

CUIA 21/22 - Tema 4: Consciencia del contexto

1. Contexto

La comunicación directa entre humanos se enriquece con información del entorno ¿Qué tipo de información del entorno usamos?

En la computación clásica los dispositivos no entienden el lenguaje natural ni son capaces de reconocer una situación a partir de los datos del entorno. Esto obliga a suministrar explícitamente dicha información al ordenador.

El suministro explícito de información rompe con la transparencia que deseamos en un sistema de computación ubicua - pero si olvidamos por un momento esa transparencia, y la suministramos de forma explícita, la comunicación user-máquina está muy lejos de la comunicación entre humanos. ¿Cómo podemos mejorar esa deficiencia?

- Mejorando el lenguaje que los humanos pueden usar para comunicarse con el ordenador
 - El objetivo es una comunicación más natural
 - Puede combinar lenguaje verbal y gestual
- Usar el contexto de un modo implícito para enriquecer la comunicación entre humano y computador
- ¿Qué elementos del contexto emplearía el ordenador?
 - Expresiones faciales
 - Hechos recientes
 - Existencia de otras personas cerca
 - ...

Contexto: "Cualquier información que puede ser usada para caracterizar la **situación** de una entidad. Una entidad es una persona, objeto o lugar que se considera relevante para la interacción entre usuario y aplicación, ambas incluidas."

A.K. Dey 2001

Situación: es el resultado de agregar los datos de contexto, por lo que se encuentra en un nivel más alto de abstracción.

"Descripción de los estados de las entidades relevantes"

A.K. Dey 2001

Tipos de contexto

Para cada situación unos tipos de contexto serán más útiles que otros

- Físico: fenómenos o medidas del mundo físico
- Humano: características de los usuarios
- Virtual: servicios disponibles

Algunos tipos de contexto suelen ser más importantes:

- **Localización:** ¿Dónde ocurre?

- **Identidad:** ¿Quién participa?
- **Tiempo:** ¿Cuándo ocurre?

La situación es una descripción de **qué** está pasando

Categorías de contexto

- Primario
 - Más importante que el secundario
 - Ej. localización, tiempo, identidad
- Secundario
 - Se puede deducir a partir del primario
 - Ej. distancia, relaciones

2. Consciencia del contexto

Un sistema es consciente de contexto si usa el contexto para suministrar servicios o información relevante al usuario, donde la relevancia depende de la tarea del usuario

¿Qué acciones realiza un sistema consciente del contexto?

- Presentación de información y servicios
 - Por ejemplo un portátil que muestra las impresoras cercanas
- Ejecución automática de servicios
 - Por ejemplo la emisión de un aviso cuando un amigo se encuentra cerca

Propósito

Una aplicación consciente de contexto usa el contexto para entender el propósito del usuario y así actuar apropiadamente.

Ej. Consciencia de contexto y propósito: un visitante de un museo lleva un rato detenido delante de una de una obra inspirada en la ciudad de Granada.

El sist analiza el contexto y puede ofrecerle al visitante más información sobre la ciudad, sobre el artista...

El visitante realmente lleva un rato parado hablando por teléfono con un amigo, delante de una obra que no le interesa

Aquí es dónde entra el propósito, el sistema debe usar el contexto (ej. está realmente el visitante mirando la obra?) para conocer su propósito

Necesitamos más información de contexto para una mejor determinación del propósito del usuario. La aplicación debería estimar un grado de certeza acerca del propósito que ha estimado

Inferencia contextual

Proceso mediante el cual un sistema consciente de contexto obtiene datos del entorno y determina la **situación** en la que se encuentra el usuario. Dicha situación se empleará para inferir el **propósito** del usuario

3. Representación del contexto

Requisitos

- **Heterogeneidad**
 - Múltiples fuentes de información con diversas frecuencias de actualización y nivel semántico.
 - Sensores, bases de datos, perfiles de usuario...
- **Movilidad**
 - La información contextual será empleada en aplicaciones móviles
 - La aplicación empleará información procedente de fuentes móviles
 - La información contextual deberá adaptarse al entorno cambiante
- **Relaciones y dependencias**
 - Se deben capturar las relaciones existentes entre los distintos datos del contexto
 - Una de las relaciones es la de dependencia, cuando entidades o hechos dependen de otra entidad contextual
- **Tiempo**
 - Puede ser necesario acceder a datos pasados o estimar futuros estados
 - La frecuencia con que se producen cambios puede dificultar su gestión
- **Imperfección**
 - La información contextual puede ser de calidad variable debido a su naturaleza heterogénea
 - Los sensores tienen una precisión limitada
 - Podemos encontrarnos con datos incorrectos o incompletos
- **Razonamiento**
 - La información contextual se usará para tomar decisiones
 - Es importante la eficiencia computacional de las técnicas de razonamiento usadas
- **Usabilidad de formalismos de modelado**
 - Los diseñadores crean modelos que les permiten manipular la información del contexto
 - Los formalismos de modelado facilitan la traducción de conceptos del mundo real en modelos y su posterior uso
- **Suministro eficiente de contexto**
 - Los modelos grandes con muchos elementos necesitan un acceso eficiente al contexto.

