

Trayectorias semánticas en aplicaciones de *Mobile Workforce Management* *

Nieves R. Brisaboa, Miguel R. Luaces, Cristina Martínez Pérez, and Ángeles S. Places

Universidade da Coruña, Laboratorio de Bases de Datos, A Coruña, Spain
{brisaboa,luaces,crisrina.martinez,asplaces}@udc.es

Abstract. Los *smartphones* actuales presentan continuamente mejoras en sus características y en la actualidad incluyen diversos sensores que capturan información de muy diversos tipos (localización, aceleración lineal, etc.). Un proceso industrial que podría beneficiarse mucho de esta información es el de *Mobile Workforce Management (MWM)*. Sin embargo, existen varios problemas que lo impiden: i) hoy en día el nivel de abstracción de las actividades que son identificadas es demasiado bajo (por ejemplo, *moviéndose* en vez de *realizando una inspección en un cliente*, o *parado* en vez de *cargando un camión en la instalación de un cliente*); ii) los trabajos de investigación se centran en el uso de algoritmos que contrastan la información geográfica con los datos del GPS, o en algoritmos de aprendizaje aplicados a los datos de los sensores, pero existen pocos resultados de investigación que combinen ambos tipos de datos; y iii) la información contextual procedente de los repositorios de información geográfica o del software MWM es raramente usada.

En este artículo se presenta una nueva metodología que convierte los datos crudos capturados por los sensores de los dispositivos móviles en trayectorias anotadas con actividades semánticas en un alto nivel de abstracción. La metodología está basada en la definición de *taxonomías de actividades* que pueden ser adaptadas fácilmente a las necesidades de cualquier empresa. Estas taxonomías describen los valores esperados para cada una de las variables que son recogidas en el sistema usando predicados definidos mediante un lenguaje de especificación de patrones.

Keywords: trayectorias semánticas, *mobile workforce management*, datos de sensores, sistemas de información geográfica

1 Antecedentes y motivación

En la última década el uso de dispositivos móviles ha crecido incesantemente y más de un 60% de las personas que viven en economías avanzadas posee un *smartphone* [8]. Como consecuencia, las características de dispositivos móviles

* Financiado por MINECO (PGE & FEDER) [TIN2013-46238-C4-3-R, TIN2013-46801-C4-3-R, TIN2015-69951-R, TIN2016-78011-C4-1-R, TIN2016-77158-C4-3-R]; CDTI and MINECO [Ref. IDI-20141259, Ref. ITC-20151305, Ref. ITC-20151247].

como *smartphones*, *tablets* y *wearable devices* han ido incrementándose constantemente. En los últimos años su poder computacional es comparable al de un ordenador de escritorio, y estos dispositivos incluyen múltiples sensores que pueden ser usados para medir diferentes variables como la posición geográfica usando el GPS, la actividad del usuario usando el acelerómetro, o información del entorno usando por ejemplo el termómetro. Un proceso industrial que podría beneficiarse especialmente del uso de esta información capturada por los dispositivos móviles es *Mobile Workforce Management (MWM)*. Los sistemas MWM son usados por las compañías para manejar y optimizar la programación de tareas de sus trabajadores (por ejemplo, asegurarse de que una empresa tenga el mínimo número de empleados activos desplegados en cualquier momento de la jornada laboral) y para mejorar la actuación de sus procesos de negocio (por ejemplo, detectar qué tareas son costosas para una compañía). Como un ejemplo, sería beneficioso para una empresa que recoge residuos orgánicos en zonas rurales detectar si los trabajadores invierten demasiado tiempo en la actividad *repostar*, en ese caso sería una buena idea decidir que se contrate a alguien para realizar esta tarea por las noches.

Este tipo de estudios son imposibles de llevar a cabo sin un histórico de los datos de los movimientos de los trabajadores y un procedimiento para detectar y analizar, a un alto nivel de abstracción, las actividades que los empleados llevan a cabo y cuanto tiempo invierten en cada una de ellas. Si el sistema MWM puede detectar qué ha pasado realmente en la jornada laboral y lo compara con lo que ha sido previamente planificado, entonces se podrá generar información muy útil para gestionar procesos de negocio y detectar aquellos puntos críticos. Sin embargo, los sistemas MWM actuales no usan la información que es recogida por los dispositivos móviles. A lo sumo, usan los servicios de localización para registrar cuando llega a la ubicación del cliente, pero no hacen uso de la gran variedad de datos recogidos por los sensores de los dispositivos móviles para responder a consultas tales como *¿Estaba el trabajador en un atasco y por eso llegó tarde?* o *¿Cuánto tiempo invierte en cada visita a la empresa del cliente?*, ya que se trata de datos masivos y de gran complejidad.

