



SWE 205: Introduction to Software Engineering

Lecture 10

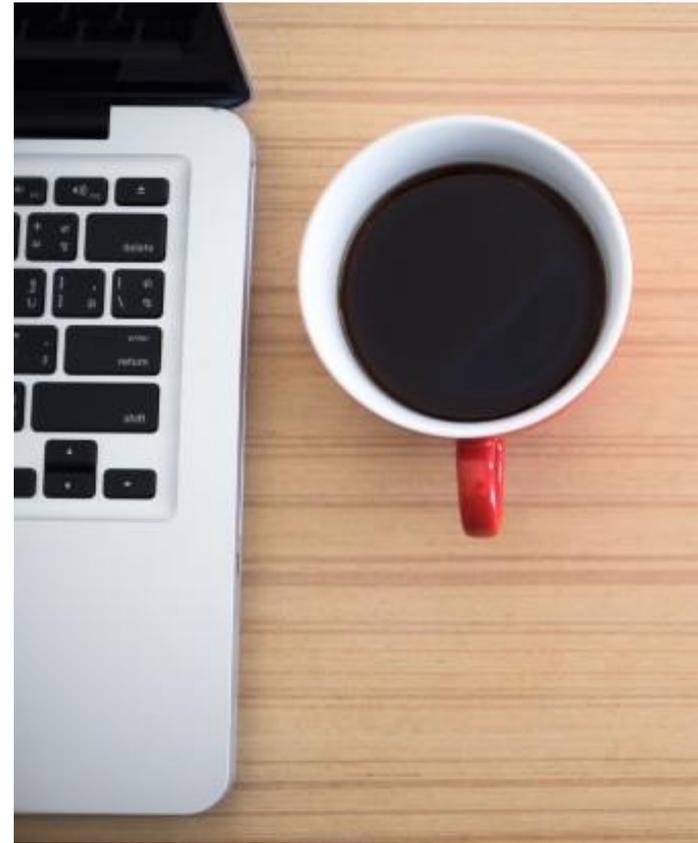
System Models

Course Topics

- ~~Introduction~~
- ~~Software Process Models~~
- ~~Requirements Engineering~~
- Modeling
- Programming Languages
- Software Construction Techniques
- Testing
- Project Management
- Refactoring
- Ethical Issues

Lecture Objectives

- ✓ Introduction
- ✓ Context Models
- ✓ Interaction modelling



System modelling

- Definition
 - The process of developing abstract models of a system
 - with each model presenting a different view or perspective of that system

- System modelling helps the analyst to understand the functionality of the system.

- Models are used to:
 - Specify these functionalities
 - Communicate and verify these functionalities with customers.

- Unified Modelling Language (UML)

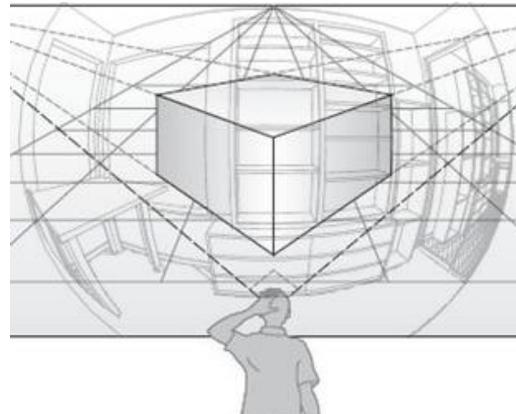
Using models



- Models can be used at
 - Requirements engineering process
 - Help derive requirements
 - Design process
 - Describe the system for implementation purposes
 - After implementation
 - Document the system structure
- 

System modelling – different perspectives

- Different models present the system from different perspectives
 1. **External** perspective: showing the context or environment of the system.
 2. **Interaction** perspective: showing the interactions between a system and its environment, or between the components of a system.
 3. **Structural** perspective: showing the organization of a system or the structure of the data that is processed by the system.
 4. **Behavioral** perspective: showing the dynamic behavior of the system and how it responds to events.



