



**Bournemouth
University**

Writing Use Cases: Comprehensibility: Guidelines and Writing Rules

- We'd like use case descriptions to be:
 - easy to read, understandable, clear, logical, etc..
- This suggests certain questions:
 - Is it possible to state what those properties really are?
 - Can we assess descriptions according to such properties?
 - How do we help people to write use cases (guidance / rules) with those properties?

- Scenarios and use cases are used in industry
- Industry has identified a lack of structure and guidance to writing use cases.
 - Few practitioners considered how to write use cases.
 - Academia has ignored this problem in general.
- Research groups:
 - CREWS project (Co-operative Requirements Engineering With Scenarios),
 - EARTH - the CP writing rules,
 - Ande et al..(templates)

- Deciding upon the properties of descriptions.
 - And how we assess conformance of descriptions?
 - Will introduce umbrella term: comprehensibility.
- Subsequently: guidelines to help support writing use cases so that they will better match these expectations.

Ideal properties of descriptions

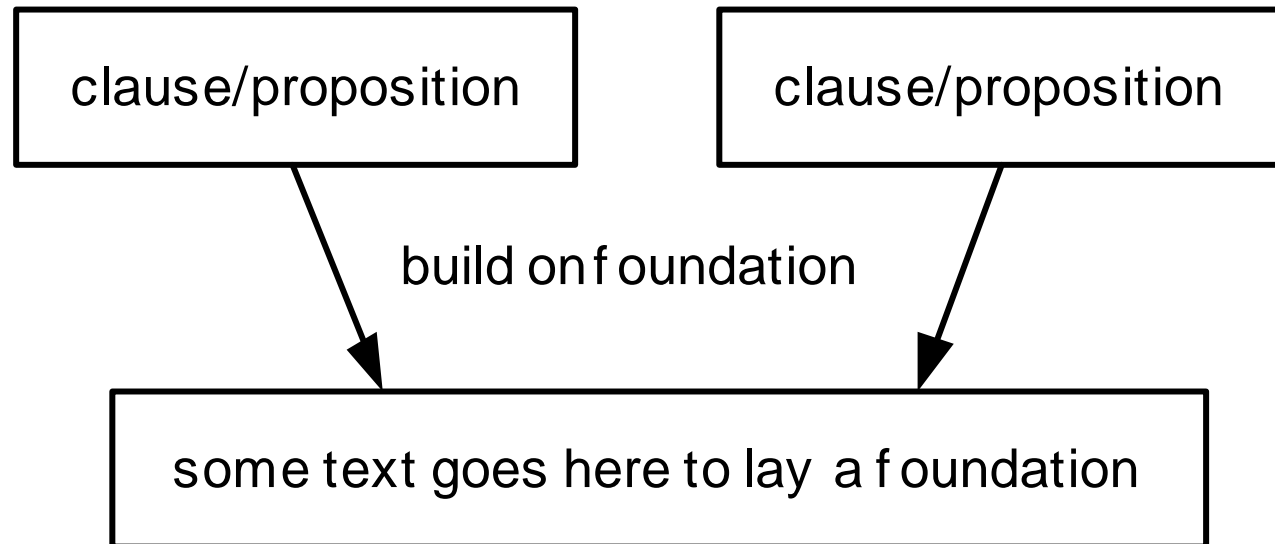
- Essentially looking at structured text.
- Consider structure (grammatical) and style.
- Consider work on text comprehension, typically work on discourse process.
- Relate text comprehension to use case descriptions (meta-model).
- Derive communicability ‘qualities’ factors.
- Example of usage