En este trabajo se presenta una metodología basada en taxonomías de actividades para el anotado de trayectorias en entornos MWM. Este sistema puede ser utilizado para recoger la información capturada por los sensores de los dispositivos móviles, analizar y anotar la trayectoria con actividades con un alto nivel de abstracción utilizando información del contexto del sistema MWM y proveedores de información geográfica, y finalmente proporcionar la información de la trayectoria semántica de nuevo al sistema MWM con el fin de apoyar el proceso de inteligencia de negocios. El resto de este documento se organiza de la siguiente manera. En primer lugar, la sección 2 describe trabajo previo relacionado. A continuación, la sección 3 describe la metodología de anotación y las taxonomías de actividades que forman la base de la metodología. Luego, la Sección 4 describe el lenguaje de especificación de patrones de actividad que se utilizan en las taxonomías para decidir qué actividad se está realizando. Finalmente, la Sección 5 presenta las conclusiones y trabajo futuro.

2 Trabajo relacionado

El interés en organizar de manera coherente y etiquetar las secuencias de puntos espacio-temporales para manejarlos comienza con los trabajos preliminares [1,11,12] donde la idea de asignar palabras semánticas clave a los datos crudos obtenidos con el GPS es presentada por primera vez. Sin embargo, esos trabajos asumen que las palabras semánticas clave son obtenidas de otras fuentes y no del GPS en si mismo. Esto es, el sistema analiza la secuencia de datos GPS y los organiza en conjuntos de manera coherente, pero, en vez de etiquetarlos, los presenta para ser anotados por el propio usuario. Desde entonces, una nueva línea de investigación ha surgido en la comunidad científica dedicada a anotar las trayectorias de objetos en movimiento, y el centro de interés se ha desplazado desde los datos crudos obtenidos del GPS a un mayor nivel de abstracción orientado a las necesidades concretas de cada aplicación.

Últimamente han sido muchos los trabajos de investigación en el campo de las trayectorias semánticas abordando los problemas de recolección, modelado, almacenamiento y análisis de estos datos. Una trayectoria semántica es un conjunto de episodios en el que cada uno consta de una hora de inicio, una hora de fin y una palabra semántica clave. Un buen resumen de la investigación y los retos en el campo se puede encontrar en [7]. Sin embargo, la investigación sobre trayectorias semánticas se centra en la detección de actividades de bajo nivel como *parado*, *moviéndose*, *caminando*, o *corriendo* en vez de actividades de alto nivel como puede ser *parado en un atasco de camino a las instalaciones del cliente*, o *cargando el camión en las instalaciones del cliente*. La razón de esta diferencia en el nivel de abstracción es que la investigación sobre trayectorias semánticas está dirigida al caso general y hay un uso limitado de información de contexto, a lo sumo contrastan los datos con la información geográfica donde se realizan paradas durante un tiempo mínimo, por ejemplo el trabajo [3], mientras que en un sistema MWM la información de contextual (por ejemplo, la tarea del trabajador), no solo la geográfica, es extremadamente importante.

Algunas de las técnicas propuestas en el campo de investigación de las trayectorias semánticas se han centrado en el uso de técnicas de aprendizaje máquina para la detección de actividades [2,9,10]. Sin embargo, estas técnicas no se pueden aplicar en sistemas MWM porque requieren datos de entrenamiento, los cuales resultan muy costosos de registrar. Por ejemplo, en un sistema MWM esto significaría que los empleados deban anotar para cada tarea que realizan durante su jornada laboral, la hora de inicio y fin, lo cual es un proceso costoso y tedioso. Además, habría un gran margen de error ya que resulta muy complicado para un empleado realizar su trabajo y, al mismo tiempo, recoger los datos de muestra para el proceso de entrenamiento. El empleado podría olvidarse fácilmente de anotar el instante inicial o el final de una actividad y por lo tanto, el conjunto de datos de muestra sería incorrecto. Por último, esto podría provocar que los empleados cometan errores en sus propias tareas. Por lo tanto, la anotación de trayectorias semánticas utilizando técnicas de aprendizaje máquina no parece demasiado apropiado.



