

# SociAALML: Lenguaje de Modelado para Escenarios de Inteligencia Ambiental

Pablo Campillo-Sanchez, Jorge J. Gómez-Sanz, and Juan Pavón

Universidad Computense de Madrid  
{pabcampi|jpavon|jgomez}@ucm.es,  
WWW home page: <http://grasia.fdi.ucm.es>

**Abstract.** El estudio y desarrollo de sistemas de inteligencia ambiental puede llegar a ser complejo y costoso, especialmente en fases avanzadas que se ejecutan en laboratorios vivientes (*living labs*). En la última década, los avances en tecnologías 3D permiten plantear entornos virtuales donde se puedan desplegar dispositivos y usuarios. Así se podría reducir costes y potenciar la investigación de este tipo de sistemas. En este trabajo se presenta SociAALML, un lenguaje específico de dominio para modelar escenarios de la vida cotidiana en el hogar, incluyendo el modelado de personas con Parkinson. Usando modelos creados con este lenguaje, se generan simuladores 3D que incluyen usuarios simulados, el propio entorno y los dispositivos que componen el sistema de inteligencia ambiental.

**Keywords:** Simulación dirigida por modelos, Inteligencia Ambiental, Modelado de la Actividad Diaria, Modelado de Parkinson

## 1 Introducción

El desarrollo de sistemas de inteligencia ambiental (AmI) como en el caso de *Ambient Assisted Living* (AAL) se ha encontrado con dificultades para proporcionar soluciones eficaces y bien aceptadas por los usuarios finales. Una parte importante del problema reside en la necesidad de entender y adaptarse a las necesidades particulares de los usuarios finales, las personas adultas y los cuidadores, como demuestran estudios recientes [1]. Una solución es el uso de laboratorios vivientes (*living labs*). Siendo un avance, su uso plantea, sin embargo, algunas limitaciones importantes, como el coste tanto por la instalación del equipamiento y la contratación de personal para las pruebas, como por el tiempo que requiere la ejecución de los escenarios de prueba.

Una alternativa al método de *living labs* sería usar simulaciones computarizadas. Estas podrían facilitar una experimentación preliminar de bajo coste, convirtiéndose en el lugar donde se conciben y ensayan las primeras versiones del sistema AAL. Sin embargo, su uso suele requerir conocimientos en complejas herramientas de desarrollo. Es por ello que se ha desarrollado un lenguaje específico del dominio para modelar escenarios de la vida cotidiana en el hogar, incluyendo el modelado de personas con Parkinson. Este lenguaje, denominado

SociAALML, permite definir el comportamiento de las personas implicadas, sus interacciones y escenarios donde a demás se puede especificar el despliegue de la solución (dispositivos Android y el software). A partir de los escenarios, utilizando herramientas de generación de código, se obtienen simulaciones computarizadas autocontenidas que pueden usarse para validar o experimentar con desarrollos AAL.

Las simulaciones son el resultado de procesar modelos creados con SociAALML. Se basan en PHAT [2], un simulador 3D que reproduce un sistema de inteligencia ambiental donde los dispositivos son sistemas Android y donde el comportamiento de los personajes corresponde con el declarado en el modelo. PHAT fue desarrollado desde cero utilizando el motor de juegos de código abierto jME3 para conseguir simulaciones realistas de entornos virtuales 3D. El uso de motor de juegos y motores físicos permite crear situaciones donde dos actores chocan al intentar cruzar a la vez el marco de una puerta y donde el mobiliario entorpece los movimientos de los actores.

El meta-modelo de SociAALML se ha especificado con la herramienta INGENME<sup>1</sup>. Se ha elegido por su estabilidad y rapidez de uso para prototipado rápido. No es un reemplazo al EMF o al MOF, sino una forma alternativa y más sencilla de experimentar con técnicas de meta-modelado. Implementa un metalenguaje inspirado en el metalenguaje GOPRR, anterior a OMG MOF. Los elementos básicos a definir son las siglas de GOPRR: Diagrama (Graph), Entidad (Object), Atributo (Property), Relación (Relationship) y Rol (Role). INGENME permite especificar el meta-modelo de un lenguaje de modelado y genera el editor correspondiente. INGENME está integrado con Maven y permite incorporar actividades de MDD en el ciclo de vida de una aplicación.

