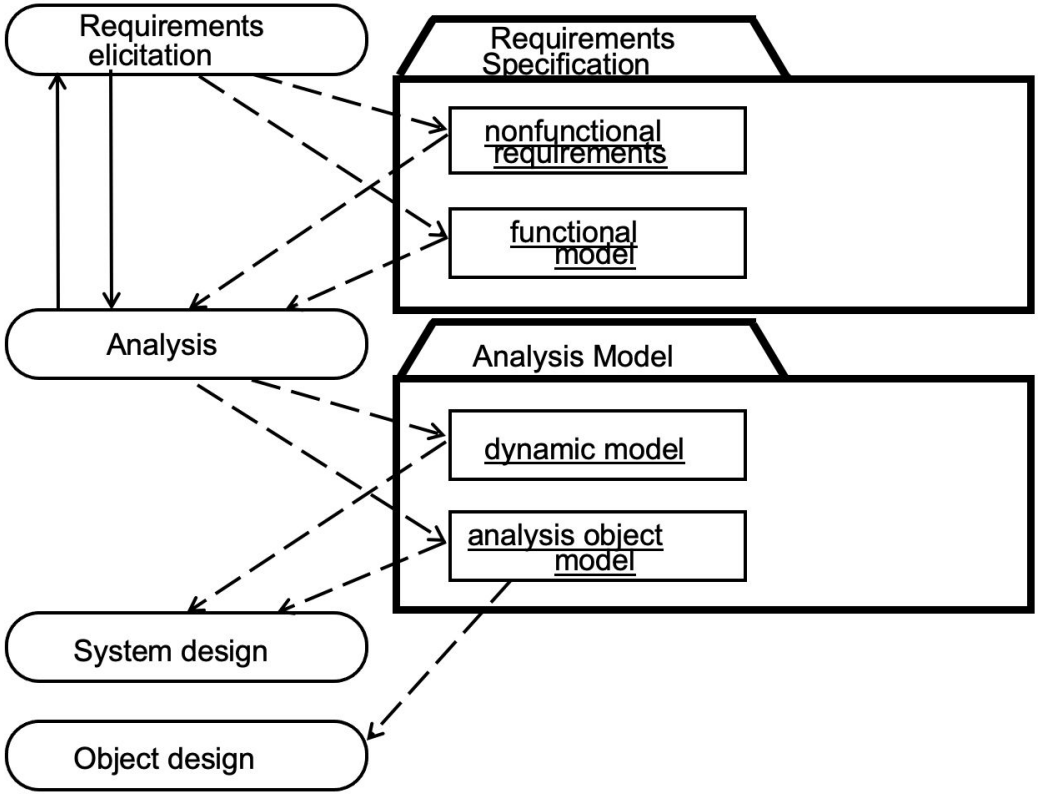
14. Analisi, Modellazione degli oggetti

Prof. Paola Barra
a.a. 2022/2023

Analisi e i suoi prodotti



Modello di analisi

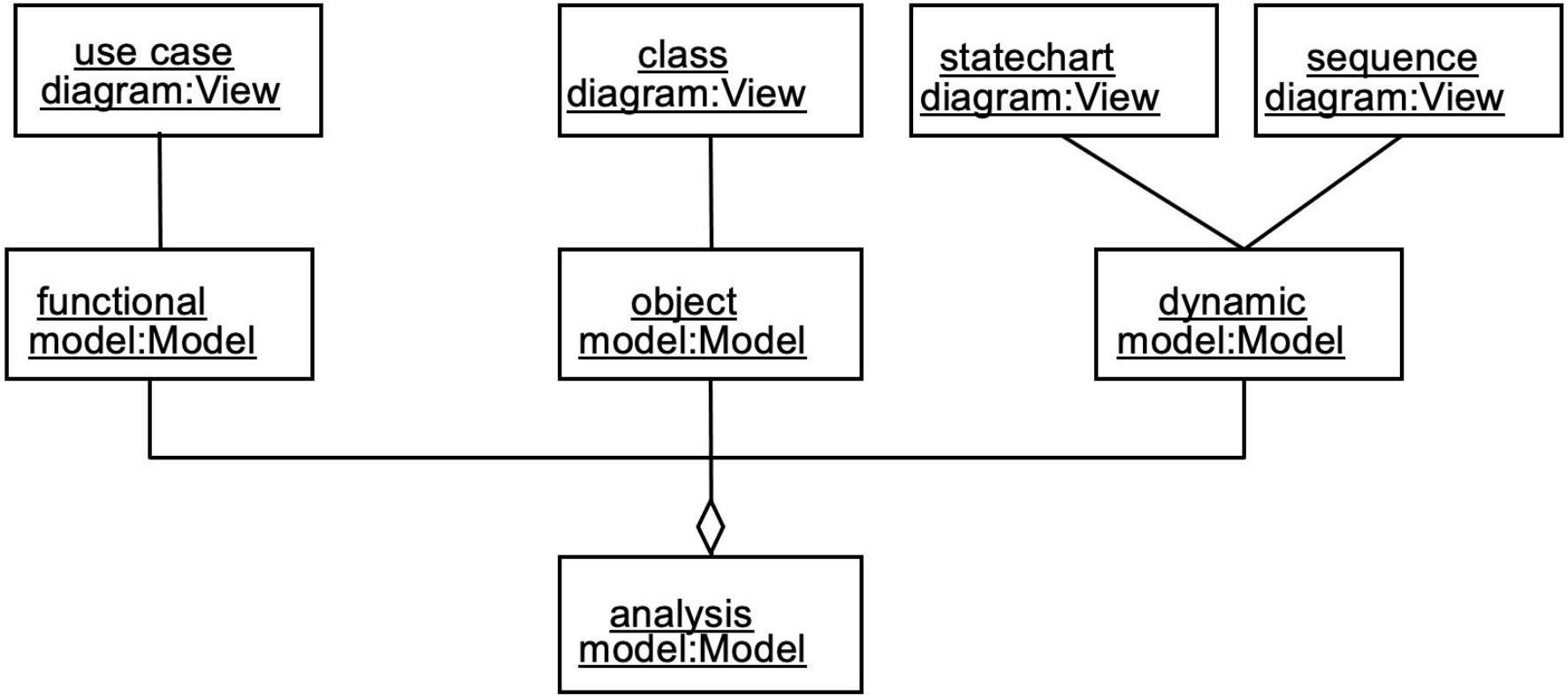
Il modello di analisi è composto da tre singoli modelli

- Il **modello funzionale**, rappresentato da casi d'uso e scenari

- Il **modello ad oggetti di analisi**, rappresentato da diagrammi delle classi e degli oggetti

- Il **modello dinamico**, rappresentato dai diagrammi di stato e delle sequenze

Il modello di analisi



Attività di analisi dei requisiti

Modellazione degli oggetti

- Attività durante la modellazione degli oggetti
- Identificazione degli oggetti
- Tipi di oggetti
- Entità (**Entity**), confine (**Boundary**), controllo (**Control**)
- Stereotipi
- Tecnica di Abbott: aiuta nell'identificazione degli oggetti

Attività durante la modellazione degli oggetti

- Obiettivo principale: trovare le astrazioni importanti
  - Passi
    1. Identificazione delle classi
       - Basata sull'assunto essenziale che possiamo trovare le astrazioni
    2. Trovare gli attributi
    3. Trovare i metodi
    4. Trovare le associazioni tra le classi
  - Ordine dei passi
    - Obiettivo: determinare le astrazioni desiderate
    - L'ordine dei passi è secondario, solo un'euristica
  - Cosa accade se troviamo le astrazioni errate?
    - Iteriamo e revisioniamo il modello

Il diagramma delle classi

Un diagramma delle classi (Class Diagram) rappresenta una vista statica del sistema. Descrive gli attributi e le operazioni delle classi. Il diagramma delle classi è il diagramma di modellazione più utilizzato per i sistemi orientati agli oggetti perché possono essere mappati direttamente con linguaggi orientati agli oggetti.

Gli scopi principali dei diagrammi di classe sono:

1. Mostrare la struttura statica dei classificatori in un sistema
2. Fornire una notazione di base per altri diagrammi di struttura prescritti da UML
3. Utile anche per sviluppatori e altri membri del team
4. Gli analisti aziendali possono utilizzare i diagrammi delle classi per modellare i sistemi da una prospettiva aziendale.

La classe e il diagramma delle classi

Una classe rappresenta un’entità logica a cui sono state associate una serie di attributi che ne descrivono le caratteristiche.

E un insieme di operazioni che ne caratterizzano il comportamento.

