Tesina de Grado

Desarrollo de GIS basado en Componentes Dinámicos

Facultad de Informática – U.N.L.P.

Carrera: Licenciatura en Informática

Alumnos: Leonardo Nomdedeu y Luciano Benítez

Directora: Dra. Silvia Gordillo

Temario

1. **Motivación de la Tesina**
2. **Propuesta**: Proceso de Desarrollo
3. **Descripción del Proceso**
4. **Presentación del Prototipo**
5. **Conclusiones**
6. **Trabajos Futuros**

1. Motivación

- Existen muchas dependencias funcionales fuertemente ligadas a los Sistemas de Información Geográfica (GISs) (ej. el sistema de entrada de datos, la representación y visualización de la cartografía, las fuentes de almacenamiento de datos o las estrategias del negocio).
  - Las Dependencias son relaciones en el código donde un cambio en una sección del mismo requiere cambios en otras secciones.

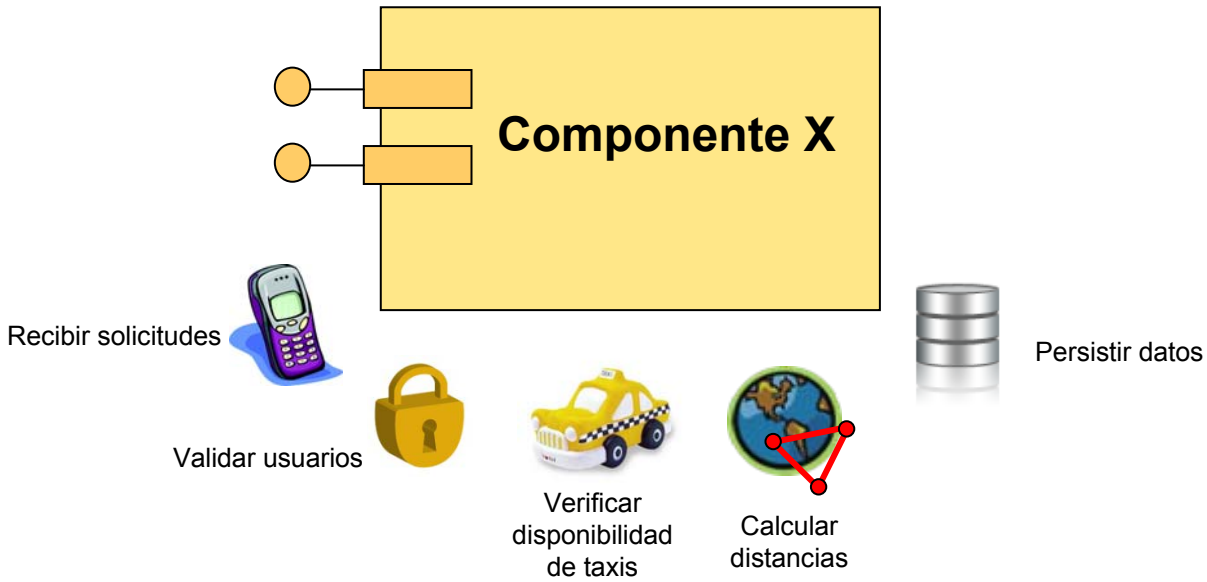
- A medida que evoluciona el GIS, estas dependencias hacen más compleja la tarea de incorporación o modificación de componentes, teniendo que realizar refactoring complejo, reingeniería o incluso migrar hacia otros productos.

- Veámoslo con un ejemplo sencillo.

