

Uso de DSL y MetaEdit+ para automatizar etapas iniciales en la construcción de Sistemas de Información Sanitaria

Cesaretti Juan¹, Paganini Lucas¹, Garbarino Gabriel¹,
Rocca Leandro¹, Leopoldo Nahuel^{1,2}, Giandini Roxana^{1,2}

¹ GIDAS, Grupo de I&D Aplicado a Sistemas informáticos y computacionales
Facultad Regional La Plata - Universidad Tecnológica Nacional

² Centro de I&D LIFIA, Laboratorio de Investigación y Formación en Informática Avanzada
Facultad de Informática - Universidad Nacional de La Plata
{jcesaretti, lpaganini, ggarbarino,
leorocca, lnahuel, rgiandini}@frrlp.utn.edu.ar

Abstract. El propósito de este trabajo es aportar una herramienta para simplificar y acelerar las primeras fases del proceso de desarrollo de sistemas de información sanitaria. Para ello se propone la utilización de un lenguaje específico del dominio que denominamos DSL_SIS. Este fue definido tomando elementos del estándar de interoperabilidad clínica FHIR, desarrollado por la organización internacional HL7 (Health Level Seven). Para implementar la herramienta, se utilizó el metaeditor MetaEdit+, que ofrece una mayor sencillez y celeridad para componer los distintos bloques de construcción, y además, cuenta con un generador integrado que permite realizar fácilmente transformaciones de modelo a texto. Así, a partir de diagramas estructurales construidos rápidamente, obtuvimos el código para generar las clases correspondientes en distintos lenguajes de programación, como Java, Php, Python y Ruby.

Keywords: DSL, Sistema de Información Sanitaria, FHIR, Automatización, MetaEdit+.

1 Introducción

La Ingeniería Dirigida por Modelos (MDE - Model Driven Engineering) se enfoca en los modelos, a los que considera los elementos centrales de la Ingeniería de Software. Un modelo es una representación simplificada de la realidad, desacoplada de los detalles de implementación. Esto provee una gran adaptabilidad frente a la evolución de las tecnologías utilizadas. En particular, el Desarrollo de Software Dirigido por Modelos abarca todas las propuestas y mecanismos para producir aplicaciones a partir de la transformación de modelos, proporcionando un alto nivel de abstracción. Se conoce por sus siglas en inglés MDD (Model-Driven Development) [1], [2].

El Diseño Específico del Dominio o DSM (del inglés: Domain Specific Modeling), es una disciplina que, en el contexto del MDD, trabaja con lenguajes propios, restringidos a cada ámbito de interés. Son los llamados Lenguajes Específicos de Dominio, también conocidos por su acrónimo: DSL (Domain-Specific Language). Esta

especialización permite una mayor automatización, que no podría lograrse usando un lenguaje de propósito general, como UML.

Un modelo puede constituirse con diagramas, utilizando símbolos gráficos definidos en un lenguaje de modelado. En un DSL, cada bloque de construcción representa algún concepto propio del dominio considerado. Luego, este tipo de lenguaje es más claro y manejable para los usuarios, y las particularidades tecnológicas son transparentes para los mismos [1].

La sintaxis abstracta de un lenguaje gráfico de modelado se define en un metamodelo que especifica los elementos, las relaciones entre ellos y las reglas de buena formación de los modelos. Dado que un metamodelo es también un modelo, debe estar expresado en un lenguaje bien definido, conocido como metalenguaje. Este determina qué elementos pueden ser usados y cómo pueden vincularse. Algunos metalenguajes conocidos son: MOF (Meta-Object Facility) [3], ECORE [4] y GOPPRR [5].

La sintaxis concreta de un lenguaje gráfico determina el aspecto visual del mismo, estableciendo una colección de símbolos que pueden utilizarse para construir los diagramas.

La organización OMG propone una arquitectura de cuatro niveles:

- **Nivel M0:** Instancias. Consta de los objetos instanciados por la aplicación. En este caso, los lenguajes de programación en los que se podrán instanciar objetos serán: Java, PHP, Python y Ruby.
- **Nivel M1:** Modelo. En el que se dibujan los diagramas para modelar el sistema. Para la construcción del editor se utilizó la herramienta MetaEdit+.
- **Nivel M2:** Metamodelo. Es donde se definen los bloques de construcción de los diagramas del nivel M1. Se consideran un subconjunto de recursos fundamentales de FHIR y se utiliza el metalenguaje GOPPRR.
- **Nivel M3:** Meta-metamodelo. Es el nivel más abstracto. Dentro del OMG, MOF es el lenguaje estándar de la capa M3.
- **Lenguaje DSL_SIS:** Lenguaje Específico de Dominio presentado en este trabajo, en el que se expresan los modelos del Nivel M1. Más adelante, en la sección 5, se exponen algunos modelos a modo de ejemplo.
- **Metalenguaje GOPPRR:** Metalenguaje en el que está escrito el metamodelo del Nivel M2.

En la Fig. 1 se ilustran estos conceptos, en el contexto del desarrollo del DSL_SIS.

