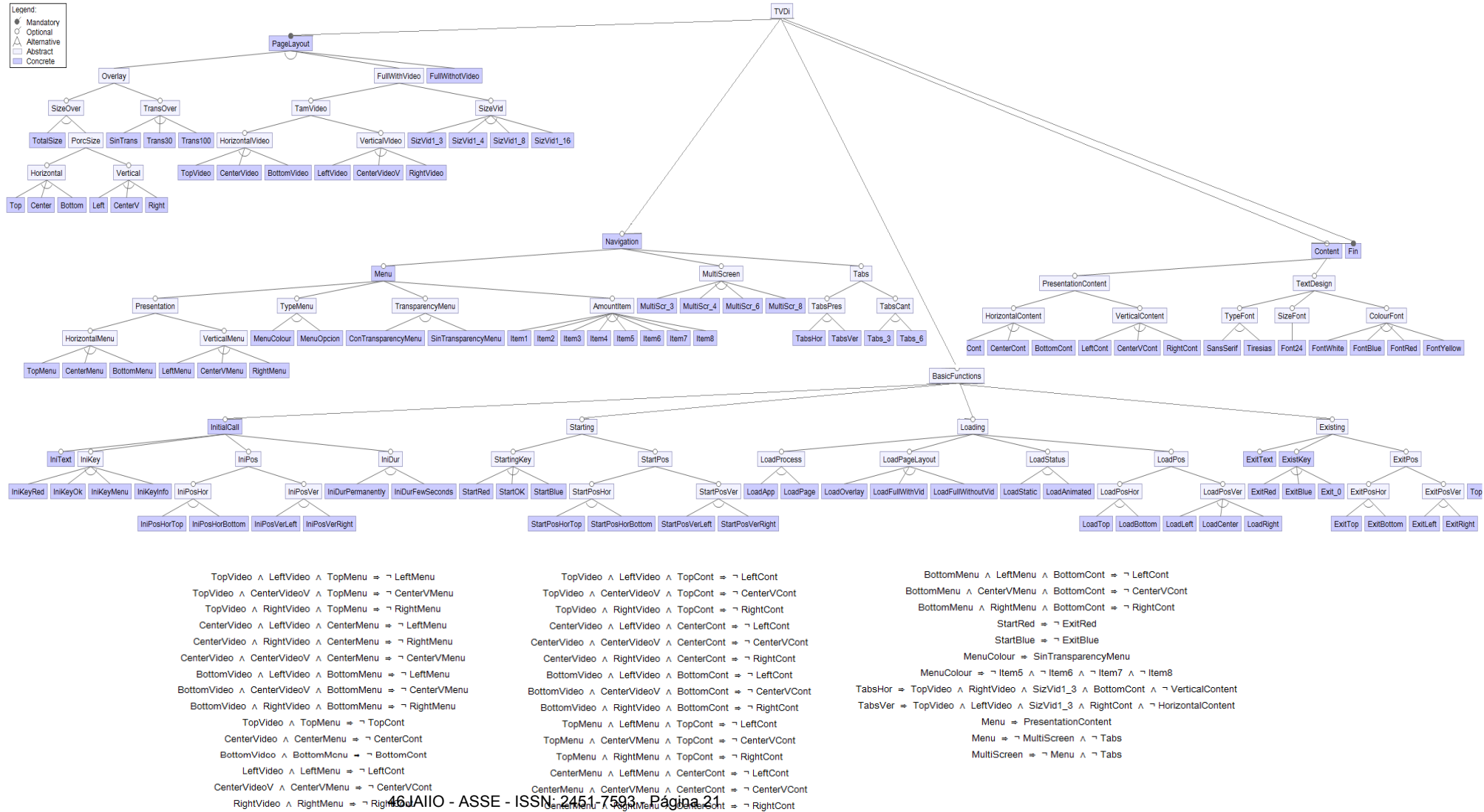
Además se definen restricciones del tipo implicación y exclusión entre características, esto permite la simpleza en la creación de una configuración porque reduce la cantidad de alternativas de características a seleccionar.

La herramienta SPL-TVDi permite la generación de productos a partir de la especificación de configuración. Una vez creado un producto, el mismo se ejecuta en un middleware Ginga NCL.

