Design of Interactive Systems (DIS)

Lecture 8: Techniques for designing interactive systems – Task Analysis

Dr. Kalpana Shankhwar, kalpana@iiitd.ac.in
Assistant Professor
Department of Human Centered Design,
IIIT Delhi

Part II Techniques for designing interactive systems

- **Chapter 7**: Understanding
- **Chapter 8**: Envisionment
- **Chapter 9**: Design
- **Chapter 10**: Evaluation
- **Chapter 11:** Task Analysis
- **Chapter 12**: Visual Interface Design
- **Chapter 12**: Multimodal Interface Design

Introduction

- **Task** is a central notion in **human-computer interaction**
- Undertaking a **task analysis** is a very useful technique for understanding **people** and how they carry out their **work**
- In human centered design, it is necessary to **look at the tasks** that people do or they will have to do because of some redesigned system.

Aims of task analysis

- Understand the difference between **goals, tasks** and **actions**
- Undertake a **hierarchical task analysis**
- Undertake a procedural **cognitive task analysis**
- Understand the importance of considering a **structural view of a domain.**

Goals, tasks and actions

- Some of the ‘task analysis’ techniques involve interviewing, observation, development of scenarios, etc.
- The concept of task, how to undertake task analysis and what benefit designers might get from such analyses is the main focus in this chapter.

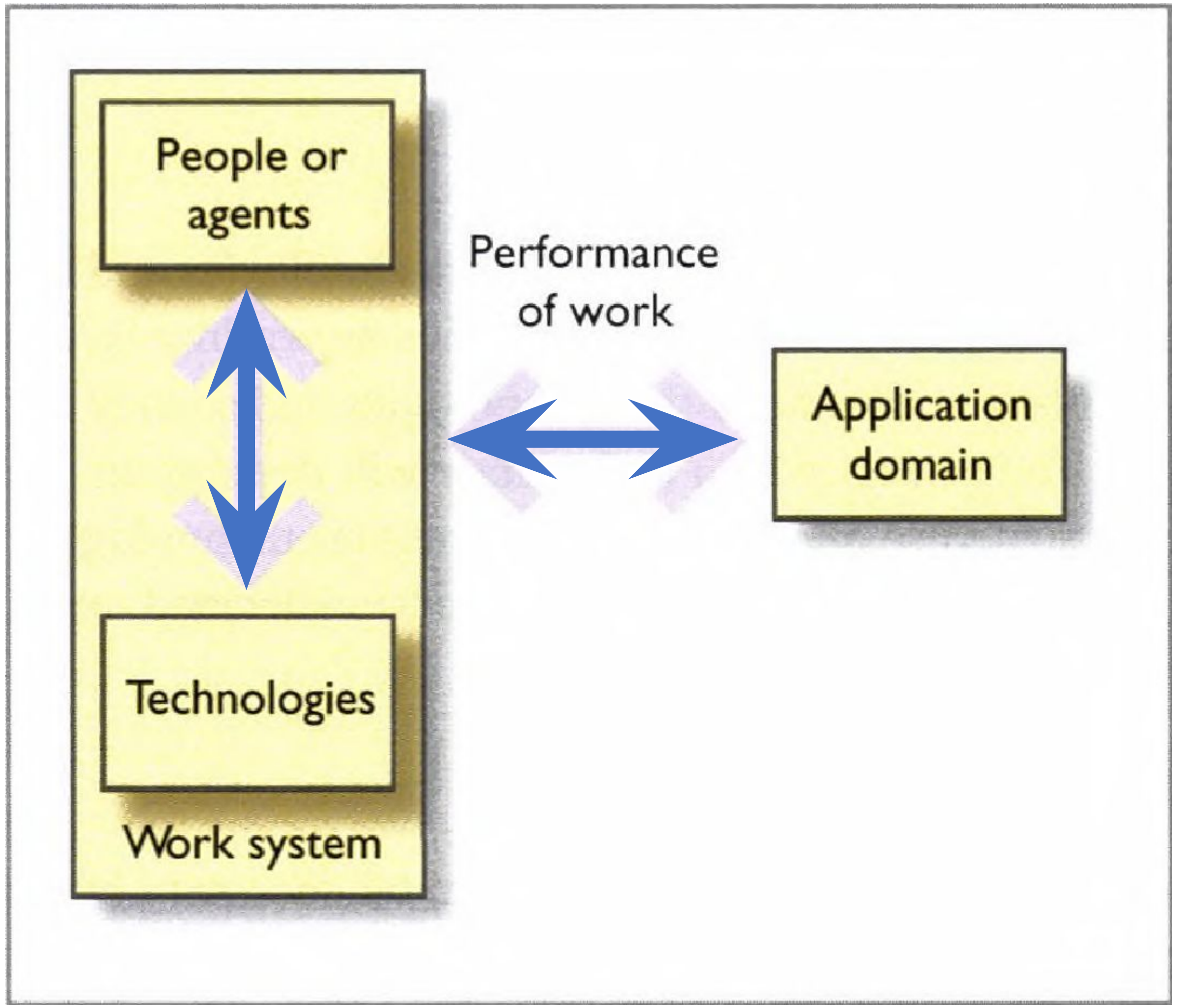
“A task is a goal together with some ordered set of actions”