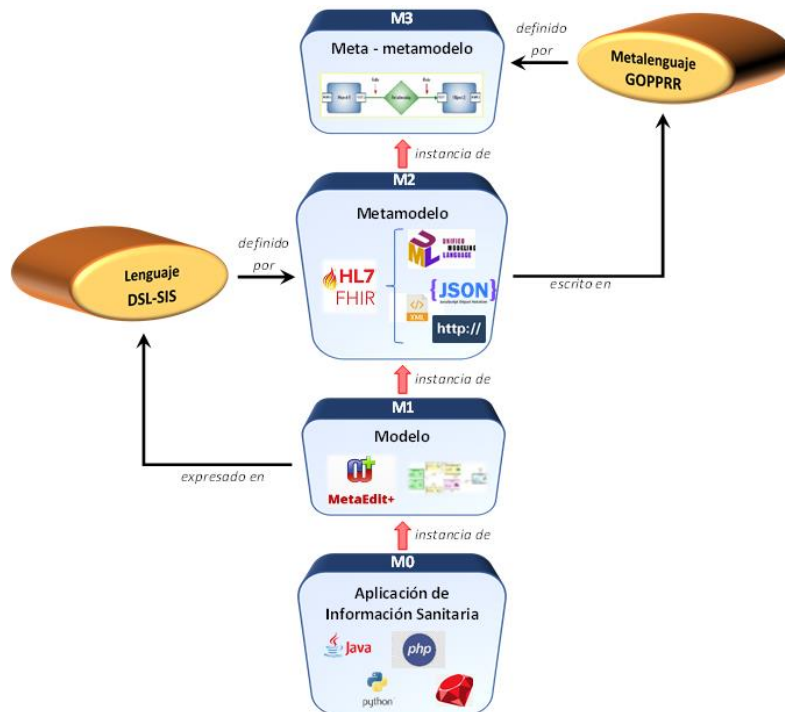


Fig. 1. Arquitectura de cuatro niveles del OMG y su integración con DSL

Un dominio con una complejidad creciente es el de los Sistemas de Información Sanitarios (SIS). Esto se debe a la progresiva digitalización de la información básica de atención médica, que debe ser accesible desde distintas organizaciones y puntos geográficos, en virtud de la movilidad de los pacientes [6]. Además, estos sistemas necesitan adecuarse constantemente a las tecnologías aplicadas. Por lo tanto, el abordaje del DSM resulta muy conveniente en este caso.

Para intercambiar la información entre distintos sistemas de gestión sanitaria, existen estándares de interoperabilidad que definen la estructura de los datos, para que puedan compartirse [7]. Diversas organizaciones se ocupan de uniformar criterios de interoperabilidad: HL7 Internacional, HIMMS o NEMA.

El último estándar de interoperabilidad clínica de HL7 es FHIR (Fast Healthcare Interoperability Resources) [8]. Es de código abierto, y amalgama lo mejor de los estándares más utilizados en la actualidad. El elemento fundamental de FHIR es el recurso, es decir, la unidad básica de interoperabilidad. Cada recurso especifica un concepto del dominio sanitario: paciente, médico, problema de salud, etc.

Si bien existe un metamodelo de HL7, llamado MIF (Model Interchange Format) [9], es muy extenso y complejo, y su aprendizaje significa un costo muy alto para los ingenieros de software. Además, MIF tiene múltiples versiones coexistentes, y esto puede causar problemas de incompatibilidad [10]. Por otra parte, algunos investigadores de HL7 Internacional han publicado un perfil de UML ajustado a MIF

[11]. No obstante, el propósito de este trabajo es presentar un DSL elemental, con un metamodelo más simple, fundado en un subconjunto relevante de recursos de FHIR.

Existen experiencias previas en el uso de HL7 en el marco de MDE [12]. Pero en ninguna de ellas se produjo una herramienta para automatizar la generación de código fuente. Esto se debe a que su propósito fue extender o especializar un lenguaje de propósito general como UML, o lograr transformaciones de modelos entre HL7 y UML. Con el DSL_SIS, es posible esta automatización, siendo uno de los aspectos más interesantes del presente trabajo.

2 Antecedentes: trabajos y prototipos realizados previamente

En trabajos que realizamos anteriormente [13], [14], consideramos sistemas básicos de gestión sanitaria donde pueden registrarse consultas médicas ambulatorias de demanda espontánea, guardia o urgencias e internaciones no programadas. Se excluyó la gestión de turnos, diagnóstico por imagen o laboratorio, y las prescripciones de medicamentos.

Se desarrolló un DSL para este dominio acotado, a través de un metamodelo construido con el metalenguaje ECORE, reutilizando algunos elementos del metamodelo de UML [15].

Cada elemento del metamodelo se corresponde con un recurso de FHIR. El subconjunto de recursos seleccionados fue:

- **Patient (Paciente)**: Sujeto receptor de los servicios de atención sanitaria.
- **Practitioner (Profesional de la Salud)**: Cualquier persona que está directa o indirectamente involucrada en el aprovisionamiento de atención sanitaria. Puede ser médico, enfermero, etc.
- **Person (Persona)**: Abstracción que reúne la información demográfica y administrativa de una persona, independiente del contexto sanitario.
- **Organization (Organización)**: Establecimiento donde se proveen servicios sanitarios. Puede ser hospital, clínica, sala de atención primaria, etc.
- **Encounter (Encuentro)**: Cualquier interacción entre un paciente y un proveedor de atención sanitaria. Puede ser una consulta ambulatoria, una práctica, etc.
- **Episode of Care (Episodio de Atención)**: Asociación temporal entre una organización sanitaria responsable y un paciente, durante la cual pueden ocurrir encuentros. Se enfoca en una enfermedad o problema de salud particular. Varias organizaciones pueden estar involucradas en el proceso, pero cada una tendrá su propio Episodio de Atención, para realizar el seguimiento de su responsabilidad con el paciente. Esto permite separar los encuentros según el problema. Por ejemplo, si un paciente está siendo atendido por una enfermedad viral y por un traumatismo de rodilla, se mantendrán dos Episodios de Atención diferentes.

Las relaciones válidas también se establecieron de acuerdo con la especificación de FHIR.

Se implementó un editor gráfico basado en el DSL propuesto. Para ello se utilizó Sirius, un framework de código abierto, que aprovecha las tecnologías EMF y GMF de la plataforma Eclipse [16]. Este editor solamente permitía crear, visualizar y modificar

diagramas, respetando la sintaxis del DSL, pero no contaba con la posibilidad de realizar transformaciones de modelo a código.