Factors influencing understanding of text

1. Coherence and Inference
2. Complex Sentences
3. Referential Continuity
4. Structure Foundations

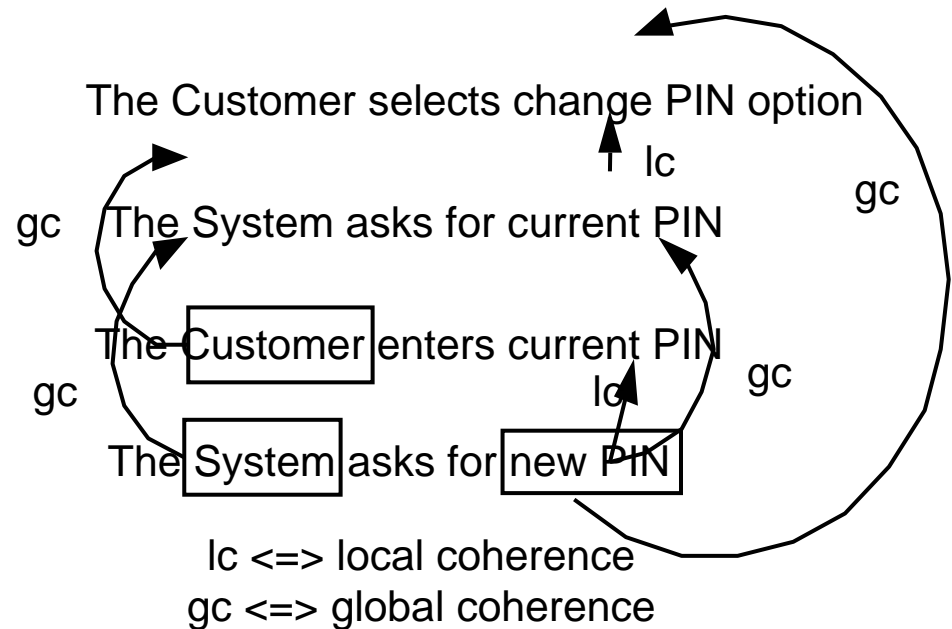
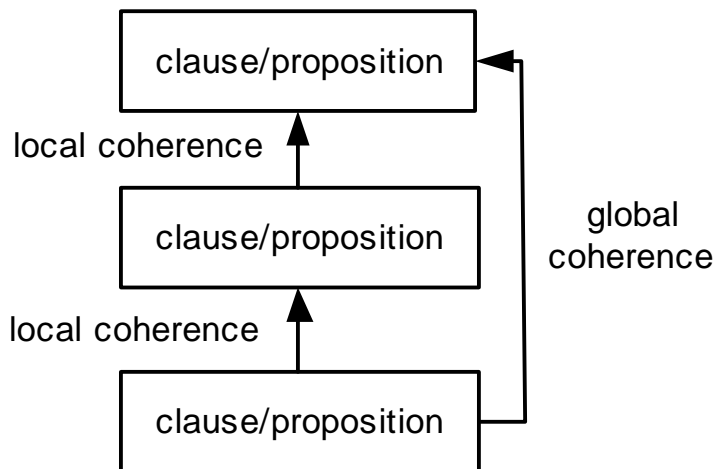
1. Understanding is the assembly of a multilevel representation.
2. Understanding is the construction of a coherent representation.
3. Understanding is a complex dynamic system.
4. Understanding is a process of managing working memory.
5. Understanding is inference generation.

Use case descriptions should continually build upon the foundations constructed at the start rather than add events that force new foundations to be laid.



Each successive sentence should logically cohere to a preceding sentence.

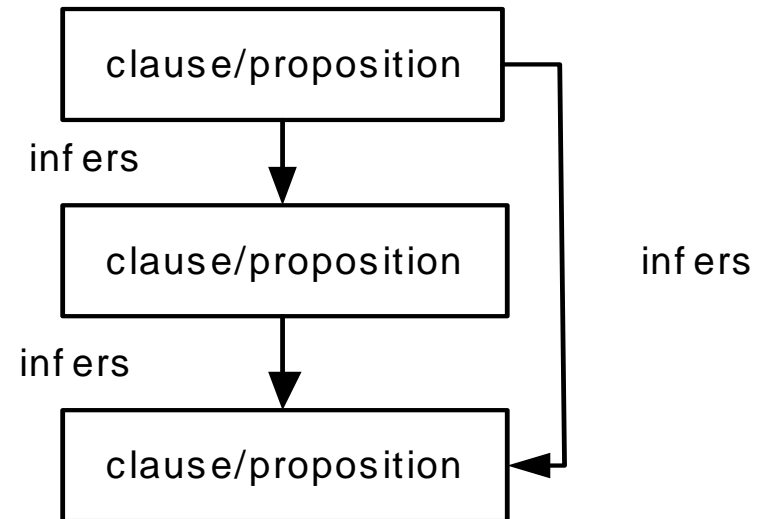
1. Customer selects ChangePIN
2. System asks for current PIN.
3. Customer enters current PIN.
4. System asks for new PIN



