

**THE LAST PIECE OF THE PUZZLE:
BRINGING INTERACTIVE SYSTEM DESIGN INTO THE
ENGINEERING FOLD**

W Newman

Rapporteur: Dr Robert Stroud

The Last Piece of the Puzzle: Bringing Interactive System Design into the Engineering Fold

William Newman
Xerox Research Centre Europe
September 1999

IV.3

Revisiting the Multi-Disciplinary Argument

- Card, Moran and Newell (1983):
The Multi-Disciplinary approach leads to engineering methods for taking the properties of users into account during system design.
- What does it mean to develop engineering methods?
To assist “*Organizing the design and construction of artefacts so as to achieve desired outcomes.*”
- Computer Science matches this description
- So does the HCI of interaction technologies
- **Interactive System Design is the exception**
- Can we bring it into the fold?

IV.4

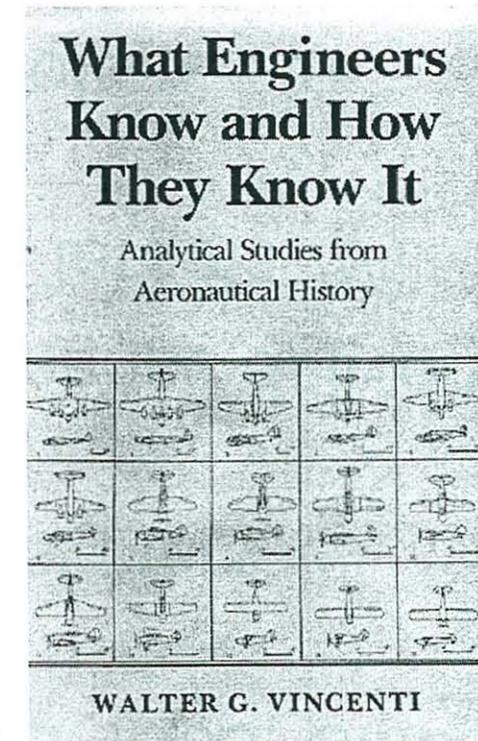
The Argument: Delivering Performance Improvements to Users

- A major challenge for HCI researchers and practitioners
- Failure to deliver regularly cited (Landauer 1995, Gibbs 1997, Hennessey 1999, ...)
- Performance often ignored by system designers (Herbsleb 1993, Newman & Taylor 1997)
- HCI methodologies largely ignore performance delivery
- But designers, given simple performance targets, will tackle them (MacLean 1991)
- Can we encourage them to do this more often – to achieve these desired outcomes?

What Engineers Know...

that equips them to achieve desired outcomes

- about existing designed artifacts
- about environments in which they're used
- about ways of describing these
- about ways of modelling behaviours
- about parameters for measuring critical aspects of performance

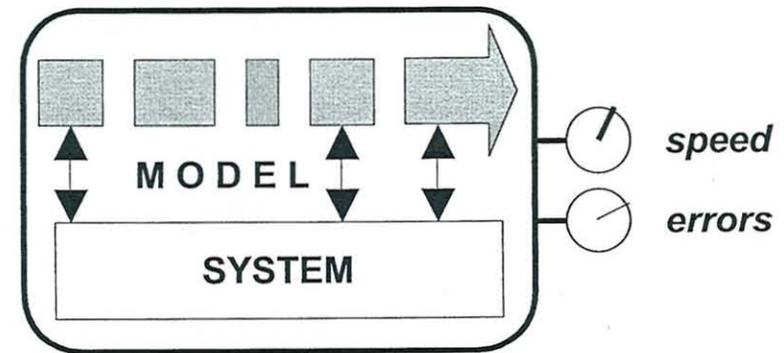
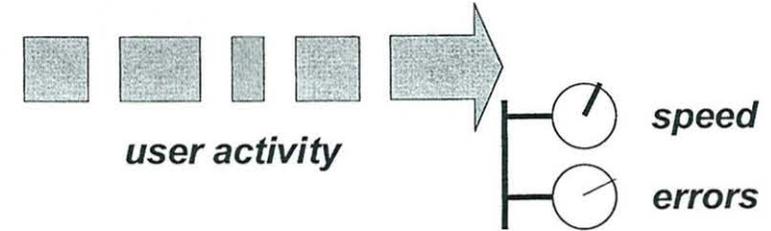