Goals, tasks and actions

- The concept of task derives from a view of **people**, or other **agents**, interacting with **technologies** trying to achieve some **change** in an **application domain**.
- Taken together, the people and technology constitute what is sometimes called a **'work system'**, which is separate from the '**application domain**'.
- Application domain (or simply 'domain') is an **abstraction** of the real world, i.e. some abstract representation (such as a database, a website or an iPhone app).

Goals, tasks and actions

- Task analysis is concerned with some aspects of the **performance** of a **work system** with respect to a **domain**.
- This performance may be the amount of **effort to learn** a system, to reach a **certain level of competence** with a system, the **time taken** to perform certain tasks, and so on.



Goals, tasks and actions

- A **work system** in HCI consists of **one or more human** and **computer components** and usually many other sorts of thing as well.
- **Tasks** are the means by which the **work system** changes the **application domain**.
- **Goals** are desired future states of the **application domain** that the work system should achieve by the tasks it carries out.
- The work system's performance is deemed **satisfactory** as long as it continues to **achieve its goals** in the application domain.

Goals

- A **goal** is a **state of** the application domain that a work system wishes to achieve
- Artificial entities such as **technologies or agents** can have goals.
- It is not just people who have goals; the work system as a whole may have goals.
- For this reason the term **‘agent’** is often used to encompass both **people and software systems** that are actively and autonomously trying to achieve some **state** of the application domain.
- The term **‘technology’** is used to encompass physical devices, information artefacts, software systems and other methods and procedures.

Goals

- For example, an agent might have a goal such as to **write a letter**, **to record a programme on TV**, or to find the **strongest mobile phone signal**.
- The assumption is that the domain is in **one state now** - no letter written, the TV programme not recorded, the signal not confirmed as the strongest - and the agent has to undertake some activities to get **required state**.
- For **recording a TV programme**, for example, an agent could use the following technologies:
  - Ask a friend to record it.
  - Press ‘Rec’ on the PVR (personal video recorder).
  - Set the timer using a manual setting.
  - Set the timer using an on-screen TV guide.

Tasks and actions

- A **task** is a structured **set of activities** required or used, by an agent to achieve a goal using a particular technology.
- A task will often consist of **subtasks** where a subtask is a task at a more **detailed level of abstraction**
- The task is broken down into more and more **detailed levels of description** until it is defined in terms of **actions**.
- Actions are '**simple tasks**'.
- Task might include some structure such as doing things in a **particular sequence**, making decisions from the alternative things (**selection**) and doing things several times (**iteration**), an action does not.
- This structure is often called a **plan** or method.

Tasks and actions

- Task analysis methods can be divided into two broad categories:
  1. Those concerned with the **logic of the task** - the sequence of steps that need to be undertaken by a work system to achieve a goal
  2. Those concerned with **cognitive aspects**.
- **Cognitive task** analysis is concerned with understanding what cognitive processes the work system will have to undertake in order to achieve a goal.

Task analysis and systems design

- Balbo *et al.* (2004) emphasize the expressive **power of different methods** in their taxonomy of task analysis techniques.
- For example, they focus on whether a technique captures **optionality** (is a task mandatory or optional in pursuing a goal?), **parallelism** (can tasks be performed in parallel?) or **non-standard actions** such as error handling or automatic feedback

Task analysis and systems design

They also classify methods along the following axes:

- ***The goal of using the notation.*** It is the stage in the development life cycle; is it best for understanding, design, envisionment or evaluation?
- ***Its usability for communication***. Some **task analysis techniques** can be very hard to **read and understand**, particularly those that are based on a grammar rather than graphical notation.
- ***Its usability for modelling tasks.*** Task analysis methods have to **fit** into the software development process and be **used** and **understood** by software engineers. Since long time software engineers do not have access to a good task analysis technique.
- ***The adaptability of a task analysis technique to new types of system, new aims or new requirements.*** To what extent is the technique extensible to other purposes? (e.g. a task analysis technique aimed specifically at website design may not be very adaptable).