Fig. 1. Trayectoria anotada con actividades semánticas

Otros trabajos de investigación se han centrado en modelar, representar, procesar, consultar y extraer las trayectorias de objetos a partir de sus datos *crudos* sin procesar. Estas líneas de investigación definen lenguajes de consulta y modelan las rutas de los objetos en movimiento para que patrones específicos puedan ser extraídos más tarde usando los lenguajes de consulta. En [4] los datos crudos son divididos en episodios *stop* y *move* [11], y la información geográfica relevante del dominio de la aplicación se asigna a cada episodio. Luego, los autores definen un lenguaje de consulta de minería de datos para trayectorias semánticas (ST-DMQL), de manera que puedan extraer conocimiento (minería de datos) de los datos sin procesar usando consultas y patrones de movimiento. Los autores de [6] proponen la idea de que una trayectoria está formada por segmentos de diferentes niveles de abstracción teniendo en cuenta diversos criterios (por ejemplo, *stops* y *moves* frente al sistema de transporte usado). En [5] los autores definen una *trayectoria simbólica* como un conjunto de unidades compuestas por un intervalo de tiempo, además de un valor que puede ser de tipo *label*, *labels*, *place* or *places*. Los autores también definen un lenguaje de consulta con operadores para filtrar y recuperar esas trayectorias simbólicas. A pesar de que todos estos enfoques pueden ser utilizados para consultar eficientemente los datos de la trayectoria, todas son técnicas de bajo nivel de abstracción y difícilmente pueden integrarse en un sistema MWM porque requeriría que un encargado de la empresa, no experto en el área, escribiera consultas complejas.

Finalmente, además de los problemas mencionados anteriormente, todas las soluciones son genéricas y no asumen ningún conocimiento adicional más allá de los datos de la trayectoria. Sin embargo, en el caso de la integración con un sistema MWM, es necesario tener en cuenta que se conoce información adicional y debe incluirse.

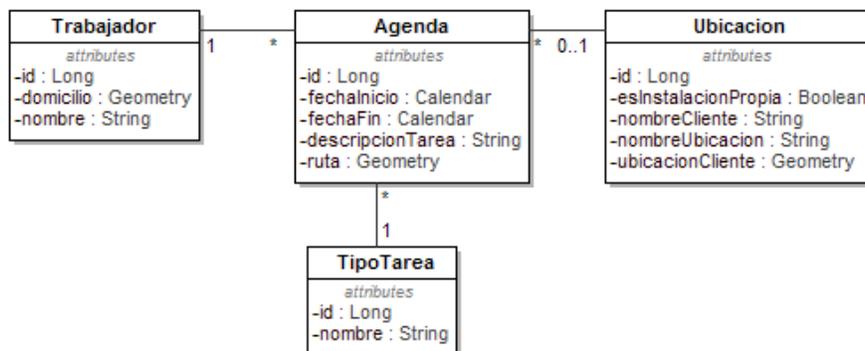


Fig. 2. Información mínima requerida procedente del sistema MWM

3 Metodología de anotación

La propuesta del proceso de anotado es dividir los datos crudos capturados por los dispositivos móviles durante la jornada de trabajo en una colección de actividades, cada una consiste en una fecha de inicio, una fecha de fin y una etiqueta que describe la actividad realizada por el trabajador. Por ejemplo, la figura 1 muestra una trayectoria anotada en la que el trabajador estaba realizando la actividad *conduciendo en ruta planificada* entre las 09:00 y las 09:10.

Los datos crudos usados para construir y anotar la trayectoria son recuperados desde 3 fuentes diferentes: *el sistema MWM*, *los sensores del dispositivo móvil*, y *la información geográfica del dominio*. Respecto al sistema *MWM*, el modelo conceptual de la Figura 2 muestra la información mínima requerida para el proceso de anotación. En primer lugar, es necesario conocer la información de todos los trabajadores de la empresa (representada en la clase **Trabajador**). Además, debemos conocer todos los lugares a los que los empleados tienen que ir en su jornada laboral (representados en la clase **Ubicacion**). Cada ubicación se describe con el nombre del cliente, el nombre y la posición geográfica de la ubicación y un valor booleano para indicar si la ubicación es una instalación de la propia empresa o es una instalación de un cliente. Finalmente, la agenda de cada trabajador es representado por la clase **Agenda**. Esta clase describe, para cada tarea realizada por cada trabajador, la hora de inicio y fin programada, el tipo de tarea (representado por **TipoTarea**), una descripción de la tarea y la ruta planificada que el empleado debe seguir para llegar a su destino planificado previamente por el sistema *MWM*.

La figura 3 muestra el modelo conceptual de la información recogida por los sensores del dispositivo móvil. Cada trabajador (representado por la clase **Trabajador**) está relacionado con todos los valores recogidos por los sensores. La información del sensor se representa mediante la clase **DatosSensor**, que almacena el tipo de sensor, el intervalo de tiempo en el que se registró el valor y