IV.6

models and parameters are essential to achieving desired outcomes

Metrics and Models: Their Roles in Performance Improvement

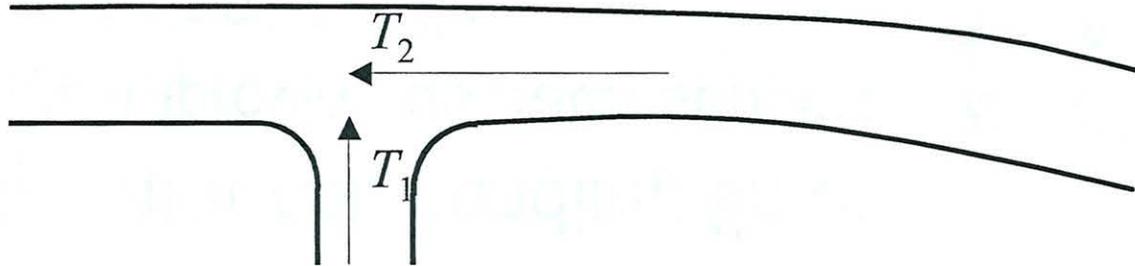
- Metrics enable performance targets to be set...
- and enable performance to be measured.
- Models enable current systems' performance to be explained...
- and future systems' performance to be predicted.
- Neither one supports performance improvement without the other.
- We need to look for metrics and models in tandem.



Critical Performance Parameters

- Routinely used in setting performance requirements in engineering design
- **Established parameters** by which designers can:
 - measure whether artefact serves its purpose
 - compare one design with another
- **Persistent** across generations of design
- **Manipulable** during design
- **Application-specific.**
- Rarely identified or used in Interactive System Design.

A critical design parameter: Lag time at T-junctions



- Designing safe junctions by determining minimum accepted lag time

Lag time = $T_2 - T_1$ = lag between arrival of minor- and major-road vehicles

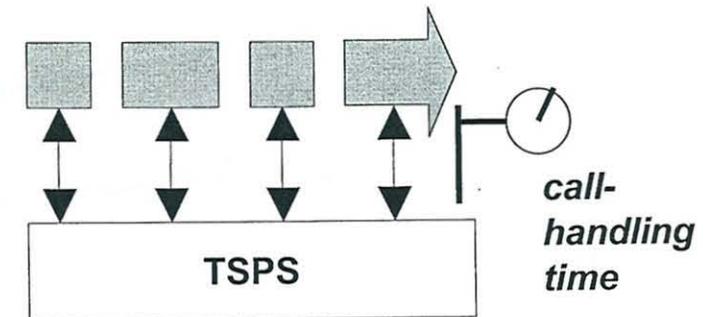
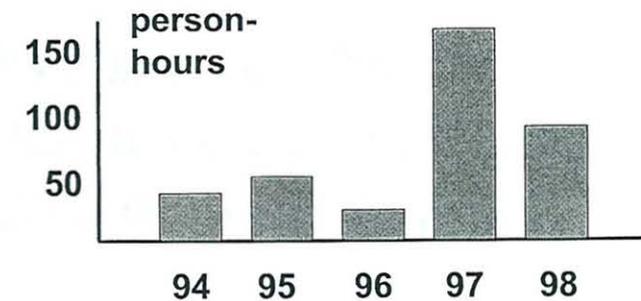
If lag time is less than minimum accepted lag time, driver will not enter main road

Typical minimum accepted lag time is 7 seconds (for 90% of drivers)

- Intersection design must ensure approaching vehicle is visible for longer than minimum accepted lag time.

