JOHANNES KEPLER
UNIVERSITY LINZ

Use Cases and Scenarios

This presentation is partly based on a tutorial by Prof. Neil Maiden from the City University London, Centre HCI Design

Where do Use Cases Come From?

Originated with Ivar Jacobson

- OOSE method described in: *Object-Oriented Software Engineering: A Use Case Driven Approach*, by Ivar Jacobson, Magnus Christerson, Patrik Jonsson & Gunnar Overgaard, Addison-Wesley,1992
- Pragmatic method based on telecoms experience

Came into the UML and the RUP process in 1998

- Requirements coverage undertaken with use cases
- Emergent weaknesses in RUP
  - Use cases become the requirements - lack of expressiveness
  - Minimal prescriptive process guidance - more focus on UML and representations instead

Use Case vs Usage Case ;-)

Please respond to ivar@rational.com

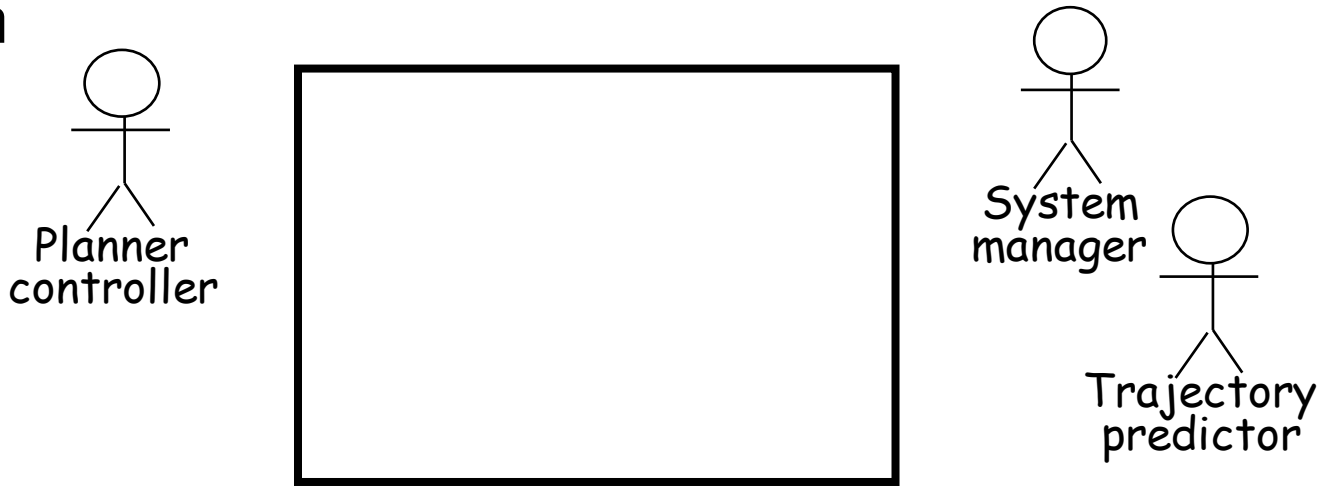
To: Bruce Anderson/UK/IBM
cc:
Subject: RE: "Use Case"

In Swedish i called 'it'
anvandningsfall (the second a has two dots above it) which means usage case.
I translated it to use case!
Ivar

> -----Original Message-----
> From: bruce_anderson@uk.ibm.com [mailto:bruce_anderson@uk.ibm.com]
> Sent: den 30 augusti 1999 14:30
> To: ijacobson@Rational.Com
> Subject: "Use Case"
>
>
>
>
> Ivar - what is the Swedish phrase of which "Use Case" is a translation?
> How would you now translate it? I remember we discussed this before the
> panel at OOPSLA this year.
> thanks,
> Bruce
>
> Bruce Anderson, Senior Consultant, IBM EMEA Object Technology Practice
> +44.181.818.4305 (switches to mobile then voicemail); home office/fax
> +44.1206.825487; mobile +44.410 981 329

Use Case Modelling

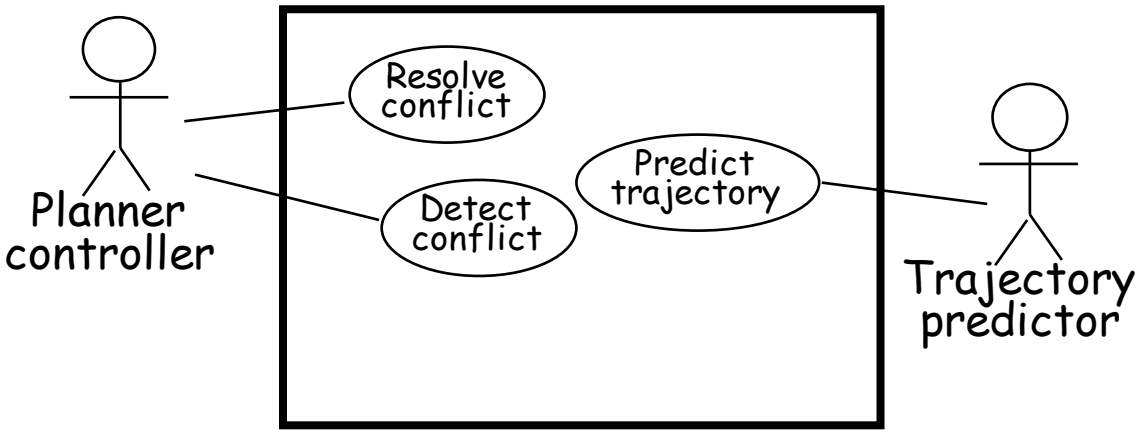
- An **actor** is a type of anything external to the software system, human or machine
  - A single person or system can instantiate several different actors by playing different roles
- Actors come in two kinds:
  - Primary actors, using system in daily activities
  - Secondary actors, enabling primary actors to use system