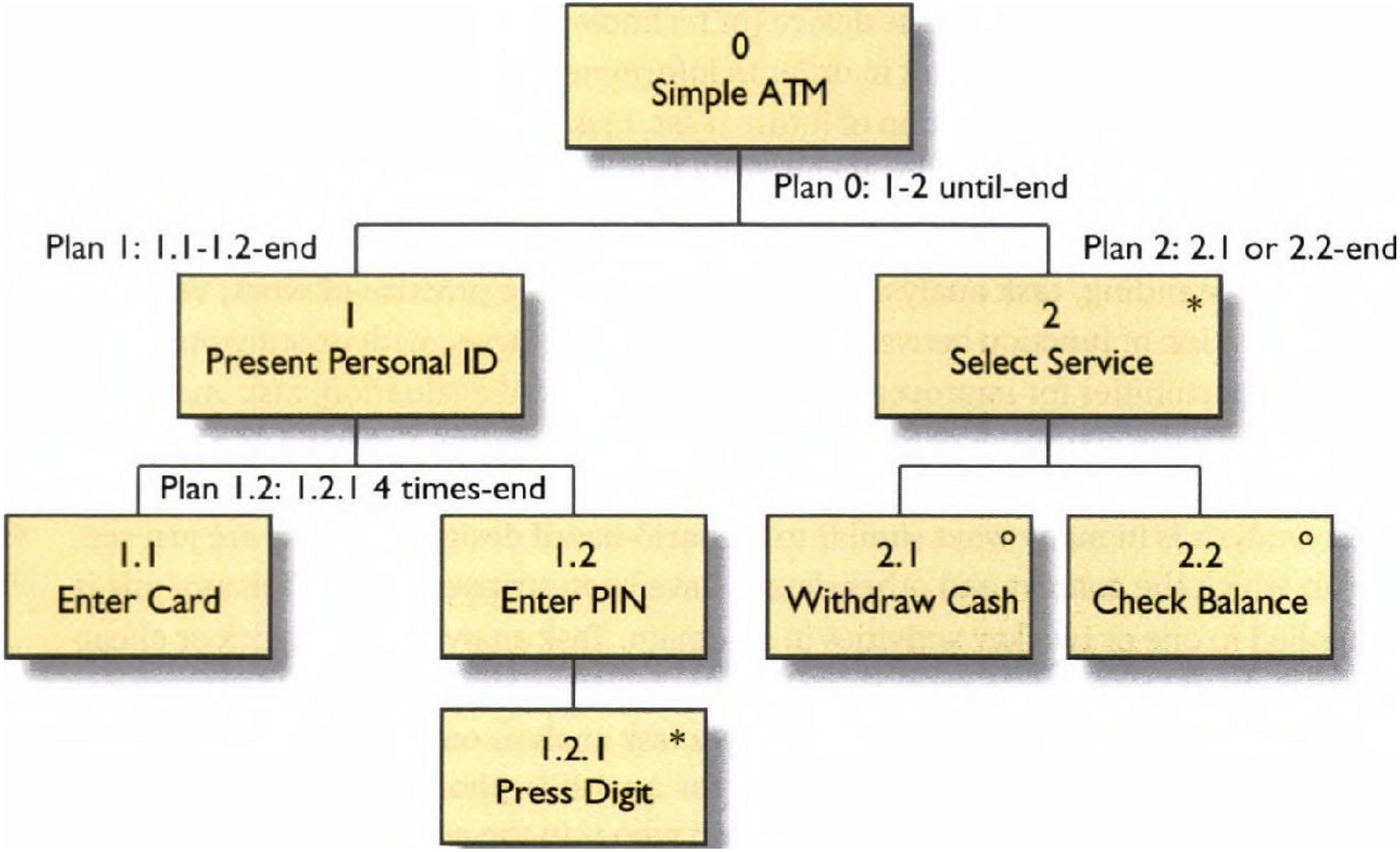
Hierarchical task analysis

- Hierarchical task analysis (HTA) is a **graphical representation** of a task structure based on a structure chart notation.
- Structure charts represent a **sequence of tasks**, **subtasks** and **actions** as a hierarchy and include **notational conventions** to show whether an action can be repeated a number of times (**iteration**) and the execution of alternative actions (**selection**)
- Sequence is usually shown by ordering the tasks, subtasks and actions from left to right.
- Annotations can be included to indicate ***plans.***