Il diagramma delle classi rappresenta le classi di un sistema e le eventuali relazioni che ci sono tra di loro.

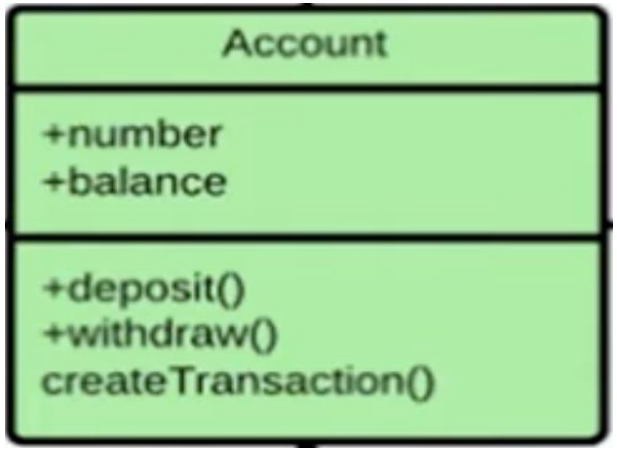
Le classi

Il simbolo grafico che rappresenta le classi UML è un rettangolo suddiviso in tre scomparti : nome della classe, attributi e operazioni.

Classe: Account

Attributi : number e balance

Operazioni: deposit, withdraw e create Transaction



Le varie classi sono in relazione tra loro con una serie di relazioni.

Identificazione delle classi

- Cruciale
  - Aiuta ad identificare le entità importanti del sistema
- Assunzioni di base
  - Possiamo trovare le classi in un nuovo sistema software (Forward engineering)
  - Possiamo identificare le classi in un sistema esistente (Reverse engineering)
- Come possiamo procedere?

Identificazione delle classi

- Approcci
  - Approccio del dominio applicativo
    - Chiedere agli esperti di identificare le astrazioni più rilevanti
  - Approccio sintattico
    - Iniziare con i casi d'uso
    - Analizzare il testo per identificare gli oggetti
    - Estrarre gli oggetti partecipanti dal flusso di eventi
  - Approccio con i design pattern
    - Usare design pattern riusabili
  - Approccio basato su componenti
    - Identificare classi di soluzioni esistenti

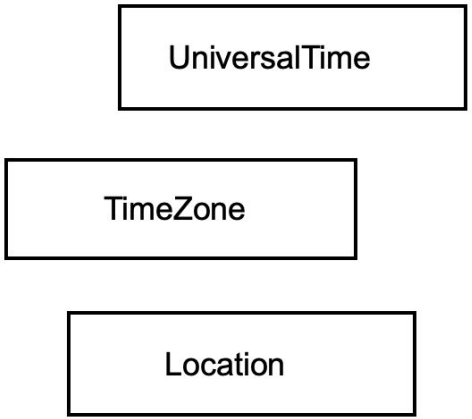
Identificazione degli oggetti, è un problema difficile

- Un problema: definizione del contorno (o confine) del sistema
  - Quali astrazioni sono all'esterno, quali all'interno?
    - Gli attori sono all'esterno
    - Classi/oggetti sono all'interno
- Un altro problema: le classi/oggetti non si trovano considerando solo una prospettiva del dominio o di una scena
  - Il dominio applicativo deve essere analizzato da diverse prospettive
  - A seconda dello scopo del sistema possono essere individuati diversi oggetti
    - Come possiamo identificare lo scopo del sistema?
    - Scenari e casi d'uso -> modello funzionale

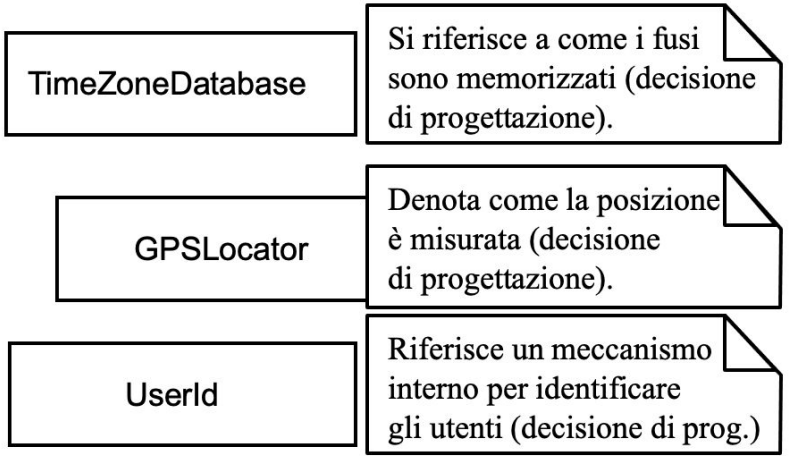
Esempi e contro-esempi di classi nel modello ad oggetto di analisi SatWatch