1. Motivación: Ejemplo

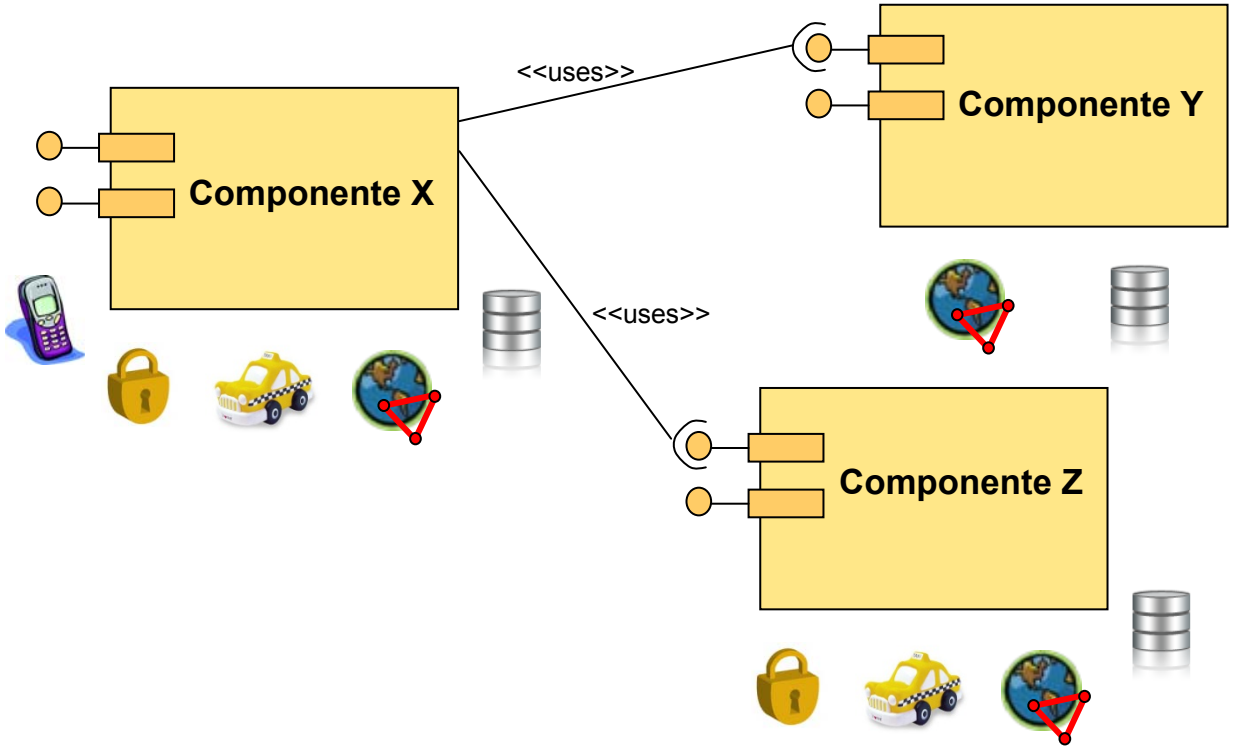
- Escenario: "*El sistema debe poder recibir una solicitud de un cliente válido y en base a la ubicación geográfica del mismo se debe asignar el taxi libre más cercano. Esta asignación debe ser almacenada en una fuente de datos*".

- Usualmente se suelen desarrollar componentes del GIS que tienen la responsabilidad de llevar a cabo tareas de distinta naturaleza. Ejemplo:



1. Motivación: Ejemplo - Continuación

- Escenario más complejo: varios componentes que trabajan juntos y que realizan tareas de distinta naturaleza.



- **El problema:** Evolución de funcionalidad dispersa en distintos componentes del GIS. Ejemplo: cálculo de operaciones espaciales.