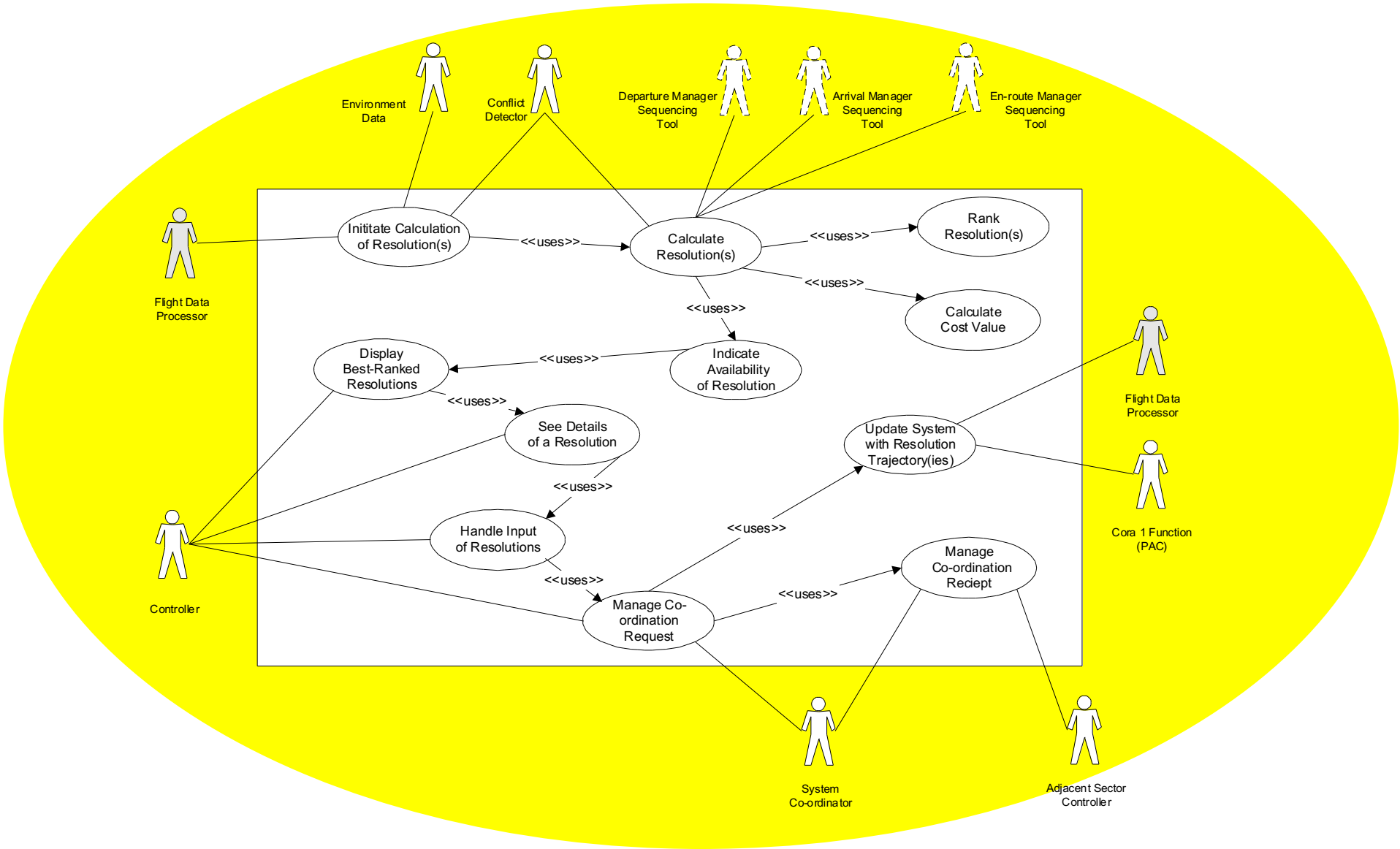
Use Case and Use Case Models

- A use case is a specification of **system behaviour** to undertake a major **system goal or service**, or to achieve a requirement

- The use case model characterises the behaviour of the whole system. It can be expressed as a use case diagram.



Example: CORA-2 Use Case Model



Audience Survey

Your experience with Use Case Analysis:

- Created some use case diagrams
- Defined the normal course of use cases
- Defined variations and alternatives courses of use cases
- Performed a scenario walkthrough
- Used use case analysis in real-world projects

