

# Has the Pattern Emperor any Clothes? A Controversy in Three Acts

Luciana d’Adderio  
Division of Informatics  
University of Edinburgh  
luciana@dcs.ed.ac.uk

Rick Dewar  
Computing and Electrical Engineering  
Heriot-Watt University, Edinburgh  
ceergd@cee.hw.ac.uk

Ashley Lloyd  
Management School  
University of Edinburgh  
Ashley@ed.ac.uk

Perdita Stevens  
Division of Informatics  
University of Edinburgh  
Perdita.Stevens@dcs.ed.ac.uk

## Abstract

The following is a light-hearted attempt to expose some difficulties in the use of patterns, including some places where naive ways of thinking about how they are used may be insufficient. We’d very much welcome comments, to feed into a more ”academic” paper on the same subject.

## Dramatis personae

**Gulliver**, a rather naive software engineering student with an eye on his CV

**Alex**, a pattern expert

**Cynthia**, a sociologist

## Act 1, Scene 1

*[A conference hall in a remote German castle. Alex is on stage; Gulliver is in the audience.]*

**Alex**: Patterns are a way to help novices to learn by experience to behave more like experts.

**Gulliver**: *[aside]* Sounds promising. I want to learn to be an expert. Decent pay, job satisfaction...

**Alex**: Why struggle with a common problem that hundreds of people have dealt with in similar contexts, resolving the forces crudely yourself, when there is a well-understood good solution just waiting to be read?

**Gulliver**: *[aside]* Why indeed? This is good stuff.

## Act 1, Scene 2

*[A few days later, at a university reception. Cynthia chatting to Gulliver.]*

**Cynthia**: How are you enjoying life as a student? Been to any good conferences lately?

**Gulliver**: Yes, actually. I heard Alex Blank – you know, the patterns expert who’s visiting here next week – explaining the benefit of design patterns. In fact, he says it’s not specific

to design – patterns can help you be an expert in anything without needing to get any experience yourself!

**Cynthia**: Oh, really? And how’s that?

**Gulliver**: Well, the expert solution is written down in a standard form, and written in a book or incorporated in a website, and then anyone can come along and consult it.

**Cynthia**: But surely, it isn’t that easy? The same solution might be good or bad in different circumstances, for example.

**Gulliver**: That’s all right. The pattern form includes a Context, which tells you the circumstances where the pattern might be useful. Also, there’s a Consequences, or Discussion, section where the pattern writer tells you things you might need to know in order to decide whether you’re in a good situation for using this pattern.

**Cynthia**: But even then, a pattern is just like a recipe. It can provide a useful list of instructions to be followed, but it can’t turn a novice into an expert cook on its own, any more than a car manual can turn anyone that reads it into a proficient driver!

## Act 2, Scene 1.

*[A week later. Cynthia’s office. Gulliver, Cynthia and Alex (who is visiting the university) sit around a table.]*

**Cynthia**: These “patterns” really can’t be a panacea. In my field we know a lot about how difficult it is to capture, store, reproduce and transfer knowledge and experience. I don’t believe any format can just do away with those problems.

**Gulliver**: How do you mean?

**Cynthia**: OK, here begins the lecture. Stop me if you need me to clarify anything.

Firstly, it’s difficult to capture an expert’s knowledge and experience. This has been argued by scholars in many disciplines – Economics, Science and Technology Studies, Organisation Science, Engineering Epistemology and AI among them. The difficulty applies to individual knowledge and also to organisational knowledge. Organisational routines, for example, have knowledge embedded in them which is tacit and

partially impossible to articulate. If you can't articulate it you can't codify it.

**Gulliver:** Tacit knowledge is, like, how we do things round here?

**Cynthia:** Yes – tacit knowledge is bound up with the individual's – and/or the organisation's – background of accumulated knowledge and experience. "Here" may be a broad community, like the object oriented design community.

From what you say, the notion of patterns assumes implicitly that expertise can be easily abstracted, embedded in a pattern catalogue and reused by practitioners when needed, whether they're novices or experts. As I said, many people in many fields have explained the limitations (and to an extent the impossibility) of capturing and codifying knowledge and expertise.

**Alex:** Well, we don't say it's easy, exactly...

**Gulliver:** But you mean, it's *impossible* to write down everything that experts know about a solution. Something will get lost in the pattern writing process.

**Cynthia:** That's right. Patterns can only contain a reduced, abstracted, subjectively filtered and restructured version of knowledge and experience.

**Alex:** That's true. In a way, it's even obvious. But if it's all you've got...

**Cynthia:** These implications alone are sufficient to invalidate the usefulness of the patterns methodology. [*Alex splutters*] But there is more. Let me put all my cards on the table and then we'll discuss it.

While knowledge capture is problematic, so are the processes of knowledge storage (knowledge is always selected and restructured in the process of articulation and codification) and reuse (codified, computer-embedded knowledge must always be reinterpreted before it can be reapplied). Authors in the fields of STS and CSCW have discussed the problems involved in re-enacting, recreating and making stored knowledge "operational".