2. Propuesta

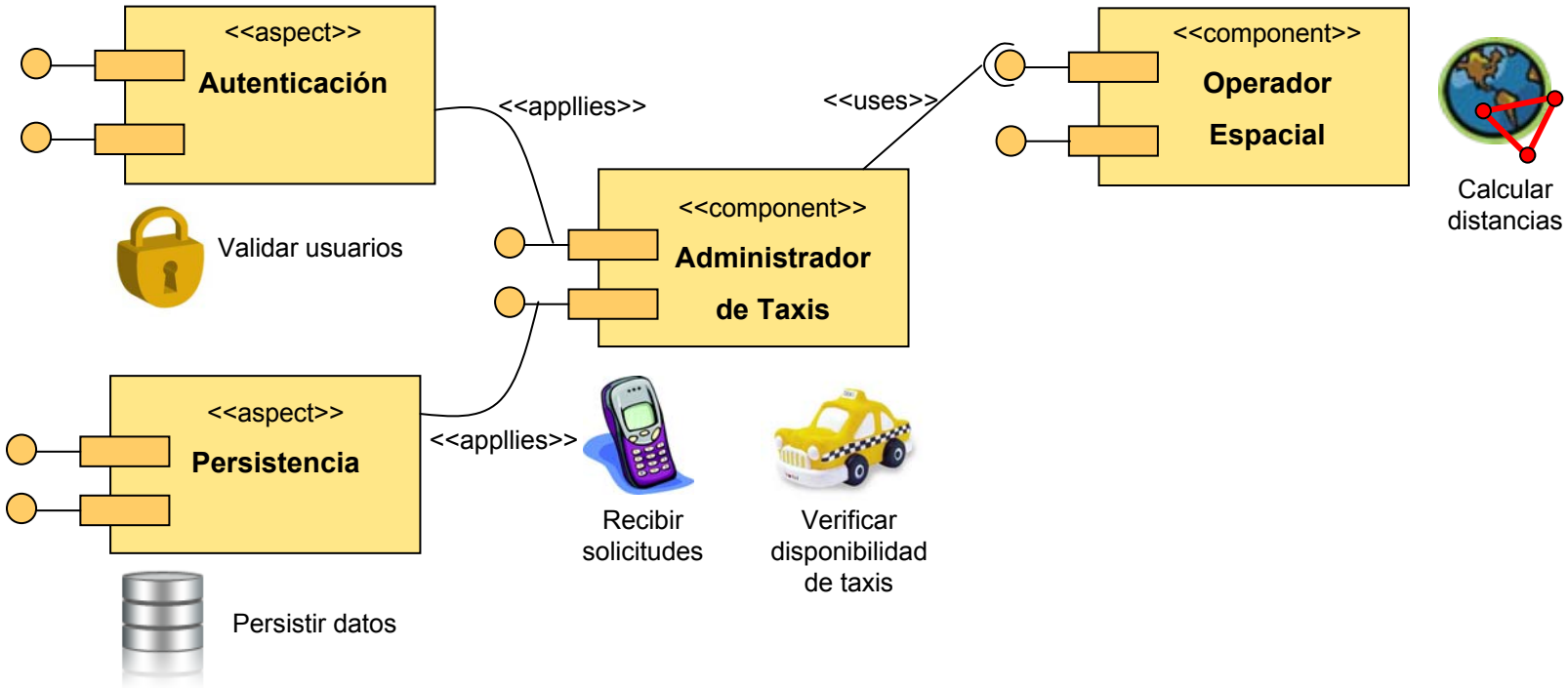
- Generar herramientas GIS flexibles que permitan adaptarse a los cambios tecnológicos y a las nuevas reglas de negocio que evolucionan en el tiempo.

- ¿Cómo se obtiene un GIS con estas características?

  Definiendo un proceso de desarrollo que permita generar una arquitectura GIS flexible, que admita la incorporación o modificación de un componente del sistema en forma dinámica sin afectar al resto de los componentes, de aquí surge el concepto de ***Componentes Dinámicos***.