- Il modello ad oggetti di analisi rappresentano concetti a livello utente e non le classi o componenti effettivi del software

Concetti del dominio che dovrebbero essere rappresentati nel modello ad oggetti di analisi



Classi software che non dovrebbero essere rappresentate nel modello ad oggetti di analisi



Tipi diversi di oggetti

- Oggetti **Entity**
  - Rappresentano le informazioni persistenti mantenute dal sistema (oggetti del dominio applicativo)
- Oggetti **Boundary**
  - Rappresentano l'interazione tra l'utente ed il sistema
- Oggetti **Control**
  - Rappresentano i compiti di controllo eseguiti dal sistema

Esempio: modellazione 2B Watch

Per distinguere i diversi tipi di oggetti in un modello possiamo usare il meccanismo UML dello Stereotipo

Oggetti Entity Oggetti Control Oggetti Boundary

Tipi di oggetti

- Le tre categorie di oggetti consentono ai modelli di essere più resilienti alle modifiche
  - L'interfaccia di un sistema cambia con maggiore probabilità rispetto al controllo
  - Il modo in cui il sistema è controllato cambia con maggiore possibilità rispetto alle entità di un dominio applicativo
- I tipi degli oggetti sono stati introdotti in Smalltalk
  - Model, View, Controller (MVC)
    - Model <-> Oggetto Entity
    - View <-> Oggetto Boundary
    - Controller <-> Oggetto Control

Trovare gli oggetti partecipanti nei casi d'uso

- Prendere un **caso d'uso** e leggere il flusso degli eventi
- Fare un'analisi testuale (analisi sostantivo-verbo)
  - I sostantivi sono candidati per gli oggetti/classi
  - I verbi sono candidati per le operazioni
  - Questa procedura è nota come tecnica di Abbott
- Dopo che gli oggetti/classi sono stati trovati, identificare i tipi
  - Identificare le entità del mondo reale di cui il sistema deve tener traccia (FieldOfficer-> Oggetto Entity)
  - Identificare procedure del mondo reale di cui il sistema deve tenere traccia (EmergencyPlan -> Oggetto Control)
  - Identificare artefatti di interfaccia (PoliceStation-> oggetto Boundary)

Esempio

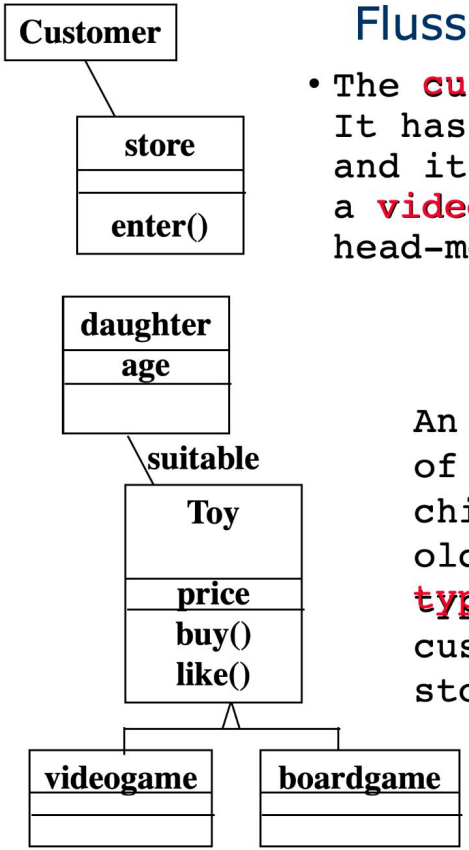
Flusso di eventi:

- The customer enters the store to buy a toy.
- It has to be a toy that his daughter likes and it must cost less than 50 Euro.
- He tries a videogame, which uses a data glove and a head-mounted display. He likes it.
- An assistant helps him.
- The suitability of the game depends on the age of the child.
- His daughter is only 5 years old.
- The assistant recommends another type of toy, namely the boardgame “Monopoly".