UML (Unified Modeling Language)



- A standard modeling language for Object-Oriented modeling.
 - **13** different diagram types that may be used to model software systems.
 - In 1997, similar object-oriented notations were integrated to create the UML language.
 - UML was adopted as a standard by the **Object Management Group** (OMG), and has been managed by this organization ever since.
 - A major revision (UML 2) was finalized in 2004.
- 

UML diagram types



- **Use case diagrams**, which show the interactions between a system and its environment.
 - **Activity diagrams**, which show the activities involved in a process or in data processing.
 - **Sequence diagrams**, which show interactions between actors and the system and between system components.
 - **Class diagrams**, which show the object classes in the system and the associations between these classes.
 - **State diagrams**, which show how the system reacts to internal and external events.
- 

Context models

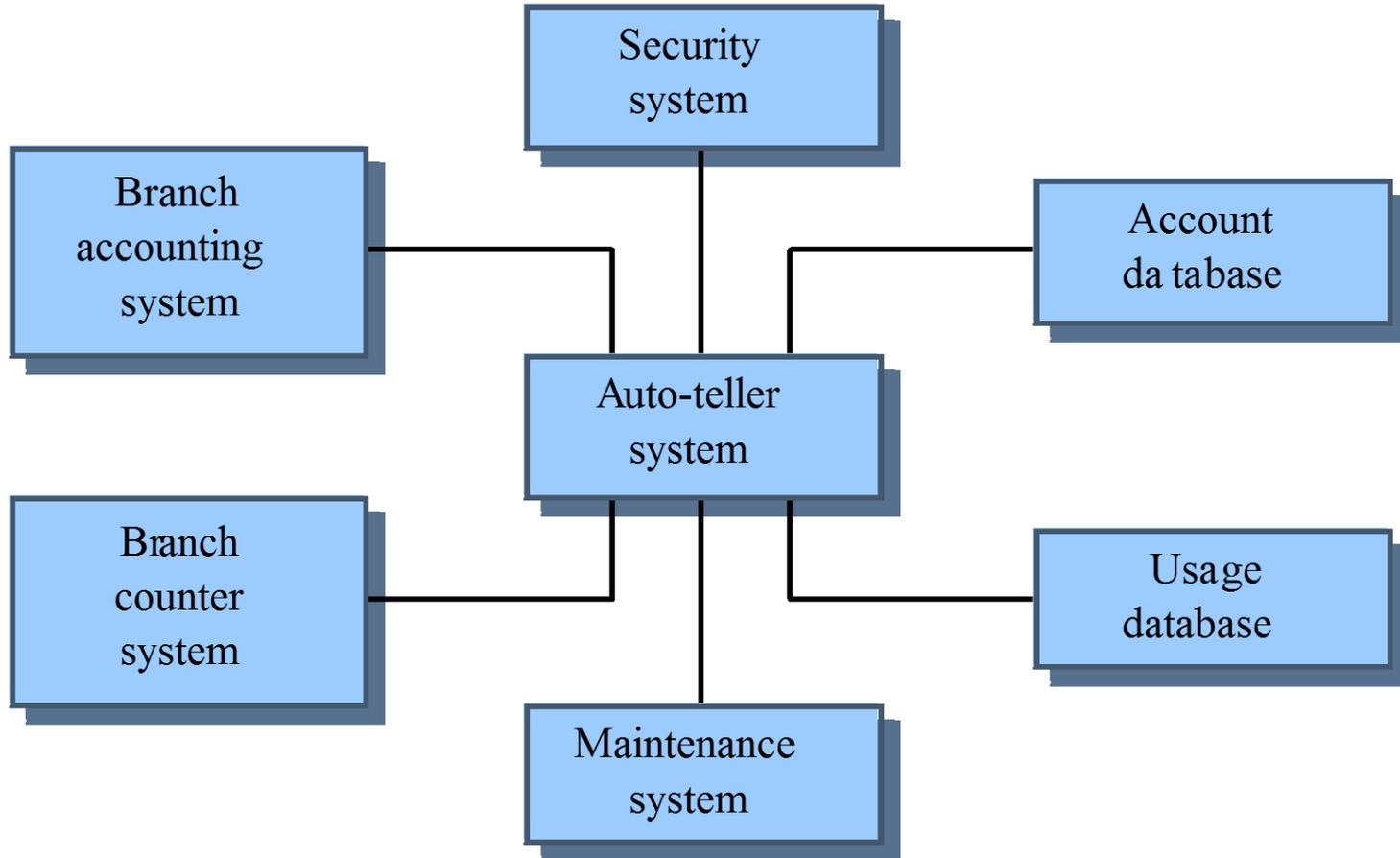


- At an early stage in the specification of a system, you should decide on the system boundaries.

- Context models are used to illustrate the operational context of a system
 - They show what lies outside the system boundaries.

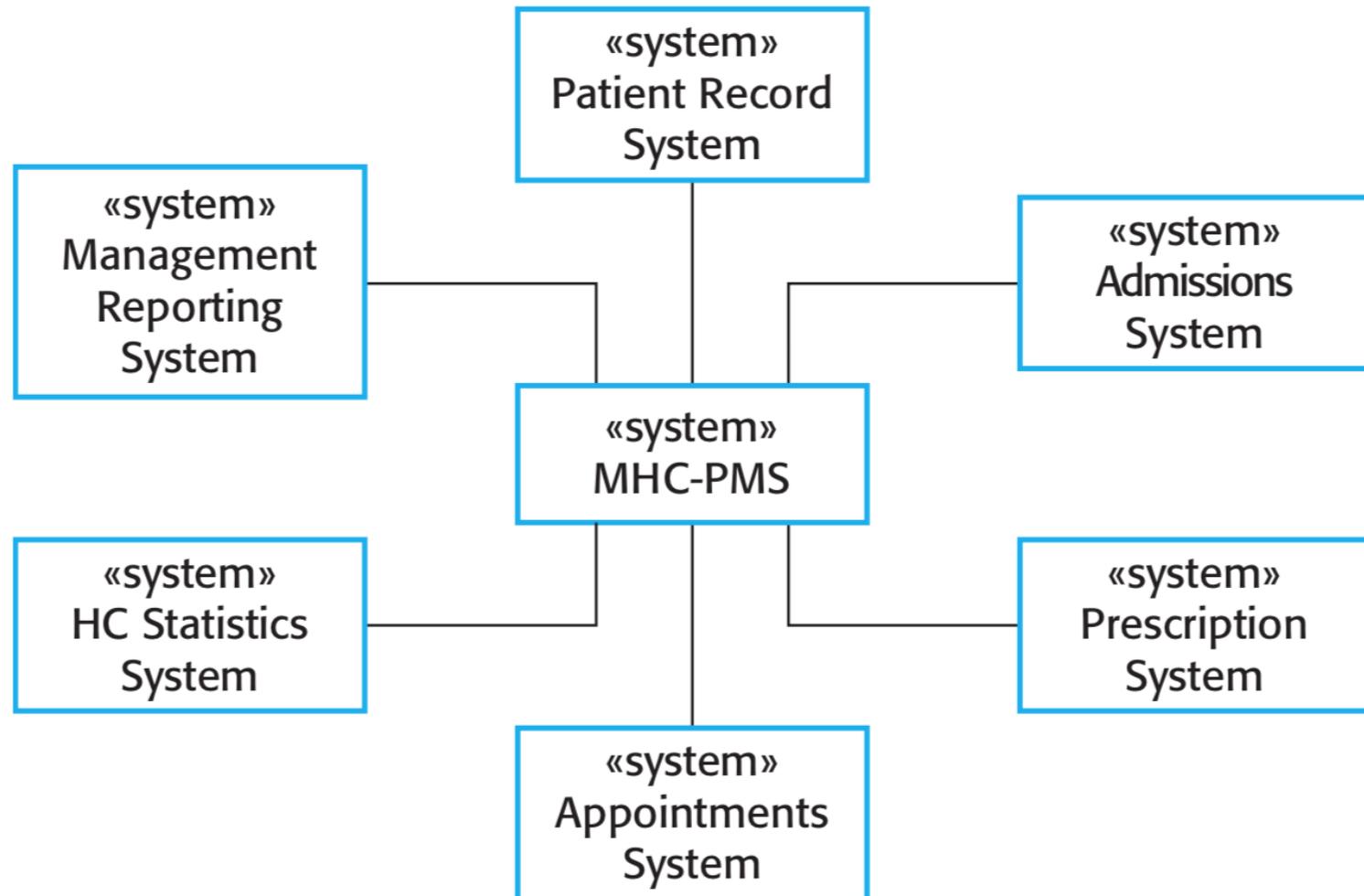
- Developing a simple architectural model is the first step.
 - Architectural models show the system and its relationship with other systems.