REFERENCES

- [1] M. Acher, B. Baudry, P. Heymans, A. Cleve, J. L. Hainaut, "Support for Reverse Engineering and Maintaining Feature Models". Seventh International Workshop on Variability Modelling of Software-Intensive Systems (VaMoS'13), pp. 20, 2013.
- [2] M. Abrutsky, F. Bobbio, D. Campos, G. Cáceres, D. Díaz, C. R. Luna, M. A. Odetti, "Transferencia de resultados de investigación en desarrollo de software para TV Digital en la provincia de Córdoba". In XVII Workshop de Investigadores en Ciencias de la Computación, 2015.
- [3] F. Balaguer, A. Zambrano, "TV Digital Interactiva Argentina. TV DIGITAL: Un dialogo entre disciplinas y multipantallas", Universidad Nacional de La Plata, ISBN 978-950-34-1048-6, pp. 33, 2013.
- [4] Página Oficial de Ginga ar: <http://tvd.lifia.info.unlp.edu.ar/ginga.ar/>
- [5] T. Kunert, "Types of Design Guidance for iTV Applications. In User-Centered Interaction Design Patterns for Interactive Digital Television Applications", Springer London, pp. 47-84, 2009.
- [6] S. Segura, J. A. Galindo, D. Benavides, J. A. Parejo, A. Ruiz-Cortés, "BeTTY: benchmarking and testing on the automated analysis of feature models". In Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems, pp. 63-71, 2012, New York, NY, USA: ACM. doi:10.1145/2110147.2110155
- [7] E. Rincón, A. Matteo, F. Losavio, "Evaluación y Análisis Jerárquico de Métodos de Análisis del Dominio en Líneas de Productos Software", 2015.
- [8] M. F. Miranda, S. I. Casas, "Desarrollo de una Línea de Producto de Software para aplicaciones de TVDi". En VI Congreso Internacional de Televisión Digital Interactiva IV Jornadas Iberoamericanas en Aplicaciones y Usabilidad de la TVi, pp. 23-32, 2015, Universidad de las Islas Baleares - Palmas, España.
- [9] M. Z. Ros, "Patrones en elearning. Elementos y referencias para la formación". Revista de Educación a Distancia, nro. 27, 2015.
- [10] F. Oyarzo, F. Herrera, S. Casas, "API TVD, a wizard for interactive applications for Digital TV", CLEI 2014 XL Latin American, pp. 1-8, 2014.
- [11] N. M. Marqués, C. Santos, "StoryToCode: Un Modelo Baseado em Componentes para Especificación de Aplicaciones de TV Digital Interativa Convergentes", Brazil, 2009.
- [12] A. Deursen, P. Klint, "Domain-Specific Language Design Requires Feature Descriptions", Journal of Computing and Information Technology, Vol 10, pp.1-17, 2002.
- [13] C. Camacho González, "Software product lines using FODA: a formal approach", 2012.
- [14] L. Gherardi, D. Brugali, "An eclipse-based feature models toolchain". In Proc. of the 6th Workshop of the Italian Eclipse Community (Eclipse-IT 2011), 2011.
- [15] J. Enríquez, S. Casas, "FUsaM: Framework, con base en una SPL, para la medición de usabilidad en aplicaciones móviles". Uniciencia, 30(2), pp. 31-45, 2016.
- [16] N. Huenchuman, A. Bucella, A. Cechich, M. Polla, M. D. S. Doldan, E. Morsán, M. Arias, "Reingeniería de una línea de productos de software: un caso de estudio en el subdominio de Ecología Marina". In XVIII Congreso Argentino de Ciencias de la Computación, 2013.
- [17] J. G. Navarro Favela, U. Juárez Martínez, "Cómo desarrollar una línea de productos de software, un enfoque práctico". 4TH International Conference on Computer Science and Its Application (CIA 2013), ISBN 978-607-9119-02-7, pp. 16-25, 2013, <http://cia.itsm.edu.mx>
- [18] A. Febles Estrada, and D. Pozo, "Evaluando la usabilidad en aplicaciones para la TVDi". In I Jornadas Iberoamericanas de Difusión y Capacitación sobre Televisión Digital Interactiva, 2013.
- [19] ŽARNIĆ, Berislav. http://link.springer.com/referenceworkentry/10.1007/978-981-287-532-7_143-1. En Encyclopedia of Educational Philosophy and Theory. Springer, 2016. [Http://link.springer.com/chapter/10.1007/978-3-642-15114-9_14](http://link.springer.com/chapter/10.1007/978-3-642-15114-9_14).
- [20] KIM, Kangtae; KIM, Hyungrok; KIM, Woomok. Building Software Product Line from the Legacy Systems' Experience in the Digital Audio and Video Domain'. En Software Product Line Conference, 2007. SPLC 2007. 11th International. IEEE, 2007. pp. 171-180. [Http://ieeexplore.ieee.org/abstract/document/4339266/](http://ieeexplore.ieee.org/abstract/document/4339266/)
- [21] R. L. Guimarães, R. M. de Resende Costa, L. F. G. Soares, "Composer: Authoring Tool for iTV Programs. In European Conference on Interactive Television", EuroITV 2008, Springer Berlin Heidelberg, LNCS 5066, pp. 61-71.
- [22] M. Arroyo, S. Schwartz, S. Cardozo, L. Tardivo, "CreaTVDigital: Composición Visual de Aplicaciones Interactivas para TV Digital", 41JAIIOSS, ISSN, 1850-2830, pp. 305-321, 2012.
- [23] I. M. Bernal, G. Cabezas, M. F. Quezada, "Sistema de Generación de Aplicaciones Interactivas para TV Digital para la evaluación de servicios masivos", Revista Politécnica, 32, 2013.
- [24] R. M. Segundo, J. C. F. da Silva, T. A. Tavares, "ATHUS: A generic framework for game development on Ginga middleware". In Games and Digital Entertainment (SBGAMES), 2010 Brazilian Symposium on pp. 89-96, IEEE, 2010.
- [25] F. A. de Sousa Neto, P. Bezerra, D. D. S. F. Dias, "ITV-Learning: a Prototype for Construction of Learning Objects for Interactive Digital Television", 2012.
- [26] S. Ochoa, A. Pillaio, F. Acosta, G. Imedo, "Template Generator: Software para la generación de aplicaciones interactivas para la televisión digital terrestre a partir de plantilla Ginga y LUA". Jauti2014 III Jornadas Iberoamericanas de Difusión y Capacitación obre Aplicaciones y Usabilidad de la TVDi. III WTVDI Webmedia 2014, ISBN 978-950-34-1188-9, pp. 109-113, 2014.
- [27] D. C. Barboza, E. W. G. Clua, "Ginga Game: a framework for game development for the interactive digital television", In Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on pp. 162-167, IEEE, 2009.