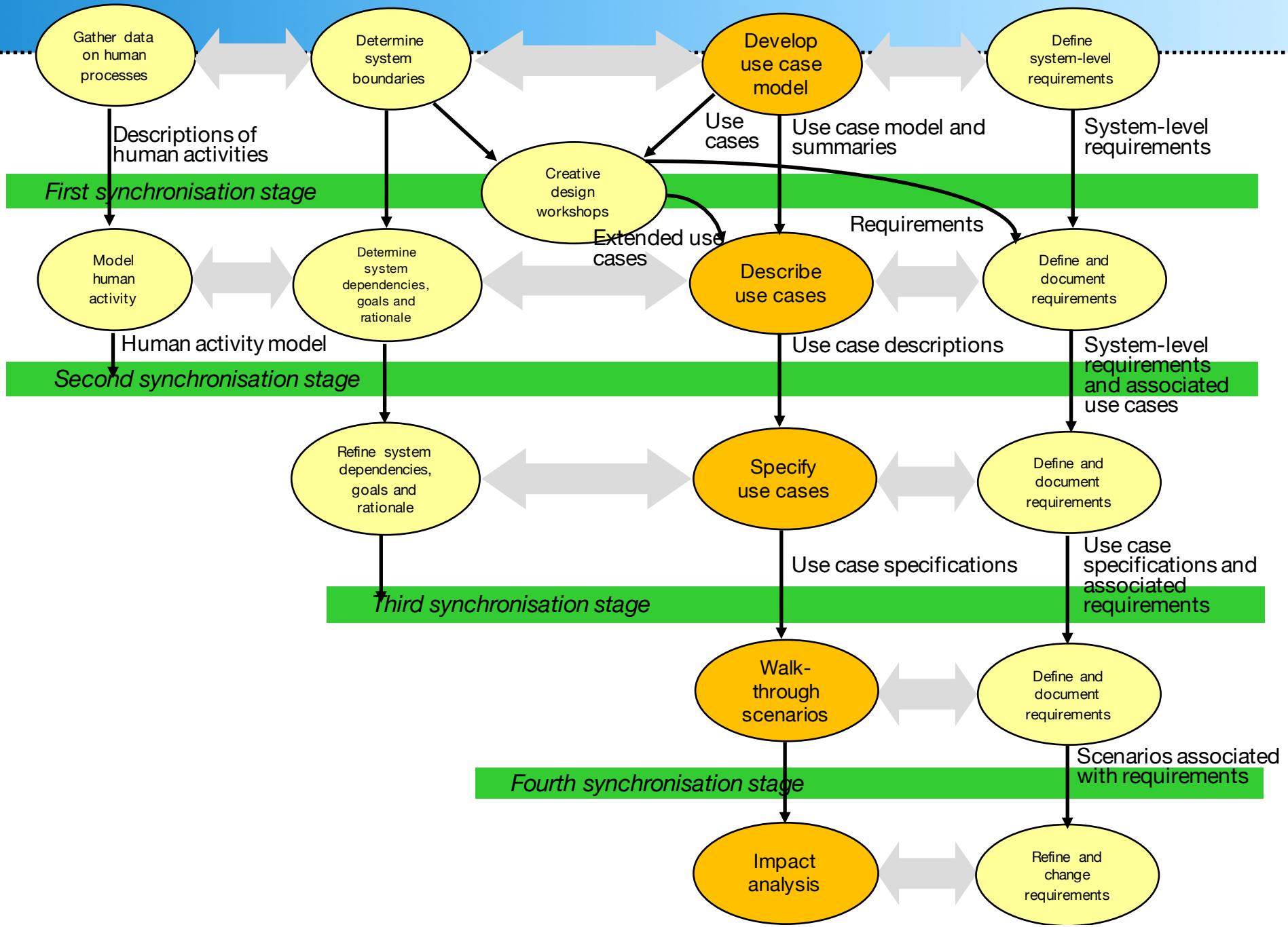
Use Case Modelling: Classroom Exercise

- Create a use case model for the example system
- Determine actors and the possible software system boundaries
  (you will need to make assumptions)
- Determine possible use cases and draw a diagram model
- Develop the model further with extends and uses relationships between use cases

RESCUE: Requirements Engineering with Scenarios for a User-centred Environment

Jones, S., and Maiden, N., RESCUE: An Integrated Method for Specifying Requirements for Complex Sociotechnical Systems, IGI Global, 2005

The Use Cases Stream in RESCUE



RESCUE Use Case Template

Guides use case description

- Through a **comprehensive template** that is both complete and simple-to-use
- Complete **one** template **for each use case** in the use case diagram, thus ensuring simple traceability
- **Living document** that is evolved as the device to structure all requirements in the requirements document

Similar to best-practice elsewhere

- E.g., Alistair Cockburn's best practices in use cases

Common Use Case Attributes

- Use case **name**, number and requirement(s)
- The **normal course** to achieve the requirement(s)
- All **variations** and **alternative** courses
- Models (activity and sequence diagrams) showing the concurrent actions and agent message passing
- Class diagrams linked to the use case
- Related background information, references and comments
- Change history for the use case
- Open and unresolved issues for the use case
- Glossary of terms

RESCUE Use Case Template

	Name of Use Case
Use Case ID	Unique ID for Use Case
Author	Name of author
Date	Date Use Case was written
Source	Source of Use Case
Actors	*Actors involved in Use Case (from the Use Case Model)*
Problem statement (now)	*Description of current problem*
Precis	*Informal scenario description*
Requirements	*Requirement that the use case **DECOMPOSES***
Constraints	*Requirements that impose **CONSTRAINS** on the required behaviour in the use case*
Added Value	*Benefit of Use Case above and beyond the original scenario from the original system*
Justification	*Why is the Use Case needed?*
Triggering event	*Event or events that can trigger the Use Case*
Preconditions	*Necessary conditions for the Use Case to occur*
Assumptions	*Explicit statement of any assumptions made in writing the Use Case*
Successful end states	*Successful outcome(s) of the Use Case*
Unsuccessful end states	*Unsuccessful outcome(s) of the Use Case*
Normal Course	*1. Event 1* *[EN] System requirement **ENABLES** Event 1* *[CO]System requirement **CONSTRAINS** Event 1*
	2. Event 2 *[EN] System requirements **ENABLES** to Event 2* *[CO] System requirement **CONSTRAINS** Event 2*
	…
	…
Variations	*1. If [condition] then [variation statement] (related to Event 1)*
	…
Alternatives	*1. If [condition] then [alternative course statement] (related to Event 1)*
	…

Writing the Use Case Precis

Informal use case descriptions

- Describe each use case using **simple attributes** (ID, source, actors) and **precis** (one paragraph) describing how actors will use the future system to achieve their goals
- Precis should be describe wider system vision

Benefits of these descriptions

- **Avoid over-commitment** by stakeholders
- Flexible representation, **simple to change**
- Focus on **envisioned behaviour** at expense of details
- Encourage people to work with and **manipulate** these representations before agreeing details

Example CORA-2 Use Case Precis

Manage Invalid Resolution(s)

Should a resolution become invalid due to e.g. the expiration of its ' minimum time for implementation' G, CORA 2 deletes it from the stored resolutions and re-orders the ranking of the remaining resolutions. Should a resolution become invalid during co-ordination, CORA 2 requests the Systems Co-ordinator to discontinue the co-ordination process.

Comments on the precis

- The precis is largely design-free, although there are a number of tacit assumptions made
- Too few use case “steps”, suggesting perhaps that this is not a complete use case

Example CORA-2 Use Case Precis

- Display Best-ranked Resolutions

  On request, CORA 2 displays a set of ranked resolutions. One of each type G is displayed (if available), i.e. Climb, Descend, Left Turn, Right Turn and Adjust Speed. In parallel, CORA 2 displays the associated 'conflict information' G (including the related 'context aircraft' G), the cost-value on which the ranking is based and an indication of the 'optimum time to implement' G the resolutions. This allows the Controller to be aware of possible solutions for a conflict, and to understand and evaluate them rapidly before making a decision, i.e. (i) chose one of the CORA 2 resolution proposals, (ii) modify it or (iii) create his/her own resolution.

- Comments on the precis
  - More description of use case “steps”
  - Too little vision for wider system behaviour?

Use Case Description: Classroom Exercise

- Purpose
  - To practice simple use case precis as a starting point for further, more structured use case description
- Task
  - For the ***example system***, write a use case precis for one of more of the use cases modelled in the previous exercise
  - Explore how simply these precis **can be changed** to incorporate new or changing requirements and design ideas

Writing and Structuring Normal Course Use Cases

Supermarket Checkout Normal Course

Normal course

1. The operator swipes the customer's club card.
2. The system reads customer information from club card.
3. REPEAT while more products to be purchased.
    3.1 The operator swipes the product using the barcode reader.
    3.2 The system displays the item name and price.
    3.3 The system records purchase details from the product barcode.
4. The operator presses the 'transaction total' button.
5. The system displays the total amount due for the purchase.
6. The customer pays the operator.
7. The operator returns change to the customer.
8. The operator presses the 'transaction paid' button.
9. The system records the entire transaction.
10. The system updates the customer's clubcard details with the purchase information.
11. The system prints a receipt with all purchase details on the receipt.