2. Propuesta – Continuación

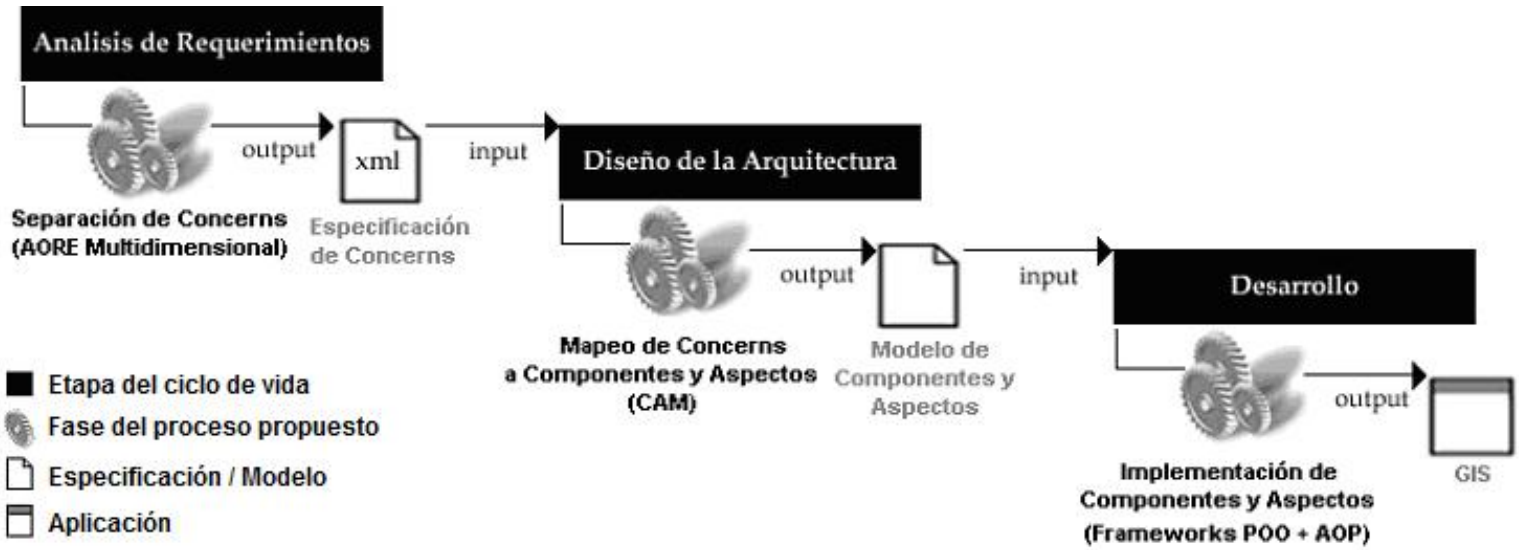
- Aplicando el proceso propuesto:
  - **Administrador de Taxis** es un componente cuya función principal es asignar taxis a las solicitudes telefónicas realizadas por los clientes de la empresa.
  - **Autenticación** es un aspecto cuya función principal es validar la autenticidad de un cliente.
  - **Operador Espacial** es un componente cuya función principal es obtener la entidad (taxis) más cercana a una coordenada geográfica determinada.
  - **Persistencia** es un aspecto cuya función principal es almacenar en la base de datos entidades del dominio.



3. El Proceso

Consta de tres fases:

1. Separación de Concerns
2. Mapeo de Concerns a Componentes y Aspectos
3. Implementación de Componentes y Aspectos



3. Fase I: Separación de concerns

- Se realiza en las primeras etapas del ciclo de vida (etapa de Análisis de Requerimientos).

- A partir de los requerimientos elicitados se realiza una identificación y clasificación de los concerns utilizando el modelo AORE Multidimensional.
  - Un concern se define como una propiedad, o punto de interés de un sistema [Filman05]. Focalizar la atención en un concern, no significa ignorar el resto de los concerns, sino que desde el punto de vista de este concern, el otro es irrelevante.

- El modelo AORE Multidimensional nos permite tener una visión individual de cada concern y a su vez nos permite determinar la influencia de un concern del GIS sobre el resto.

- Mediante este modelo se puede definir una especificación formal de los concerns del GIS y sus relaciones.

