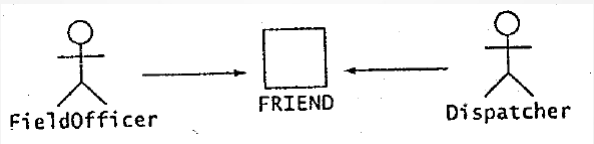
Use Case to Analysis Model

Analysis Modeling

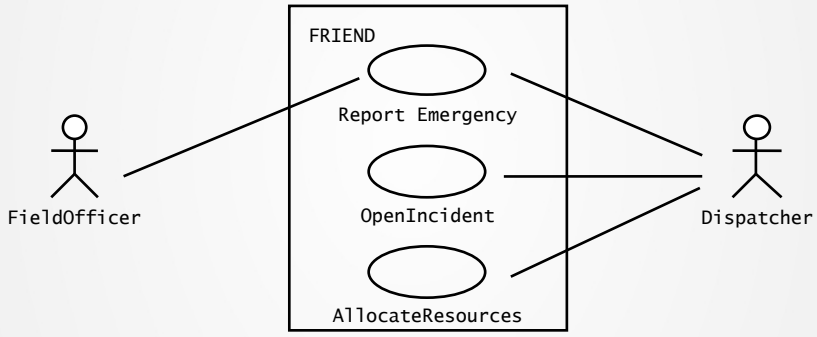
- About use case realization (analysis level)
  - "implements" a use case at the analysis level (using analysis objects/classes)
- Consists of
  - Identifying Domain objects
  - Developing Interaction diagrams, followed by
  - Developing Analysis class diagram (where each class is an abstraction of a design class (or design classes))
    - Tends to focus on functional requirements (and non-functional are relegated to design)
    - Tends to provide abstract operation interface as a text description of behavior
    - Attributes defined on a conceptual level
    - Relationships are more general – cf. navigability, generalizations,
    - Analysis classes are one of three basic stereotypes: boundary, control, entity

An Example

- FRIEND: Accident Management System
  - A distributed information system for accident management. It includes many actors such as *FieldOfficer*, who represent the police, fire officers who respond to accidents, and *Dispatcher*, the polish officer responsible for answering 911 calls and dispatching resources to an accident. FRIEND supports both actors by keeping track of incidents, resources, and task plans. The *FieldOfficer* and the *Dispatcher* interact through different interface – *FieldOfficer* interacts FRIEND through a mobile personal assistant, and *Dispatcher* access FRIEND through a workstation.



Example: Use Case Diagram (FRIEND)



Use case Report Emergency

Use case name	**ReportEmergency**
Actors	FieldOfficer, Dispatcher
Entry condition	1. The FieldOfficer activate the “report Emergency” function of her terminal
Flow of event	2. System responds by presenting a form with different details to filled-in 3. FieldOfficer completes the form. She may also describe possible responses to the situation and request specific resources. She submits the form 4. The Dispatcher reviews the information and creates an Incident in the DB by invoking **OpenIncident** use case. All the information received from FieldOfficer is then stored in the DB. The Dispatcher selects a response and allocates resources to the Incident (with **AllocateResources** use case) and acknowledge the Emergency Report by sending a FRIENDgram to the FieldOfficer
Exit condition	5. The FieldOfficer receives the acknowledgement and the selected response

Use case Report Emergency

- Entity Objects
- In general terms, entity objects encapsulate the business policy and data for
  - the logical structure of the business, such as product lines, departments, sales, and regions
  - physical items, such as warehouses, employees, and equipment

Dispatcher	Police officer who manage incidents. A Dispatcher opens, documents, and closes Incidents in response to emergency Report. Dispatchers are identified with batch numbers
Emergency Report	Initial incident information report
Field Officer	Police officer on duty and on the spot. She is responsible for reporting the incident. Recognized by her batch number.
Incident	All relevant information of the incident

Identifying Boundary Objects

- Represent system interface with the actors of the system
- Each actor must interact with at least one boundary object

Identifying Boundary Objects

Heuristics

- Identify user interfaces for actor-system interactions
- Identify forms in which data is to be filled in
- Identify communication/messages between actor and systems
- When multiple actors are involved in a use case, identify actor terminals
- Do not model visual aspects of the interface
- Use user/customer language for describing interfaces