CREWS-SAVRE Style Guidelines

S1 Each action is undertaken by one agent

S2 Write normal course as discrete actions on new lines

S3 Avoid sentences with more than 2 clauses

S4 Use consistent agent, object and action names

S5 Avoid synonyms and homonyms

S6 Use present tense and active voice

S7 Avoid use of negations, adverbs and modal verbs

S8 Bring out and make explicit the sequences, conditions and iterations in use case descriptions

S9 Separate out the contextual information

S10 Make clear all calls to other use cases

S11 Use graphics, storyboards and video clips to support actions that cannot be expressed solely with text

Use Case Description Classroom Exercise

Purpose

- To practice style and content guidelines for writing use case normal course descriptions

Task

- For the *example system*, write a use case descriptions for one or more of the use cases for which you wrote a precis in the previous exercise
- Demonstrate the use of the style and content guidelines

Establishing the Start Event for a Use Case

Establishing the Use Case Start Event

- Critical for effective use case authoring
  - Establishes the real external driver (business, problem or physical world) for the system
  - Establishes the scope of the use case, the agents involved, and hence the requirements discovered and expressed
  - Provokes more creative thinking about use cases (is this the right way to do something?)
- Some consequences
  - You might change or introduce new actors in the use case model to undertake the real start event

Starting Events

Finding the start event

- Starting events happen *out there* in the environment or are undertaken by some adjacent actor
- One or more actors must receive **notification of the event** in the form of some **communication**
- There might be further events which lead to **communication across system boundaries** between adjacent and other actors

Examples

- CORA-1 detects conflict between two or more aircraft in CORA-2 system
- Radar systems provide data that for conflict detection
- CORA-2 will involve communication will pilots

Use Case Starting Point Classroom Exercise

Purpose

- To discover the most correct starting event for use cases

Task

- For *the example system*, reconsider each of your use cases and establish different possible start events
- Consider the pros and cons of each possible start event, in terms of system boundaries, requirements on different agents, etc

Structuring and Writing Use Case Variations and Alternative Courses

Variations and Alternative Courses

- What are **variations**?
  - Different normal course actions, expressed as sequences, that also lead to the success end state for the use case
  - Needs domain expertise and judgement to distinguish normal and variation actions
  - Represent in separate section to normal course actions
- What are **alternative courses**?
  - Actions, and their pre-conditions, that lead to an unsuccessful end state for the use case
  - Essential to ensure requirements completeness

How to Discover and Write Variations

- Discover variations using simple questions
  - Are there other means of achieving the use case goal?
  - Can these action(s) be undertaken by another agent?
  - Can these action(s) be undertaken in a radically-different sequence?
- Write each variation as a mini-normal course
  - Action number: Normal course action(s) which the variation defines different behaviour to
  - Precondition: WHEN-statement defining the precondition for the variation
  - Action statements: Sequence of numbered action statements that describe how the actors shall behave when the precondition is true

Supermarket Checkout Variations Example

Consider portion of the use case

6. The customer pays the operator.
7. The operator returns change to the customer.
8. The operator then presses the 'transaction paid' button.

Some obvious variations on payment methods

6-8. IF the customer pays with credit card

6. The customer gives the credit card to the operator.
7. The operator swipes the credit card using the barcode reader.
8. The system reads the credit card number from the card.
9. The system validates the credit card number.
10. The system prints the credit payment slip.
11. The operator takes the credit payment slip from the system.
12. The customer signs the credit payment slip.
13. The operator validates the customer signature.
14. The operator presses the 'transaction paid' button.

How to Write Alternative Courses

An alternative course follows a regular structure

- **Action number**: Normal course action(s) to which the alternative course pertains
- **Precondition**: IF-statement defining the precondition for the alternative course
- **Action statements**: Sequence of (unnumbered) action statements that describe how the actors shall behave when the precondition is true

Thus each alternative course is independent

- Independent of all other alternative courses
- Do not attempt flow charts and UML sequence and activity diagrams - ***it will become too complicated***
- Alternative courses ensure complete coverage and provide a simple partial rationale for requirements

Discovering Alternative Courses

- Events (*not happening, too frequent, too infrequent, wrong order*)
- Actions (*insufficient information, not completing*)
- Cognitive exceptions (*slips, mistakes, lack of knowledge/skill*)
- Other human exceptions (*age, size, gender, disabilities*)
- Machine exceptions (*power failures, breakdowns, blockages*)
- Human-machine exceptions (*misinterpret interface*)
- Machine-machine exceptions (*communication failure, scrambled messages*)
- Environmental exceptions (*light, heat, humidity, noise*)

Some Supermarket Checkout Alternative Courses

1. **IF the customer does not have a club card**, do not do action.
2. **IF the customer does not have a club card**, do not do action.

3.1. **IF the item does not have a bar code**. The operator places the item on the scale. The operator enters a 4-digit number which is a unique identifier for the fruit or vegetable. The operator presses the button. The system calculates the price.

3.1 **IF the bar code reader cannot read the product code**. The operator enters the code using the keyboard.

3.1. **IF one product is bought several times**. The operator swipes the bar code once. The operator enters the number of products to be purchased. The system multiplies the price by the number of products.

3.1. **IF there is a product code error**. The operator cancels the last bar code swiping operation. The system removes the last product from the current transaction.

3.1. **IF the product has a special price reduction**. The operator enters the special reduction price of the product.

3.3. **IF the item does not have a bar code**. The operator enters the bar code number of the product using the keyboard.