Critical Parameters: Examples

- Conference papers assignment: speed and adequacy:
 - **critical** to maintaining programme quality
 - **persistent** across every conference
 - **manipulable** ?
- Phone-operator call-handling time:
 - **critical** to improving operator support
 - **persistent** since 1950s
 - **manipulable** by designers of support systems since TSPS (1960s)

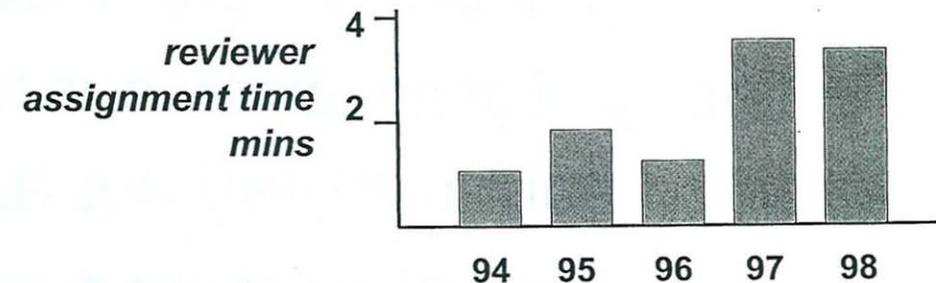
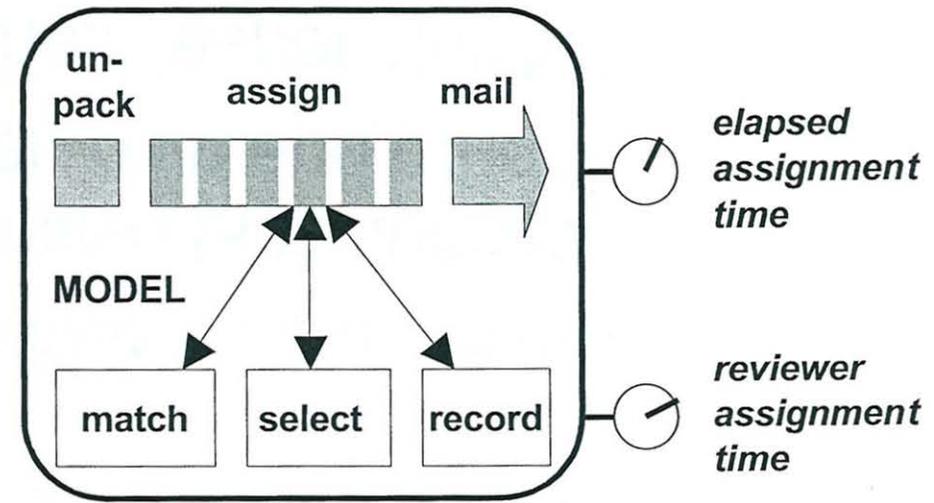


Identifying Critical Parameters in the Workplace

- Critical parameters measure the performance of work:
 - time invested / to completion, error rates, quality of outputs, etc.
- Designers' knowledge of these measures varies between
 - widely known and accepted, e.g., call handling time and
 - little known and largely unidentified, e.g., in papers assignment
- But workers often have extensive tacit knowledge of performance measures – their own and others'
- They rely on these to meet their responsibilities
- Ethnographic study and analysis can elicit this knowledge

How models make performance explainable and predictable

- Models of work structure show distributions of time, errors etc.
- Can be extended to model influence of technology
- Critical parameters can be identified for components of work
- In conference example, **reviewer assignment time** was critical.



Another example: Primary Health-Care Support



- Goal of study:
Understand how to re-design GP system to provide better support
- Candidate critical parameters:
 - Length of consultation
 - Quality of treatment
 - Maintenance of control over consultation
- Each of these parameters is **critical** and **persistent**
- Only parameter that is potentially **manipulable** through technology re-design is Maintenance of Control

A Study of Videotaped GP Sessions (Newman and Button)

- Applying ethnomethodology to analysing GP's use of records
- What studies revealed:
 - during lengthy accesses to records, patients / doctors usually interrupt
 - some interruptions disrupt for >30s
 - interruptions peak at 10 seconds' silence: seizing the "free turn."
- Component critical parameter:
% accesses completed within 10 secs
- A system that assists accessing within 10 secs will reduce loss of control, and thus help in managing length of consultation.

Examples (previous silence, secs)

(12.0) Patient: I can't sleep, doctor
Doctor (typing): mmm.

(11.0) Doctor (while reading) They do pass the buck, don't they?
Patient (laughs): Heh heh.

(8.0) Doctor: Poor old you!
Patient (mishears): How old?
Doctor: Poor old you, I said!
Patient (laughs): Oh yes.

