

SIMULADOR PARA CIUDADES INTELIGENTES

Ana Rubio Ruiz, Tecnóloga, Universidad de Castilla-La Mancha

David Villa, Profesor Contratado Doctor, Universidad de Castilla-La Mancha

Maria J. Santofimia, Profesora Contratada Doctora, Universidad de Castilla-La Mancha

Félix J. Villanueva, Profesor Contratado Doctor, Universidad de Castilla-La Mancha

Juan Carlos López, Catedrático, Universidad de Castilla-La Mancha

Resumen: Citisim es un simulador de Smart City que, construido sobre Blender, permite la generación automática de un plano 3D de cualquier ciudad a partir de sus coordenadas geográficas. Una vez construido el plano, Citisim puede ser utilizado con tres propósitos: la monitorización de eventos que ocurran en la Smart City, la simulación de eventos en esa ciudad virtual o como interfaz de usuario para la interacción con los dispositivos reales. El resultado de esta plataforma será una herramienta que permitirá a desarrolladores de servicios para Smart City la validación de los mismos sin necesidad de un previo despliegue de sensores reales en la ciudad, el despliegue virtual de sensores para la validación de servicios que impliquen algún aspecto de monitorización o el manejo de los sensores reales a través de una 3D-UI.

Palabras clave: Blender, OpenStreetMap, Ciudad, Inteligente, Smart, City, Simulador, Eventos, Distribuido.

INTRODUCCIÓN

El panorama actual de ciudad inteligente encabezado por ciudades como Santanderⁱ, Barcelonaⁱⁱ, Málaga o Rivas Vaciamadrid nos ofrece un conjunto de servicios generalmente inconexos, totalmente acoplado a protocolos o tecnologías específicas, que impiden la integración directa de soluciones diseñadas para otros contextos u otras ciudades. Así, el principal obstáculo para el pleno desarrollo del paradigma de la Smart City podemos encontrarlo en la heterogeneidad de soluciones actualmente implementadas por los desarrolladores de este tipo de soluciones.

De la misma manera que el uso de middlewares vino a resolver un problema similar en el contexto de los sistemas distribuidos, este proyecto plantea una solución similar, como es recurrir al uso de un middleware de comunicaciones para abstraer la heterogeneidad de los sistemas hardware y software presentes en una ciudad inteligente. Como prueba de concepto, este sistema no se limitará a proponer un middleware para la Smart City, sino que además se desarrolla una plataforma virtual para validar el mismo.

Esta plataforma virtual recibirá el nombre de Citisim y pretende integrar sensores y servicios para componer servicios más elaborados. Citisim puede dar servicio a la integración de cualquier tipo de sensor y servicio, como el control de los elementos que intervienen en la circulación de la ciudad, control de temperatura en algunas zonas, monitorización del paso por lugares determinados u otros aspectos que sean de interés.

Citisim será una herramienta de virtualización que dependerá de la estructura que tenga cada ciudad, de ahí la necesidad de manejar un callejero de la misma. La representación manual del callejero de cada ciudad sobre la que se quisiera trabajar haría que un software de este tipo fuera poco útil. Es necesario, por lo tanto, dar respuesta a la necesidad de representar automáticamente callejeros a partir de unas coordenadas geográficas. De esta forma se conseguirá la representación automática de la ciudad sobre la que realizar la simulación o monitorización, así como información adicional relacionada con la situación de determinados objetos o su comportamiento.

Como se verá más adelante se pretende conseguir que la representación del escenario de trabajo sea realista, en el sentido de que éstas estén basadas en datos reales de la ciudad, o ficticia, si los datos necesarios para llevar a cabo la representación no existen.

En definitiva este proyecto proporcionará una plataforma para la representación de ciudades a las que poder añadir diferentes tipos de sensores y dispositivos para realizar simulaciones, en entornos reales o ficticios, de forma sencilla y, en su mayor parte, de forma automática.

MATERIAL Y MÉTODOS

A continuación se describirán la metodología y los medios utilizados en el proceso de desarrollo del proyecto Citisim.