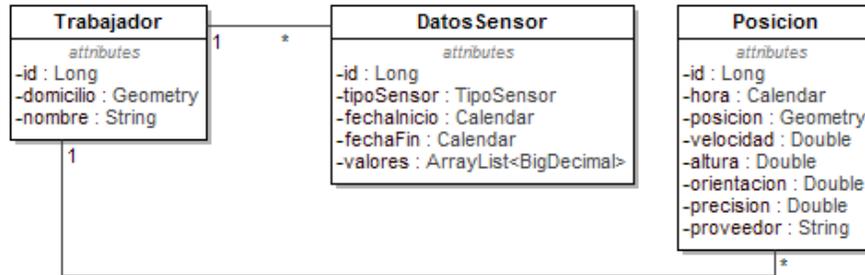


Fig. 3. Datos crudos registrados por el dispositivo móvil

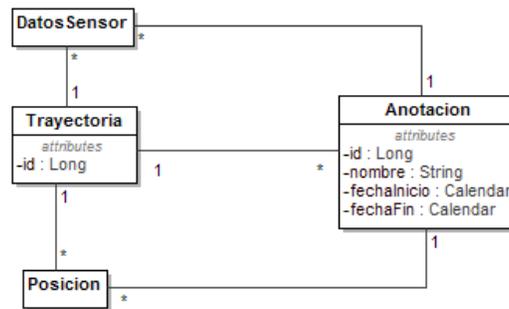


Fig. 4. Trayectoria anotada

el propio valor. Además, se almacenan todas las ubicaciones de los trabajadores (representadas por la clase `Posicion`) almacenando las variables recogidas por el GPS: el instante en el que se registra el valor, la ubicación geográfica, la velocidad, la altura, la orientación, la precisión y el proveedor de la ubicación.

Además de la información proporcionada por el software MWM y la información recogida por los sensores del dispositivo móvil, el proceso de anotación necesita la información geográfica del dominio para comparar la ubicación del trabajador con el contexto (la red de carreteras o puntos de interés específicos del dominio tales como gasolineras). Esta información se considera externa al sistema de anotación, pero se utiliza en este proceso de etiquetado y debe estar disponible para que el sistema pueda usarla. Hay dos maneras posibles de recuperar esta información: la primera es accediendo a una base de datos local mediante consultas SQL y la segunda mediante consultas a OpenStreetMap utilizando la API Overpass¹. En ambos casos, el resultado es un conjunto de objetos geográficos que pueden ser utilizados en la evaluación del patrón de actividad.

¹ <https://overpass-api.de/>

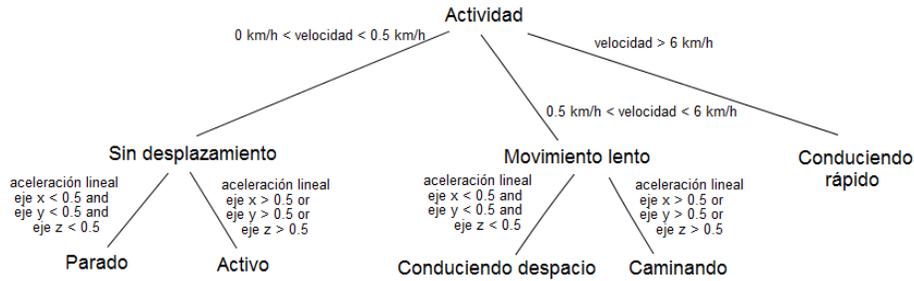


Fig. 5. Ejemplo de una taxonomía de actividades

El modelo conceptual de la figura 4 muestra el resultado esperado del proceso de anotación. Una trayectoria cruda es un conjunto de datos crudos representados por las clases `DatosSensor` y `Posicion`. Una trayectoria anotada consiste en un conjunto de anotaciones (representadas por la clase `Anotacion`). Cada anotación almacena el nombre de la actividad, el inicio y fin de la actividad y hace referencia a los datos crudos que componen la actividad.

La base del proceso de anotación es una *taxonomía de actividades*. Para cada uno de los tipos de trabajadores en la empresa debe definirse una taxonomía de actividades que incluya todas las actividades que puedan ser realizadas por el tipo de trabajador y que incluye las reglas para decidir qué actividad se realiza en cada intervalo de tiempo. La Figura 5 muestra un ejemplo de una taxonomía de actividad. Los nodos hoja de la taxonomía son las actividades que puede realizar un trabajador (por ejemplo, *parado*, *activo*, *conduciendo despacio*, *caminando*, *conduciendo rápido*). Cada nodo intermedio está formado por un conjunto de aristas. Cada arista de la taxonomía está representada con un *patrón de actividad* que indica los valores esperados para las variables que se registran en el sistema (por ejemplo, $0\text{ km/h} < \text{velocidad} < 0,5\text{ km/h}$).

La taxonomía de actividades para cada trabajador se almacena en un catálogo cuyo modelo conceptual se muestra en la figura 6. En este catálogo, cada trabajador de la empresa (representado por la clase `Trabajador`) está asociado con una taxonomía de actividades representada por la clase `Taxonomia`. La definición del predicado asociada a cada arista de la taxonomía se realiza mediante un lenguaje de especificación que se describe en la Sección 4 y que está representado por la clase `PatronEspecificacion`. Por último, una taxonomía puede estar asociada a más de un empleado ya que las actividades identificadas para todos los empleados de la empresa de la misma categoría serán las mismas.

