

Modelos de contexto: Un enfoque dinámico



TESINA DE GRADO

EMILIANO GUILLERMO PÉREZ

DIRECTOR:

DR. GUSTAVO ROSSI

CO-DIRECTOR:

DRA. SILVIA GORDILLO

Motivación

2

- **Utilizado en sistemas informáticos:**
 - Para adaptar la presentación de información
 - Para adaptar el comportamiento de la aplicación
 - Para recolectar información

- **Contexto:**
 - Definido por su relevancia sobre un sujeto
 - Da identidad a ese sujeto
 - Concepto subjetivo
 - Información sobreentendida

Programa

3

- El problema de modelar el contexto
 - Interpretaciones
 - Posturas de Dourish
- Enfoques actuales sobre el problema
- Enfoque dinámico: Guías de diseño
- Construyendo modelos dinámicos

¿Qué es el contexto?

4

- ¿Qué se puede considerar como contexto?
 - Listas:
 - ✦ Ubicación del usuario, temperatura ambiente, compañía, hora, etc...
 - Categorías :
 - ✦ Ubicación, tiempo, identidad y actividad.
 - Descripciones:
 - ✦ *“Situación del usuario”, “Estado del entorno de la aplicación”.*
- *“[...] el contexto es cualquier información que **se pueda** usar para caracterizar una entidad. Una entidad es una persona, lugar u objeto que se considere **relevante** para la interacción entre el usuario y la aplicación, incluyendo al usuario y a la aplicación en sí[...].”*

¿Cómo modelar el contexto? I

5

- Sistemas concretos que incorporan información contextual.
 - Los requerimientos de cada aplicación determinan *qué* se considera como contexto.
 - Las necesidades de cada aplicación determinan *cómo* se representa cada pieza de información contextual.
- Suelen implementarse utilizando técnicas no diseñadas específicamente para manejar este tipo de información.
 - OWL, XML, Bases de datos, entre otros.

¿Cómo modelar el contexto? II

6

- Arquitecturas que permiten modelar las particularidades de la relación contextual.
 - Se busca permitir que eventualmente cualquier tipo de información pueda ser contextual.
 - Los mecanismos de modelado están inspirados en cómo se comporta la relación.
- Suelen estar expresados mediante mecanismos de modelado abstracto.
 - Meta-modelos (CML).
 - Modelos gráficos (UML).

¿Cómo modelar el contexto? III

7

- ¿Hasta que punto se pueden contemplar los aspectos dinámicos del contexto en una representación informática?

La relación contextual

8

- Sujeto absoluto -> Sujeto relativo -> Sujeto + Contexto
- Al interpretar el sujeto este se hace parcial y relativo.
 - Se toman solo las características de su entorno que son relevantes.
- La relevancia contextual define la relación del sujeto con los elementos de su entorno.
- La relevancia está determinada por:
 - El observador que interpreta la escena
 - El momento actual

¿Cómo modelar el contexto? IV

9

- ¿Qué significa que un modelo contemple la relación contextual?
 - Que permita identificar entidades como sujetos sensibles al contexto.
 - Que permita establecer relaciones contextuales.
 - Que la relación pueda depender del momento actual y del observador.

Dualidad de Dourish

10

- El modelado de contexto
 - ¿Un problema de representación o de interacción?

Postura Positivista

11

¿Cómo representamos la información en un sistema informático?

- El contexto es una forma de información
- El contexto es determinable
- El contexto es estable
- El contexto y la actividad son separables

Postura Fenomenológica

12

¿Cómo se genera el contexto a partir de las interacciones?

- El contexto es una propiedad que relaciona objetos y actividades
- El contexto no puede ser determinado
- La estructura del contexto es dinámica
- El contexto es consecuencia de la actividad

Las posturas de Dourish

13

Positivista

- Información
- Determinable
- Estable
- Contexto <> Actividad

Fenomenológica

- Relaciones
- Indeterminable
- Dinámica
- Actividad -> Contexto

¿Existe un punto intermedio?

14

Delimitado por la Representación
Problema de Representación
Nivel Concreto
Vista Técnica

Postura
Positivista

Balance

Postura
Fenomenológica

Vista Social
Nivel Abstracto
Problema de Interacción
Delimitado por el observador