3. Fase I: Modelo AORE-Multidimensional

- Pasos más importantes del modelo:
  - Identificar y clasificar concerns
  - Identificar relaciones entre concerns (grano fino y grueso)
  - Especificar dimensiones de concerns
  - Abstraer meta-concerns

- Output: Especificación de concerns y sus relaciones

```
<Concern name="Operador Espacial" type="Negocio" >
  <Relationships>
    <Relation name="Administrador de Taxis" type="Contribución" />
  </Relationships>
</Concern>
<Concern name="Persistencia" type="Calidad" >
  <Relationships>
    <Relation name="Administrador de Taxis" type="Contribución" />
</Relationships>
</Concern>
<Concern name="Autenticación" type="Calidad" >
  <Relationships>
    <Relation name="Administrador de Taxis" type="Condición" />
</Relationships>
</Concern>
```

3. Fase II: Mapeo de Concerns a Componentes y Aspectos

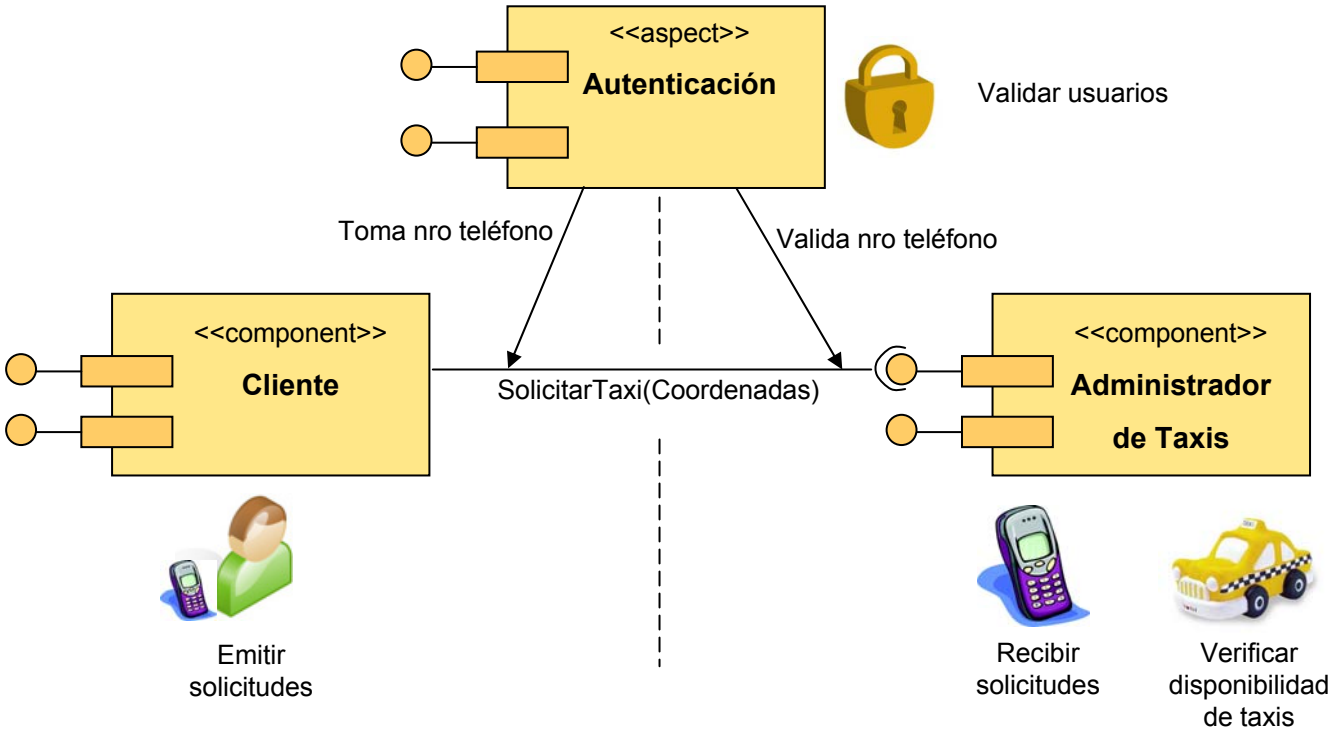
- Se realiza en la etapa de Diseño de la Arquitectura.
- Se toma la especificación de concerns del modelo AORE Multidimensional y se realiza un mapeo de concerns a entidades de la arquitectura lógica del GIS, que pueden ser:
  - Componentes (paradigma CBSD)
  - Aspectos (paradigma AOSD)