Siguiendo los lineamientos de estas experiencias previas sobre el modelado específico del dominio, abordamos una nueva investigación utilizando MetaEdit+ y haciendo uso del generador integrado de código, que permite realizar fácilmente transformaciones de modelo a texto.

3 Un Lenguaje Específico de Dominio para Sistemas de Información Sanitaria

A continuación, se presenta la sintaxis abstracta del lenguaje propuesto, DSL_SIS, en el estado actual del proyecto.

El metamodelo del DSL_SIS fue realizado en el Metalenguaje GOPRR, utilizando el mismo metaeditor MetaEdit+ (Ver Fig. 2).

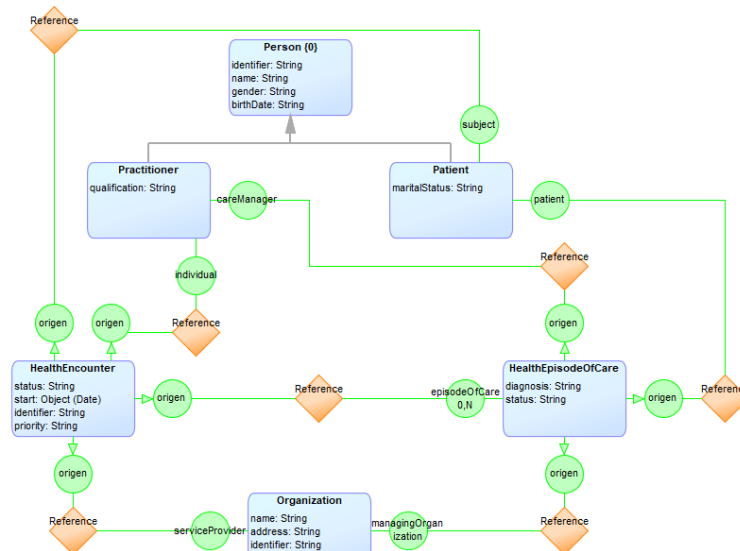


Fig. 2. Metamodelo del DSL_SIS

GOPRR es un acrónimo creado a partir de los metatipos básicos, que son:

- **Gráfico:** es un diagrama, una colección de objetos y relaciones.
- **Objeto:** es cada uno de los elementos (conceptos del dominio) que componen un diagrama.
- **Puerto:** es la especificación opcional de una parte de un objeto a la que puede conectar un rol determinado.
- **Propiedad:** es una característica asociada a los otros metatipos.
- **Relación:** es una conexión explícita entre objetos.
- **Rol:** es la especificación de cómo participa un objeto en una relación.

El metamodelo de DSL_SIS, hasta el momento, consta de un solo gráfico: un diagrama estático. Dicho diagrama está compuesto por los objetos tomados de la especificación de FHIR: Person, Practitioner, Patient, Organization, Encounter y EpisodeOfCare. Las propiedades de cada objeto también se tomaron del mismo estándar de interoperabilidad, así como la única relación (Reference) por la que pueden conectarse los diferentes objetos, con los roles allí definidos.

4 Diseño y Desarrollo del Editor del DSL_SIS con MetaEdit+

Se desarrolló una herramienta basada en el DSL_SIS, utilizando el metaeditor MetaEdit+ [17]. Se escogió este metaeditor porque provee un entorno sencillo y amigable para construir editores en un tiempo muy corto. Y en este sentido es mucho más potente que las herramientas de Eclipse que nosotros mismos utilizamos previamente en este proyecto de investigación [18], [19]. En la Fig. 3 se muestra la ventana principal de MetaEdit+.

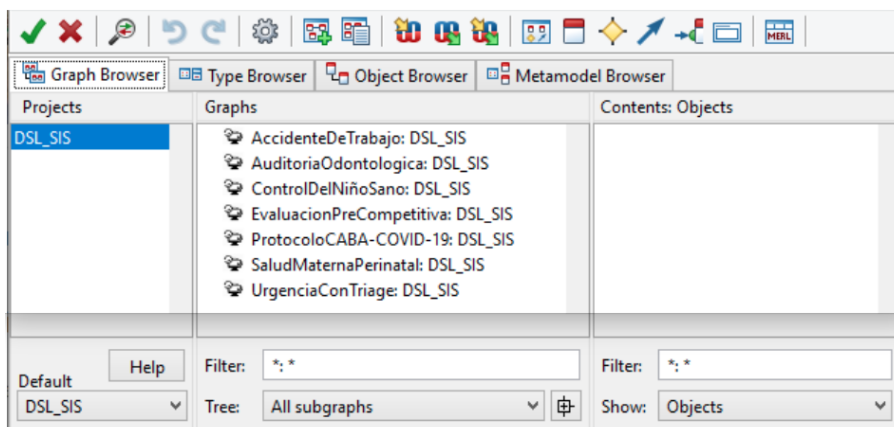


Fig. 3. Ventana principal de MetaEdit+

El primer paso para construir el editor, una vez creado el proyecto, fue definir los elementos, sus relaciones y los roles, utilizando la Herramienta de Gráficos, como muestra la Fig. 4. Los elementos definidos fueron: HealthPatient, HealthPractitioner, HealthOrganization, HealthEncounter, HealthEpisodeOfCare, correspondiéndose con los descriptos anteriormente. La única relación establecida está basada en la que vincula los distintos recursos de FHIR: Reference. Y los nombres de los roles fueron tomados también de dicho estándar.