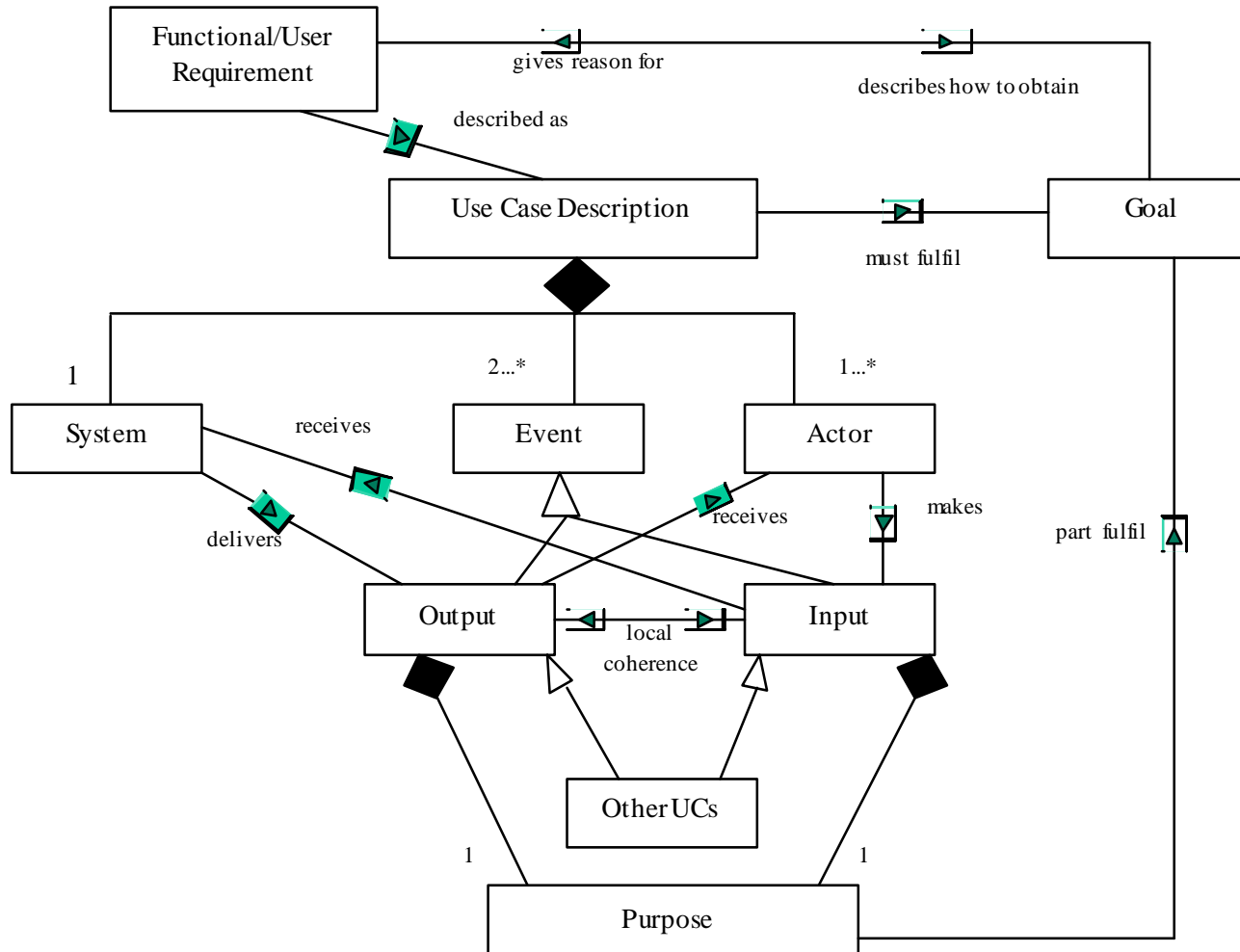
- There has to be syntax but of a relatively simple structure. E.g, adjacency pair
 - “Input -> Response to input,” or
 - “Interaction -> Response to interaction” – this focuses on describing specification.
- If the audience is the user, might be rephrased as:
 - action->response or question->response
 - because users might not be aware of the input output idea.)