IV.14

Towards Improving the Methodology

- Attempting to improve current methods, not to introduce a new 'Methodology'
- Basic method involves application-specific study/analysis:
 - identify parameters in the work **overall**, and in its **components**
 - model the influence of **component performance** on **overall performance**
- Drawing on a range of study / analysis methods: ethnography, ethnomethodology, task analysis, etc.
- Closing the loop:
Current XRCE research aims to identify parameters in authoring work and test ability to design for improvement

Implications for Researchers and Users

- New roles for researchers:
 - parameter identification
 - model building
 - rethinking the relationship with design teams
- New powers for users:
 - dictating performance requirements
 - insisting on value for money
 - making suppliers more accountable
- All it takes to get started:
 - during design, ask “Are there critical parameters here?”

DISCUSSION

Rapporteur: Dr Robert Stroud

Lecture One

Professor Nygaard felt that the role of System Design as a single discipline was a very important thing to discuss.

Professor Shepherd felt that Mr Newmann was assuming that the Engineering approach was always right but it might be wrong in some situations. Mr Newmann replied that in practice there needed to be a balance.

Professor Martin argued that design studies were about understanding the problem, whilst engineering was about providing solutions. Mr Newmann replied that if the design studies were not closely coupled with the engineering solutions, then a bad result was inevitable.

One of the ICL delegates felt that CHI practitioners who proposed radical solutions must have some model in mind. Was Mr Newmann's criticism that they didn't make the model explicit or did he think that there was no model? Mr Newmann replied that most people didn't know much about the environment for which such novel solutions were intended and that this weakened the model. This was a long and difficult problem to address but a particular concern was that people developing novel solutions didn't have any body of documented designs to draw on.

Professor Martin remarked that as well as documenting designs, it was equally important to document their context and institutional usage. Many of the real problems were non-incremental and knowing what was the same and what was different for big changes would help to provide some context. Mr Newmann agreed with this point and felt that clearly identifying the test conditions could help.

Dr Davis felt that the issue of innovative vs creative designs arose because of the plethora of known designs that failed. Thus, he suggested that novel designs were based on observed failures of existing designs in use. But Mr Newmann disagreed, saying that his work wasn't based on a knowledge of how existing solutions worked but arose out of an interest in exploiting new technologies.

Professor Nygaard suggested that perhaps the problem was that the reward system for engineers and CHI researchers was different. CHI researchers were credited for originality and novelty, not sound experimental work. Mr Newmann agreed that this might be a possible explanation - hence the need to achieve a balance between the two different kinds of approach. But he didn't agree that HCI was risky - building novel systems that wouldn't get tested in the real world wasn't risky.

Professor Hall suggested that the HCI community valued innovative systems and were disrespectful of Microsoft interfaces. Mr Newmann agreed that the engineering value system was different.

Mr Brenner drew an analogy with the work in the patterns community on capturing known designs and strategies, and Mr Newmann agreed that this was relevant.

Professor Kopetz asked whether Mr Newmann was implying that there were two disjoint communities, one building real systems, the other building unreal systems. Mr Newmann agreed with this statement, saying that some HCI systems were indeed disconnected from the user and weren't expected to pay off because the technology was moving too fast.

Lecture Two

Dr Neumann asked how Mr Newmann's view of engineering went down at Xerox. Mr Newmann jokingly replied that he had to introduce the "e-word" carefully because Xerox was a Software Engineering free zone of its own!

Dr Kay asked what Mr Newmann meant by performance? He replied that he meant things like speed of carrying out tasks, and ability to achieve results without errors.

Professor Nygaard asked whether program proving techniques were outside this kind of engineering. Mr Newmann said that he considered program proving to be something that could be incorporated into an engineering discipline but not something that was used in the design of interactive systems.

Professor Randell asked whether Mr Newmann had looked at military systems design. There were obvious problems with getting access to such information but there was a long tradition of evaluation in the UK and he had the impression that the military community made more of an attempt to get feedback on their systems. Mr Newmann thought that Professor Randell might be right but said that he hadn't looked into such systems. How could he gain access? Professor Randell suggested approaching DERA who first starting building computers in 1947. If the military community knew how to tackle performance, why can't others do the same thing?