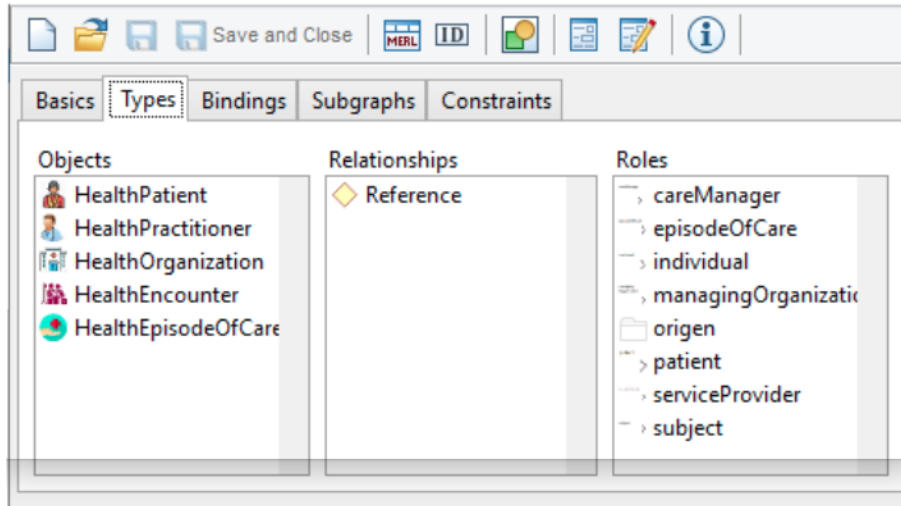


Fig. 4. Herramienta Gráfica para los elementos visuales del DSL_SIS

Las propiedades de cada elemento se editaron usando la Herramienta de Objetos, como muestra la Fig. 5 para el caso de un Encuentro (HealthEncounter).

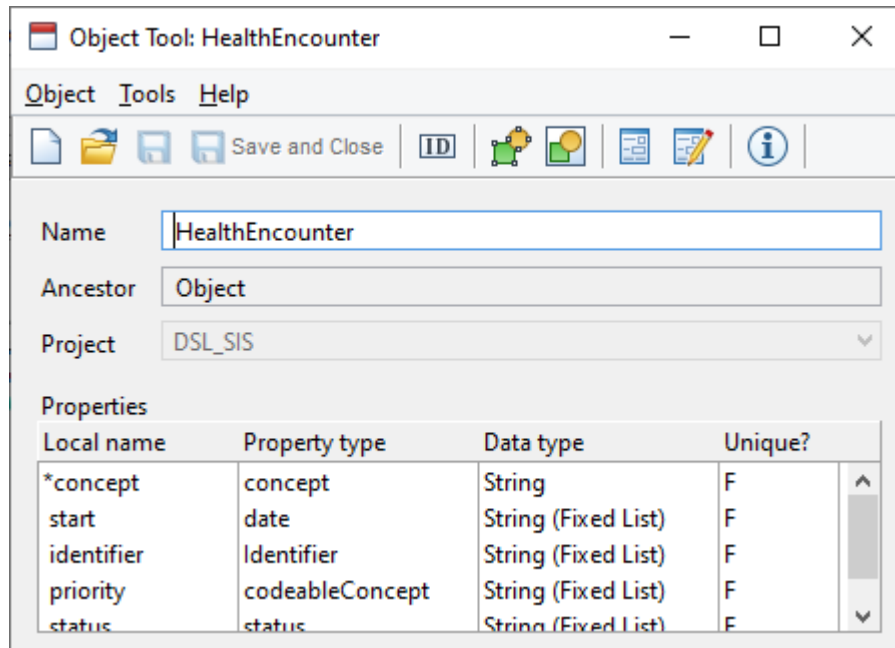


Fig. 5. Herramienta de Objeto para el elemento del dominio: Encuentro (HealthEncounter)

Se editó la representación gráfica de cada elemento usando el Editor de Símbolos, como muestra la Fig. 6, para el caso de un HealthPractitioner.

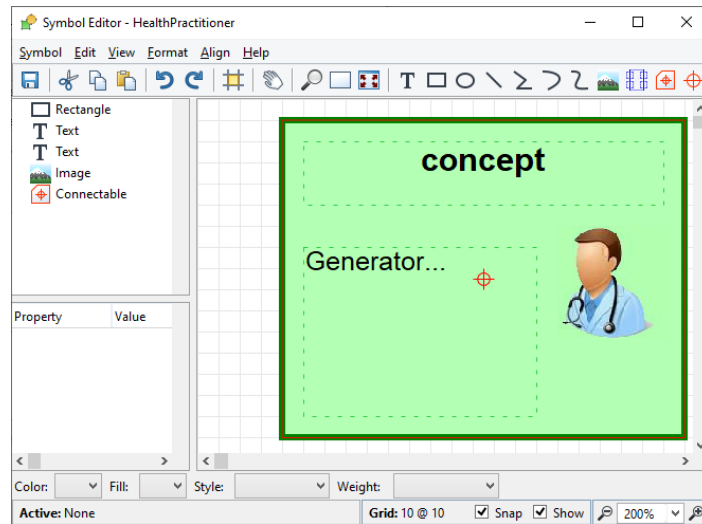


Fig. 6. Editor de Símbolos

Luego se definieron las relaciones válidas, a través de “bindings” que guardan qué tipo de objetos pueden conectarse y con qué cardinalidad, en cada uno de los diferentes roles que se establecieron previamente, como muestra la Fig. 7.