Use Case Description Exercise

- **Purpose**
  - To practice discovering and describing variations and alternative courses for use cases
- **Task**
  - For the ***example system***, write possible variations and alternative courses for the earlier use cases
  - Demonstrate again the use of relevant style and content guidelines

RESCUE Scenario Walkthrough Environment

Scenario Walkthrough Environment

- Prepare the walkthrough environment carefully
  - Provide relevant use case-driven requirements **documents** to participants
- Clear roles of the facilitators
  - ***Facilitator*** leads the walkthrough, communicating with the participants and encouraging their involvement
  - ***Scribe*** records all results from the walkthrough, including comments, new or changed requirements, and scenario changes
  - ***Participants*** provide stakeholder input to the walkthrough process, to ensure requirement completeness and correctness
  - These roles and the walkthrough rules are **communicated** to all participants beforehand

Planning Scenario Walkthrough

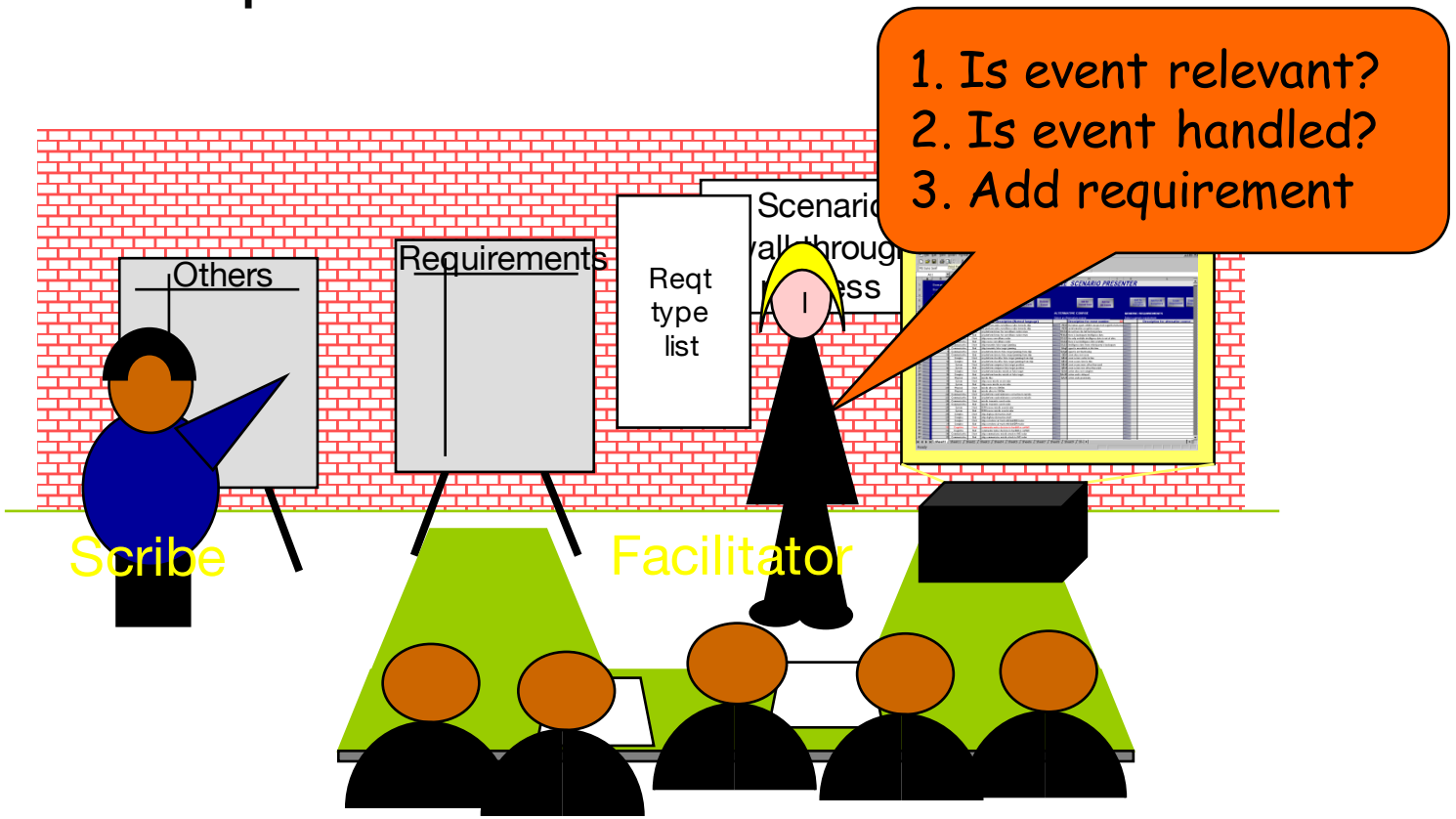
- Plan for half-day walkthrough
  - **15-minute** introduction, **150-minute** walkthrough session, **15-minute** wrap-up session
- Which participants to invite
  - One **facilitator** and one **scribe** are essential
  - Representatives of all **stakeholders/actors** who undertake actions in the scenario
  - Other key domain and technology **experts**
- Create the right environment
  - Walkthroughs should be **creative** and innovative, so provide a relaxing environment which does not inhibit involvement

Before Scenario Walkthrough

- All participants prepare for walkthrough
  - To familiarise themselves with the scenario
  - To edit the scenario so that it is agreed by all participants
- Each participant reviews the scenario individually
  - Recognise possible new **requirements**
  - Send **comments** on scenario to facilitator
  - Propose **candidate changes** to scenario to facilitator
- Facilitator changes scenario in light of comments
  - Consider all candidate comments and make agreed changes to the scenario
  - Publicise scenario and changes to it to all participants