Métodos

Aunque para el desarrollo del proyecto se ha trabajado con metodologías ágiles (que permiten que el proceso sea adaptativo) también se establecieron distintas fases generales basadas en los objetivos a alcanzar, que veremos más adelante:

1. **Elección de los medios a utilizar.** Se llevó a cabo un estudio sobre cuáles eran los medios, hardware y software, que se adaptaban mejor al proyecto y que veremos en el siguiente apartado.
2. **Representación de la ciudad.** Una vez elegidos los medios, utilizando los mismos se procedió a la representación del escenario de trabajo, pues era esencial para poder incorporar el resto de elementos que componían el sistema.
3. **Establecimiento de los elementos a emplear.** Se especificaron cuáles eran los sensores, dispositivos y objetos necesarios para que el sistema pudiese funcionar correctamente: las formas de representación en la escena (semáforos, farolas), qué tipos de sensores se iban a desplegar para poder llevar a cabo las simulaciones (sensor de presencia, temperatura) y se indicaron los dispositivos a emplear.
4. **Desarrollo del middleware.** Implementación de un middleware de comunicaciones que permitiese, mediante eventos, que sensores y demás elementos que conforman el sistema pudieran comunicarse a través de eventos independientemente de qué sensor/dispositivo se utilizara.

Material

Para la realización de este proyecto, además de la adquisición de una estación de trabajo que fuese capaz de soportar el renderizado de imágenes de la Smart City y sus representaciones 3D, se realizó un análisis de distintos programas, lenguajes y otros a utilizar:

- **Fuente de información.** Los datos que se emplearían para la representación de nuestra Smart City en 3D serían los contenidos en los archivos *OSM* que proporciona *OpenStreetMap*ⁱⁱⁱ. *Google Maps* también fue una de las opciones que se contemplaron para el desarrollo del proyecto, pero *OpenStreetMap* fue el elegido por tratarse de software libre y por el formato en el que proporciona los datos del mapa, más fáciles de procesar.
- **Representación 3D.** *Blender*^{iv} es el medio a través del cual representaremos nuestra Smart City en 3D junto con los sensores y dispositivos que vayamos a desplegar en ella. Aquí, al igual que para la elección de la fuente de información se analizó otro software llamado *Unity*, pero *Blender* fue elegido finalmente por ser, una vez más, software libre y por utilizar Python como lenguaje de programación.
- **Middleware de comunicaciones.** Como lenguaje para realizar las comunicaciones a través de eventos entre los distintos objetos que se utilicen para la representación de los dispositivos o sensores en la ciudad se emplearía *ZeroC Ice*^v.
- **Alojamiento.** *Bitbucket* es un sistema de alojamiento basado en web que es utilizado en nuestro caso para el control de versiones.

EL PROYECTO

Aunque ya se ha descrito de forma general cuál es el objetivo de Citisim a continuación se detallarán cuáles fueron los objetivos que se establecieron en un principio de acuerdo a los resultados que se deseaban obtener:

1. **Representación de la ciudad.** Uno de los objetivos principales era la automatización del proceso por el cual se construye el escenario en el que se trabaja, una ciudad en nuestro caso. A partir de unas coordenadas geográficas se obtiene dicha ciudad, o el fragmento de la misma, sobre el cual se lleva a cabo la simulación. Esas coordenadas permiten obtener el archivo OSM de *OpenStreetMap* con los datos relativos a calles, edificios, semáforos, farolas u otros elementos relevantes en función de la simulación a realizar.
2. **Desarrollo de un middleware de comunicaciones.** Integrar distintos servicios que en otros proyectos se habían tratado de forma individual. Para ello se necesitaba un middleware de comunicaciones que diese soporte a distintas plataformas de sensorización que se pudiesen desplegar en nuestra ciudad virtual, ya fuera por medio de los datos recogidos de los archivos OSM (datos reales) o a través de algún otro medio que diese la posibilidad de representar situaciones ficticias en nuestra ciudad. Este middleware permitiría la comunicación independientemente del tipo de dispositivo o sensor.
3. **Establecimiento de un mecanismo para la simulación de escenarios concretos.** Como se quiere dar la posibilidad al usuario de representar situaciones ficticias, es decir, situaciones en las que los datos necesarios para su representación no se encuentran contenidos en los archivos OSM de *OpenStreetMap* por requerir de elementos inexistentes en la ciudad real, se tuvo que establecer un medio para especificar dónde situar cada uno de esos elementos para poder simular nuestro escenario concreto.
4. **Establecimiento del medio para la representación de determinados elementos.** Para especificar qué tipo de sensor o dispositivo ficticio queremos desplegar, en qué lugar y cómo queremos que funcione también se tendría que establecer un modo por el cual recoger los datos necesarios para su representación con un formato que el sistema pudiese procesar para añadir dichos elementos a nuestra ciudad, de igual forma que si éstos figurasen en el archivo OSM.