Mapping parti del parlato ai componenti del modello

Example	*Part of speech*	*UML model component*
"Monopoly"	Proper noun	object
Toy	Improper noun	class
Buy, recommend	Doing verb	operation
is-a	being verb	inheritance
has an	having verb	aggregation
must be	modal verb	constraint
dangerous	adjective	attribute
enter	transitive verb	operation
depends on	intransitive verb	Constraint, class, association

Generare il diagramma delle classi dal flusso degli eventi



Flusso di eventi:

- The **customer enters** the **store** to **buy** a **toy**. It has to be a toy that his **daughter** likes and it must cost **less than 50** Euro. He tries a **videogame**, which uses a data glove and a head-mounted display. He likes it.

An assistant helps him. The suitability of the game **depends** on the **age** of the child. His daughter is only 5 years old. The assistant recommends another **type of toy**, namely a **boardgame**. The customer buy the game and leaves the store

Modi per trovare oggetti

Investigazione sintattica con la tecnica di Abbott

- Flusso di eventi nei casi d'uso
- Descrizione del problema

Usare altre sorgenti di conoscenza

- Conoscenza dell'applicazione: gli utenti e gli esperti conoscono le astrazioni del dominio applicativo
- Conoscenza della soluzione: astrazioni nel dominio della soluzione
- Conoscenza generale del mondo: la vostra conoscenza e intuizione

Ordine delle attività per l'identificazione degli oggetti

1. Formulare pochi scenari con l'aiuto di un utente o esperto del dominio applicativo
2. Estrarre i casi d'uso dagli scenari, con l'aiuto di un esperto del dominio applicativo
3. Poi procedere in parallelo con:
   - Analizzare il flusso di eventi in ogni caso d'uso usando la tecnica di analisi testuale di Abbott
   - Generare il diagramma delle classi

Passi nella generazione del diagramma delle classi

1. Identificazione della classe (analisi testuale, esperto del dominio)
2. Identificazione di attributi e operazioni (talvolta prima che le classi siano trovate)
3. Identificazione di associazioni tra classi
4. Identificazione delle molteplicità
5. Identificazione dei ruoli
6. Identificazione dell’ereditarietà

Esempio

Dopo un primo esame del caso d'uso **ReportEmergency** si impiegano la conoscenza del dominio applicativo e le interviste con gli utenti per identificare gli oggetti **Dispatcher**, **EmergencyReport**, **FieldOfficer** e **Incident**

E' da osservare che l'oggetto EmergencyReport non è menzionato esplicitamente nel caso d'uso ReportEmergency

- Il passo 5 del caso d'uso si riferisce ai rapporti di emergenza come "informazioni sottomesse dal FieldOfficer".
- Dopo una revisione con il cliente si scopre che a questa informazione ci si riferisce solitamente come "emergency report" e si decide quindi di chiamare il corrispondente oggetto EmergencyReport

Nome	ReportEmergency
Attori Partecipanti	Iniziato dal FieldOfficer Comunica con il Dispatcher
Flusso eventi:	1. Il FieldOfficer attiva la funzione “Report Emergency” del terminale 2. FRIEND risponde presentando una form al FieldOfficer. *La form include un menu del tipo di emergenza (emergenza generale, incendio, trasporto) e la località, la descrizione dell’incidente, la richiesta di risorse, i campi dei materiali nocivi.* 3. Il FieldOfficer riempie la form *specificando almeno il tipo di emergenza e i campi* descrizione. Il FieldOfficer descrive anche possibili risposte alla situazione di emergenza *e può richiedere risorse specifiche*. Appena finito Il FieldOfficer invia la form 4. FRIEND riceve la form e notifica al Dispatcher *con un finestra pop-up.* 5. Il Dispatcher rivede le informazioni sottomesse dal FieldOfficer e crea un incidente nel DB invocando il caso d’uso OpenIncident. *Tutte le informazioni contenute nella form del FieldOfficer sono incluse automaticamente in Incident. Il Dispatcher seleziona una risposta allocando le risorse all’incidente (con il caso d’uso AllocateResources) e notifica il rapporto di emergenza inviando un breve messaggio al FieldOfficer.* 6. FRIEND visualizza l’accettazione e la risposta selezionata al FieldOfficer
Condizioni di uscita	Il FIeldOfficer riceve l’accettazione e la risposta selezionata