Scenario Walkthrough Environment

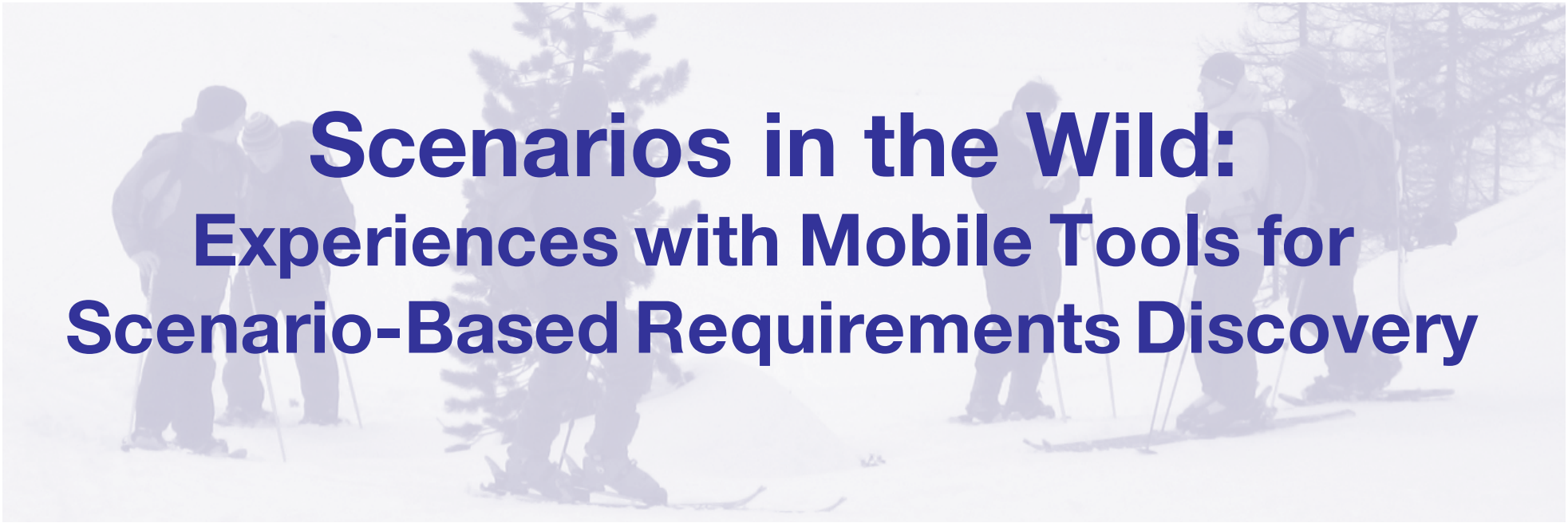
- Physical layout of room is important
  - Advise use a structure based on JAD/RAD workshops



Scenario Walkthrough Process

- **At start of** scenario walkthrough
  - Facilitator introduces scenario and originating use case
  - Facilitator summarises participant comments and earlier changes made to the scenario
  - Participants agree and make final changes to scenario
- **During** scenario walkthrough
  - Walk through normal course to discover requirements for each event in turn
  - Walk through each alternative course for each normal course event in turn, to discover new requirements
- **At end** of scenario walkthrough
  - Consider all requirements discovered during the walkthrough, to add, edit or delete requirements

JOHANNES KEPLER
UNIVERSITY LINZ



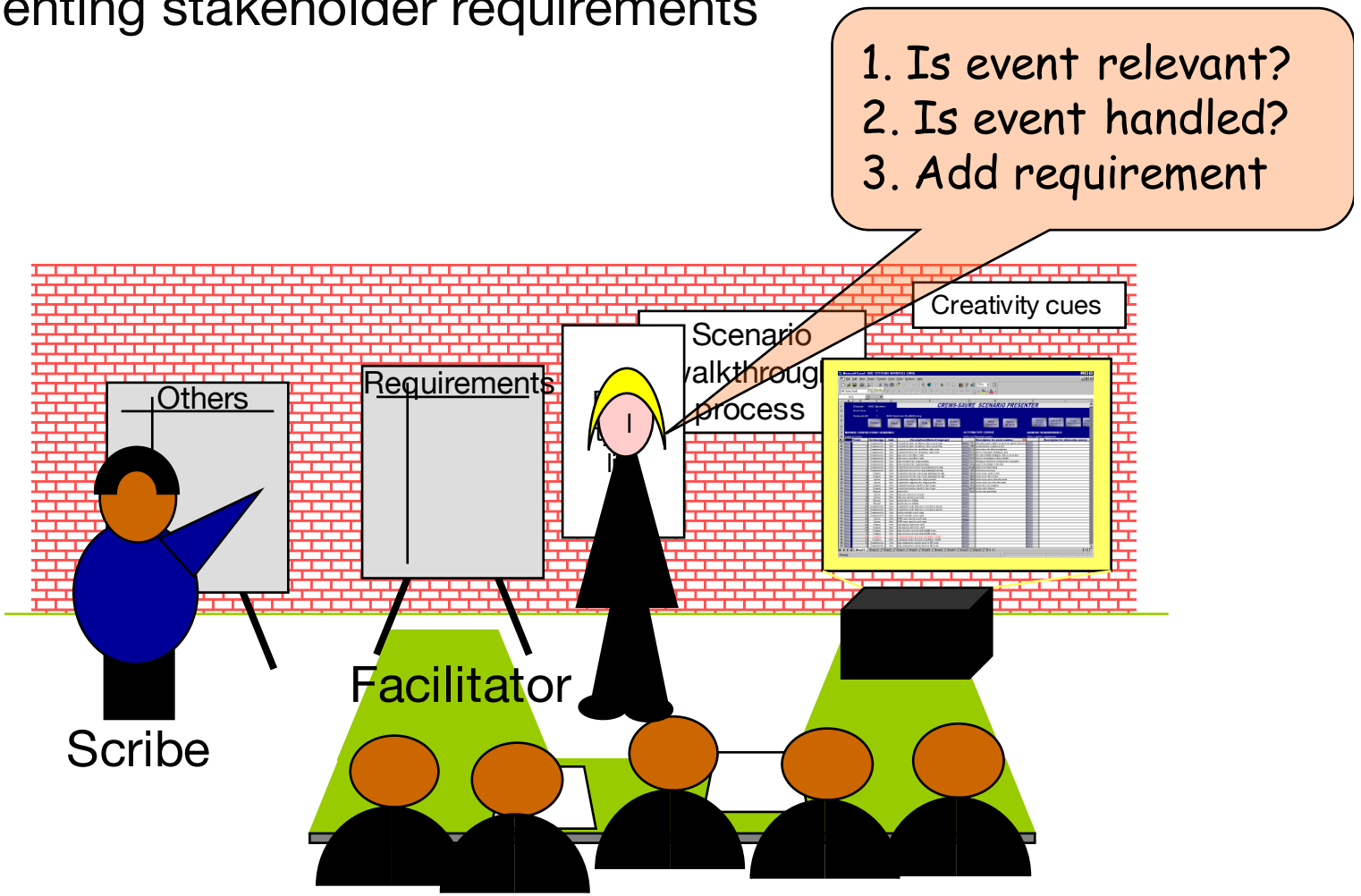
Scenarios in the Wild:
Experiences with Mobile Tools for Scenario-Based Requirements Discovery

Florian Graf **Paul Grünbacher** Norbert Seyff

Systems Engineering and Automation
Johannes Kepler University Linz
www.sea.uni-linz.ac.at