Enfoques

15

Postura
Fenomenológica

Arquitecturas
Abstractas

Modelos
Gráficos

Modelos Lógicos

Ontologías

Diseños OO

Bases de datos

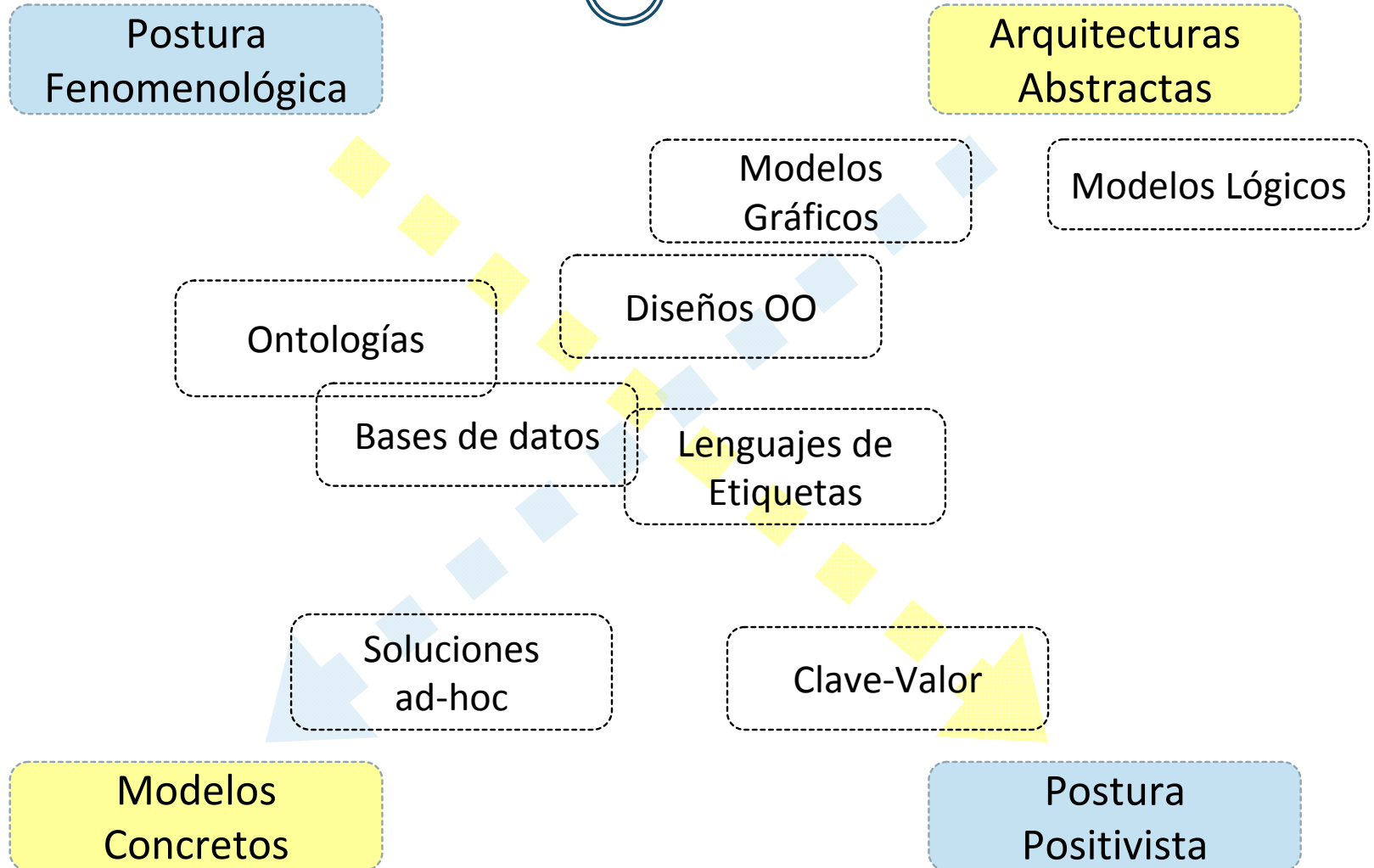
Lenguajes de
Etiquetas

Soluciones
ad-hoc

Clave-Valor

Modelos
Concretos

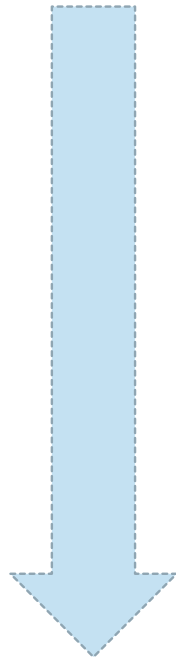
Postura
Positivista



Enfoque propio

16

Postura
Fenomenológica

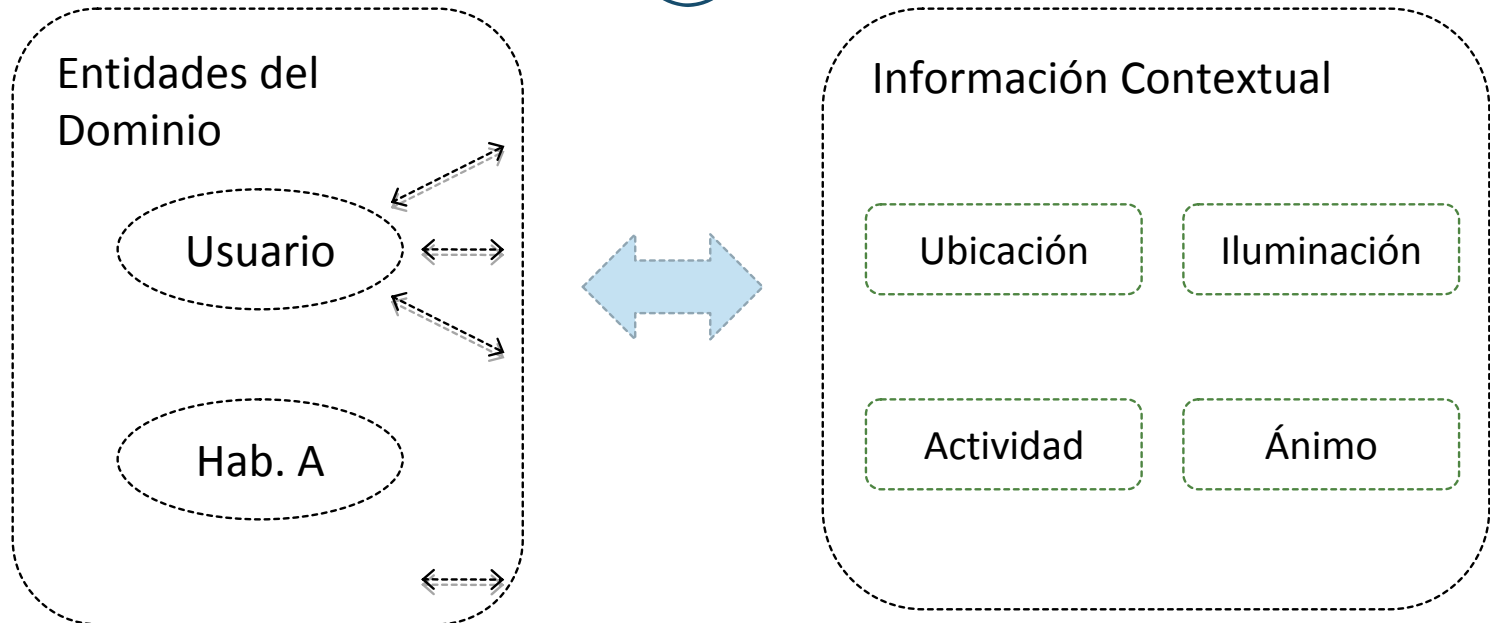


Modelo
Concreto

- ¿Qué es el contexto?
- Análisis de la relación contextual
- Principios de diseño
- Arquitectura
- Implementación

¿Cómo se relaciona el contexto a las entidades?

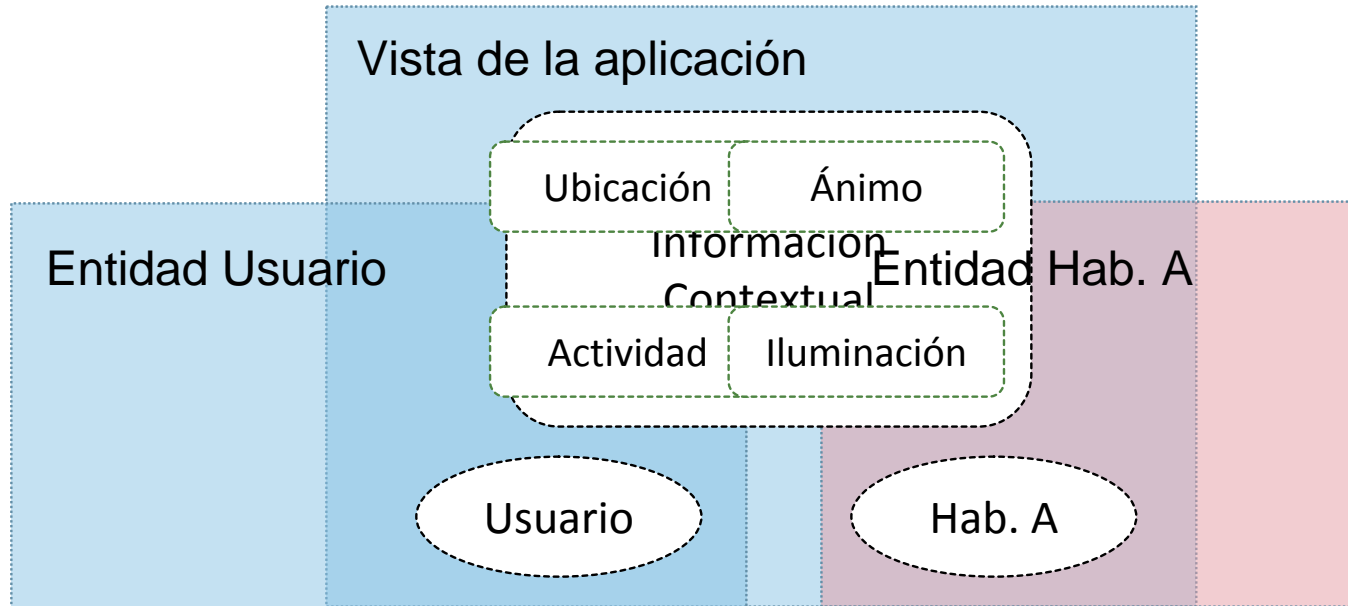
17



- En de **La información contextual necesita estar asociada a un sujeto para existir.**

¿Cómo organizamos el contexto en nuestros sistemas?