The context of an ATM system



Architecture models do not show the relationship between the other systems in the environment.

Context of a Patient Management System (PMS)



Process perspective



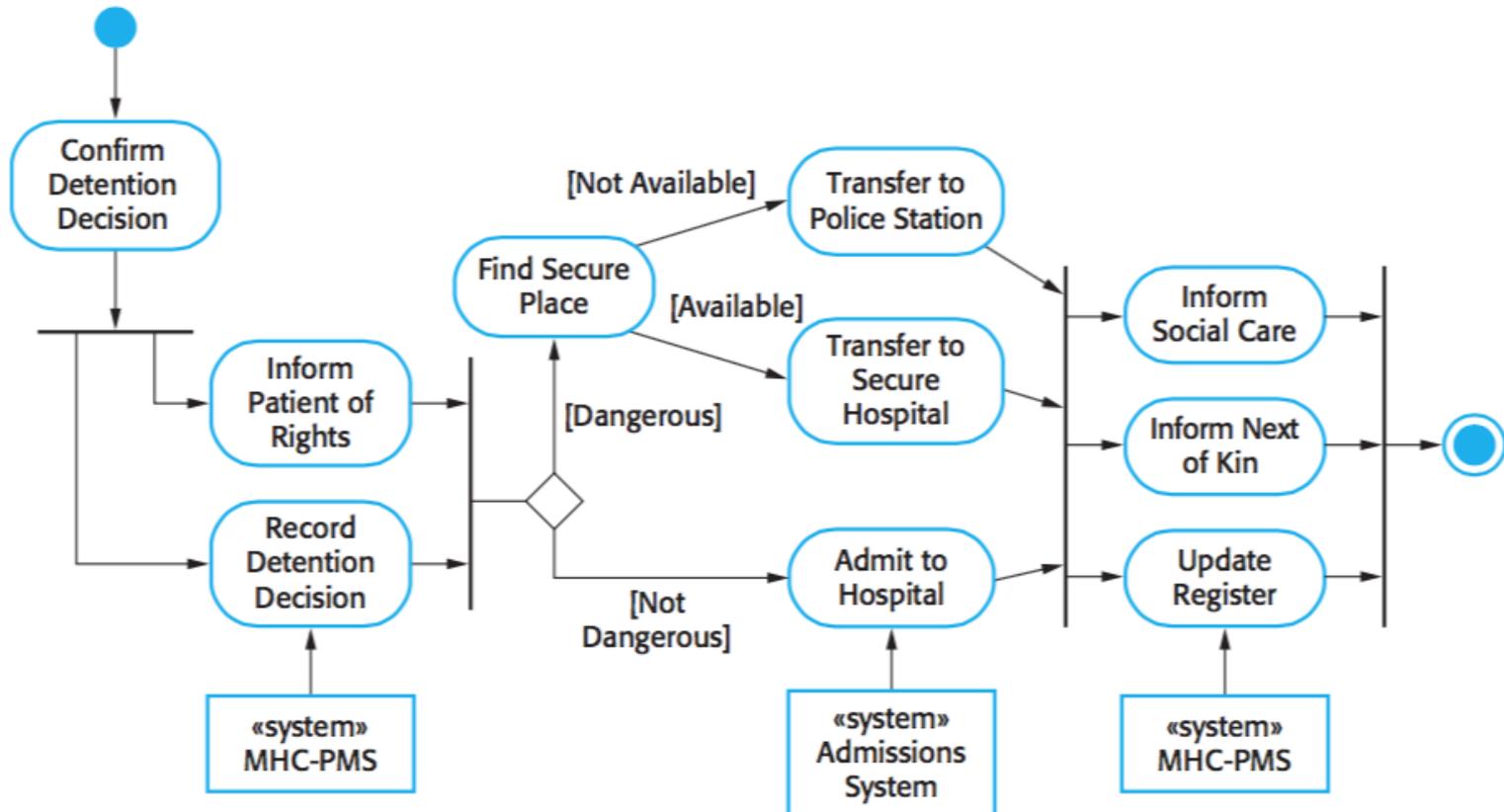
- Context models simply show the other systems in the environment, not how the system being developed is used in that environment.
- Process models reveal how the system being developed is used in broader business processes.
- **UML activity diagrams** may be used to define **business process models**.