Representación

En primer lugar se representa de forma automática la ciudad sobre la que realizar la monitorización o simulación. Para ello se utilizan los archivos *OSM* de *OpenStreetMap*, que nos proporcionan los datos necesarios para la representación, y *Blender*, software de diseño 3D de código abierto para visualizar los edificios, carreteras y elementos (farolas, semáforos u otros) de dicha ciudad.

La aplicación desarrollada permite generar la ciudad a partir de un archivo OSM descargado previamente de la web, en la que ya habremos seleccionado el área a representar. Este archivo contiene las coordenadas de los nodos que componen los elementos de la ciudad (o de nuestro fragmento de ciudad) y, mediante etiquetas, la especificación del tipo de nodo indicando el objeto al que pertenecen: carreteras, caminos o edificios, compuestos por un conjunto de nodos; o farolas y semáforos, por ejemplo, formados por un único nodo. La aplicación procesa el archivo *OSM* y representa en *Blender* los nodos que contiene, pasando las coordenadas geográficas a datos que pueda entender este software para posicionar cada objeto correctamente en la escena^{vi}.

Adición de objetos

Además de mostrar en *Blender* la ciudad, entendiendo que ésta está formada por vías y los edificios, lo que se pretende es dar la posibilidad al usuario de llevar a cabo monitorizaciones y simulaciones representando otros objetos. La forma en la que montaremos la escena dependerá de cuál sea nuestro objetivo: trabajar en un escenario real o ficticio.

1. **Escenario real.** *OpenStreetMap* nos da todo lo que necesitamos para construir la ciudad sobre la que trabajar, pero también puede proporcionarnos información sobre otros elementos de la misma que no sean los edificios y las vías: la situación de pasos de cebra, semáforos, farolas o incluso postes de luz. Además de determinar las coordenadas geográficas de dichos elementos *OpenStreetMap* da la posibilidad de indicar otras muchas características. Uno de los casos con los que trabajamos es la representación de los semáforos que hay en un área determinada gracias a que, mediante etiquetas, podemos saber el tipo de objeto y su orientación (0º = norte).

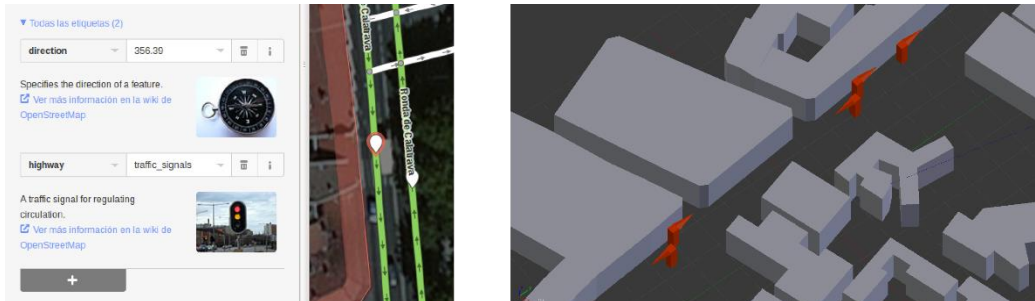


Figura 1. A la izquierda edición de las etiquetas con iD (editor del navegador). A la derecha área en Blender en la que se han importado "semáforos" (representados aquí por una flecha para que se vea más claramente su orientación).

2. **Escenario ficticio.** En vez de importar en la escena objetos que se encuentren en *OpenStreetMap*, como en el caso anterior, una vez que el usuario haya indicado la información necesaria (posición y orientación, por ejemplo, en el caso del semáforo) y sea almacenada serán esos objetos (que no existen en realidad) los que se representen en la escena.

Renderizado de imágenes

Además de generar la ciudad e importar objetos en ella se da la posibilidad de renderizar la escena completa, generándose así una imagen de nuestra ciudad.