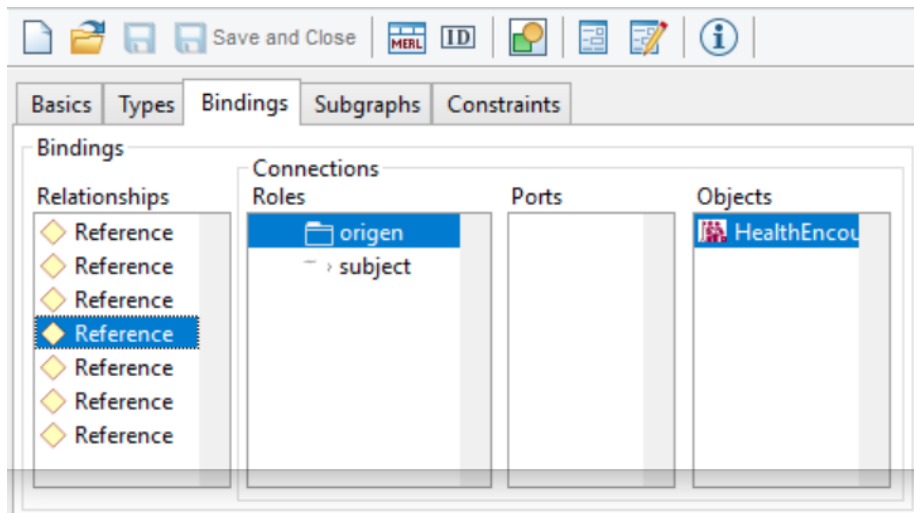


Fig. 7. Relaciones válidas definidas en el DSL_SIS para Encuentro (HealthEncounter)

Pueden definirse otras restricciones (de conectividad, ocurrencia, puertos o unicidad) dentro de la Herramienta de Gráficos, en la solapa correspondiente, como muestra la Fig. 8.

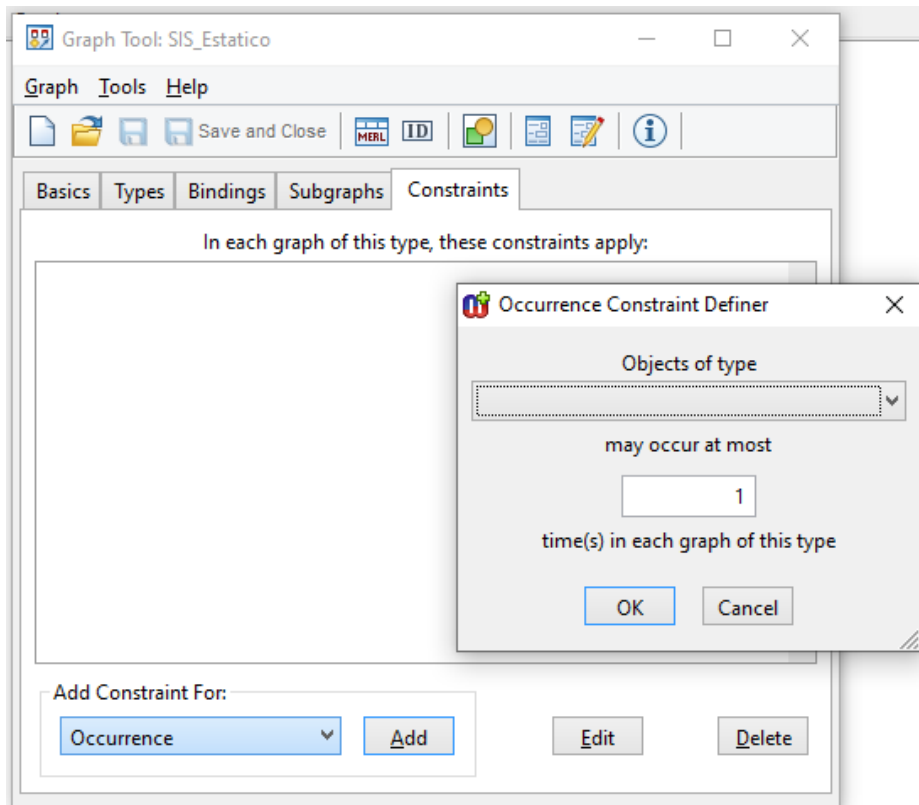


Fig. 8. Definición de restricciones en la Herramienta de Gráficos

Así, la paleta de elementos y relaciones que definimos quedó integrada a la Barra de Herramientas como puede verse en la Fig. 9.



Fig. 9. Paleta de elementos y relaciones

Una vez completados estos pasos, el editor estuvo listo para crear diagramas usando el DSL_SIS, arrastrando elementos desde la paleta de elementos y relaciones, hacia el paño (ver Fig. 10).

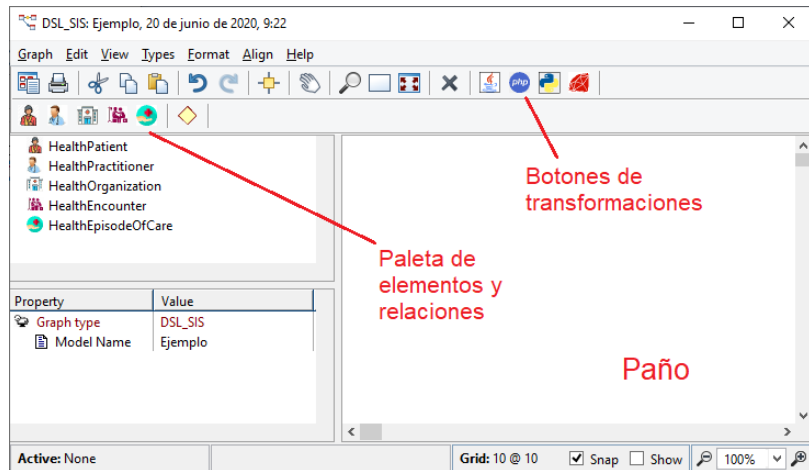


Fig. 10. Editor para el Modelado de Sistemas de Información Sanitaria basada en DSL_SIS

Aprovechando el generador integrado que posee MetaEdit+, se realizaron transformaciones de modelo a texto (ver Fig. 11) para producir código automáticamente en diferentes lenguajes (Java, Php, Python, Ruby), con sólo presionar un botón de la barra de herramientas.