- UC description should not be complex; the structures should be straightforward and coherent.
 - Korn (2000) [43] suggests that adjectives and adverbs introduce subjective possibilities of success into scenarios.
 - Subjectivity can lead to misunderstanding.
 - Removal of unnecessary adverbs, adjectives and pronouns

- Reader infers that an event take place based upon information already comprehended.
 - Using the adjacency pair rule should help inference
 - But **DANGER** of inferences.



Meta model



The 7 Cs of Communicability

1. Coverage.
2. Competent Logically.
3. Coherent Logically.
4. Consistent Abstraction.
5. Consistent Structure.
6. Consistent Grammar.
7. Consideration of Alternatives.

1. Coverage

- 1.1.Span: The use case should contain all that is required to answer the problem. That is, is there enough information in the description or is some detail missing?
- 1.2.Scope: The use case should contain detail only relevant to the problem statement. Extra unnecessary information provided is out of problem scope and is not required.

2. Cogent

- 2.1. Text Order: The use case should follow a logical path. Is this path logical or are events in the description in the wrong order?
- 2.2. Dependencies: The use case should complete as an end-to-end transaction (which can include alternative / exceptional flows). Does the actor reach a state that stops the transaction from terminating as we expect?
- 2.3. Rational Answer: The logic of the use case description should provide a plausible answer to the problem. Are there any events that appear out of place or you recognise as incorrect?

3. Coherent

- Coherent. The sentence you are writing now should *repeat a noun* in the last sentence or a previous sentence, if possible.
- The description is easier to read and quicker to understand if there is logical coherence throughout.
- Are there any events in the description that do not cohere to others?

4. Consistent Abstraction

- The use case should be at a consistent level of abstraction throughout.
- Mixing abstraction levels (problem domain, interface specification, internal design mixes) said to cause difficulty in understanding.
 - Is abstraction consistent?
- *Interestingly some evidence to suggest that mixed abstraction might help*

5. Consistent Structure

- 5.1. Variations: Alternative paths should be excluded from the main flow. Inclusion of alternative paths in the main flow reduces readability.
- 5.2. Sequence: Numbering of events in the main flow should be consistent. Are there any inconsistencies?

6. Consistent Grammar

- Simple present tense should be used throughout.
- Adverbs, adjectives, pronouns, synonyms and negatives should be avoided.
- Have they been used?

7. Consideration of Alternatives

- 7.1. Separation: There should be a separate section for any alternative/exceptional paths to the main flow.
- 7.2. Viable: Alternatives should make sense. Are they viable?
- 7.3. Numbering: Alternative numberings should exactly match the numbers in the main flow. Do they or is there inconsistency?