El procedimiento para anotar las trayectorias se describe en el algoritmo 1. El primer paso es obtener todos los trabajadores de la empresa. Entonces, para cada trabajador, se recupera la taxonomía de actividad correspondiente al tipo de trabajador. Después de eso, los datos sin procesar se recuperan para el trabajador. Como se muestra en la Figura 3, los datos del sensor consisten en una

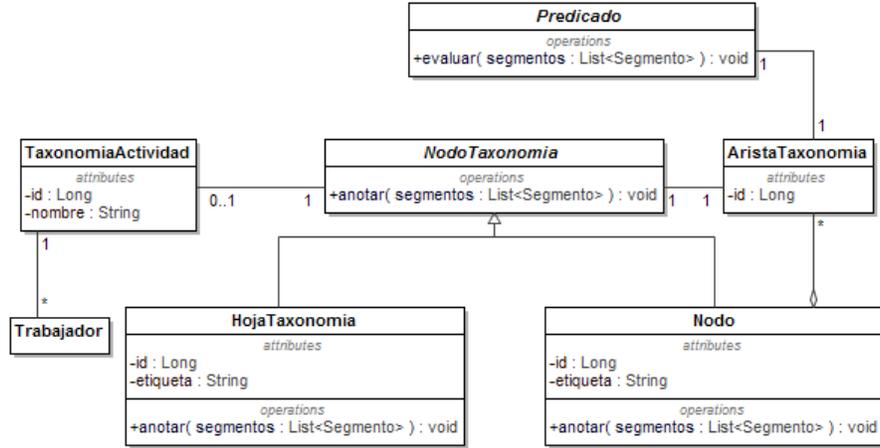


Fig. 6. Catálogo de taxonomía de actividades

Algorithm 1 Algoritmo para anotar trayectorias

```

function ANNOTATETRAJECTORIES(currentDate: Timestamp)
    workers ← retrieveWorkers()
    for all aWorker ∈ workers do
        ▷ Se recupera la taxonomía de actividades
        activityTaxonomy ← retrieveActivityTaxonomy(aWorker)
        rootNode ← activityTaxonomy.getRootNode()
        ▷ Se recupera y segmenta la trayectoria
        rawTrajectory ← retrieveTrajectoryData(aWorker, currentDate)
        segmentedTrajectory ← segmentTrajectory(rawTrajectory)
        ▷ Se anota la trayectoria (ver Algoritmo 2)
        rawAnnotation ← annotate(segmentedTrajectory, rootNode)
        ▷ Se unen los segmentos consecutivos de la misma actividad
        annotation ← aggregateSegments(rawAnnotation)
        ▷ Se almacena la trayectoria anotada
        for all segment ∈ annotation do
            annotate(rawTrajectory, segment)
        end for
    end for
end function
  
```

colección de objetos `DatosSensor`, cada uno de ellos presenta un intervalo de tiempo y un valor de sensor. Del mismo modo, los datos de la ubicación consisten en una colección de objetos `Posicion`, cada uno de ellos presenta una marca de tiempo y las variables GPS. Sin embargo, estos datos pueden no estar alineados en el sentido de que los tiempos de inicio y fin de los intervalos no son necesariamente los mismos para todos los tipos de sensores y las marcas de tiempo de ubicación no coinciden con ningún intervalo de tiempo. Por lo tanto, es necesario un paso en el algoritmo para homogeneizar los intervalos de tiempo. El resultado es una colección de objetos `Segmento`, cada uno de ellos consiste en un intervalo de tiempo, un valor de sensor para cada tipo de sensor y una `Posicion`.

Algorithm 2 Algoritmo para evaluar la trayectoria en un nodo de la taxonomía

```

function ANNOTATE(segmentedTrajectory: List of segments, node: TaxonomyNode)
  if node.getType()  $\neq$  "leaf" then
     $\triangleright$  Esto es un nodo intermedio
    edges  $\leftarrow$  node.getTaxonomyEdges()
    for all anEdge  $\in$  edges do
       $\triangleright$  El predicado de la arista es evaluado
      evaluation  $\leftarrow$  anEdge.getPredicate().evaluate(segmentedTrajectory)
       $\triangleright$  Los segmentos cuya evaluación ha sido cierta pasan al nodo hijo
      trueSegments  $\leftarrow$  removeTrueSegments(evaluation, segmentedTrajectory)
      annotate(trueSegments, anEdge.getChildNode())
       $\triangleright$  Solo los segmentos cuya evaluación devuelve falso permanecen en segmentedTrajectory
      removeFalseSegments(evaluation, segmentedTrajectory)
    end for
     $\triangleright$  Los segmentos que permanecen al final son anotados como indefinidos
    annotate(segmentedTrajectory, undefined)
  else
     $\triangleright$  Esto es un nodo hoja
    for all segment  $\in$  segmentedTrajectory do
       $\triangleright$  Todos los segmentos son anotados con la actividad del nodo
      segment.activity  $\leftarrow$  node.getActivityName()
    end for
  end if
end function