Professor Cockton wondered whether the military really took performance into account or whether they just did evaluation. But Professor Randell said that there was evidence that they did proper performance evaluation.

During his talk, Mr Newmann mentioned an increase in the time taken by the CHI Programme Committee to assign papers to reviewers. He attributed this to the use of a Web interface to manage the process, which had actually slowed things down considerably if you measured the time this had taken over previous years.

Professor Jones said that he understood the desire for metrics but wasn't there a temptation that people would only measure easy things. What about something hard like how good a referee assignment turned out to be in practice rather than how long it took to make in the first place? Mr Newmann replied that he would be the first to admit that there was a long path ahead but that he thought it was still worth trying to make incremental progress by tackling the issues one by one. However, there was a danger of complacency. What about the danger that the metric might become how many metrics do you have? No, it should be how many metrics can you model.

Professor van Rijsbergen was intrigued by the idea of parameter identification. In the field of Aeronautics, you solved fluid equations by finding parts of the problem you could ignore by using an appropriate parameterisation of the real world. What Mr Newmann seemed to be proposing was the other way round - extracting the model from the data. Mr Newmann replied that if you could draw on science, this was obviously better but there weren't always scientific models available for human behaviour. However, thinking more about models gave you insights into parameters, and this was what psychologists had been doing in developing models of cognition which gave very generic parameters that you could then apply in detail.

A member of the audience suggested that in cases where there were several parameters to be discovered, experimental techniques such as regression could help.

Professor Cockton wanted to respond to Professor van Rijsbergen's comment. Returning to the problem of assigning referees to papers for CHI, it was the underlying roles and responsibilities of the Programme Chairs that made this take a significant time. This kind of qualitative issue was what made HCI different. The parameters in the GP case study weren't

simple scalars that could be fitted to a model and this was what radically distinguished HCI from engineering. Sticking to metrics that were simple scalars was cherry picking. Yes - but what about people who did have a sense of what the performance measures might be? They shouldn't be ignored.

Mr Rowley from ICL remarked that 20 years ago when he went into an Accounts department it was to computerise whatever they did. But nowadays, he found himself more and more doing things they didn't know they wanted. So how can you parameterise something that you don't know exists? The quickest way was to give people a good technique for completing their existing tasks. Mr Newmann replied that although this might be true for mundane tasks, he felt that in professional work, there were measures that persisted - these were based on experience of doing the job and would last. However, since it might be difficult to measure the advantages of a new system pre-delivery, old techniques shouldn't be abandoned.

Professor Nygaard remarked that Mr Newmann always gave him something to think about. Although Professor Nygaard's background was partly in CS/Informatics, he had also worked in Operations Research (OR). As an OR person, object oriented programming had emerged quite naturally. As far as Mr Newmann's point was concerned, as an OR person, he agreed wholeheartedly - what Mr Newmann was presenting was very much an OR perspective and it was indeed the case that big successes in OR came from a sudden/surprising understanding of the critical parameters. Speaking now from the other side, as a CS/Informatics person, he was interested in how to transform design into an engineering discipline. He was pleased to hear that formal methods would have a place, presumably experimental, but where was the balance? What about resolving conflicts? In order to define our field, we needed to take a broad look at what was in the underlying science. Perhaps an engineering perspective was too limited.

Mr Newmann responded by saying that he thought engineering had something to teach us but that software engineering had been an attempt to define what it means to apply engineering to software and people had got it completely wrong. A purely empirical approach to designing and testing systems could not work - for more complex systems, all you could do was look for faults. This is what usability design was about, not whether the product was better.

Mr Budgen contrasted the aeronautical example (where there was a physical model and fairly quick feedback) with a software design. The problem with software was that you couldn't measure it until you'd built the system. How important was this? Mr Newmann replied that one of his aims was to speed up the design cycle by using these techniques to evaluate designs.

Professor Randell wanted to bring the discussion to a close but couldn't resist having the last word. In particular, he wanted to react to Mr Newmann's comments about Software Engineering. A lot of people had blamed Peter Naur and him for inventing the term "Software Engineering" but he wanted to set the record straight. They had merely pointed out that Software Engineering was an aspiration, not a reality. And they had then had nothing to do with it for the next ten years!