Theoretical Underpinning

7 C's		Rationale
Coverage	Scope	Jackson (1995) [36] – refined notions of completeness for requirements.
	Span	
Cogent	Text Order	Gernsbacher (1996, 1997 [24, 25]) – structure building framework Graesser et al. (1996) [29] – inference building (Question -> Reply to Question)
	Dependencies	Jacobson et al. (1992) [39], OMG (2001)[51] – representing a complete transaction. e.g. Trabasso et al. (1989) [64], Goldman et al. (1996) [27] – local and global coherence. Garnham and Oakhill (1996)[22] – referential continuity. Graesser et al. (1996) [29] – inference building (Question -> Reply to Question)
	Rational Answer	Gernsbacher (1996, 1997 [24, 25]) – structure building framework Grasser et al. (1996) [29] – inference building (Question -> Reply to Question) Anda et al. (2001) [5] – the realism of the use case
Coherent		e.g. Trabasso et al. (1989) [64], Goldman et al. (1996) [27] – local and global coherence.
Consistent Abstraction		E.g. Anda et al. (2001) [5]
Consistent Structure	Variations	Kulak and Guiney (2000) [45], CREWS – keep variations to a separate section [1].
	Sequence	e.g. Schneider and Winters (1998) [63], CREWS – consistent sequential numbering [1]
Consistent Grammar		e.g. Pooley and Stevens (1999) [57] – avoid passive voice; the consensus is there are many grammatical elements to avoid. Some structures might improve comprehension e.g. Graham (1998) SVDPI [30].
Consideration of Alternatives	Separation	Kulak and Guiney (2000) [45] – keep variations to a separate section. Alexander (2003)[4] – failure to deal with exceptions leads to system failures.
	Viable	Alexander (2003) [4] – failure to deal with exceptions leads to system failures.
	Numbering	Cockburn (2001) [14], CREWS [1] – there should be consistency in numbering.

- These are in two parts: General Style Rules and Specific Structure Rules.
- The Style Rules are applicable across the description. There are 7 Style Rules.
- The Structure Rules are specific to individual sentences in the description.
- There are 2 Structure Rules.

- **Style 1:** Each sentence in the description should be on a new, numbered line. Alternatives and exceptions should be described in a section below the main description and the sentence numbers should agree. For example:
 - *Main Flow*
 - 1. *The patient record appears on the screen.*
 - 2. *The doctor enters the patient's new address.*
 - *Alternative Flow*
 - 2. *The doctor deletes the patient's record.*
- The alternatives go below the main flow and the sentence numbers agree (2 and 2).

- **Style 2:** All sentences are in present tense format. The use case should describe events and actions in the here and now, not the past or the future. Some examples:
 - *The operator presses the button.*
 - *The checkout operator enters the amount.*

- **Style 3:** Avoid using adverbs and adjectives, these add unnecessary clutter to the description and give values that are difficult to quantify. Only use negatives in alternative and exceptional flows of events. Avoid using pronouns (E.g. he, she, it, we, their etc) and synonyms. Examples:
 - *Doctor writes the prescription slowly.*
 - *slowly* is an adverb - we don't need to know how the doctor writes the prescription, just that *the doctor writes the prescription*.
 - *Patient swallows the big pill*
 - *big* is an adjective and is unnecessary; you should write *the patient swallows the pill*.

- **Style 3 (contd.)**

- *The patient stands next to the doctor.*
- *He puts the prescription in his pocket.*
- Who is “he”? Whose pocket is “his”? Write proper nouns / names instead:
 - *The doctor puts the prescription in the patient’s pocket.*
 - *The GP puts the prescription in the customer’s pocket.*
- This sentence is at fault because it uses synonyms (GP for doctor and customer for patient). Only use the agreed language of the domain since a synonym does not convey the same meaning.

- **Style 4:** Give explanations if necessary. Explanations should be enclosed in brackets:
 - *The librarian enters the borrower's details (details are: name, address, phone number, library card number).*
- Don't overuse explanations. If you use too many explanations then you have too much information; you need to break the use case down into smaller use cases.

- **Style 5:** There should be logical coherence throughout the description. The sentence you are writing now should refer to something in the last sentence or a previous sentence, if possible. We understand the use case better this way.
 - 1. *The cat sits on the mat.*
 - 2. *The mat belongs to Fred.*
- The *mat* in (2) coheres to *mat* in sentence (1).