```

En el paso siguiente, la taxonomía de la actividad se evalúa en función de la trayectoria segmentada. El algoritmo 2 muestra el algoritmo utilizado para la evaluación. El proceso comienza en la raíz de la taxonomía. Se recupera la primera arista de la taxonomía y se evalúa su patrón de actividad en cada uno de los segmentos de la trayectoria. Todos los segmentos donde la evaluación del predicado ha devuelto cierto se pasan al nodo hijo de la arista de la taxonomía para ser evaluados recursivamente. Este procedimiento recursivo termina cuando se alcanza un nodo de hoja de la taxonomía. En este caso, todos los segmentos

recibidos se anotan usando el nombre de actividad en el nodo hoja. Así bien, todos los segmentos que no se evalúan como verdaderos en un nodo interno se evalúan en la siguiente arista de la taxonomía. Los segmentos que permanecen después de la evaluación de todas las aristas de la taxonomía son etiquetados como *indefinido*.

Después de evaluar la taxonomía de la actividad con la trayectoria segmentada puede suceder que muchos segmentos contiguos en el tiempo sean anotados con la misma actividad. El siguiente paso del algoritmo 1 es agregar todos estos segmentos en uno solo. El paso final del algoritmo es almacenar la anotación en una base de datos siguiendo el modelo conceptual descrito en la figura 4.

4 Lenguaje de especificación de patrones

Como se describe en la sección 3, cada arista de la taxonomía de actividades está asociada con un predicado que se utiliza para evaluar cada segmento de la trayectoria y decidir la actividad concreta con la que se anota el segmento. Cada predicado se define a través de un *lenguaje de especificación de patrones* que describe los valores esperados para cada una de las variables que se recogen en el sistema. El lenguaje consta de siete tipos diferentes de predicados que reciben una trayectoria representada como una lista de segmentos y devuelven el resultado de evaluar el predicado como una lista de valores booleanos anotados con un intervalo de tiempo. Los tipos de predicados se agrupan en dos niveles, el primero engloba aquellos predicados que son básicos y el segundo nivel agrupa aquellos predicados que son compuestos, es decir, que combinan varios predicados para construir predicados más complejos. El significado de cada predicado individual es el que sigue:

– Predicados básicos

- **PredicadoSensor**: Devuelve cierto si los valores del sensor en el segmento satisfacen la condición indicada con el operador de comparación.
- **PredicadoGPS**: Devuelve cierto si en el segmento los valores del GPS satisfacen la condición del operador. Por ejemplo, si la velocidad es mayor que 10 km/h.
- **PredicadoEspacial**: Evalúa si la posición GPS satisface una relación espacial respecto a la información geográfica del contexto. Un ejemplo, se comprueba si la ubicación registrada por el GPS coincide con la ubicación de la empresa del cliente donde el trabajador debe acudir a realizar una inspección.
- **PredicadoTarea**: Devuelve cierto si la información de la agenda del trabajador en el MWM satisface la condición indicada en el operador. A modo de ejemplo, se busca si en la agenda del trabajador existe alguna tarea que sea “Revisar alarma”.

– Predicados compuestos

- **PredicadoLogico**: Permite combinar diferentes predicados a través de operadores lógicos.

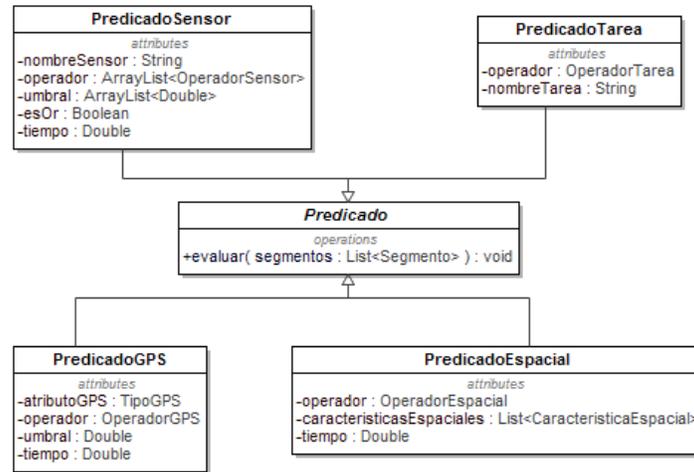


Fig. 7. Tipos de predicado básico

- **PredicadoDecision:** Puede usarse para crear un árbol de decisiones. Por ejemplo, si la condición del predicado se cumple se evalúa el predicado asociado a *esCierto*, sino se cumple se evalúa el predicado asociado al objeto *esFalso*, por último si es indefinido se evalúa el predicado asociado a *esIndefinido*.
- **PredicadoConstante:** Devuelve un valor constante (verdadero, falso o indefinido) independientemente de los valores del segmento. Es útil como predicado hijo de un **PredicadoDecision**.