Use case Report Emergency

- Boundary Objects
  - AcknowledgementNotice, DispatcherStation, ReportEmergencyButton, EmergencyReportForm, FieldOfficerStation, IncidentForm

AcknowledgementNotice	
DispatcherStation	Computer used by the Dispatcher
ReportEmergencyButton	
EmergencyReportForm	
FieldOfficerStation	Mobile computer used by the FieldOfficer
IncidentForm	This form is presented to the Dispatcher on the DispatcherStation when the EmergencyReport is received

Identify Control Objects

- Responsible for coordinating boundary and entity objects
- Do not have a concrete counterpart in the real world
- Life time from beginning to end of the use case

Identify Control Objects

Heuristics

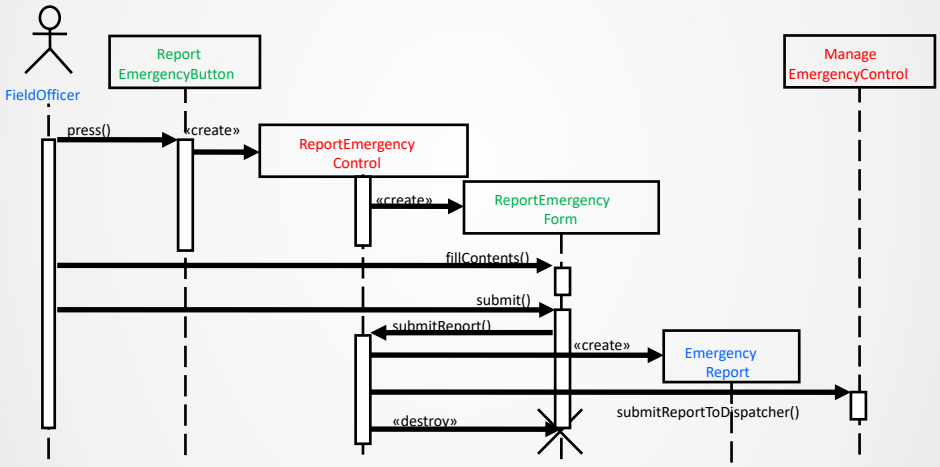
- Identify one control object per use case
- Identify one control object per actor in the use case
- Life span of a control object should cover the extent of the use case or extent of a user session

Use case Report Emergency

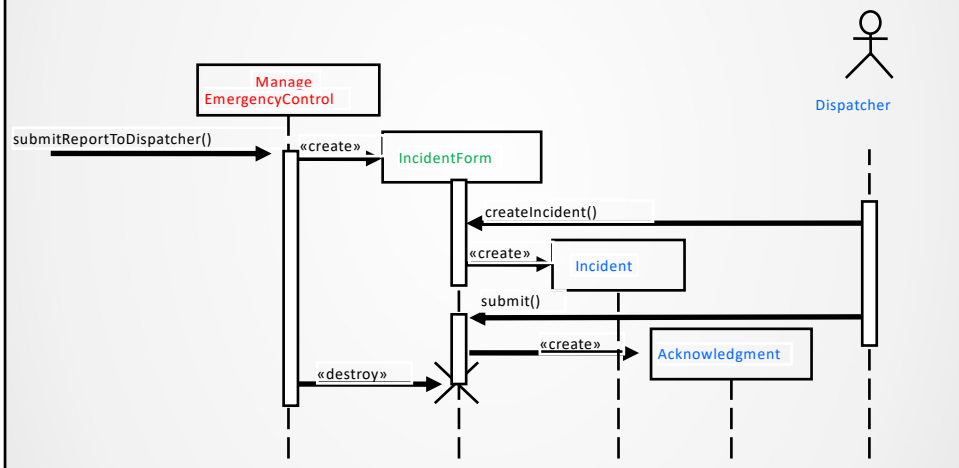
- Control Objects
  - ReportEmergencyControl, ManageEmergencyControl