CP Style Rules

- **Style 6:** When an action occurs there should be a meaningful response to that action. For example, when there is an input there should be a response to that input somewhere in the use case, usually immediately. This makes sure we do not forget to respond to any action in the use case description.
 - *1. The doctor enters the patient's record identification number.*
 - *2. The system displays the patient's record.*
- Sentence 2 gives an immediate response to sentence 1.

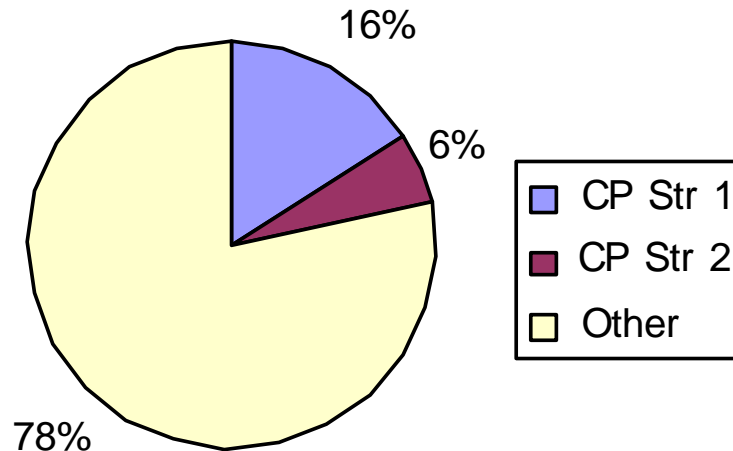
- **Style 7:** Underline other use case names.
 - *The user makes a new equipment request.*
- When it is necessary to include a use case or have a use case extended by another use case, then write the use case name in the sentence and simply underline it.

- **Structure 1:** Subject verb object.
 - *The operator presses the button.*
- Note that **verb** refers to present tense as described in **Style 2**.

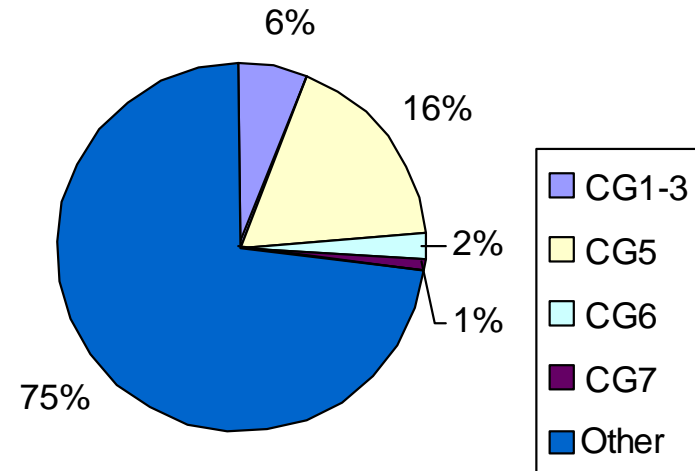
- **Structure 2:** Subject verb object prepositional phrase.
 - *The operator gives the tool **to the mechanic.***
 - *The builder puts the bricks **on top of the pile of rubbish.***
 - *The system reminds the operator **to save all the open files.***
- The **bold** text in the examples are prepositional phrases.

- There has been a rather chaotic mix of grammar used in writing descriptions:

Percentage of CP Structures found



Percentage of CREWS Structures found



- The CP Rules should guide the writer in producing a potentially more coherent description.