Process models



- Process models show the overall processes that are supported by the system.
- **Activity Diagram** may be used to show the processes and the flow of information from one process to another.

Hospital system process – Activity Diagram



Identify the main symbols in the activity diagram.

Interaction models



- Modeling user interaction is important as it helps to identify user requirements.
 - Modeling system-to-system interaction highlights the communication problems that may arise.
 - Modeling component interaction helps us understand if a proposed system structure is likely to deliver the required system performance and dependability.
 - Use case diagrams and sequence diagrams may be used for interaction modelling.
- 

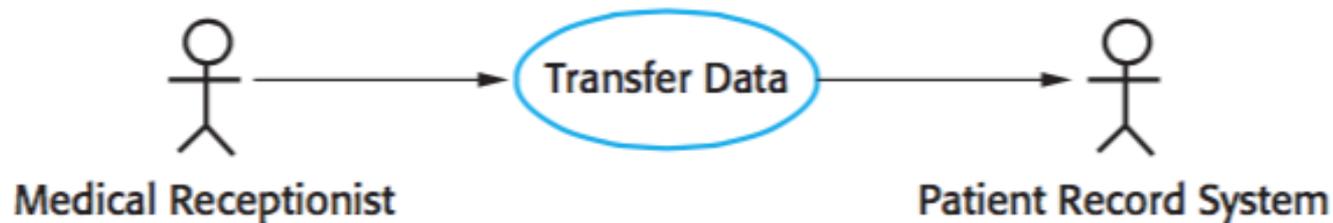
Use case modeling



- Use cases were developed originally to support requirements elicitation and now incorporated into the UML.
 - Each use case represents a discrete task that involves external interaction with a system.
 - Actors in a use case may be people (roles) or other systems.
 - Represented diagrammatically to provide an overview of the use case and in a more detailed textual form.
- 

Transfer-data use case

- A use case in the MHC-PMS



- Use case diagrams give a fairly simple overview of an interaction so you have to provide more detail. This detail can either be:
 - a simple textual description
 - a structured description in a table
 - or a sequence diagram

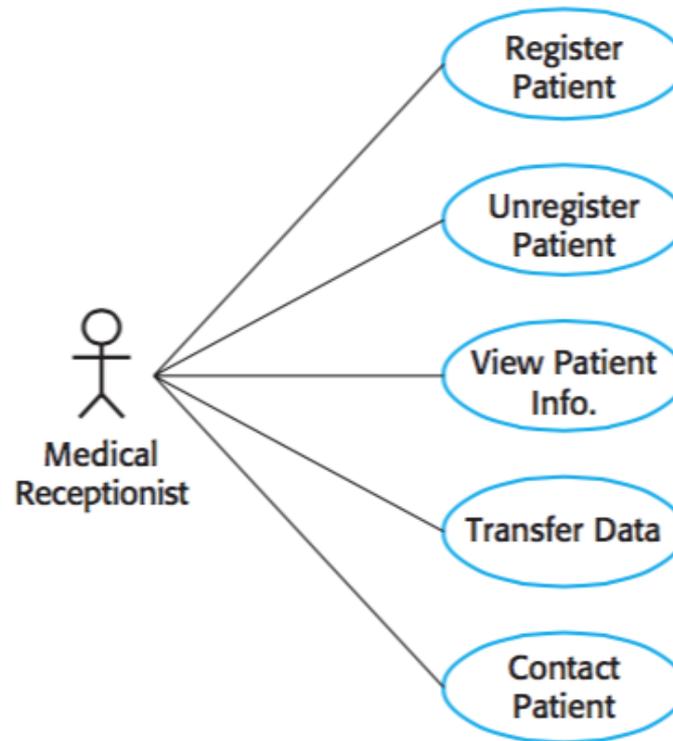
Tabular description of the 'Transfer data' use-case