ANEXO MODELO DE CARACTERÍSTICAS DE LA SPL PARA APLICACIONES TVDI



La Tabla describe las características representadas en el Modelo de Características, muestra las características superiores, correspondientes a los grupos de los patrones de diseño. La columna Relación describe el tipo de característica, obligatoria, opcional, alternativa o disyuntiva.

Características	Descripción	Relación
TVDi	Aplicación de TV Digital Interactiva	-----
PageLayout	Diseño de Pantalla	Obligatoria
Overlay	Pantalla con Video de fondo y Aplicación encima	Alternativa
FullWithVideo	Pantalla completa con video y aplicación	Alternativa
FullWithoutVideo	Pantalla completa sin video, solo aplicación	Alternativa
Navigation	Pantalla de Formas de Navegar	Opcionales
Menu	Presentación de diversos contenidos	Opcionales
MultiScreen	Video Múltiple pantalla	Opcionales
Tabs	Tabulación para el acceso a los datos	Opcionales
BasicFunctions	Funciones Básicas	Opcionales
InitialCall	Llamada inicial a la aplicación	Opcionales
Starting	Inicio de la llamada a la aplicación	Opcionales
Loading	Carga de la aplicación	Opcionales
Existing	Salida de la aplicación	Opcionales
Content	Presentación del Contenido	Opcionales
PresentationContent	Disposición del texto en pantalla	Opcionales
TextDesign	Diseño del texto	Opcionales