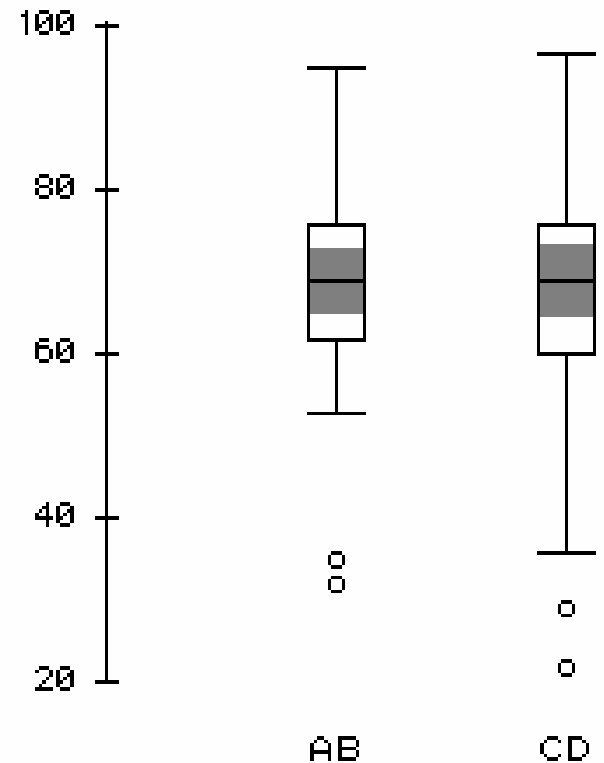
Use Case Question Set

- The notion is to interrogate each event in a use case description to determine:
 - Dependencies (pre- and post-conditions)
 - Actors
 - Interfaces
 - System responsibilities
 - and, only then, classes
- Surprisingly, there is little detailed work on this important area. It seems to be taken for granted.

Experiment 1: Writing Descriptions

- Comparing CP against CREWS, 2 treatments ATM and Retail, 4 groups, 15 subjects per group:
- Hypothesis 1: Counts of Rule Usage
 - CP use significantly less negatives than CREWS
 - CREWS use weakly significantly less pronouns than CP
 - Both use a high number of non-present tense structures
 - CP (Retail) uses significantly more ‘subject verb object’ than CREWS (Retail).
 - CP uses significantly more ‘svo prepositional phrase’ than CREWS equivalent.

- Hypothesis 2: Communicability
 - CP (ATM) has a significantly better Consistent Structure.
 - CP (Retail) significantly more Competent Logically.
- Overall, there is little difference between the groups.
- CP (AB) have a narrower range than CREWS (CD).
- CP Retail results best - opposite to the pilot study (where CP ATM best).



Experiment 2: Comprehending Descriptions

- 2 groups (CP/CREWS ATM), 24 subjects per group.
- H1: CREWS UC more comprehensible
 - Allowable internal design important
- H2: Dependency discovery - no difference.
 - Approx. 2/3 dependencies undiscovered.
- H3: Actor identification - no difference.
- H4: CREWS UC discovers significantly more classes.
 - Low average number of classes identified by groups:
 - CP (2); CREWS (3)

- Use cases really good for testing
- No one has done this kind of modelling before.
 - CP Rules and 7 C's applied as action research.
 - CP Rules used in 87% of events (33 use cases, average 11 events in the main flow per use case).
 - Both CP and 7 C's need tool support.
 - No feedback due to project schedules on these.
- Question Set is important for detailed design
 - But needs to be rationalised to,
 - focus on the important events
 - be more data-centric
 - be more problem specific

Case Study Feedback

Categories		Occurrences	Sources
Time		11	5
Culture		3	2
Business	Requirements	6	3
	IT Knowledge	4	4
	IT Dept Work	3	4
	Process	3	4
Requirements		5	3
Process		5	5

- Issues
 - Experiment results poorer than expected,
 - Students not experienced
 - Experimenter expectations too high
 - Experiments are difficult
 - Abstraction in use case descriptions
 - Mixing abstraction levels in descriptions appears to help in discovery of more classes than sticking to the original notion of only black box or only white box descriptions.
 - Industry problems
 - No Time! If only there was time to do this really neat requirements analysis and design stuff!
 - Business and Marketing departments are the real problem?

Summary

- Industry requires guidance in writing use cases.
- CREWS have proposed a set of guidelines
- CP writing rules have been suggested as well.
- The goal is to make *reading* use cases easier - this is done by imposing writing structure and style upon the description