  } Componentes Dinámicos
- ¿Cuáles son las reglas de mapeo utilizadas?
  - Concerns de tipo Negocio o Contexto → Componentes
  - Concerns de tipo Calidad → Aspectos
  - Concerns de tipo Objetivo, Riesgo o Físico → No se mapea

3. Fase II: ¿Que es un aspecto?

- Es una abstracción de un concern no funcional que tiende a ser transversal a múltiples componentes del sistema.

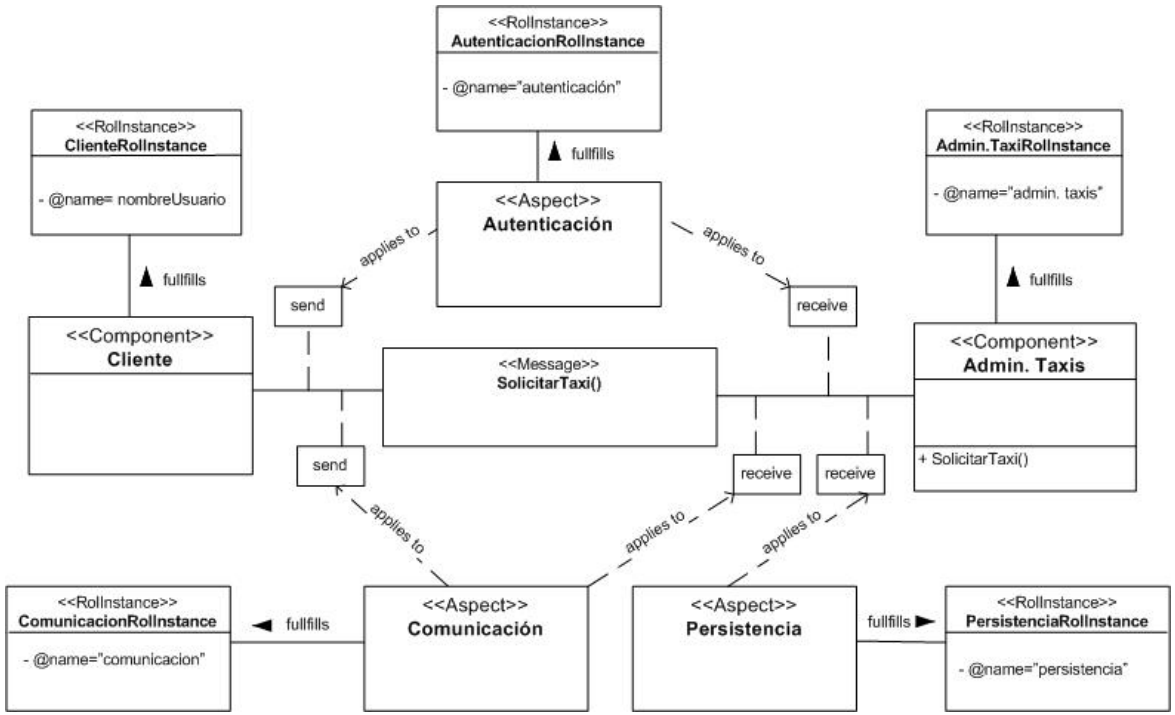
- Tiene la capacidad de aplicar comportamiento en determinados puntos de intercepción (por ej. antes y/o después de la invocación a un mensaje).