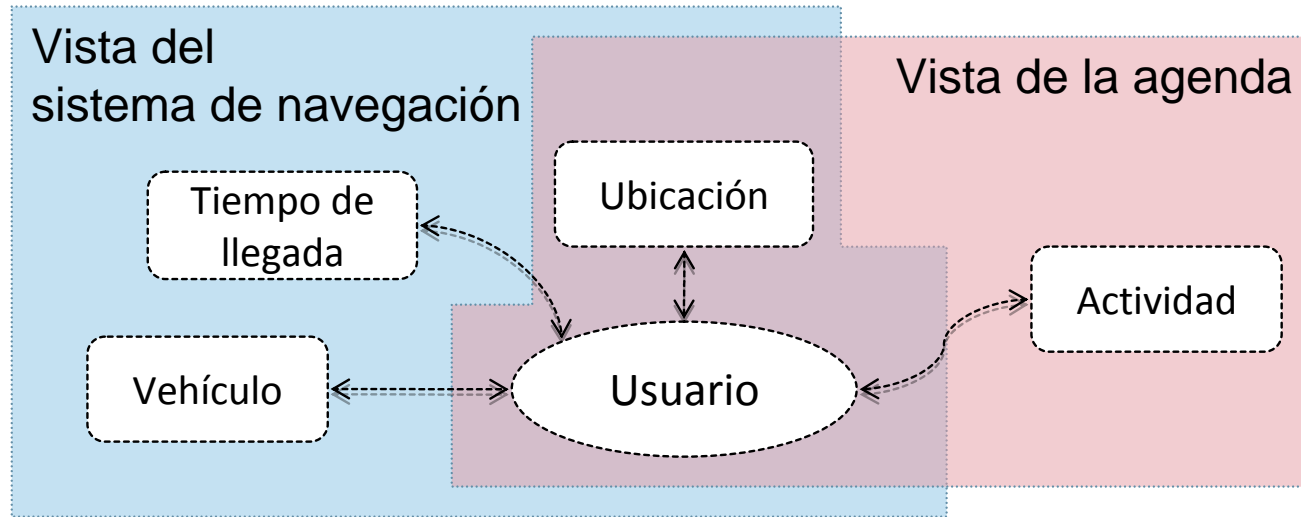
18



El contexto no es una única pieza de información, sino que se encuentra distribuido en los objetos de nuestra aplicación.

¿Qué aspectos componen el contexto de un sujeto?

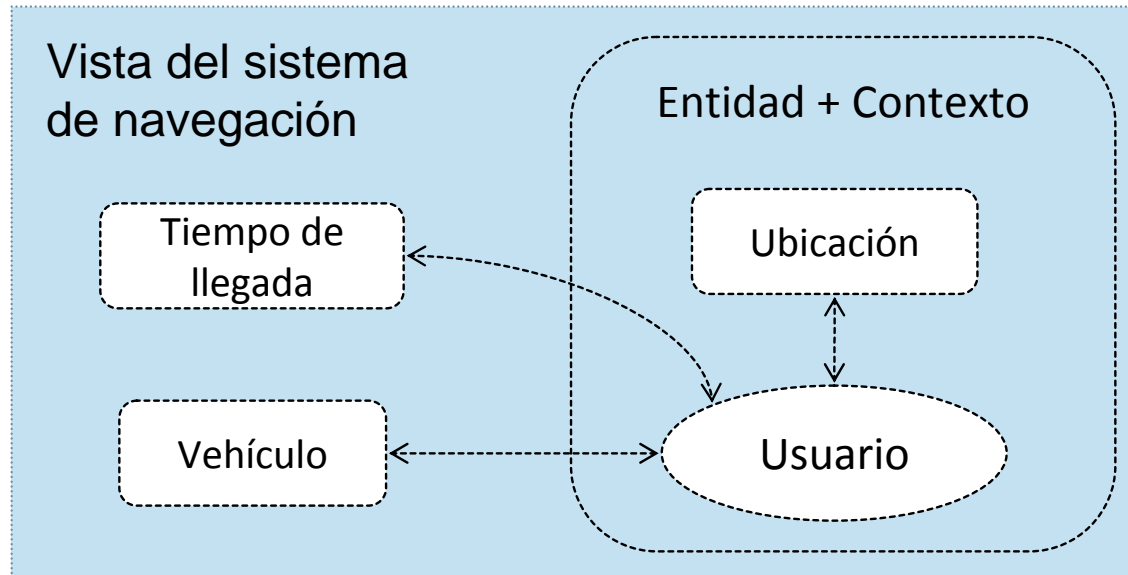
19



- Al ser el contexto un concepto subjetivo, cada
El modelo debe permitir distintas representaciones
contextuales del mismo sujeto.
para la misma entidad.

¿Es todo el contexto visible por igual?

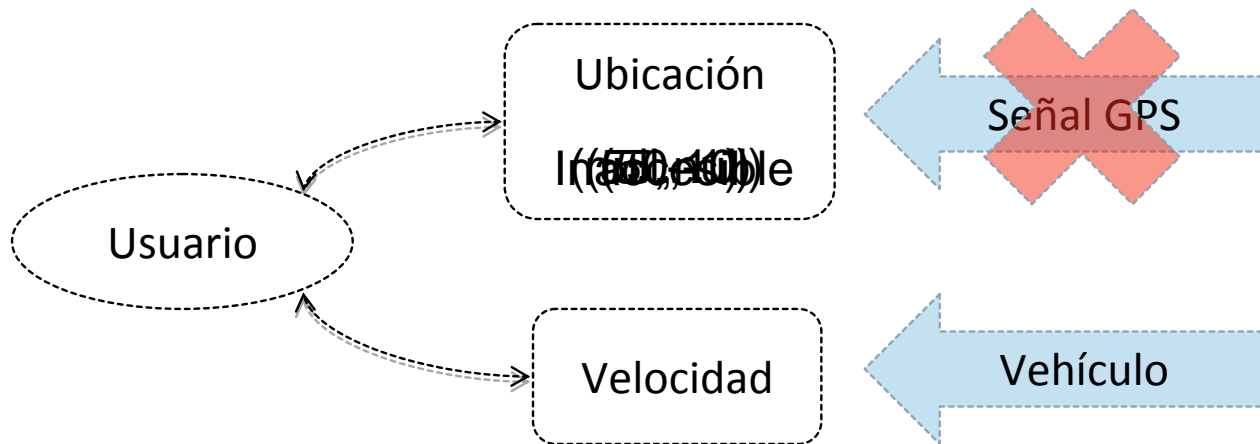
20



- Algunos contextos son específicos de una aplicación o sistema.
- El contexto puede ser parte interna de la entidad o ser específico a una aplicación.
- Otros son inherentes al sujeto con el que se relacionan.

¿Cómo se comporta el contexto en el tiempo?

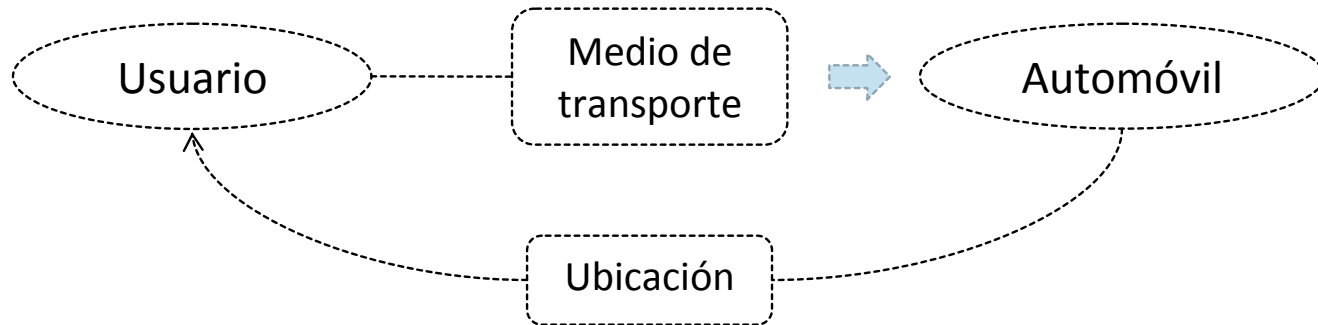
21



- El contexto depende de la configuración, la actividad actual y los participantes de la misma.
- La estructura del contexto asociada a una entidad debe ser modificable en ejecución.
- El contexto, pueden cambiar constantemente.

¿De qué manera afectan los cambios contextuales a la estructura del contexto?