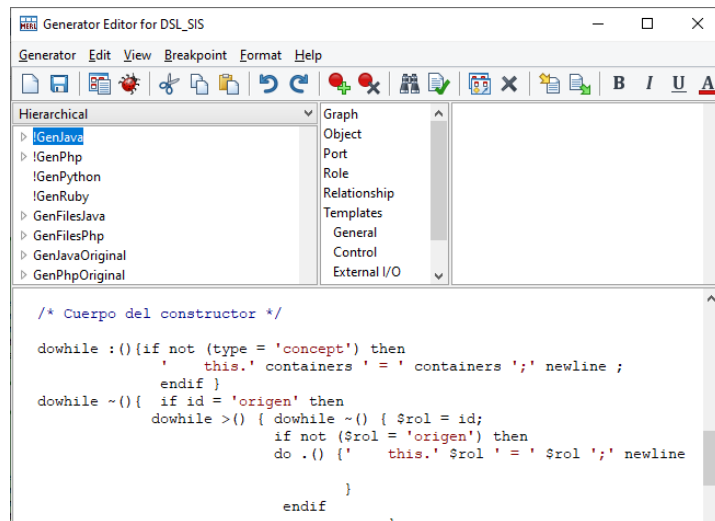


Fig. 11. Editor del Generador para DSL_SIS

Esto fue posible gracias a un lenguaje propio del metaeditor: MERL (MetaEdit Report Language), que permite navegar por los elementos de los diagramas, generando salidas con distintos formatos. En la Fig. 12 se muestra un archivo de salida, obtenido a partir de un diagrama.

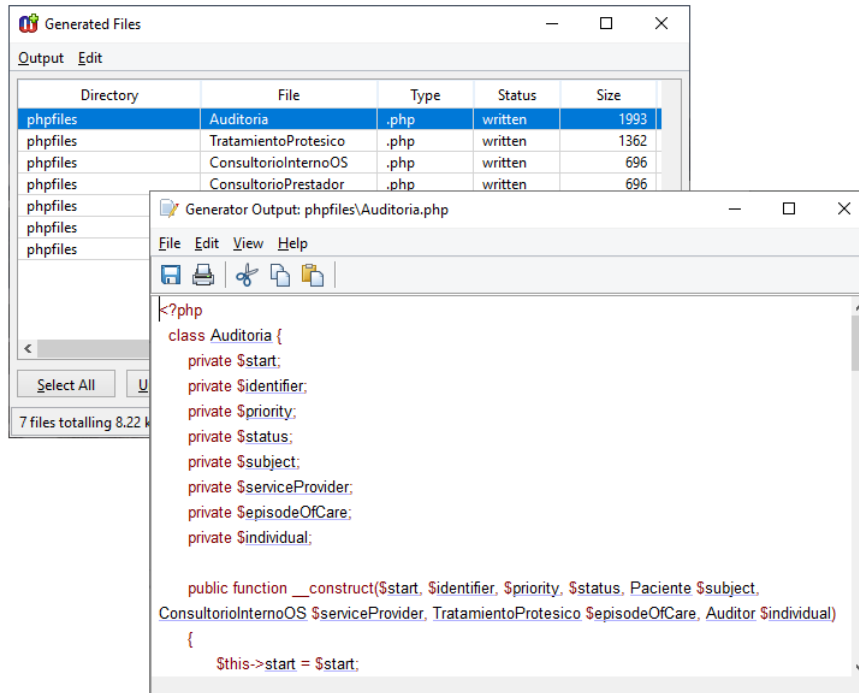


Fig. 12. Código en Php generado a partir de un diagrama

5 Aplicando el DSL_SIS a situaciones reales

En esta sección se presentan tres dominios específicos que pueden modelarse usando el DSL_SIS. Para obtener más detalles, se ofrece un video demostrativo [20]. Los archivos de código fuente, generados automáticamente para cada uno de estos modelos, también se encuentran disponibles [21].

5.1 Modelado del Protocolo COVID-19 en UFU (Unidad Febril de Emergencia)

El Ministerio de Salud de la ciudad de Buenos Aires agregó Unidades Febriles de Emergencia a gran parte de los hospitales de agudos y pediátricos, en el contexto de la pandemia de COVID-19 [22]. En la Fig. 13 puede verse un diagrama simplificado que modela la atención en una UFU de un paciente presuntamente contagiado.