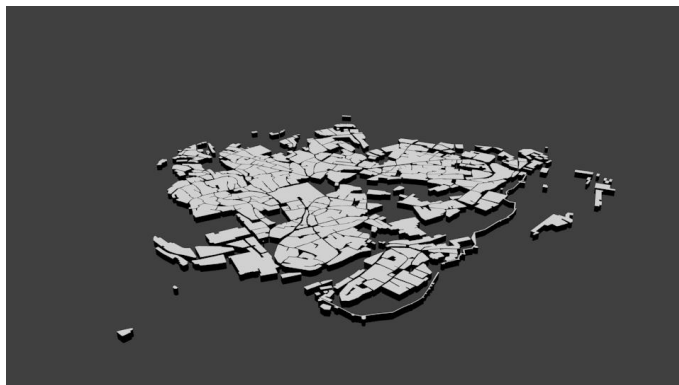


Figura 2. Renderizado del casco antiguo de la ciudad de Toledo.

Usos y comunicación

Citism integra sensores y actuadores que pueden ser reales o artefactos puramente virtuales. Todas las comunicaciones se basan en un middleware orientado a objetos y es, por tanto, independiente de la localización o de los detalles de implementación. Sus principales usos son tres:

1. **Simulador.** Cualquier Smart City tiene que tratar con diferentes tipos de detalle que requieren una determinada infraestructura, incluso en los casos más sencillos. El “Simulador Citisim” permitirá controlar acciones complejas, como las relacionadas con el tráfico. Esas acciones se detectarán por medio de sensores desplegados en la ciudad virtual, para ser transformadas en eventos que mandará al middleware. Los servicios serán ejecutados en un computador que utilice esos eventos virtuales y que reaccione para solucionar posibles problemas u otras cuestiones, simplificando así el proceso de desarrollo y testing de nuevos servicios.

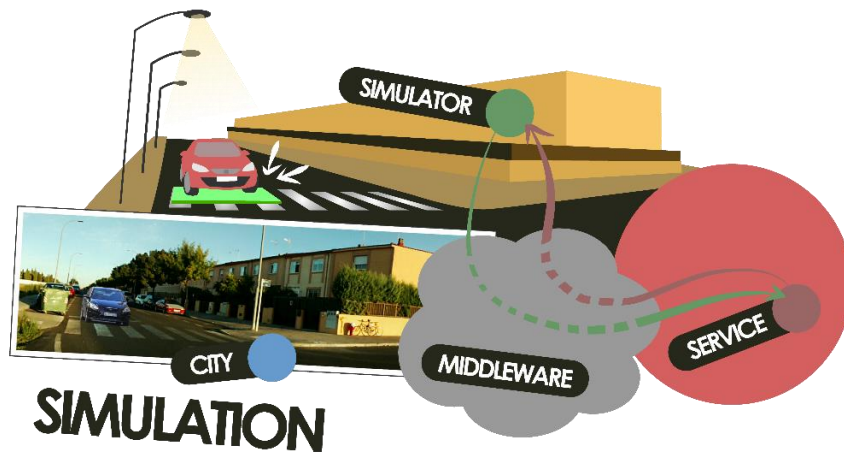


Figura 3. Diagrama de comunicaciones del Simulador.

2. **Monitor.** Su objetivo será el de manejar y visualizar grandes cantidades de información heterogénea. El “Monitor Citisim” utilizará la información generada por los sensores y actuadores desplegados en la ciudad real para representar la situación actual en el mismo entorno 3D. El usuario, además, podrá interactuar con los dispositivos representados para obtener información.

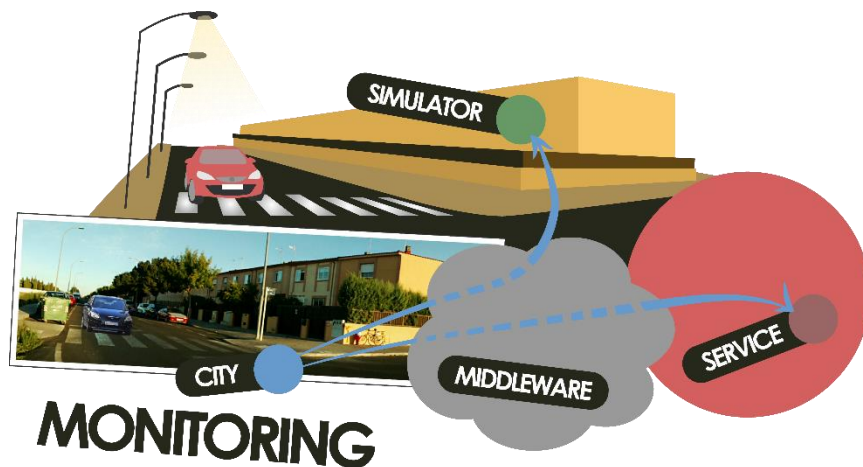


Figura 4. Diagrama de comunicaciones del Monitor.