Identificazione oggetti boundary

Gli oggetti boundary rappresentano l'interfaccia del sistema con gli attori

In ciascun caso d'uso, ogni attore interagisce con almeno un oggetto boundary

L'oggetto boundary raccoglie le informazioni dagli attori e le traducono in una forma che è possibile usare con ambo gli oggetti entity e control

Euristiche per identificare oggetti boundary

Identificare gli elementi dell'interfaccia utente di cui l'utente necessita per iniziare il caso d'uso (es., ReportEmergencyButton)

Identificare le form con le quali l'utente immette dati nel sistema (es., EmergencyReportForm)

Identificare avvisi e messaggi che il sistema usa per rispondere all'utente (AcknowledgmentNotice)

Quando multipli attori sono coinvolti in un caso d'uso, identificare i terminali degli attori (es., DispatcherStation) per riferirsi all'interfaccia utente in questione

Non modellare aspetti visuali dell'interfaccia con oggetti boundary

Usare sempre i termini dell'utente finale per descrivere le interfacce; non usare termini dai domini implementativi

Oggetti boundary del caso d'uso ReportEmergency

AcknowledgmentNotice	Avviso usato per visualizzare l'accettazione del Dispatcher al FieldOfficer.
DispatcherStation	Computer usato dal Dispatcher.
ReportEmergencyButton	Pulsante usato dal FieldOfficer per iniziare il caso d'uso ReportEmergency
EmergencyReportForm	Form usata per inserire il ReportEmergency. Questa form è presentata al FieldOfficer sulla FIeldOfficerStation quando è selezionata la funzione "Report Emergency". La EmergencyReportForm contiene i campi per specificare tutti gli attributi di un rapporto di emergenza e un pulsante (o altro controllo) per sottomettere la form completa
FieldOfficerStation	Computer portatile usato dal FieldOfficer
IncidentForm	Form usata per la creazione di Incidenti. Questa form è presentata al Dispatcher sul DispatcherStation quando l'EmergencyReport è ricevuto. Il Dispatcher usa anche questa form per allocare le risorse a per accettare il rapporto del FieldOfficer

Oggetti boundary del caso d'uso ReportEmergency

- *IncidentForm* non è menzionato esplicitamente nel caso d'uso *ReportEmergency*

- L'oggetto è stato identificato osservando che il Dispatcher ha bisogno di una interfaccia per vedere il rapporto di emergenza sottomesso dal FieldOfficer e per spedire un'accettazione

- I termini usati per descrivere gli oggetti boundary nel modello di analisi dovrebbero seguire la terminologia dell'utente

Identificare gli oggetti control

Gli oggetti control sono responsabili per coordinare gli oggetti boundary ed entity

Solitamente non hanno una controparte concreta nel mondo reale

Spesso esiste una relazione stretta tra un caso d’uso e un oggetto control

Un oggetto control solitamente è creato all’inizio di un caso d’uso e cessa di esistere alla terminazione dello stesso

E’ responsabile di raccogliere informazioni dagli oggetti boundary agli oggetti entity

Identificare gli oggetti control

Inizialmente, si modella il flusso di controllo del caso d'uso *ReportEmergency* con un oggetto control per ogni attore

- ***ReportEmergencyControl*** per il *FieldOfficer* e ***ManageEmergencyControl*** per il *Dispatcher*

La decisione di modellare il flusso di controllo del caso d'uso *ReportEmergency* con due oggetti control scaturisce sapendo che *FieldOfficerStation* e *DispatcherStation* sono due sottosistemi che comunicano su un link asincrono

Rendendo questo concetto visibile nel modello di analisi consente di mettere a fuoco comportamenti eccezionali come la perdita di comunicazione tra ambo le stazioni