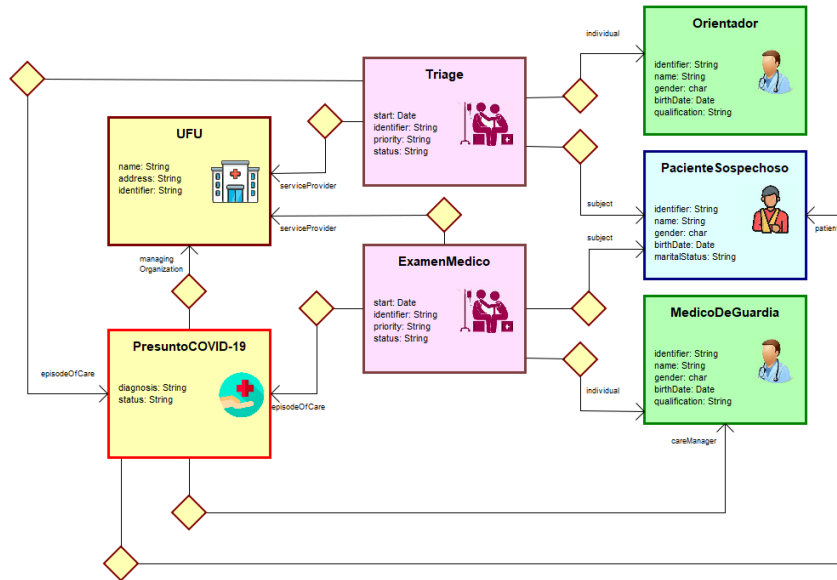


Fig. 13. Modelo Específico de Dominio (DSL_SIS): Atención en una UFU

5.2 Modelado del Control del Niño Sano

La AAP (American Academy of Pediatric) elaboró un programa de visitas periódicas que recomienda cumplir para el cuidado de un niño sano, el seguimiento de su crecimiento y desarrollo, y la prevención de enfermedades [23]. En la Fig. 14 se muestra una de dichas consultas en un Centro de Atención Primaria (CAP), modelada con DSL_SIS.

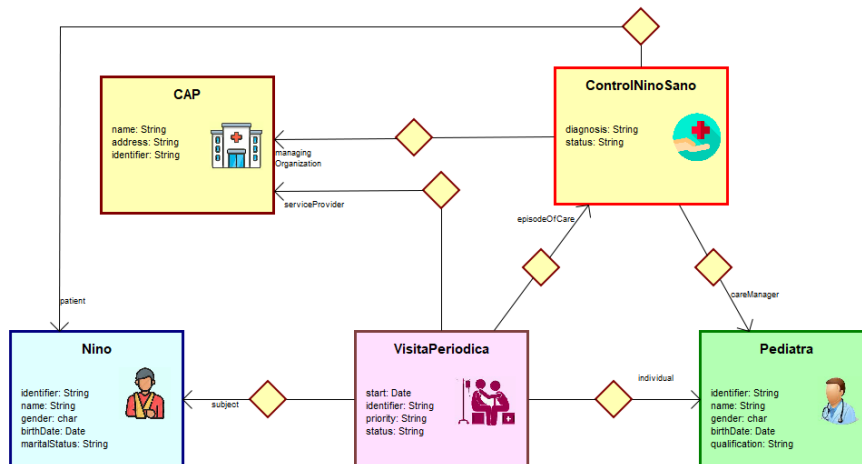


Fig. 14. Modelo Específico de Dominio (DSL_SIS): Control del Niño Sano

5.3 Modelado de los Controles de Salud Materna y Perinatal

La Organización Mundial de la Salud (OMS) promueve los controles previos y posteriores al parto, para preservar la salud de las mujeres durante el embarazo, y en el período puerperal [24]. En la Fig. 15 se representa la atención médica de una madre, por parte de un ginecólogo particular, en las instancias pre y post-parto.

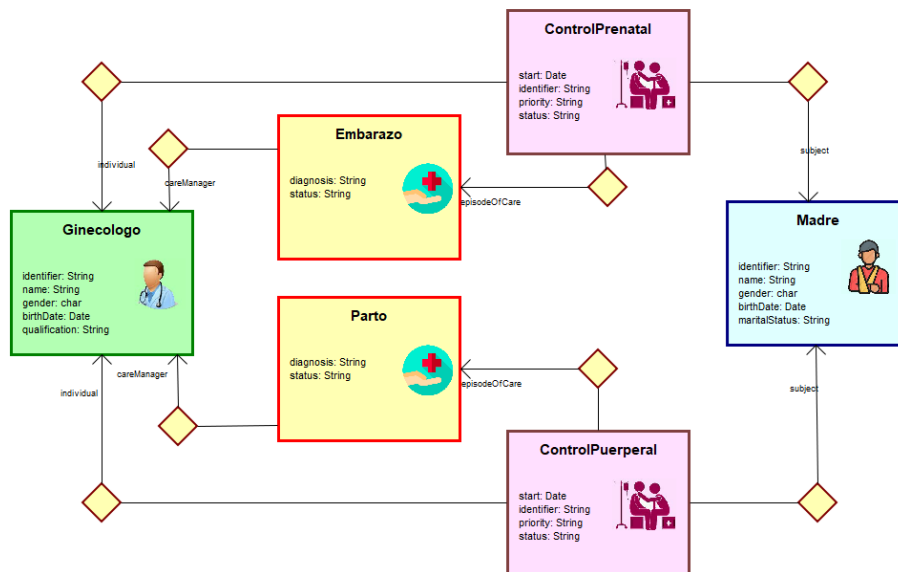


Fig. 15. Modelo Específico de Dominio (DSL_SIS): Control de Salud Materna y Perinatal

6 Conclusiones

Los sistemas de gestión sanitaria plantean dos grandes retos. Por un lado, deben ir adaptándose con el tiempo a las constantes actualizaciones tecnológicas. Y por otro lado, deben posibilitar la integración de toda la información y su disponibilidad en todos los puntos en que se necesite acceder a ella.

La primera cuestión puede resolverse abordando el problema con el enfoque del DSM, en vez de aplicar las metodologías tradicionales de desarrollo de sistemas. El poder de abstracción que provee el DSM, permite manejar la complejidad creciente de una manera rápida y clara. Ocultando los detalles de implementación, se consigue una mayor adaptabilidad. Por eso resulta muy beneficioso disponer de un lenguaje como DSL_SIS.

Y el segundo problema, el de la comunicación entre distintos sistemas, puede solucionarse recurriendo a un estándar de interoperabilidad clínica. Por eso definimos los elementos del DSL_SIS basándonos en un subconjunto de recursos significativos de FHIR.