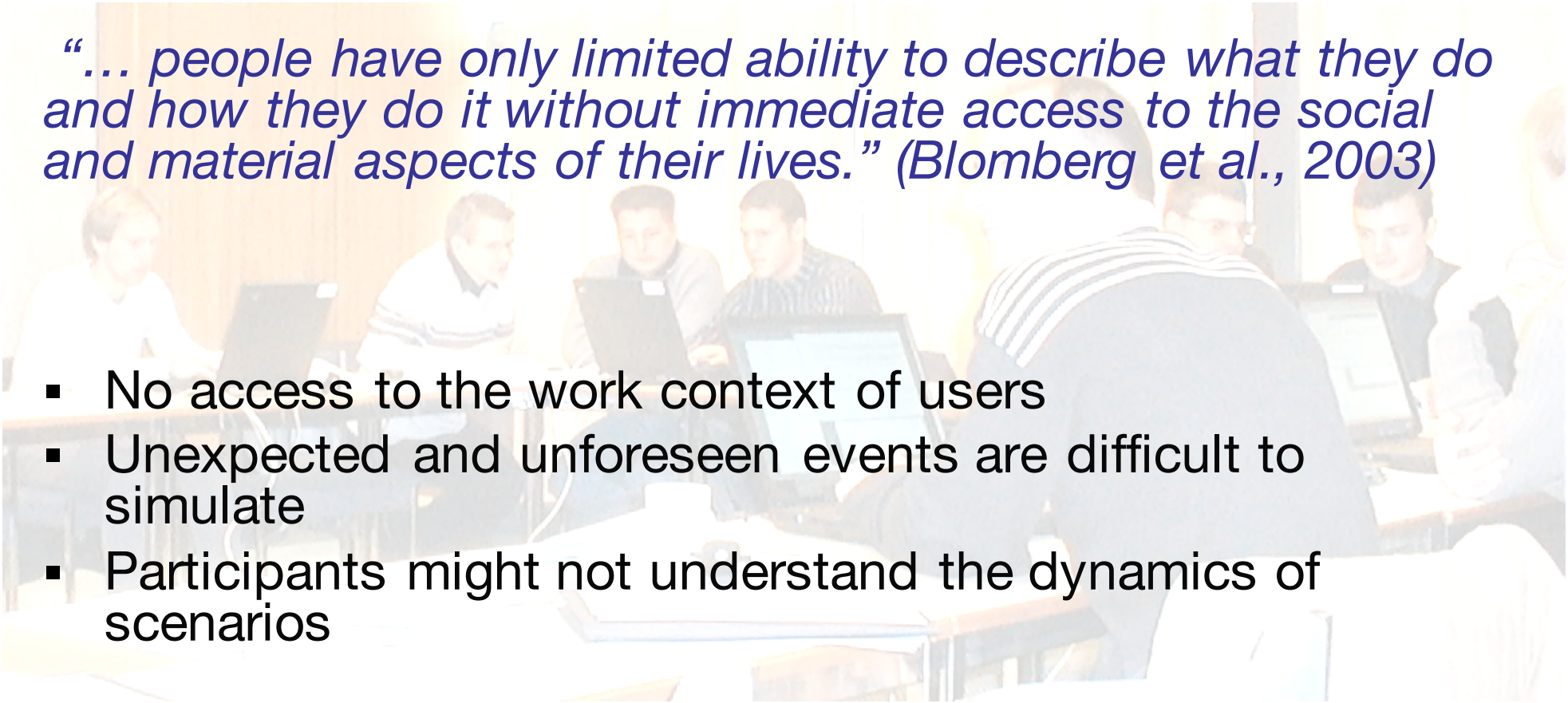
Using Scenarios in Workshops

- ART-SCENE (Analysing Requirements Trade-offs: Scenario Evaluations) is a scenario-driven technique for discovering and documenting stakeholder requirements



Disadvantages of workshops

"… people have only limited ability to describe what they do and how they do it without immediate access to the social and material aspects of their lives." (Blomberg et al., 2003)



- No access to the work context of users
- Unexpected and unforeseen events are difficult to simulate
- Participants might not understand the dynamics of scenarios

- High costs and effort
- Not all stakeholders available

Potential of Mobile Computing for Improving Scenario-based RE

- Allow **in situ use** of tool-supported RE methods
- Capture requirements of future system users directly at their work place
- Use **multimedia** to capture requirements
- Consider **mobility** and changing work context of future users to capture better requirements for mobile and ubiquitous systems

Contextual Inquiry is an Approach Addressing Some of these Issues

- Understand users' tasks in their workplace
  - Observation
  - Structured Dialogue

http://www.usabilitynet.org/tools/contextualinquiry.htm

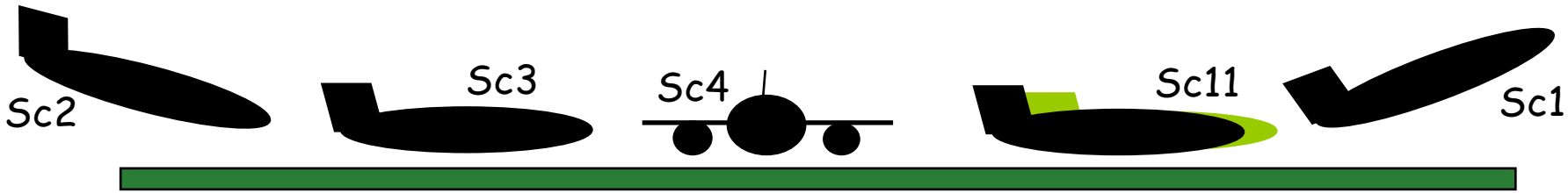
Mobile Scenario Presenter (MSP)

- Unobtrusive use of scenario techniques in the work context
- Mobile capturing of requirements
- Multimedia descriptions of requirements (e.g., audio)
- Complement existing ART-SCENE tool suite

Seyff, N., Maiden, N.A.M., Grünbacher, P., Otojare, O., Mitteregger, K., Making Mobile Requirements Engineering Tools Usable and Useful, Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06), 2006, 29-38.