22



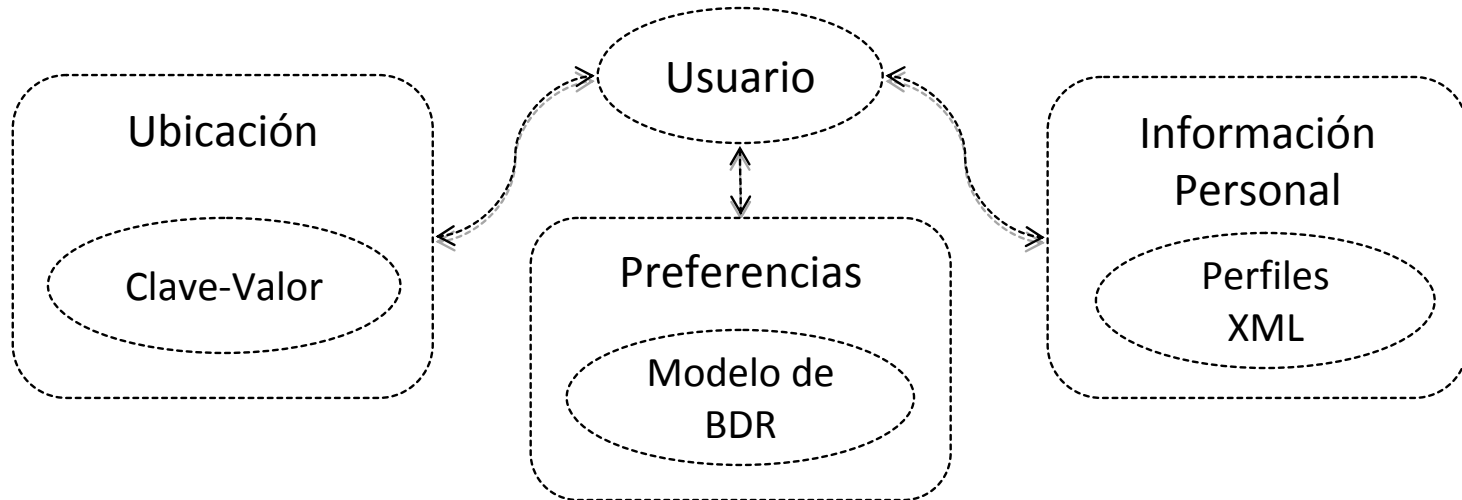
- Los cambios en la información contextual pueden generar a su vez otros cambios

La información contextual y su relación con los sujetos pueden depender de los cambios en el contexto.

• En los valores de los datos.

¿Una única implementación para todo el contexto?

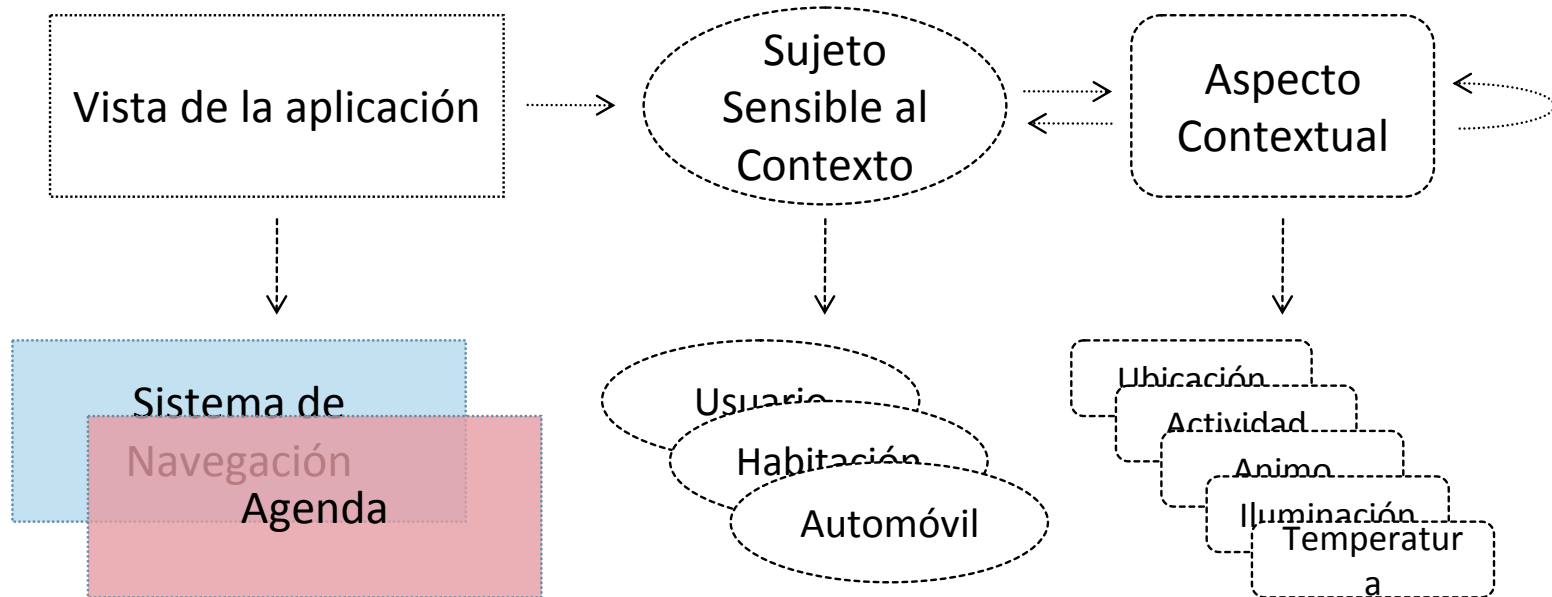
23



- Cada tipo de información se comporta de manera
Cada aspecto contextual debe poder ser modelado mediante
una técnica adecuada a su dominio particular.
pueden beneficiar a un mismo modelo.

¿Podemos definir con anterioridad cómo se comportará el contexto?

24



- Es imposible definir de antemano todos los tipos de

El modelo de contexto deberá definir las estructuras básicas y ser adaptable a requerimientos cambiantes.

que se pueda extender para instanciar aplicaciones.

Principios de diseño

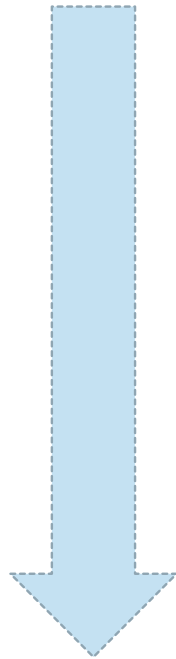
25

- **Definen:**
 - Cómo se debe pensar en el contexto a la hora de modelarlo.
 - Características específicas que el modelo debe respetar.
 - ✦ Estructura
 - ✦ Comportamiento
 - ✦ Reutilización
- **Mantienen un nivel abstracto**
 - Se pueden aplicar a arquitecturas genéricas o a modelos concretos.
 - Se pueden traducir a cualquier paradigma o lenguaje.
 - Plantean un marco básico, se pueden extender.

Enfoque propio

26

Postura
Fenomenológica



Modelo
Concreto

- ¿Qué es el contexto?
- Análisis de la relación contextual
- Principios de diseño
- Arquitectura
- Implementación