ReportEmergency Control	The object is created when the FieldOfficer selects the "Report Emergency" button. It then creates an EmergencyReportForm and presents it to the FieldOfficer. After getting all the information, it forwards this information to the DispatcherStation. It then waits for an acknowledgement. When received, it creates an acknowledgementNotice and displays it to the fieldOfficer
ManageEmergency Control	This object is created when an EmergencyReport is received. It then creates an IncidentForm and displays it to the Dispatcher. Once the Dispatcher has created an incident, allocated resources, and submitted an acknowledgement, it forwards the acknowledgement to the FieldOfficerStattion

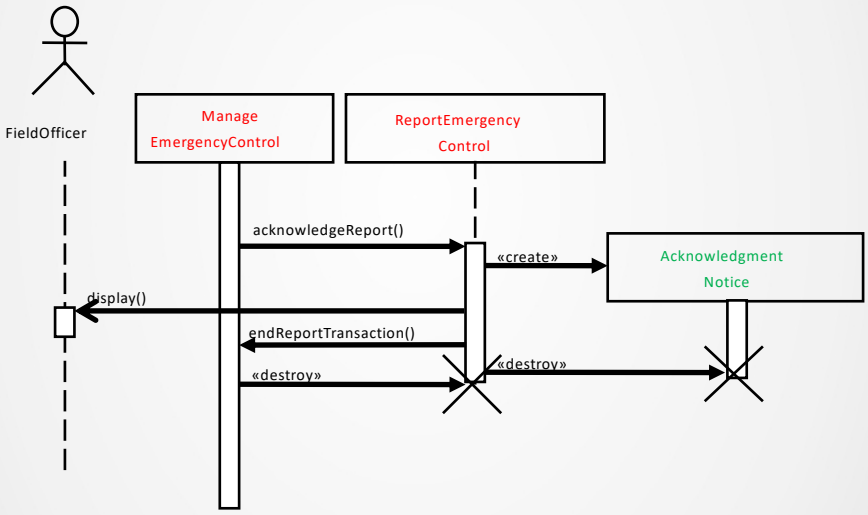
Sequence diagram for the **Report Emergency** use case.



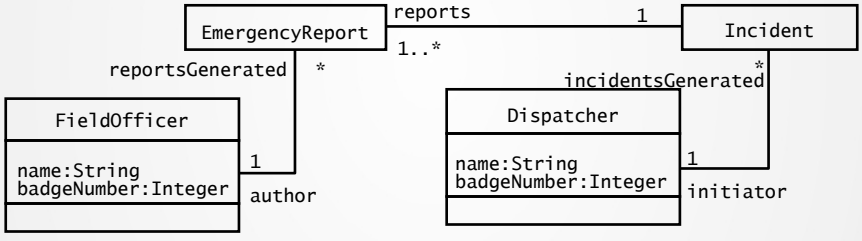
Sequence diagram for the **Report Emergency** use case.



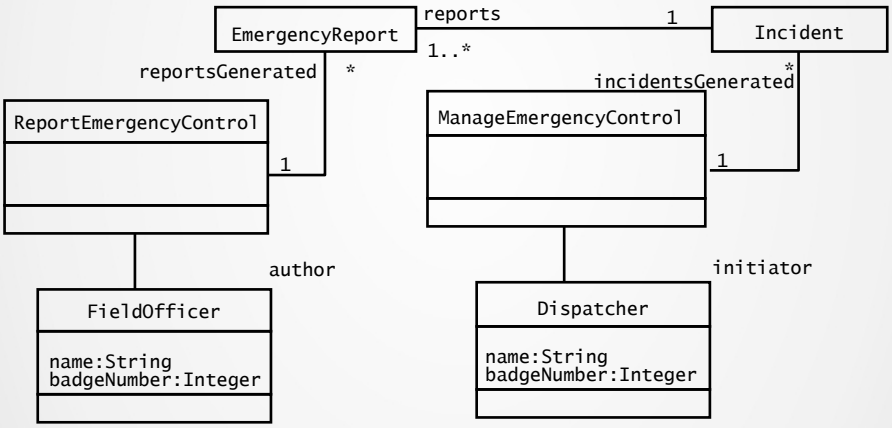
Sequence diagram for the **Report Emergency** use case.



Classes participate in the **Report Emergency** use case



Classes participate in the **Report Emergency** use case



Thank You