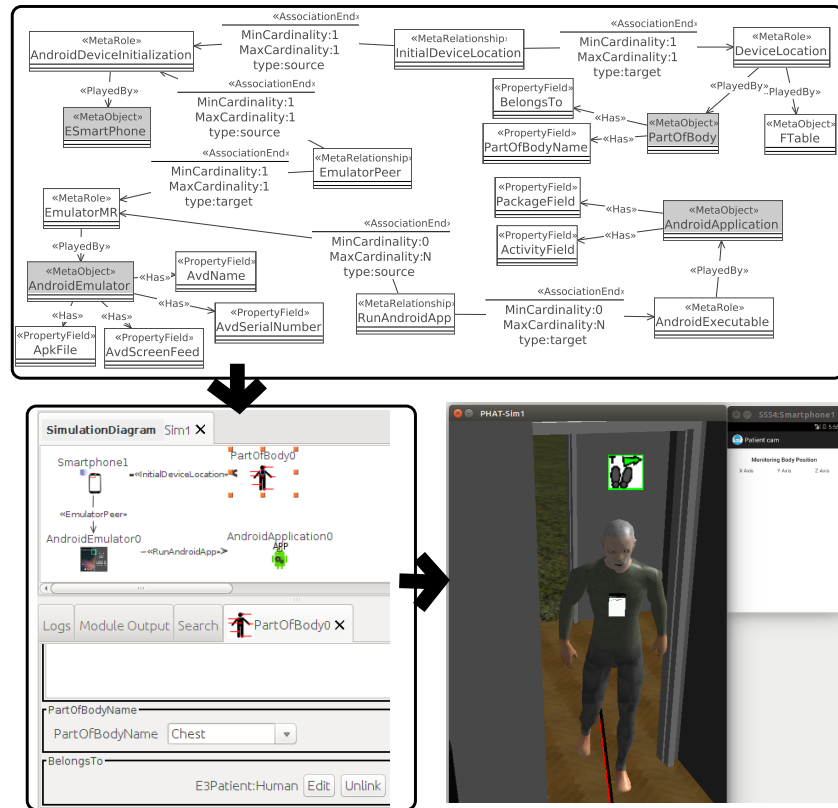
En este artículo se explica los elementos principales del lenguaje de modelado SociAALML. Su uso se ilustra con un ejemplo de cómo se recoge el efecto del Parkinson en una persona y de cómo se integran los dispositivos basados en Android en el modelo.

## 2 SociAALML y su transformación en código PHAT

SociAALML se creó desde a partir de un análisis de un trabajo de campo donde se entrevistó a 20 enfermos de Parkinson junto con sus familiares, además de a unos cuantos especialistas como neurólogos y trabajadores sociales. El análisis consistió en extraer las situaciones más preocupantes identificando sus elementos y características más representativas. (1) *La casa y sus habitaciones con muebles y objetos*. Para ello se dispone de 3 diagramas. El *BindingDiagram* especifica las plantas del edificio, el *FloorSpecDiagram* las habitaciones y su conexión y el *RoomSpecDiagram* los muebles y objetos de la habitación. La información que recoge estos elementos es necesaria para poder hacer referencia a ella posteriormente cuando se modela el comportamiento de las personas. Por ejemplo, para indicar a qué habitación se desea que vaya el personaje o en qué silla sentarse.

---

<sup>1</sup> INGENME website: <http://ingenme.sf.net>



**Fig. 1.** Arriba: una sección del meta-modelo de SociAALML. Abajo-izquierda: un modelo donde se asigna un móvil al pecho del paciente con un software de detección de caídas. Abajo-derecha: la simulación resultante del modelo.