Hierarchical task analysis

- For example, making a call using a mobile phone has two main routes through the hierarchy of tasks and subtasks.
    1. If the person’s number is in the phone’s address book then the caller has to find the number and press ‘call’.
    2. If it is not, the caller has to type the number in and press ‘call’.
- HTA uses a structured diagram representation, showing the various **tasks** and **actions** in **boxes** and using **levels** to show the **hierarchy**.
- **Asterisk** in the box to show that an action may be repeated a number of times (**iteration**) and a small **‘o’** to show **optionality**. The

Hierarchical task analysis



Hierarchical task analysis

- HTA appears in many different methods for interactive systems design.
- For example, Stanton (2003) uses it as part of his method for error identification.
- At the **action level** (bottom level of an HTA), people might make a slip such as **human error** and action slips pressing the wrong button. What happens if they do this? At the task and subtask levels the analyst can consider what type of task it is and hence what types of error might occur.

GOMS: a cognitive model of procedural knowledge

- GOMS focuses on the cognitive processes required to achieve a goal using a particular device.
- The aim is to describe tasks in terms of the following:
  1. ***Goals.*** What are people trying to do using some system (e.g. make a call using a cell phone.)
  2. ***Operators***. These are the actions that the system allows people to perform, such as clicking on menus, scrolling through lists, pressing buttons and so on.
  3. ***Methods.*** These are sequences of subtasks and operators. Such as ‘select name from address book’ or ‘enter phone number’.
  4. ***Selection rules.*** These are the rules that people use to choose between methods of achieving the same subtask. For example, to select a name from an address book a person could scroll through the names or type in the first letter and jump to a part of the address book.

GOMS: a cognitive model of procedural knowledge

- GOMS is not a suitable analytical method where people are problem-solving.
- Also it is mainly applicable to systems being used by a single person where it can give accurate estimates of performance and help designers think about different designs.

GOMS

GOMS goal hierarchy	Observed behavior
goal: handle-calls	
. **goal: handle-call**	
. . **goal: initiate-call**	
. . . **goal: receive-information**	
. . . . listen-for-beep	Workstation: Beep
. . . . read-screen(2)	Workstation: Displays source information
. . . **goal: request-information**	
. . . . greet-customer	TAO: 'New England Telephone, may I help you?'
. . **goal: enter-who-pays**	
. . . **goal: receive-information**	
. . . . listen-to-customer	Customer: Operator, bill this to 412-555-1212-1234
. . . **goal: enter-information**	
. . . . enter-command	TAO: hit F1 key
. . . . enter-calling-card-number	TAO: hit 14 numeric keys
. . **goal: enter-billing-rate**	
. . . **goal: receive-information**	
. . . . read-screen(1)	Workstation: previously displayed source information
. . . **goal: enter-information**	
. . . . enter-command	TAO: hit F2 key
. . **goal: complete-call**	
. . . **goal: request-information**	
. . . . enter-command	TAO: hit F3 key
. . . **goal: receive-information**	
. . . . read-screen(3)	Workstation: displays credit-card authorization
. . . **goal: release-workstation**	
. . . . thank-customer	TAO: 'Thank-you'
. . . . enter-command	TAO: hit F4 key

Structural knowledge

- Task analysis is about **procedures**. But before a person sets about some procedure, they need to know **what types of things** can be accomplished in a domain.
- For example, if I am using a drawing package, I need to know that there is a facility for changing the thickness of a line, say, before I set about working out how to do it. I need some conception of what is possible, or what is likely.
- Instead of focusing on the steps that people have to go through to achieve a goal, it can be focused upon what structural knowledge people have.

Structural knowledge

- Payne (2012) proposes that people need to keep in mind two mental spaces and the relationships between them.
- A **goal space** describes the state of the domain that the person is seeking to achieve.
- The **device space** describes how the technology represents the goal space.