CÓDIGO NCL GENERADO POR SPL-TVDI PARA APLICACIÓN ANKE LITE NIGHT 4

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Inicio de la Aplicacion -->
<ncl id="Inicio" xmlns="http://www.ncl.org.br/NCL3.0/EDIVProfile">
<head>
<ruleBase>
<rule id="rule1" var="page" comparator="eq" value="1"/>
<rule id="rule2" var="page" comparator="eq" value="2"/>
<rule id="rule3" var="page" comparator="eq" value="3"/>
<rule id="rule4" var="page" comparator="eq" value="4"/>
</ruleBase>
<regionBase>
<region id="regBotonSig" top="0%" height="0%" width="0%" left="0%"/>
<region id="regBotonVolverIni" top="0%" height="0%" width="0%"/>
<region id="regBotonEsc" top="0%" height="0%" width="0%" left="0%"/>
<region id="regPanFin" width="100%" height="100%"/>
<region id="regTopPanFin" height="100%"/>
<region id="regBotonVolverIniPanFin" top="93%" height="5%" width="5%" left="80%" zIndex="1"/>
<region id="regBotonEscPanCont" top="93%" height="5%" width="5%" left="80%" zIndex="1"/>
</region>
<region id="regIni" width="100%" height="100%"/>
<region id="top" height="100%"/>
<region id="regBotonTexto" top="5%" height="5%" width="15%" right="5%" zIndex="1"/>
<region id="regBotonSigVideo" top="5%" height="5%" width="5%" right="5%" zIndex="2"/>
</region>
<region id="regStart" width="100%" height="100%"/>
<region id="topStart" height="100%"/>
<region id="regBotonStart" bottom="5%" height="5%" width="5%" left="5%" zIndex="2"/>
</region>
<region id="reg1" zIndex="2" width="33%" height="33%" top="15%" left="5%"/>
<region id="reg2" zIndex="1" width="100%" height="100%"/>
<region id="regb1" width="20%" height="8%" bottom="12%" left="5%" zIndex="2"/>
<region id="regb2" width="20%" height="8%" bottom="22%" left="5%" zIndex="3"/>
<region id="regb3" width="20%" height="8%" bottom="32%" left="5%" zIndex="4"/>
<region id="regb4" width="20%" height="8%" bottom="42%" left="5%" zIndex="5"/>
<region id="regc1" width="50%" height="70%" top="15%" right="0%" zIndex="1"/>
<region id="regc2" width="50%" height="70%" top="15%" right="0%" zIndex="2"/>
<region id="regc3" width="50%" height="70%" top="15%" right="0%" zIndex="3"/>
<region id="regc4" width="50%" height="70%" top="15%" right="0%" zIndex="4"/>
<region id="regTopPanFin3" height="100%"/>
<region id="regBotonVolverIniPanFin3" top="93%" height="5%" width="5%" left="80%" zIndex="1"/>
</regionBase>
<descriptorBase>
<descriptor id="desBotonSig" region="regBotonSig"/>
<descriptor id="desBotonVolverIni" region="regBotonVolverIni"/>
<descriptor id="desBotonEscPanCont" region="regBotonEscPanCont"/>
<descriptor id="desBotonEsc" region="regBotonEsc"/>
<descriptor id="desIni" region="top"/>
<descriptor id="desBotonSigVideo" region="regBotonSigVideo"/>
<descriptor id="desInitialCallText" region="regBotonTexto"/>
<descriptor id="desStart" region="topStart" explicitDur="5s"/>
<descriptor id="desBotonStart" region="regBotonStart"/>
<descriptor id="des1" region="reg1"/>
<descriptor id="des2" region="reg2"/>
<descriptor id="desb1" region="regb1" focusIndex="ixB1" moveRight="ixB2" moveLeft="ixB4"/>
<descriptor id="desb2" region="regb2" focusIndex="ixB2" moveRight="ixB3" moveLeft="ixB1"/>
<descriptor id="desb3" region="regb3" focusIndex="ixB3" moveRight="ixB4" moveLeft="ixB2"/>
<descriptor id="desb4" region="regb4" focusIndex="ixB4" moveRight="ixB1" moveLeft="ixB3"/>
<descriptor id="desc1" region="regc1"/>
<descriptor id="desc2" region="regc2"/>
<descriptor id="desc3" region="regc3"/>
<descriptor id="desc4" region="regc4"/>
<descriptor id="desPanFin3" region="regTopPanFin3"/>
<descriptor id="desBotonVolverIniPanFin3" region="regBotonVolverIniPanFin3"/>
</descriptorBase>
<connectorBase>
<importBase documentURI="connectorBase.ncl" alias="connector"/>
</connectorBase>
</head>

```

BasicFunctions

PageLayout

Navigation

Content

BasicFunctions

PageLayout

Navigation

Content