MHC-PMS: Transfer data

Actors	Medical receptionist, patient records system (PRS)
Description	A receptionist may transfer data from the MHC-PMS to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, phone number, etc.) or a summary of the patient's diagnosis and treatment.
Data	Patient's personal information, treatment summary
Stimulus	User command issued by medical receptionist
Response	Confirmation that PRS has been updated
Comments	The receptionist must have appropriate security permissions to access the patient information and the PRS.

Use cases in the MHC-PMS

■ Medical Receptionist



Key Points (1/2)



- System modelling gives an easy way for analysts and designers to understand the functionality of the system.
 - Context models show the position of a system in its environment with other systems and processes.
 - Process models show the overall processes that are supported by the system.
- 

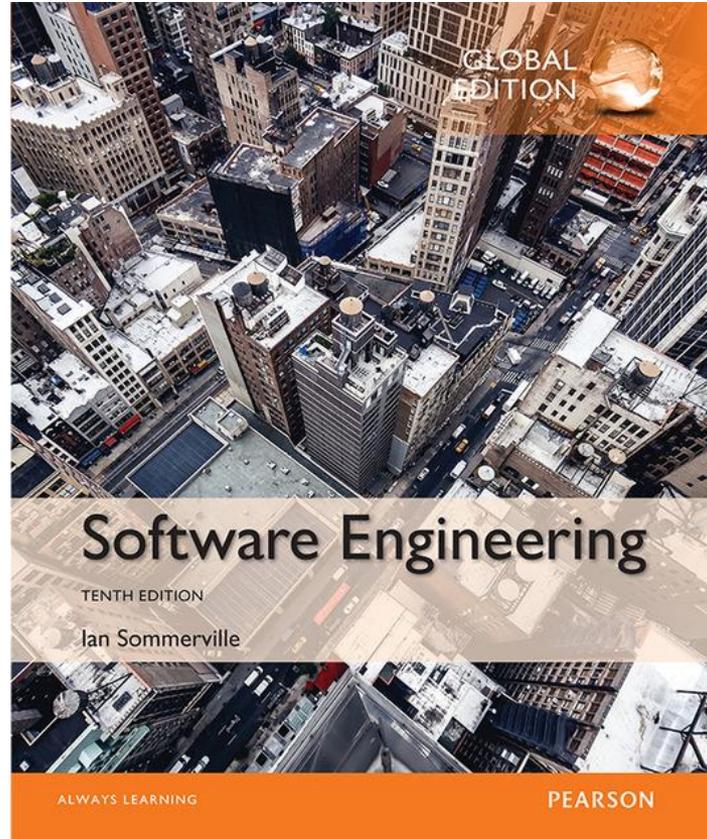
Key Points (2/2)



- Interactions models are used to describe the interactions between users and systems in the system being designed.
- Use cases describe interactions between a system and external actors;

Read

Chapter 5



References



- Ian Sommerville, “Software Engineering”, 10th Edition, Addison-Wesley, 2015.
 - Timothy C. Lethbridge and Robert Laganière, “Object-Oriented Software Engineering: Practical Software Development using UML and Java”, 2nd Edition, McGraw Hill, 2001.
 - R. S. Pressman, Software Engineering: A Practitioner’s Approach, 10th Edition, McGraw-Hill, 2005.
- 

Context diagram of a food-ordering system that shows high level details of interactions with the environment