Structural knowledge

- Payne (2012) also discusses the concept of a ‘**mental map**’ and discusses how an analysis of mental models can be useful in highlighting **differences** between **people’s views of a system**.
- Green and Benyon (1996) describe a method called **ERMIA** (entity-relationship modelling of information artefacts)
- ERMIA models structural knowledge and so can be used to represent the concepts that people have in their minds.

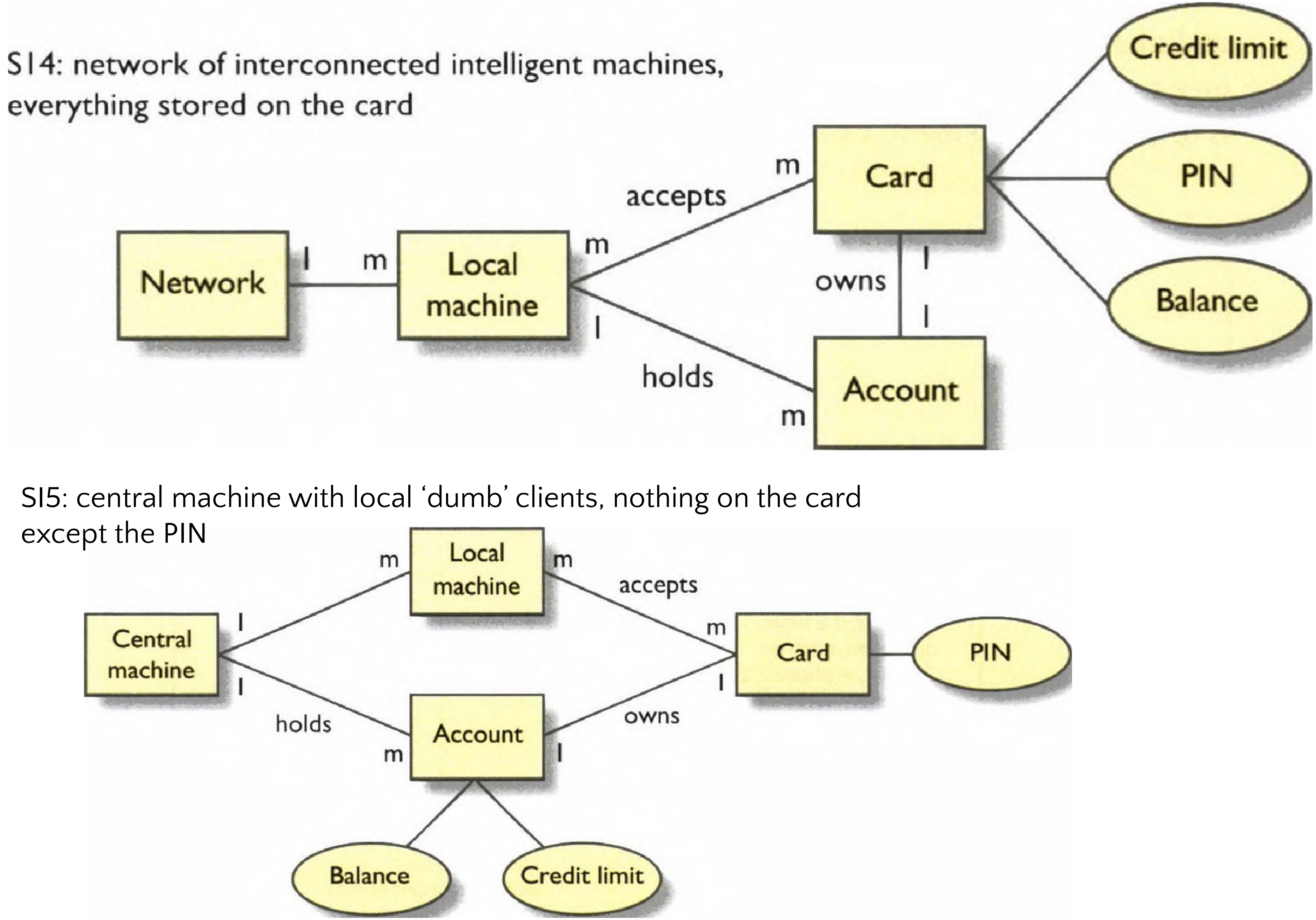


Figure 11.4 Comparison of two mental models of ATMs described by Payne (1991)

Structural knowledge

In Figure 11.5 we can see a typical menu interface. An important part of this type of analysis is that **it helps to expose differences** between the **designer's model, the system image** and **the 'user's' model**.