Las figuras 7 y 8 muestran un modelo conceptual de estos predicados. Cada predicado es una especialización de la clase abstracta **Predicado**, la cuál define un método para evaluar el predicado en una lista de segmentos de la trayectoria. Cada subclase de **Predicado** define atributos específicos para el predicado concreto y redefine el método de evaluación. Finalmente, **PredicadoLogico** y **PredicadoDecision** son especiales porque requieren un conjunto de predicados hijos (los argumentos del operador lógico en el primer caso, el predicado de decisión y los predicados para cada resultado de decisión).

La figura 9 muestra de manera conceptual el resultado de la evaluación de algunos predicados. La parte superior de la figura muestra las posiciones GPS de la trayectoria y un *objeto geográfico* utilizado para la evaluación. La parte inferior de la figura muestra los resultados de la evaluación del predicado. El eje horizontal representa el tiempo, el eje vertical representa el resultado de la evaluación del predicado usando el valor cero si la evaluación es falsa, el valor uno si la evaluación es cierta y ningún valor si la evaluación es indefinida. El predicado de la parte superior, *dentro de la instalación del cliente*, es un **PredicadoEspacial** que devuelve cierto cuando la posición GPS está dentro de la *característica espa-*

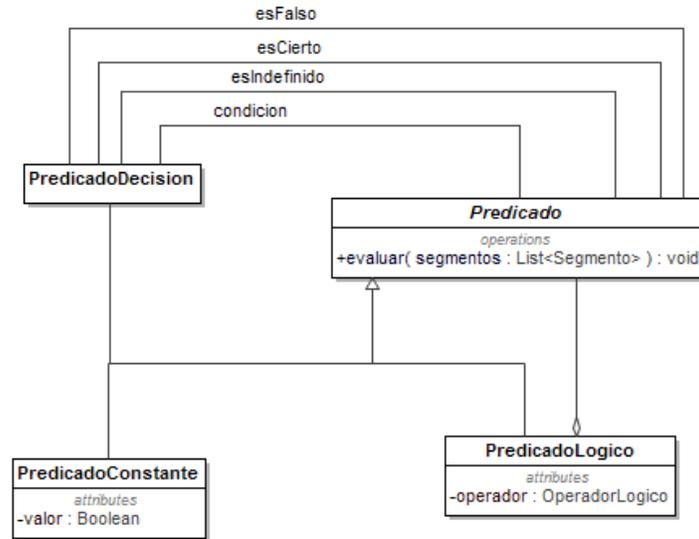


Fig. 8. Tipos de predicado compuesto

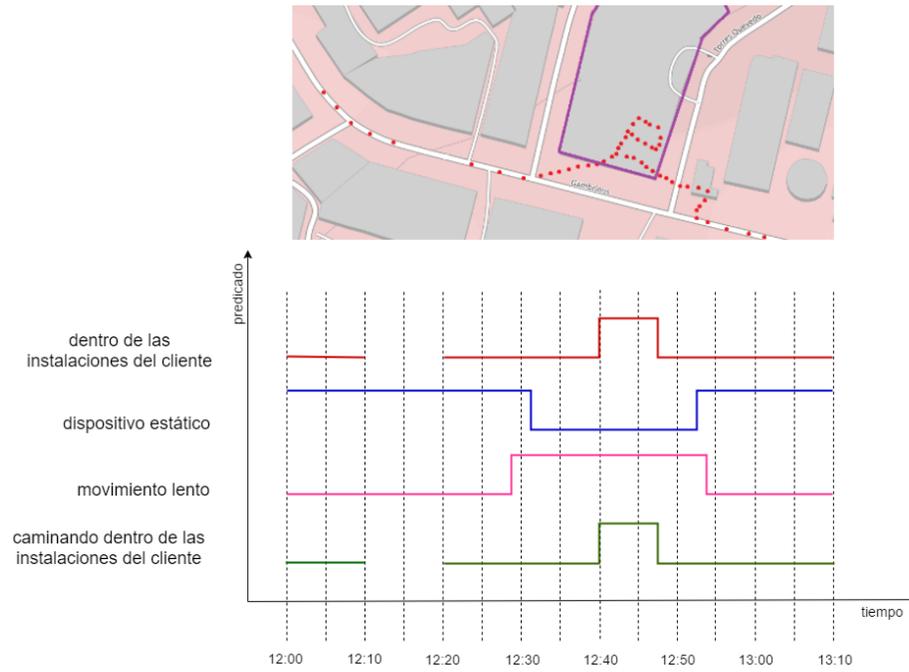


Fig. 9. Evaluación del predicado *caminando dentro de las instalaciones del cliente*