Example 1: Belfast City Airport

- Requirements for VANTAGE air traffic management system
  - New airport operation systems to reduce environmental impact (noise, gas)



Scenario	Scenario form	Way of use
Sc2 Arrival/approach sequence control	ART-SCENE	Workshop
Sc3: Ground movement control arrivals	ART-SCENE & visualization	Workshop
Sc1: Ground movement control departures	ART-SCENE	Workshop
Sc11: Coordinate flight departures	ART-SCENE	Workshop
Sc4: On-stand operations	ART-SCENE	Work context

Belfast City Airport Scenario Example

1: The in-bound pilot safely parks the aircraft on-stand
3: The in-bound pilot switches off the engines
5: The BEST system records information about throughout the following activities
7: The ramp staff plug-in aircraft to ground power
9: The ramp staff insert 'chocks'
11: The ramp staff insert steps for passengers
13: The ramp staff open aircraft hold doors and offload bags and cargo
15: The in-bound cabin crew prepare to deplane the aircraft
17: All passengers leave the aircraft
19: The customer service agent assist special needs passengers
21: The customer service agent directs arriving passengers to the arrivals area to collect bags
23: The dispatcher co-ordinates aircraft servicing
25: The cleaners clean the aircraft
27: The caters cater for the aircraft
29: The fuelers re-fuel the aircraft

…

69: The outbound cabin crew closes aircraft doors
71: The check-in system relays passenger details to the destination airport
73: The outbound pilot calls ATCO to confirm flight plan details, slot time, and airfield weather
75: The outbound pilot calls ATCO to request engine start-up and push back

- What if Pilot is unavailable or malfunctioning during this action?
What if this event does not occur in this scenario?
- What if this event occurs earlier or later in time than expected?
- What if this event occurs more or less frequently than expected?
- What if a different event occurs instead of this event in the scenario?
- What if this event repeats at an earlier or later time in this scenario?
- What if Pilot is physically unable to undertake this action?
- What if Pilot has some unusual physical characteristics that affect his/her behaviour during this action?
- What if something unusual occurs in Pilot environment during this action?

Using the MSP @ Belfast City Airport

Maiden, N., Seyff, N., Grünbacher, P. Otojare, O. and Mitteregger, K.: Determining Stakeholder Needs in the Workplace: How Mobile Technologies Can Help. IEEE Software, Vol. 24, 3/2007.

Requirements Analysis

- Requirements by subject actor
  - All scenarios generated **61 requirements on VANTAGE**
  - *Sc4 On-stand operations* generated **12 requirements on dispatch coordinator**
  - *Sc4 On-stand operations* generated **20 of 26 requirements on dispatcher**
- Requirements by theme
  - *Sc4 On-stand operations* generated **14 requirements** on operations in observed **bad weather**
- Requirements by geographical airport features
  - **24 of 48 requirements** in *Sc4 On-stand operations* scenario - observed air bridge, dispatch office, etc.

Scenario Walkthrough Cost-Effectiveness

- Scenario walkthroughs were **expensive**
  - Consumed valuable airport and airline staff
- Investigated scenario **cost-effectiveness**

Scenario type	Stakeholders	Duration	Number requirements	Reqt per hour per stake-holder
Workshop	8	2.5	22	1.1
Work context	7.2	5	59	8.2

  - Themed **structured interviews** lasted up to **15 minutes**
  - **Relevance** determined by observed events
  - Stakeholders in **workplace**, return to work immediately

Example 2: Navigation System for Ski Hikers

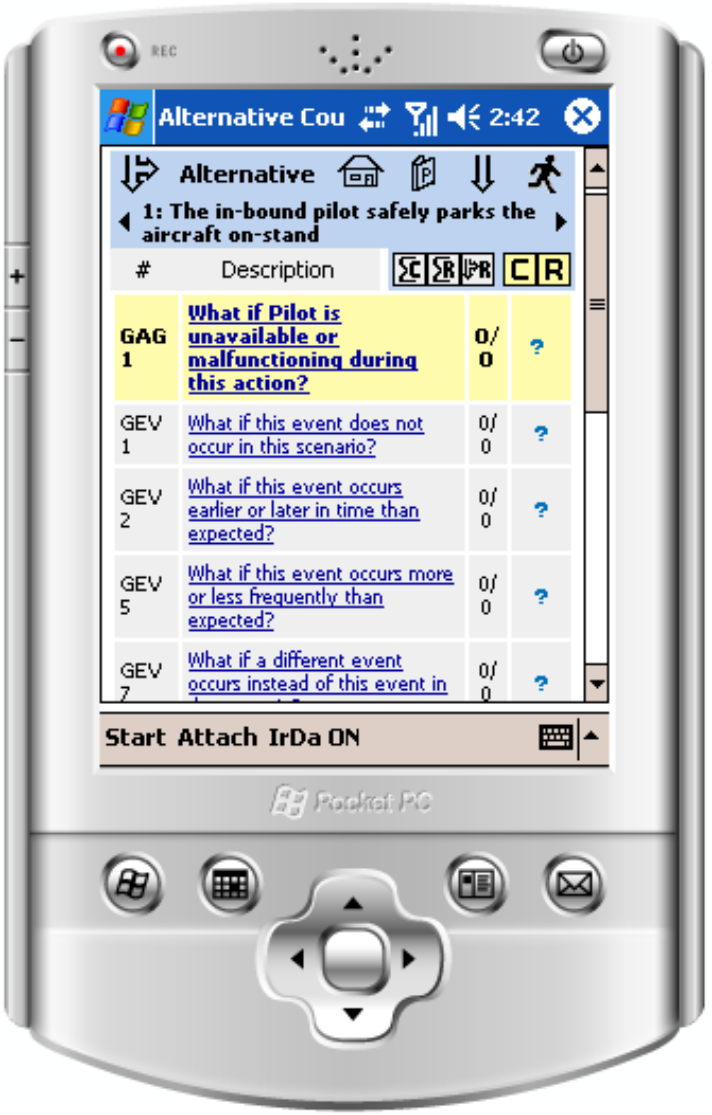
Experiences with Mobile App

Problems

- **Concurrent events** difficult to observe
- Screen **difficult to read with too many alternative course** events
- Limited **battery power**

Documenting requirements

- **Analyst inferred** requirements to handle observed events
- Scribe marked events to **interview stakeholders** later
- **Recognition cues** instead of fully specified requirements



Summary

- Scenario walkthrough in work context trigger **additional requirements**
  - New stakeholders, different types, different themes
  - Analysts learned about stakeholder work
- Scenario walkthroughs in work context did trigger **larger number of requirements**
  - Facilitator directly inferred requirements
  - Increased bandwidth from documents, evidence, etc
  - Heavy use of audio capturing