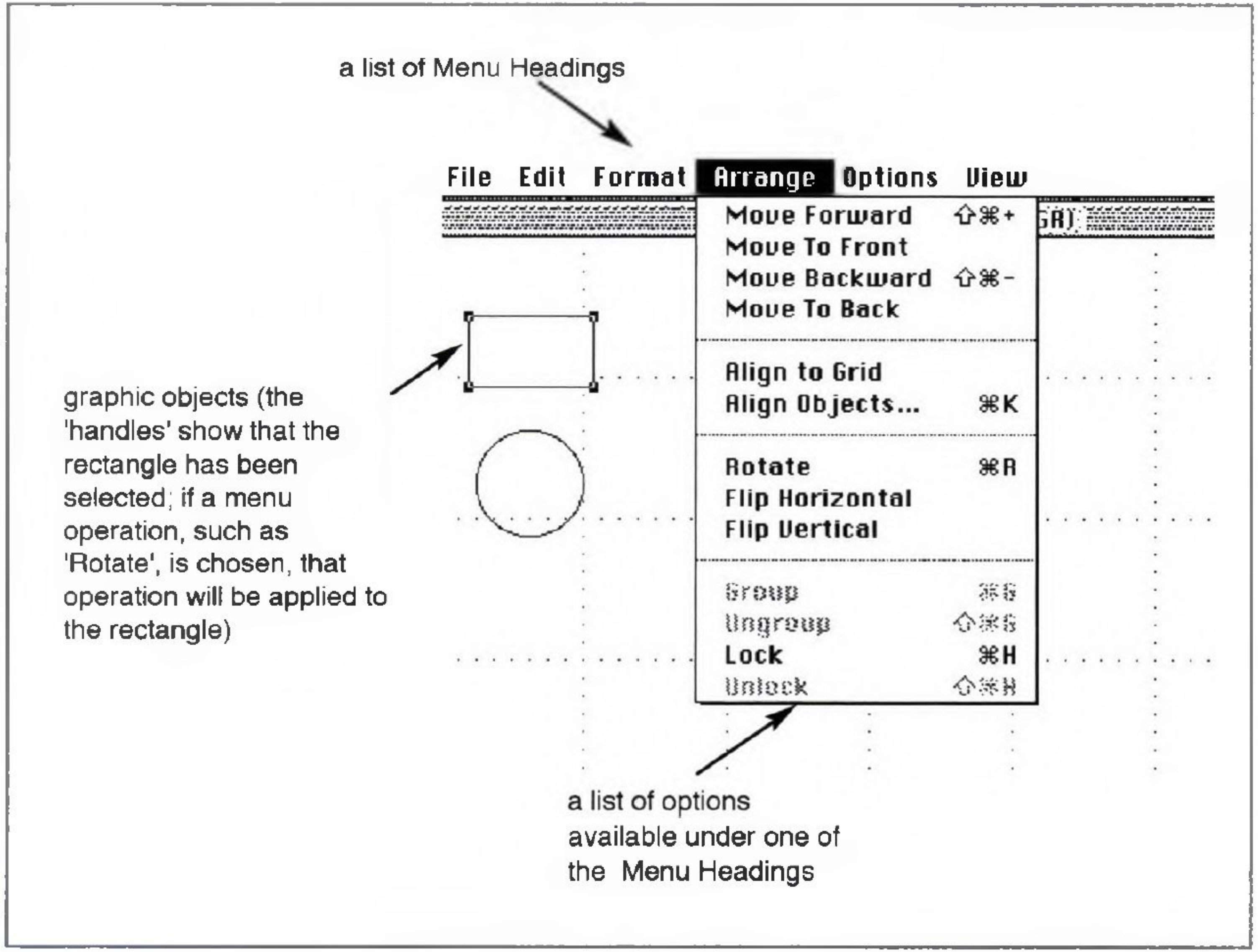


Figure 11.5 A simple drawing program, showing the document being created (a drawing, currently consisting of a rectangle and a circle) and the interface to the application