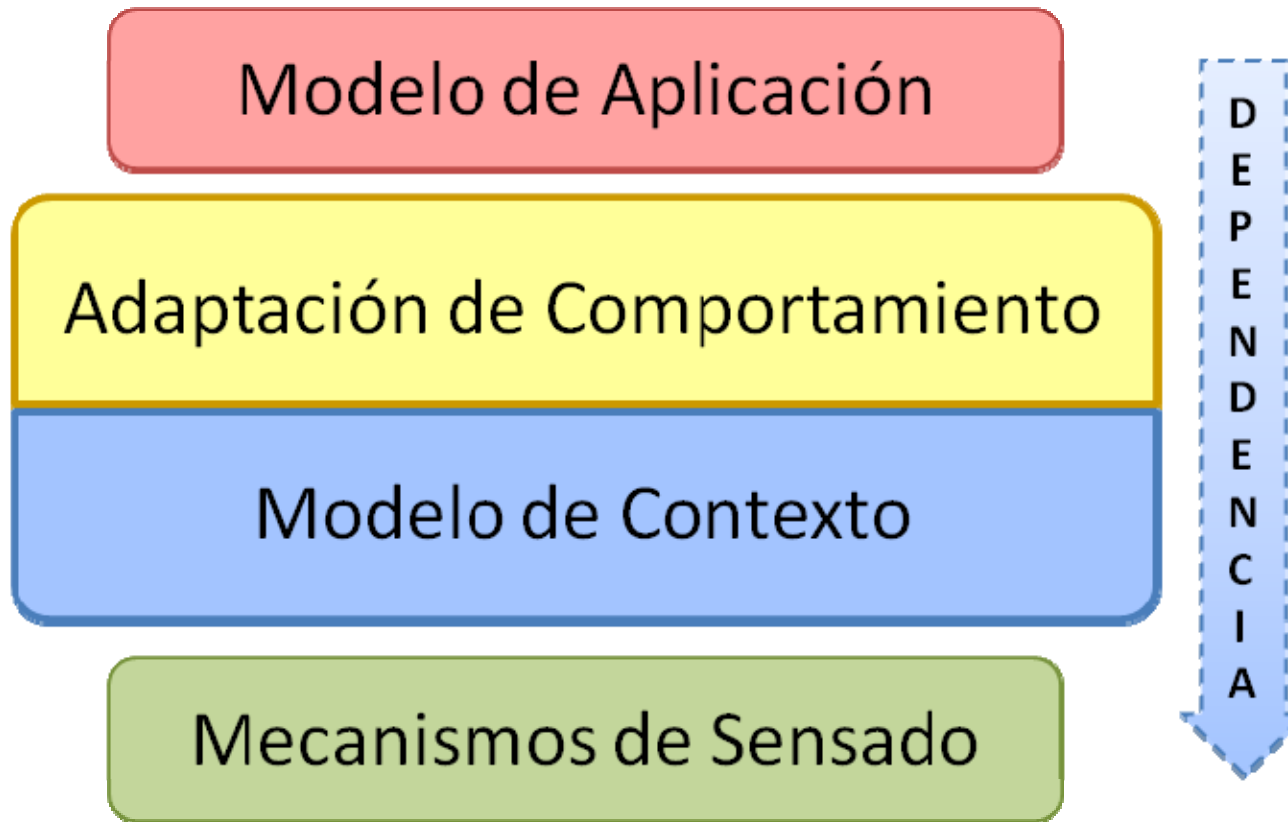
Arquitectura

27

- Traduce los principios de diseño a abstracciones para definir modelos de contexto.
- Diseño OO
 - Expresividad
 - Reutilización y encapsulamiento
 - Aplicación de patrones
 - Reflexión
 - Fácil implementación en distintos lenguajes

Arquitectura

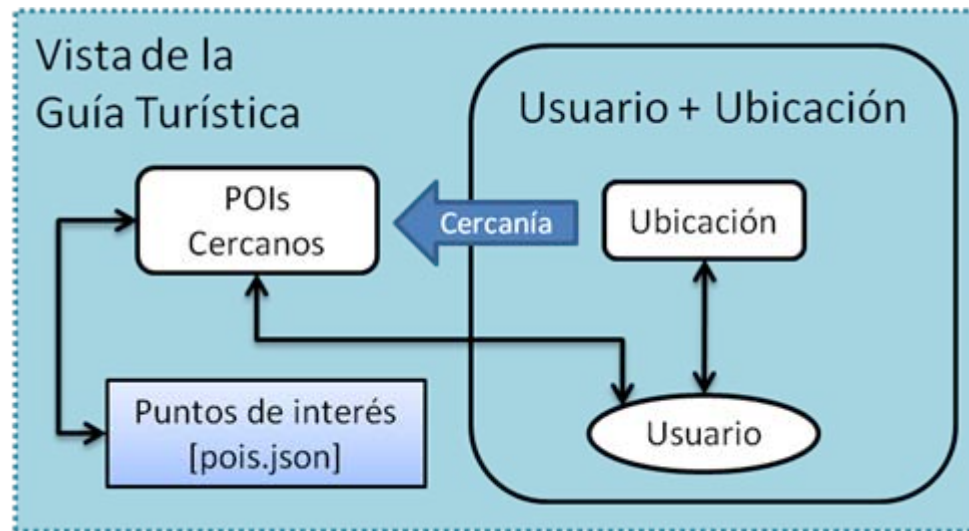
28



Ejemplo motivador: Guía Turística

29

- Un usuario camina por una ciudad que incluye determinados puntos turísticos de interés.
- A medida que el usuario se acerca a los puntos de interés la aplicación le informa este hecho al usuario.



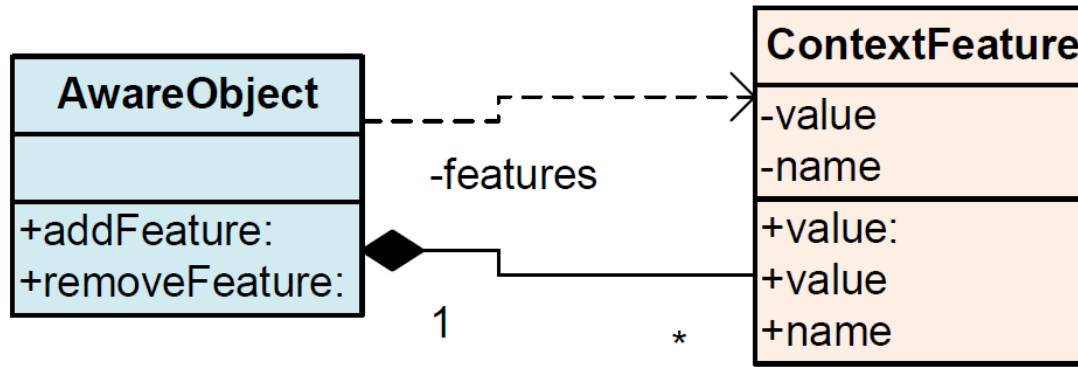
Sujeto sensible al contexto

30

- AwareObject: Entidad del modelo que se vincula con aspectos contextuales.
- ContextFeature: Aspectos contextuales que representan una relación contextual entre dos entidades.
- Se implementan como contenedores de valores asociados a un identificador.

Sujeto sensible al contexto

31

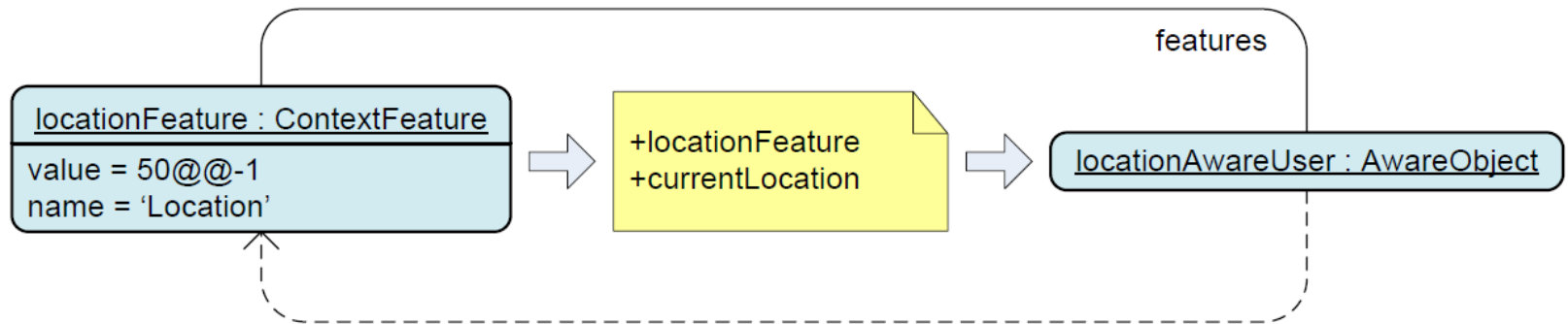


```
locationAwareUser := AwareObject new.  
locationFeature :=  
    ContextFeature  
        named: 'Location'  
        value: (50@@-1).  
locationAwareUser addFeature: locationFeature.
```