Formas de representación del contexto

- **Pares clave/valor**
 - Pros: Muy fácil de gestionar
 - Contrás: Pobre expresividad, poco eficiente, problemas con valores ausentes y atributos multivaluados
- **Lenguaje de marcado**
 - Pros: Capaz de gestionar información incompleta y heterogénea, acceso a la información mediante un lenguaje de consulta
 - Contrás: Débil formalismo
- **Grafos**
 - Pros: Más expresivo que los pares clave/valor y el lenguaje de marcado
 - Contrás: Gestión de información incompleta, soporte de modelos distribuidos

- **Lógica**
 - Pros: Fuerte formalismo, expresividad en la estructura
 - Contrás: Gestión de datos incompletos, inciertos y heterogéneos, estructurado simple
- **Ontología**
 - Pros: Estructurado expresivo, representación de información heterogénea
 - Contrás: Gestión de incertidumbre, escalabilidad

Inferencia contextual

Con frecuencia la información contextual disponible no es suficiente y nos encontramos con problemas de **ambigüedad** e **incertidumbre**

Ambigüedad contextual

- Origen:
 - Sensores defectuosos
 - Sensores con precisión limitada
 - Entornos sin sensores
 - Sistemas de inferencia contextual que no pueden alcanzar conclusiones precisas
- ¿Cómo actuamos ante ambigüedades?
 - Sistemas que suponen que el mundo no es ambiguo
 - Sistemas capaces de tratar con la ambigüedad

Inferencia contextual

Hemos visto antes que a partir de los datos del contexto obtenemos la situación. ¿Cómo estimamos el propósito una vez inferida la situación? Tenemos dos alternativas

- **Sistemas basados en reglas:** la decisión de qué acción realizar ante una situación determinada viene dada por un conjunto de reglas
 - Pros:
 - Las reglas son fáciles de construir ya que el formato usado es homogéneo
 - Existen multitud de motores de SBR
 - Contrás:
 - Al añadir nuevas reglas pueden producirse conflictos entre reglas provocados por dependencias ocultas
 - Los sistemas con muchas reglas son difíciles de depurar
 - Las reglas son muy rígidas
- **Aprendizaje automático:** se recopila información de los tipos de situaciones que el usuario puede experimentar y cuál sería el propósito adecuado.
Se emplean técnicas de aprendizaje automático para aprender la relación entre las situaciones y los propósitos.
- Aún es necesario inferir la situación a partir del contexto.
 - Pros:
 - Podría usarse aprendizaje automático para determinar el propósito directamente del contexto.
 - Contrás:
 - El aprendizaje puede ser muy lento y necesitar muchos datos
 - Las relaciones aprendidas pueden ser muy difíciles de depurar

- Los resultados pueden no ser intuitivos para el desarrollador o el usuario final

Razonamiento con incertidumbre

- Lógica difusa
- Lógica probabilística
- Redes bayesianas
- Modelos ocultos de Markov
- Teoría de la evidencia de Dempster-Shafer

4. Usuario final

Hay dos aspectos que el desarrollador debe tener en cuenta

- **Inteligibilidad**
 - Puede ser difícil conseguir que el usuario entienda por completo el comportamiento de la aplicación por la comunicación implícita que se establece
 - Puede que el usuario no sepa que el sistema ha actuado
 - No siempre habrá un mecanismo para comunicar que se esté llevando a cabo una acción
 - En sistemas con aprendizaje automático se acentúa el problema porque no es habitual la generación de explicaciones
- **Control**
 - Las aplicaciones conscientes de contexto necesitan personalizarse para los usuarios y no funcionar para un "usuario estándar"
 - Debe permitirse al usuario controlar el modo en que va a comportarse la aplicación

Privacidad

Los sistemas conscientes de contexto recopilan mucha información sobre los individuos

- Riesgo de que la información caiga en malas manos
- Riesgo de que la información se use en una situación no adecuada
- La interconexión casi global de dispositivos agrava las consecuencias de estos problemas
- Los desarrolladores deben tener muy en cuenta estos detalles para garantizar la privacidad