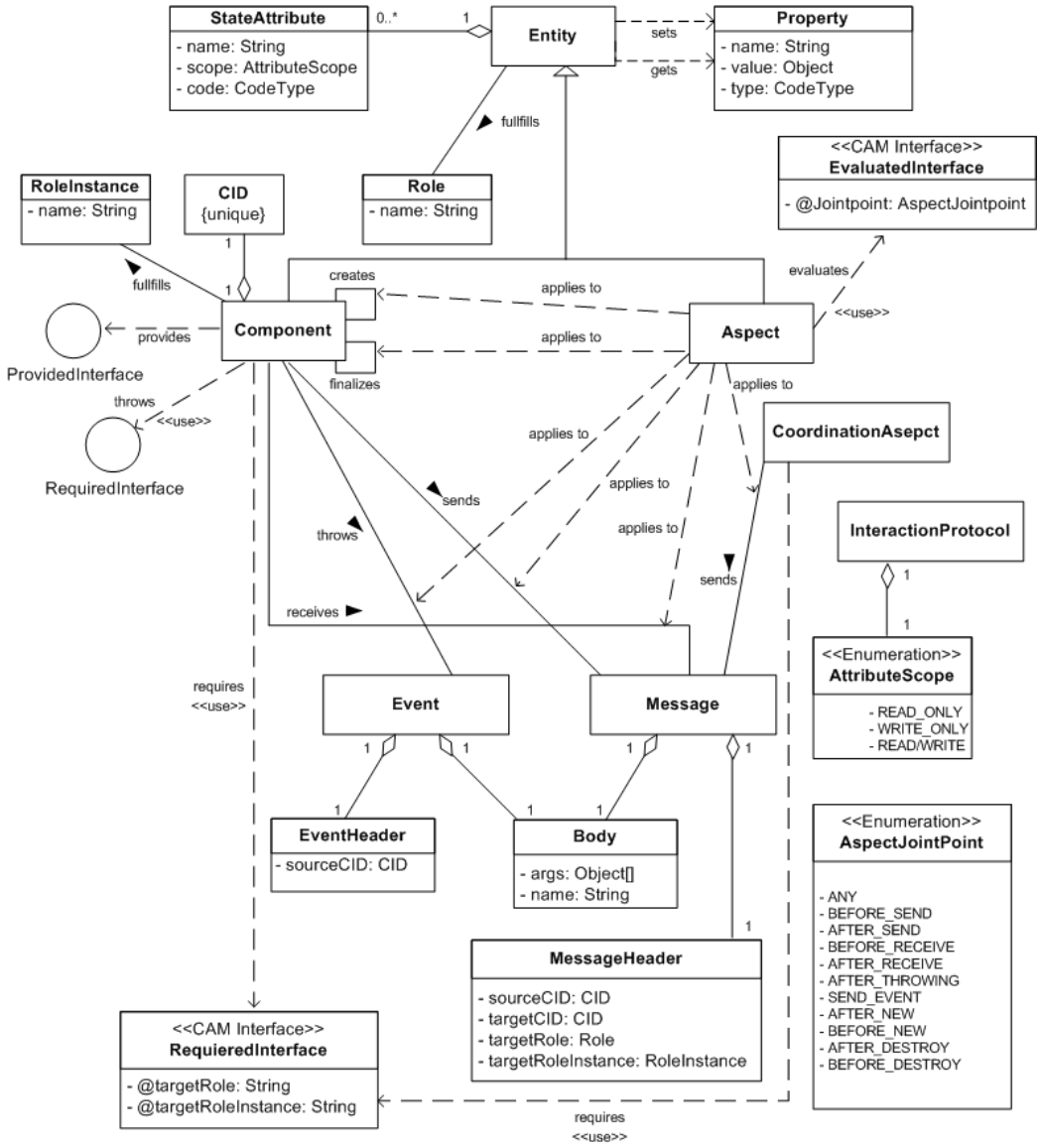
3. Fase II: Modelo CAM

- Para modelar los componentes, aspectos y sus relaciones se utiliza el modelo CAM (Componente and Aspect Model) [Pinto05].
  - Permite aplicar aspectos solo a las interfaces de los componentes respetando el principio de encapsulamiento de CBSD.