Structural knowledge

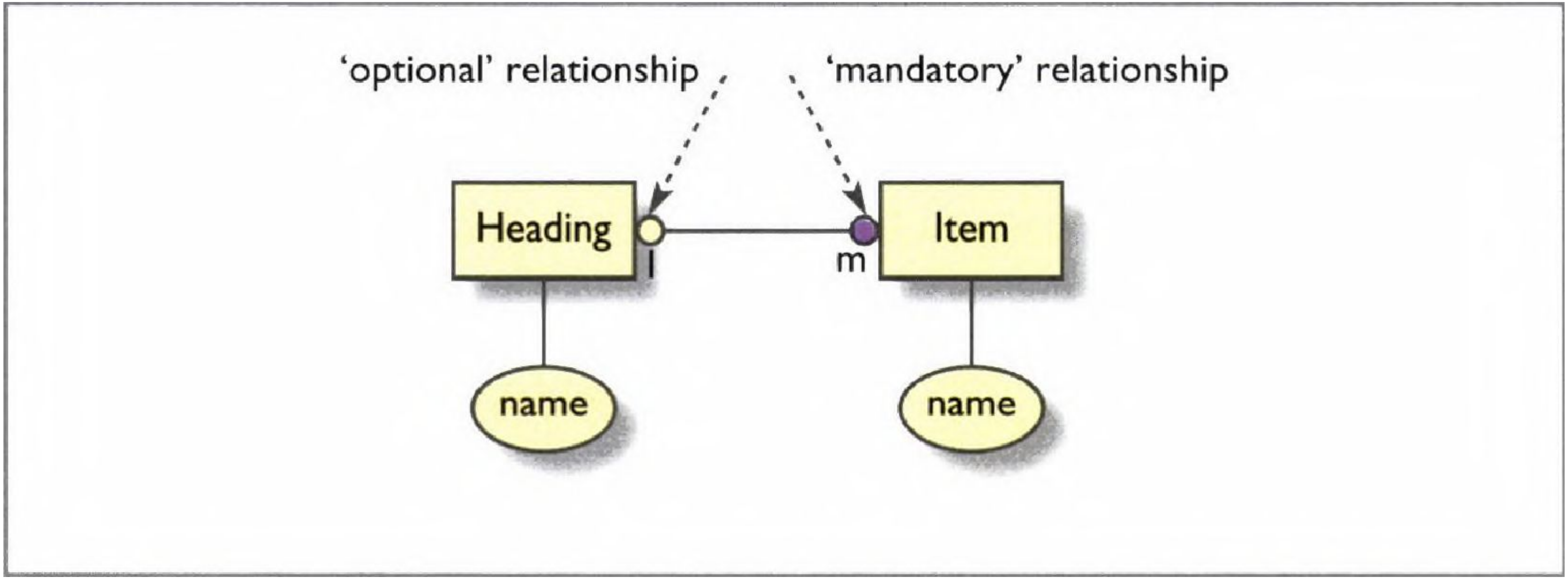


Figure 11.6 ERMIA structure of a menu system containing headers and items. The relationship is between heading and item is I:m (that is, each heading can refer to many items, but an item can be associated with only one heading). For items, the relationship is mandatory (that is, every item must have a heading), but a heading can exist with no associated items

Cognitive work analysis (CWA)

- CWA was originally formulated to help in the design of systems concerned with the **domain of process control**, where the emphasis is on controlling the physical system behind the human-computer interface
- CWA has been used in the analysis of complex real-time, mission-critical work environments, e.g. power plant control rooms, aircraft cockpits and so on.
- One principle underlying CWA is that when designing computer systems or any other ‘cognitive artefact’ we are developing a complete **work system**, which means that the system includes **people and artefacts**.

Cognitive work analysis (CWA)

- Another key principle of CWA is that it takes an ecological approach to design. Taking an ecological approach recognizes that people ‘**pick up**’ information directly from the objects in the world and their interaction with them, rather than having to **consciously process** some symbolic representation.
- CWA techniques include such things as task analysis (including **sequencing and frequency**) and workload analysis (**flow of work, identification of bottlenecks**).
- A work domain analysis has five further levels of abstraction, describing
  - The functional purpose of the system
  - The priorities or values of the system
  - The functions to be carried out by the system
  - The physical functionality of the system
  - The physical objects and devices.

Summary and key points

- Task analysis fits very closely with requirements generation and evaluation methods.
- Task analysis focuses on goals, tasks and actions.
- Task analysis is concerned with the logic, cognition or purpose of tasks.
- A structural analysis of a domain and worksystem looks at the components of a system and how the components are related to one another.

Class activities

1. Undertake an HTA-style analysis for purchasing a T-shirt from Myntra App using your phone.
2. Write a GOMS-type description for the simple ATM.