(2) *Las personas*. Estas se modelan a partir de diagramas de perfil. Se dispone de perfil social (*HumanProfileSpecDiagram*) que recoge datos personales, características físicas y relación social con otras personas. El *ADLSpecDiagram* referencia a las actividades y tareas que realiza la persona a lo largo del día. El *InteractionDiagram* modela el comportamiento de la persona ante determinados eventos como que llamen al timbre, el paciente se cae, etcétera. El *Parkinson-SpecDiagram* modela la enfermedad del paciente, es decir, cómo evolucionan sus síntomas y las limitaciones que le provocan. Las limitaciones se modelan *Filter-Diagram* que básicamente altera la forma de hacer las tareas dependiendo del grado de los síntomas. Por ejemplo, si el nivel de temblor es alto y la persona va a encender la televisión esa tarea no la puede realizar y el filtro la reemplaza por la tarea de pedir ayuda. (3) *Escenarios*. Los escenarios se especifican en el *SimulationDiagram*. En este diagrama se especifica el día y hora de la simulación. Las personas que intervendrán así como su posición inicial, síntomas

iniciales y dispositivos que llevarán indicando la aplicación que correrá durante la simulación.

La generación de código consiste en recorrer los diagramas del modelo SociAALML con la API que ofrece INGENME y completar las plantillas de clases Java que extienden el simulador PHAT[2]. Esta plantilla se construye con una solución propia y tiene métodos de la API de PHAT para inicializar el entorno, la casa, los cuerpos de los personajes (su posición inicial), dispositivos (su colocación, emulador asignado, software a instalar) y los personajes que intervendrán. El resultado es código directamente ejecutable donde se puede seleccionar el escenario a simular. Al iniciar un escenario, se lanzan las máquinas virtuales Android necesarias por cada dispositivo virtual y se instala el software, además se abre la ventana de visualización 3D donde el usuario de forma pasiva puede comprobar el comportamiento del sistema AAL ante el escenario modelado como muestra la Figura 1.

### 3 Conclusiones

El uso de técnicas de metamodelado y generación de código puede ser bastante eficaz cuando se emplea en dominios bien definidos, como se demuestra en este caso. La herramienta INGENME permite definir sin dificultad el lenguaje de dominio concreto, en este caso SociAALML. Los resultados de SociAALML, así como demostraciones de lo que se puede generar, se pueden revisar en <http://grasia.fdi.ucm.es/social>.

### Agradecimientos

Este trabajo ha sido financiado por el gobierno de la Región de Madrid a través del programa de investigación MOSI-AGIL-CM (ayuda P2013/ICE-3019, cofinanciada por FSE y FEDER), por el proyecto “Social Ambient Assisting Living (SociAAL)”, del Ministerio de Ciencia e Innovación, con ref. TIN2011-28335-C02-01, y por el Programa de Creación y Consolidación de Grupos de Investigación UCM-Banco Santander (GR3/14) para el grupo UCM-GRASIA (921354).

### References

1. M. Arroyo, L. Finkel, and J. Gomez-Sanz. Requirements for an intelligent ambient assisted living application for parkinson patients. In J. Corchado, J. Bajo, J. Kozlak, P. Pawlewski, J. Molina, V. Julian, R. Silveira, R. Unland, and S. Giroux, editors, *Highlights on Practical Applications of Agents and Multi-Agent Systems*, volume 365 of *Communications in Computer and Information Science*, pages 441–452. Springer Berlin Heidelberg, 2013.
2. P. Campillo-Sanchez, J. Gómez-Sanz, and J. Botía. Phat: Physical human activity tester. In J.-S. Pan, M. Polycarpou, M. Woźniak, A. de Carvalho, H. Quintián, and E. Corchado, editors, *Hybrid Artificial Intelligent Systems*, volume 8073 of *Lecture Notes in Computer Science*, pages 41–50. Springer Berlin Heidelberg, 2013.