- Output: Modelo de Componentes y Aspectos



3. Fase II: Metamodelo CAM



3. Fase III: Implementando Componentes Dinámicos

- Se realiza en la etapa de Desarrollo.
- Consiste en implementar los componentes dinámicos como unidades de software utilizando una tecnología determinada.
- Tecnologías más utilizadas que soportan la implementación de componentes:
  - J2EE
  - Microsoft .Net
- Tecnologías analizadas que soportan la implementación de aspectos:
  - AspectJ, Aspect#
  - JBoss AOP
  - Spring AOP (respeta los principios del modelo CAM)

4. Prototipo

- En esta sección se reproduce un video mostrando el prototipo

5. Conclusiones

- El proceso propuesto resulta ser una muy buena aproximación a tener en cuenta a la hora de comenzar el desarrollo de un proyecto GIS que sea propenso a evolucionar en el tiempo.
  - Buena modularización.
  - Bajo nivel de acoplamiento entre los componentes y aspectos del GIS.

- Ventajas
  - Reuso de componentes.
  - Evolución individual de cada componente sin afectar al resto
  - Se aceleran los tiempos de refactoring reduciendo costos de diseño e implementación.

- Desventajas
  - Implica comprender el paradigma de Programación Orientada a Aspectos (paradigma reciente).

6. Trabajos futuros

- *Mecanismo de mapeo automático*: analizando los tipos de concerns y los tipos de relaciones del modelo AORE-Multidimensional es posible implementar un mecanismo de mapeo automático de concerns a componentes y aspectos del modelo CAM.

- *Meta-concerns de GIS*: investigar en detalle y modelar los concerns que suelen estar presentes en la mayoría de los GIS.

7. Referencias de la presentación

- **[Filman05]** Filman R., Elrad, T., Clarke, S. y Aksit, M., *"Aspect-Oriented Software Development"*, Addison Wesley, Boston. 2005.

- **[Moreira_05]** Moreira Ana, Rashid Awais, Araujo Joao. "*Multi-dimensional Separation of Concerns in Requirements Engineering*". 2005. The 13th International Conference on Requirements Engineering (RE'05), August 29, September 2, 2005, Paris, France. IEEE Computer Society.

- **[Pinto05]** Mónica Pinto, Lidia Fuentes and José María Troya*, "A Dynamic Component and Aspect-Oriented Platform"*, The Computer Journal, 2005.

3. Ejemplo de Identificación y Separación de Concerns

```
<Concern name="Administrador de Taxis">
 …
 <Requirement id="4">
  Debe gestionar solicitudes de los clientes
  <Requirement id="4.1">Debe recibir del cliente el nombre de usuario, el número de telefono, la ubicación y las prefencias del
      taxi</Requirement>
  <Requirement id="4.2">Debe almacenar la solictud del cliente en una base de datos</Requirement>
  <Requirement id="4.3">Debe enviar al cliente un mensaje de recepción exitosa de la solicitud</Requirement>
 </Requirement>
 …
 <Requirement id="6">
  Debe autenticar el cliente en el sistema …….
  <Requirement id="6.1">Debe enviar el número de telefono del cliente</Requirement>
  <Requirement id="6.2">Debe recibir los datos personales del cliente</Requirement>
 </Requirement>
 <Requirement id="7">
  Debe asignar un taxi a una solicitud realizada por un cliente
  <Requirement id="7.1">Debe encontrar el taxi libre más cercano a la ubicacón del cliente que cumpla con las
      preferencias</Requirement>
  <Requirement id="7.2">Debe enviar al taxi la solicitud</Requirement>
  <Requirement id="7.3">Debe recibir la confirmación del taxi indicando si acepta o no la solicitud</Requirement>
 </Requirement>
…
</Concern>
```

```
<Concern name="Autenticación">
 <Requirement id="1">
  Se debe poder autenticar un usuario a través de un numero de telefono
 </Requirement>
 <Requirement id="2">
  Se debe poder autenticar un usuario a través de un nombre de usuario y contraseña
 </Requirement>
</Concern>
```

Volver