Sujeto sensible al contexto

32

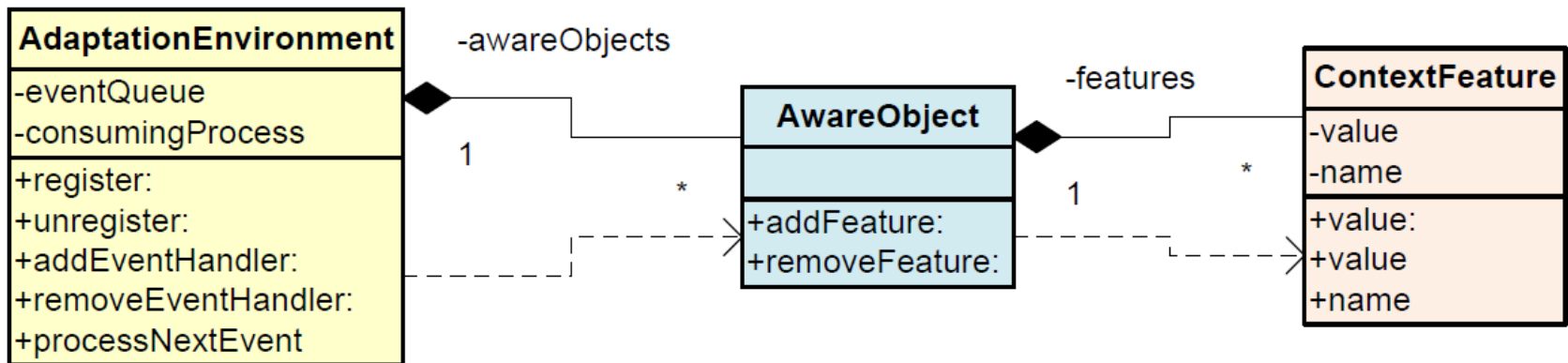
- Cada aspecto “suma” su comportamiento contextual al sujeto.
- El sujeto amplía su protocolo y publicará los cambios en su contexto.



La entidad observadora

33

- La vista de la aplicación cumple el rol del observador en la relación contextual.
- AdaptationEnvironment: Abstracción que cumple dos funciones.
 - Proveer un ambiente idóneo para administrar los sujetos.
 - Ofrecer una interpretación específica de un sujeto.



La entidad observadora

34

- Los sujetos se “registran” en un ambiente y este recibe sus actualizaciones.
- Se pueden instalar “manejadores” que representan acciones a llevarse a cabo cuando cambia el contexto.

```
environment := AdaptationEnvironment new.  
environment register: locationAwareUser.
```

```
handler := PluggableHandler  
           forAwareObject: locationAwareUser  
           featureName: 'location'.
```

```
handler actionBlock: [:event | |Transcript show:  
                      ('La ubicación del usuario ha cambiado a: ',  
                       event newValue printString)].
```

```
environment addEventHandler: handler.  
environment run.
```

El origen de los datos

35

- Conexión entre la capa de sensado y el modelo
- SensingConcern: Abstracción que procesa e inyecta datos utilizables por el modelo en un ContextFeature.

```
policy := AnnouncementsPushPolicy on: sensor
announcement: SensorAnnouncement
accessor: #value.
trans := CoordParseTransformation new.
concern := SensingConcern
policy: policy
transformation: trans
receiver: locationAwareUser locationFeature.
```

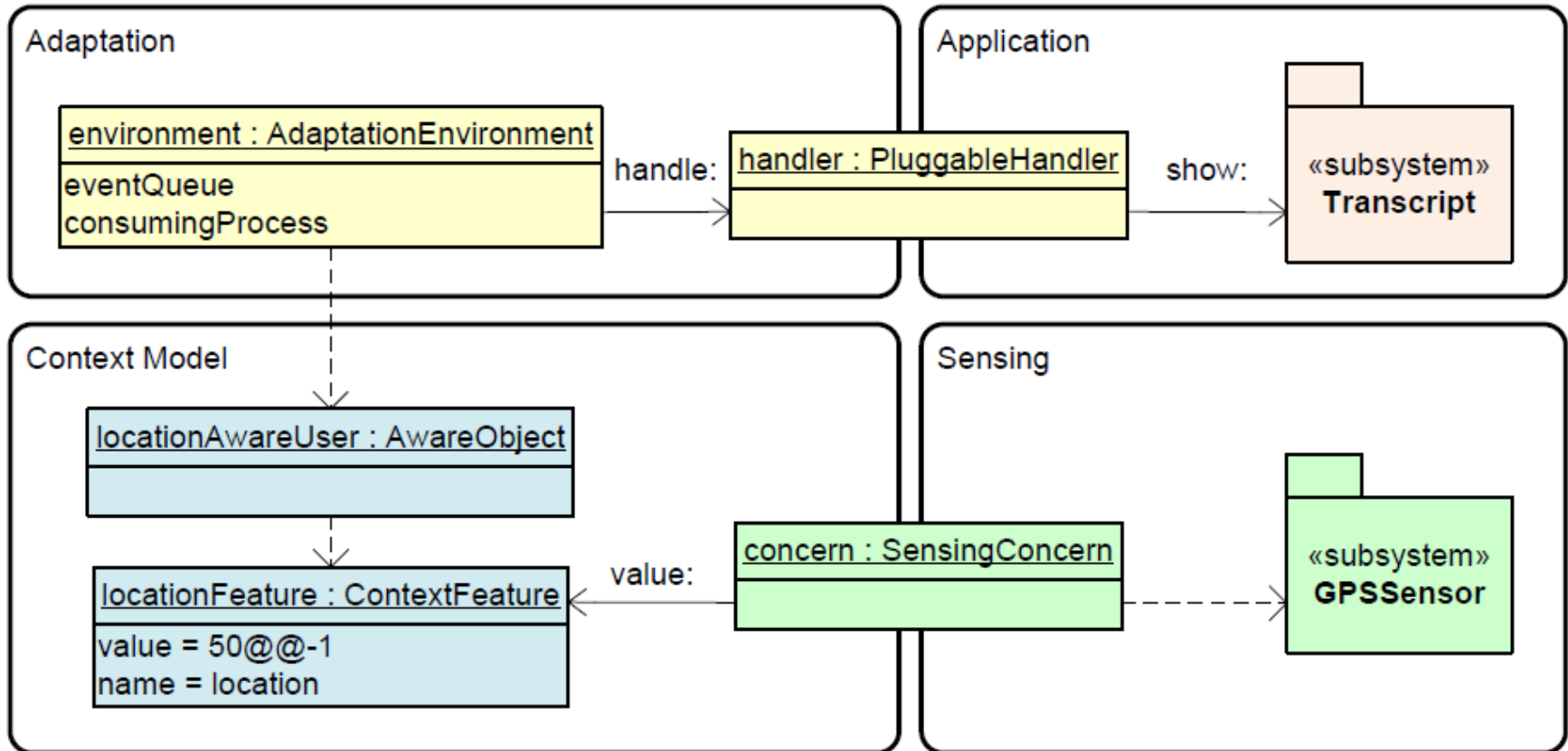
Política de publicación de los valores

36

- La función básica de un ContextFeature es publicar los cambios en su valor.
- Existen casos donde una aplicación puede no requerir todas las actualizaciones, sino solamente las que cumplan un criterio en particular:
 - Cuando el valor nuevo no es nulo.
 - Cuando el valor nuevo es efectivamente distinto al anterior.