Oggetti control del caso d'uso ReportEmergency

ReportEmergencyControl	Gestisce la funzione di reporting del ReportEmergency sulla FieldOfficerStation. Questo oggetto è creato quando il FieldOfficer seleziona il pulsante "Report Emergency". Esso dopo crea una EmergencyReportForm e la presenta al FieldOfficer. Dopo aver sottomesso la form, tale oggetto raccoglie le informazioni dalla form, crea un EmergencyReport, e lo invia al Dispatcher. L'oggetto control dopo aspetta un'accettazione proveniente dalla DispatcherStation. Quando è ricevuta l'accettazione, l'oggetto ReportEmergencyControl crea un AcknowledgeNotice e la visualizza al FieldOfficer.
ManageEmergencyControl	Gestisce la funzione di reporting del ReportEmergency sulla DispatcherStation. Questo oggetto è creato quando è ricevuto un EmergencyReport. Esso dopo crea una IncidentForm e la visualizza al Dispatcher. Una volta che il Dispatcher ha creato un Incident, allocato Resourse e sottomesso un'accettazione, ManageEmergencyControl invia l'accettazione alla FieldOfficerStation

Euristiche per identificare gli oggetti control

Identificare un oggetto control per caso d'uso

Identificare un oggetto control per attore nel caso d'uso

La durata di vita di un oggetto control dovrebbe coprire l'estensione del caso d'uso o l'estensione di una sessione utente. E' difficile identificare l'inizio e la fine dell'attivazione di un oggetto control, il caso d'uso corrispondente probabilmente non ha condizioni di entrata e di uscita ben definite

Identificare gli oggetti control

Nel modellare il caso d'uso ReportEmergency, è stata modellata la stessa funzionalità usando gli oggetti entity, boundary e control

Spostandosi dalla prospettiva del flusso di eventi ad una prospettiva strutturale, si è incrementato il livello di dettaglio della descrizione e selezionato termini standard per riferirsi alle entità principali del dominio applicativo e al sistema

Generalizzazione e specializzazione

L’ereditarietà consente di organizzare i concetti in gerarchie

- In cima alla gerarchia c’è il concetto generale
- In fondo alla gerarchia troviamo i concetti più specializzati
- Nel mezzo possono esserci diversi livelli intermedi che rappresentano concetti più o meno specializzati

Le gerarchie consentono di riferirsi a molti concetti in modo preciso. Ad esempio, in FRIEND

- Quando ci riferiamo ad un *Incident* intendiamo tutte le istanze del tipo *Incident*
- Quando ci riferiamo a *Emergency* ci riferiamo solo ad un incidente che richiede una risposta immediata

Generalizzazione e specializzazione

La generalizzazione è l'attività di modellazione che identifica i concetti astratti dai concetti di basso livello

Esempio

- Durante la fase di reverse engineering per un sistema di gestione emergenze, scopriamo l'esistenza di schermi per gestire sinistri legati al traffico e agli incendi
- Osservando le caratteristiche comuni tra i tre concetti, creiamo una classe astratta *Emergency* per descrivere le caratteristiche comuni (e generali) di incidenti legati al trasporto e agli incendi