cial de la instalación del cliente (es decir, entre 12:40 y 12:47). También devuelve *indefinido* de 12:10 a 12:20 porque no hay datos de GPS. El segundo predicado, *dispositivo estático*, está definido mediante un `PredicadoSensor` que devuelve cierto cuando los valores devueltos por el acelerómetro lineal del dispositivo están por debajo de 1 m/s^2 (por lo tanto, el dispositivo se encuentra relativamente estático y el usuario no está ni caminando ni corriendo). El siguiente predicado, *moviéndose lentamente*, se define usando un `PredicadoGPS` que devuelve cierto cuando la velocidad registrada por el GPS es menor que 10 km/h . Finalmente, el predicado *andando dentro de la instalación del cliente* se define usando un `PredicadoLogico` el cual combina los predicados anteriores usando el operador lógico *and* y negando el predicado *dispositivo estático* de la siguiente manera:

$$\begin{aligned} &\text{caminando dentro de las instalaciones del cliente} = \\ &\quad \text{dentro de las instalaciones del cliente} \wedge \\ &\quad \wedge \neg \text{dispositivo estático} \wedge \text{movimiento lento} \end{aligned}$$

5 Conclusiones

En este trabajo se presenta una nueva metodología para convertir datos crudos recogidos de los sensores de dispositivos móviles en trayectorias anotadas con actividades semánticas de alto nivel de abstracción. La metodología se basa en el concepto de *taxonomía de actividades* que hacen que el sistema sea altamente flexible porque pueden adaptarse fácilmente a las necesidades de cualquier empresa. Además, la taxonomía de actividades describe los valores esperados para cada una de las variables que se recogen en el sistema utilizando predicados definidos en un *lenguaje de especificación de patrones*, muy expresivo y que tiene en cuenta no sólo los datos del sensor sin procesar sino también los datos almacenados en el sistema *MWM*, y de la *información geográfica del contexto relacionada con el dominio*. Finalmente, destacar que se trata de un módulo que puede ser fácilmente integrado en sistemas *MWM* y en el flujo de trabajo de cualquier empresa.

Como trabajo futuro, estamos terminando la implementación de todos los componentes del sistema y planeamos hacer una evaluación experimental completa con dos empresas reales en el contexto de un proyecto de investigación.

References

1. Baglioni, M., Macedo, J., Renso, C., Wachowicz, M.: An Ontology-Based Approach for the Semantic Modelling and Reasoning on Trajectories, pp. 344–353. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
2. Bayat, A., Pomplun, M., Tran, D.A.: A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science* 34, 450 – 457 (2014), the 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC’14)

3. Bogorny, V., Avancini, H., de Paula, B.C., Kuplich, C.R., Alvares, L.O.: Wekastpm: a software architecture and prototype for semantic trajectory data mining and visualization. *Transactions in GIS* 15(2), 227–248 (2011), <http://dx.doi.org/10.1111/j.1467-9671.2011.01246.x>
4. Bogorny, V., Kuijpers, B., Alvares, L.O.: St-dmql: A semantic trajectory data mining query language. *International Journal of Geographical Information Science* 23(10), 1245–1276 (2009)
5. Güting, R.H., Valdés, F., Damiani, M.L.: Symbolic trajectories. *ACM Trans. Spatial Algorithms Syst.* 1(2), 7:1–7:51 (Jul 2015)
6. Ilarri, S., Stojanovic, D., Ray, C.: Semantic management of moving objects. *Expert Syst. Appl.* 42(3), 1418–1435 (Feb 2015)
7. Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M.L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., Yan, Z.: Semantic trajectories modeling and analysis. *ACM Comput. Surv.* 45(4), 42:1–42:32 (Aug 2013)
8. Pew Research Center: Smartphone ownership and internet usage continues to climb in emerging economies. <http://www.pewglobal.org/2016/02/22/smartphone-ownership-and-internet-usage-continues-to-climb-in-emerging-economies/>, (Accessed on 12/12/2016)
9. Read, J., Žliobaitė, I., Hollmén, J.: Labeling sensing data for mobility modeling. *Information Systems* 57, 207 – 222 (2016)
10. Rehman, M.H.u., Liew, C.S., Wah, T.Y., Shuja, J., Daghighi, B.: Mining personal data using smartphones and wearable devices: A survey. *Sensors* 15(2), 4430 (2015)
11. Spaccapietra, S., Parent, C., Damiani, M.L., de Macedo, J.A., Porto, F., Vangenot, C.: A conceptual view on trajectories. *Data Knowl. Eng.* 65(1), 126–146 (Apr 2008)
12. Yan, Z., Spremic, L., Chakraborty, D., Parent, C., Spaccapietra, S., Aberer, K.: Automatic construction and multi-level visualization of semantic trajectories. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. pp. 524–525. GIS '10, ACM, New York, NY, USA (2010)