Interacción entre los componentes basicos

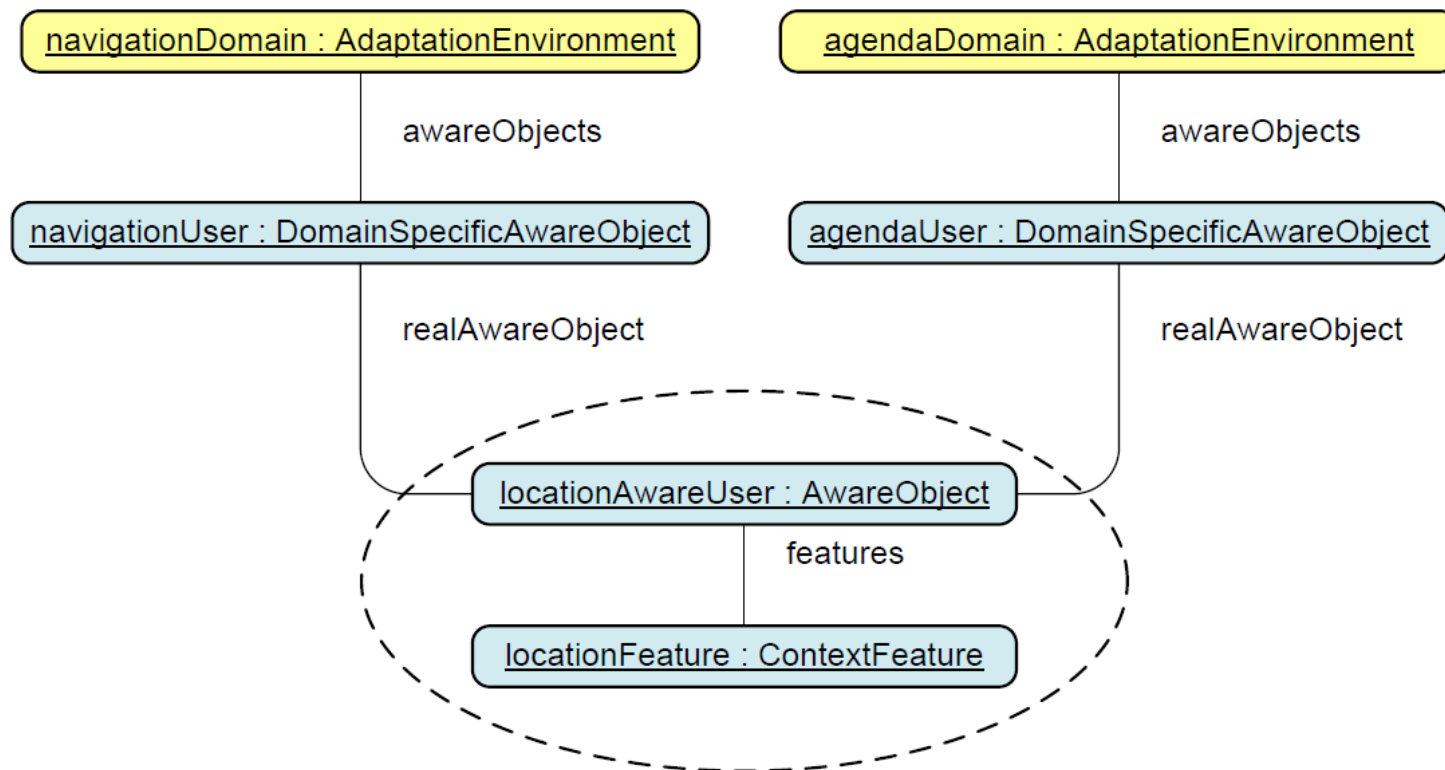
37



Sujetos reales y específicos al dominio

38

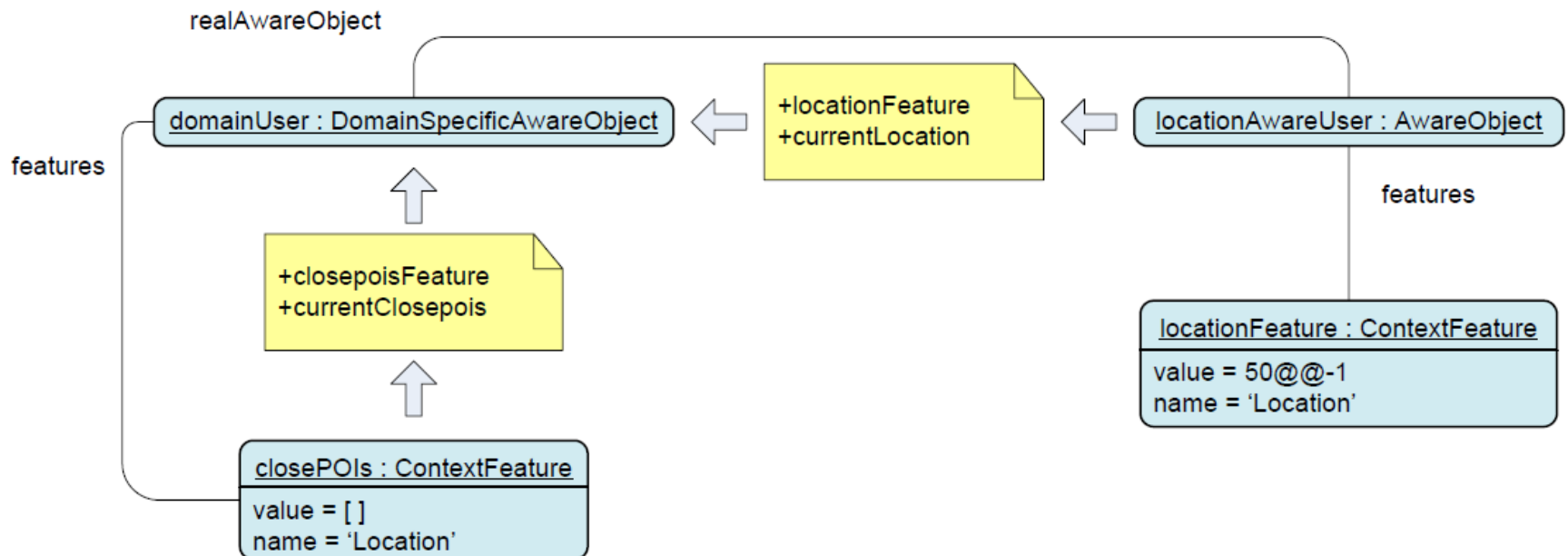
- El segundo rol del AdaptationEnvironment es proveer una interpretación sobre un AwareObject.



Sujetos reales y específicos al dominio

39

```
domainUser := environment register: locationAwareUser.  
closePOIs :=  
  ContextFeature  
    named: 'Close POIs'  
    value: OrderedCollection new.  
domainUser addFeature: closePOIs.
```