Para desarrollar una herramienta de modelado que permitiera crear, visualizar y editar diagramas mediante el lenguaje DSL_SIS, utilizamos el metaeditor MetaEdit+ que permitió probar la expresividad del editor construido, modelando algunos casos concretos y de situaciones reales, de interés actual. Además, gracias al generador incluido en el metaeditor, se realizaron transformaciones de modelo a texto, para obtener código en diferentes lenguajes de programación. De esta manera, se logró el objetivo de automatización planeado en trabajos anteriores.

Es importante subrayar que una herramienta de estas características es muy útil para simplificar el trabajo de los analistas de negocios y diseñadores de sistemas, en las primeras fases del proceso de desarrollo del software. Cada elemento de la paleta de edición se identifica con algún objeto reconocible del dominio, y es además una unidad de interoperabilidad.

Otro aspecto importante es que, al intentar conectar dos elementos entre sí, el editor comprueba automáticamente que sea una relación válida. Y si no lo es, impide su realización, evitando errores tempranos en las etapas iniciales, que son las de mayor riesgo y desconocimiento. Esto constituye el diseño e implementación de las reglas de buena formación de los modelos basados en el DSL_SIS.

Comparando MetaEdit+ con las herramientas utilizadas anteriormente en la plataforma Eclipse, se puede afirmar que el tiempo invertido para construir el editor es notablemente menor en MetaEdit+. Además, su entorno es mucho más intuitivo y simple.

Como trabajo futuro, dando continuidad a este proyecto, se plantea la posibilidad de incorporar una vista dinámica al DSL_SIS, que permita graficar diagramas de máquina de estados para los elementos del dominio, que ofrezcan propiedades internas posibles de modelarse con estados discretos, que son los de tipo HealthEncounter. Al mismo tiempo, se espera realizar una evaluación de usabilidad y calidad interna del editor construido.

Referencias

1. Pons, C., Giandini, R., Pérez, G.: Desarrollo de Software Dirigido por Modelos: conceptos teóricos y su aplicación práctica. EDULP&Mc-Graw-Hill, La Plata (2010)
2. Kelly, S., Tolvanen, J.: Domain-Specific Modeling: Enable Full Code Generation. Wiley-IEEE Computer Society, Hoboken (2008)
3. Especificación de MOF (MetaObject Facility) <https://www.omg.org/mof/>
4. Definición del Package ECORE <http://download.eclipse.org/modeling/emf/emf/javadoc/2.6.0/org/eclipse/emf/ecore/package-summary.html>
5. Especificaciones del lenguaje de modelado de GOPRR https://www.metacase.com/support/45/manuals/mwb/Mw-1_1.html
6. Guanyabens Calvet, J.: Las TIC como instrumento estratégico en los sistemas de salud. Todo hospital (267), 320-324 (2010)
7. Foro de la OMS sobre la Estandarización y la Interoperabilidad de los Datos Sanitarios. Organización Mundial de la Salud, Ginebra (2012)
8. Resourcelist – FHIR v4.0.1, <https://hl7.org/fhir/resourcelist.html>
9. Spronk, R., Ringholm, C.: The HL7 MIF - Model Interchange Format (2010), http://www.ringholm.com/docs/03060_en_HL7_MIF.htm

10. Villegas, A., Olivé, A.: UML Profile for MIF Static Models. Version 1.0 (2013), http://www.vico.org/HL7_Tooling/Submission/MIF.pdf
11. Model Interchange Format, HL7 Wiki page for this project, https://wiki.hl7.org/index.php?title=Model_Interchange_Format
12. Martínez García, A.: Resolviendo el Diseño de Modelos de Dominio HL7 mediante Soluciones Guiadas por Modelos. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla (2016)
13. Cesaretti, J., Paganini, L., Rocca, L., Caputti, M., Zugnoni, I.: Herramienta basada en Lenguaje Específico de Dominio para Sistemas elementales de Información Sanitaria. JAIIO 48, Salta (2019)
14. Rocca, L., Caputti, M., Zugnoni, I., Paganini, L., Cesaretti, J., Nahuel, L., Giandini, R.: Marco de trabajo para el diseño y desarrollo de Herramientas de modelado conceptual basado en DSL utilizando tecnologías GMF. CIITI, Buenos Aires (2018)
15. OMG Unified Modeling Language™ (OMG UML), Infrastructure. OMG (2011), <https://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF>
16. Sirius – The easiest way to get your own Modeling Tool, <https://www.eclipse.org/sirius/>
17. MetaEdit+ Workbench – Build your own modeling tool <https://www.metacase.com/mwb/>
18. El Kouhen, A., Dumoulin, C., Gerard,S., Boulet,P.: Evaluation of modelling tools adaptation. CNRS HAL hal-00706701 (2012).
19. Tolvanen, J., Kelly. S.: Effort Used to Create Domain Specific Modeling Languages. ACM/IEEE 21th International Conference on Model Driven Engineering Languages and Systems (MODELS '18), Copenhagen, Denmark (2018).
20. DSL-SIS Un Lenguaje de Modelado Especifico del Dominio de Sistemas de Información Sanitaria, https://youtu.be/WTs6f_ZNqf8
21. Código fuente generado automáticamente con DSL_SIS, para los modelos de ejemplo, https://drive.google.com/drive/folders/11b2QuFZNqQfu_pY6MKq7jtGmypoqUM-k?usp=sharing
22. Unidades Febriles de Urgencia (UFU), <https://www.buenosaires.gob.ar/coronavirus/unidades-febriles-de-urgencia-ufu>
23. Bright Futures, <https://brightfutures.aap.org/Pages/default.aspx>
24. Salud Sexual y Reproductiva. Salud Materna y Perinatal, https://www.who.int/reproductivehealth/publications/maternal_perinatal_health/es/