3. **3D-UI.** Citisim podrá, también, actuar como una avanzada Interfaz de Usuario, permitiendo la interacción directa con los dispositivos desplegados. Por ejemplo, para controlar la iluminación en una vía simplemente tocando la pantalla.

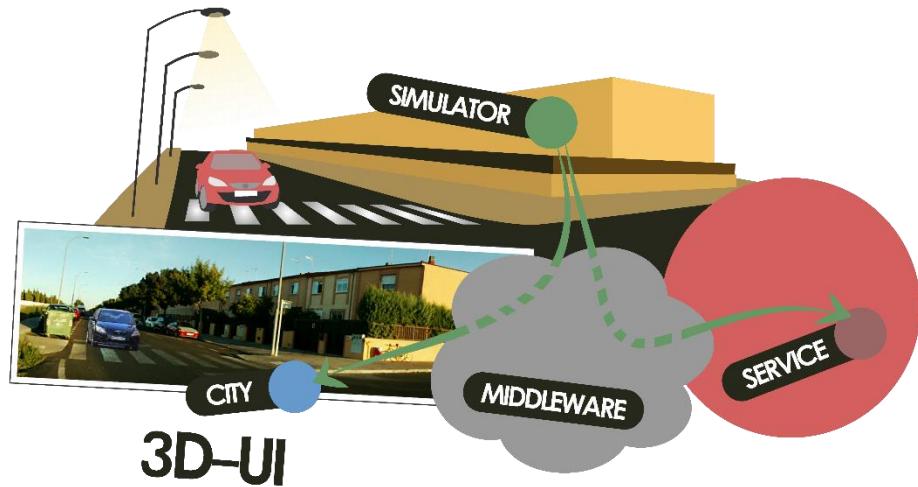


Figura 5. Diagrama de comunicaciones del 3D-UI.

RESULTADOS Y CONCLUSIONES

En definitiva, esta plataforma es una herramienta que permite a desarrolladores de servicios para Smart City la validación de los mismos sin necesidad de un previo despliegue de sensores o actuadores reales en la ciudad. Por otro lado, permitirá también el despliegue virtual de sensores para la validación de servicios que impliquen algún aspecto de monitorización o, incluso, el manejo de los distintos dispositivos realmente desplegados en la ciudad a través de una 3D-UI, facilitando el control los servicios integrados. Todo ello dando la posibilidad de que el conjunto de servicios mencionado no vayan necesariamente acoplados a protocolos o tecnologías específicas permitiendo la integración directa de soluciones diseñadas para diferentes contextos y ciudades.

Los resultados obtenidos pueden verse en el vídeo *CITISIM Project*^{vii} del grupo de investigación de Arquitectura y Redes de Computadores (ARCO) de la Universidad de Castilla-La Mancha, donde se muestran algunas de las posibilidades que ofrece *Blender* y el uso del proyecto para (en este caso) el control del tráfico en las calles de Ciudad Real.

AGRADECIMIENTOS

Este proyecto ha sido financiado por el Ministerio de Economía y Competitividad Español bajo el proyecto REBECCA (TEC2014-58036-C4-1-R) y por la Junta de Comunidades de Castilla-La Mancha bajo el proyecto SAND (PEII_2014_046_P).

ⁱ www.smartsantander.eu

ⁱⁱ www.smartbarcelona.cat

ⁱⁱⁱ <http://wiki.openstreetmap.org/wiki>

^{iv} <https://www.blender.org/>

^v <https://zeroc.com/>

^{vi} La construcción de la ciudad y el posicionamiento de objetos son posibles gracias al plugin *io_import_scene_osm.py* de **Vladimir Elistratov**, que importa edificios y vías de *OpenStreetMap*.

^{vii} <https://www.youtube.com/watch?v=yBp7-udU6Js>