Aspectos contextuales complejos

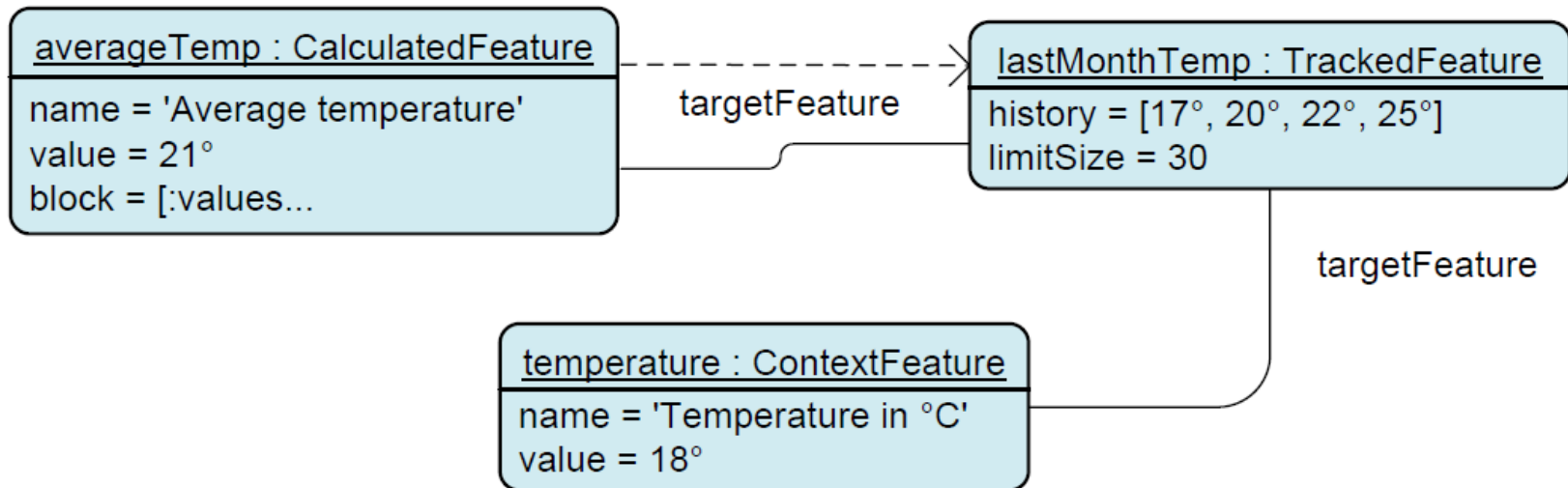
40

- Los ContextFeatures simples son solo contenedores de valores que reciben actualizaciones desde un sensor, y publican el cambio.
- Los aspectos complejos representan interacciones entre el sujeto y otros aspectos contextuales.
- No obtienen sus valores de un SensingConcern, sino de datos ya existentes en el modelo.
 - No todos entienden el mensaje #value:
 - Se actualizan automáticamente mediante el mecanismo de eventos (Patrón *Observer*).

Aspectos contextuales complejos

41

- Aspectos registrados
 - Aspectos que guardan un historial con sus valores anteriores.
- Aspectos calculados
 - Aspectos que calculan su valor en base a los valores historicos.

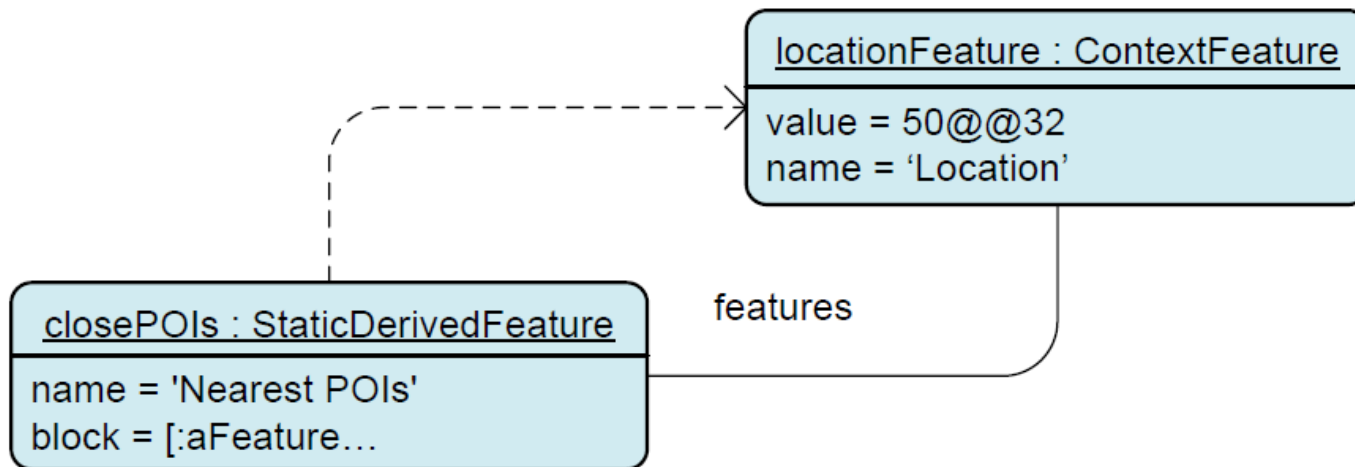


Aspectos contextuales complejos

42

- Aspectos derivados

- Aspectos que calculan su valor utilizando como entrada datos de otros aspectos.

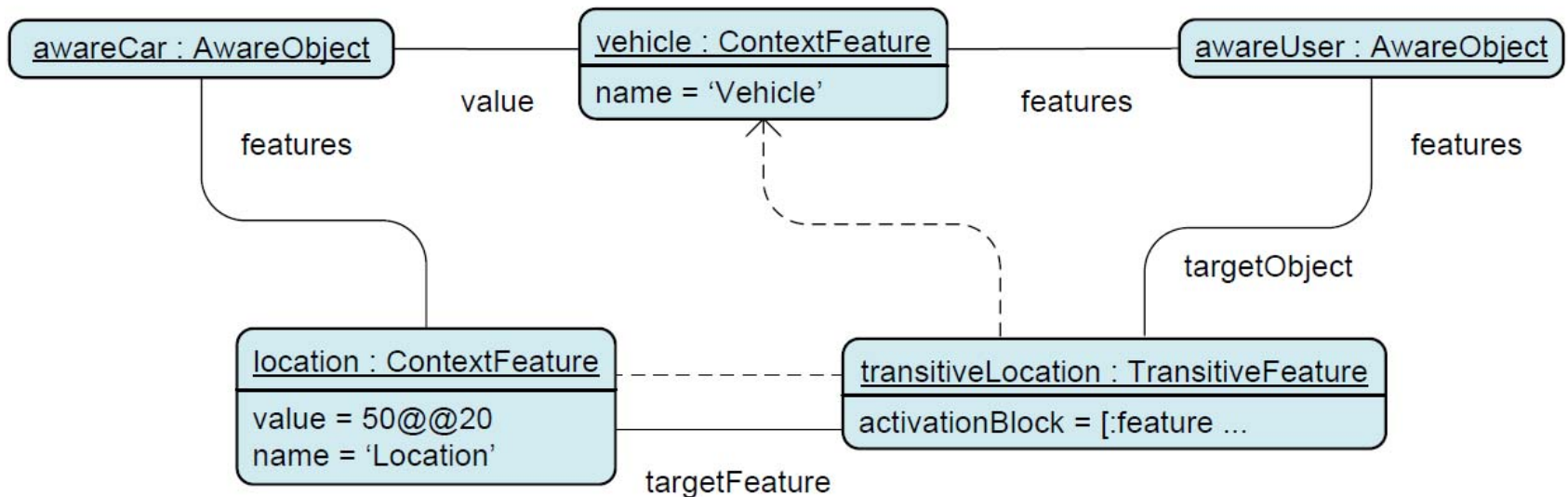


Aspectos contextuales complejos

43

- Aspectos transitivos

- Aspectos que se comparten entre dos entidades cuando se cumple una condición.



Conclusiones

44

- La información contextual se comporta de manera dinámica y responde a distintos tipos de interacciones.
 - Las distintas técnicas utilizadas para modelarla se adaptan mejor a un tipo de información particular.
- Los aspectos dinámicos del contexto pueden contribuir a obtener mejores modelos para representarlo.
 - Expresivos.
 - Reutilizables.
 - Flexibles.
 - Escalables.

Trabajos Previos



- *“Rethinking Context Models”* (2009)
 - ✦ E. Pérez, A. Fortier, G. Rossi, and S. Gordillo.
- *“Un enfoque orientado a objetos para software context-aware”* (2005)
 - ✦ A. Fortier.
- *“An Object-Oriented Approach for Context-Aware Applications”* (2006)
 - ✦ A. Fortier, N. Cañibano, J. Grigera, G. Rossi, and S. Gordillo.

Principales Referencias



- *“What we talk about when we talk about context”* (2004)
 - ✦ P. Dourish.
- *“Contextuality: A philosophical paradigm, with applications to philosophy of cognitive science”* (2002)
 - ✦ C. Gershenson.
- *“A context modeling survey”* (2004)
 - ✦ T. Strang and C. Linnho-Popien.