Generalizzazione e specializzazione

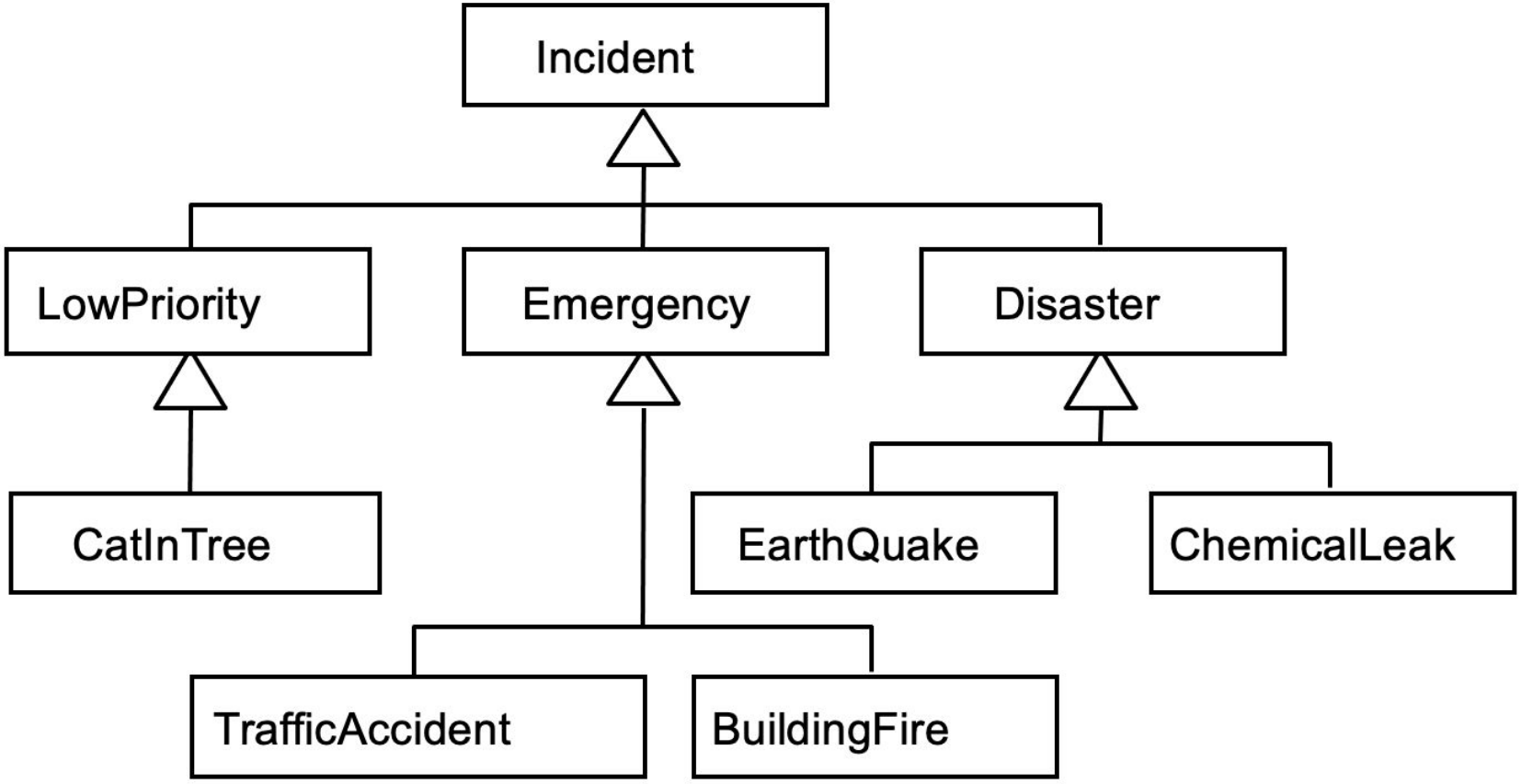
La specializzazione è l'attività che identifica concetti più specializzati da quelli ad alto livello

Esempio: stiamo costruendo un sistema di gestione emergenze cominciando da zero e stiamo discutendo le funzionalità con il cliente

Il cliente ci introduce il concetto di incidente e poi descrive tre tipi di incidenti (*Incidents*)

- Disastri (*Disasters*), che richiede la collaborazione di diverse agenzie
- Emergenze (*Emergencies*), che richiede una risposta immediata ma può essere gestita da una singola agenzia
- Incidenti con priorità bassa (*LowPriorityIncidents*), che non deve essere necessariamente gestita se le risorse sono richieste da altri incidenti con maggiore priorità

Esempio di gerarchia di generalizzazione



Chi usa il diagramma delle classi?

Scopo del diagramma delle classi: La descrizione delle proprietà statiche di un sistema

I principali utenti dei diagrammi delle classi

- Gli esperti del dominio applicativo.

  Usano i diagrammi delle classi per modellare il dominio applicativo (incluse le tassonomie)

  Durante la scoperta e l'analisi dei requisiti

- Lo sviluppatore.

  Usa i diagrammi delle classi durante lo sviluppo di un sistema

  Durante l'analisi, la progettazione del sistema, la progettazione degli oggetti e l'implementazione

Chi non usa il diagramma delle classi?

Il cliente e l'utente solitamente non sono interessati ai diagrammi delle classi

- I clienti si focalizzano su problematiche relative alla gestione del progetto
- Gli utenti sono più interessati nelle funzionalità del sistema

Riepilogo

Modellazione del sistema

•Modellazione funzionale + modellazione oggetti + modellazione dinamica

Modellazione funzionale

•Da scenari ai casi d’uso agli oggetti

Modellazione oggetti è attività centrale

•Identificazione classi è l’attività principale della modellazione degli oggetti

•Facili regole sintattiche per trovare classi e oggetti

•Tecnica di Abbott

I diagrammi delle classi sono il centro dell’universo per lo sviluppatore orientato agli oggetti

•L’utente si focalizza più sul modello funzionale e l’usabilità