**Alex:** Patterns don't automatically turn someone into an expert, of course.

**Gulliver:** [*aside*] But didn't you say...?

**Cynthia:** This takes us to the issue of reinterpretation.

According to authors in the interpretivist literature, some ambiguity is not necessarily a bad thing in knowledge creation as it can allow for multiple interpretations and meaning formation. In fact a certain amount of ambiguity can promote inter-functional collaboration and co-ordination of work – this is idea behind "boundary objects" in Science and Technology Studies. In this sense, a tool that aims to reduce or to eliminate ambiguity, such as patterns, does not favour interpretation (or rather it favours one standardised interpretation over other competing ones). Therefore it's often less useful than other more informal tools or methods such as anecdotes.

This need for the pattern user to interpret the pattern leads us to an interesting paradox: those practitioners who are the best qualified to interpret and reuse patterns (the more experienced designers) are also the ones who will least benefit from patterns and who are less likely to want to use them in the first place.

**Alex:** In fact, the patterns community does recognise that it isn't easy to write, interpret, choose, adapt and reuse patterns.

**Cynthia:** But do you think there's still value to be had out of using patterns in that case? That's what surprises me.

**Alex:** The proof of the pudding is in the eating, surely. Patterns have taken the OOD community by storm. Everyone's buying pattern books and training courses and tool vendors have taken to using patterns as a marketing point. Students are learning patterns at university, researchers are applying them to everything in sight.

**Cynthia:** But perhaps they're just on a bandwagon. Let me go on.

There is a further issue related to the reuse of the knowledge that has been embedded in a pattern. I've already implied that knowledge is always transformed in the process of writing a pattern – it's abstracted and reordered according to a new rationale or objective. But also, the knowledge embedded in a pattern has to undergo a further transformation every time it's applied. You can't take for granted that a pattern can be applied in a new context. The knowledge embedded in a pattern has to be recreated in the light of new knowledge and objectives each and every time it is reused. And there is more.

Finally, there is the problem of how pattern use can be incorporated into an organisation's existing practices. This issue is analogous to the issue of artificial language creation and adoption.

**Alex:** Yes – we haven't talked about it today, but people do try to develop "pattern languages" which are collections of patterns with their relationships explained. A pattern language can be a powerful tool, much more so than an unstructured collection of patterns.

**Cynthia:** You see, while the idea of creating a "pattern language" can be attractive, there are important limitations involved in attempting to impose a standardised, artificial language onto an existing organisational community of practitioners. The issues involved in the creation of an artificial language have been identified by a number of scholars including anthropologists, language theorists, and education and learning theorists. These contributions point to the fact that the processes of language creation, uptake and use are highly problematic. For instance, how can expert engineers be persuaded to abandon the methodologies they have learned over years of practice for patterns? There are many examples in the literature of standardised methodologies and knowledge repositories that are being left unexploited because they

aren't relevant, up to date, or just because they don't represent the way people work in an organisation. Why should patterns perform any better? Is there any evidence that they do?

**Alex:** Hmm. I see that what you're saying is right, but somehow I don't feel as though it's a threat. Maybe what we've missed out is the question of exactly *how* people use patterns really. I think it's maybe not as simple as you suggest: it's not just experts writing things down and novices reading them and applying them correctly.

## Act 2, Scene 2

[Later the same day. Cynthia's office. Gulliver, Alex and Cynthia sit around a table]

**Cynthia:** So as I was saying before lunch, knowledge can't straightforwardly be written down and then reused.

**Alex:** Not as hard-and-fast rules about what to do, I agree. But I like Davenport and Prusak's definition of knowledge: "Knowledge is a fluid mix of framed experience, values, contextual information and expert insight that provides a framework for evaluating and incorporating new experiences and information."

**Gulliver:** How does that apply to patterns?

**Alex:** A pattern is arguably a framed experience, that contains contextual information and expert insight. Using patterns to form a pattern language gives us a framework for evaluating and incorporating new experiences and information. Resolving "forces" requires application of values (which I tend to think of as defined in terms of competitiveness for the organisation).

**Cynthia:** Yes, but doesn't their definition work as much against patterns as in their favour? Something written down in a standard form doesn't fit well into something which is a "fluid mix".

**Alex:** It does, provided you don't think that *everything* has been written down. Software engineers are quite used to consulting written material and combining it with other information.

**Cynthia:** Yes, but once you introduce a standard, you create a source of rigidity. For example, the standard format may get in the way of seeing how to apply the solution to a specific case. Also, this newly introduced standard may clash with people's existing practices. How do you get engineers to give up what they thought they knew and use patterns instead?

**Alex:** You don't. That's another important contributing factor to the success of patterns, I think. They add to an existing methodology rather than replacing it: you can use as many or as few patterns as you like and just when they're useful. They're fine grained, in a sense.

**Gulliver:** So I still have to be an expert already?

**Alex:** Up to a point. Remember, although patterns should be rather thoroughly thought-out suggestions, they're still suggestions, not rules. Even once you have chosen to use a certain pattern you still haven't determined everything about your solution. Alexander said something like that you could use the same pattern a thousand times and never do the same thing twice. Same in software design, or any other field that uses patterns.

**Cynthia:** That's OK, but it just reinforces the point that the more experienced the designer is, the better use he can make of the pattern.

**Alex:** Sure. So what?

**Cynthia:** So, a novice would not have sufficient experience to interpret the solution embedded in a pattern and to adapt it to a different context.

**Alex:** That's a different claim altogether. In fact, novices sometimes *have to* solve problems. The question isn't whether using patterns lets them do it as well as experts would – of course it doesn't. The question is whether using patterns helps people solve problems better than they could without them. You seem happy that this could be the case for experts. I'm claiming – OK, without much in the way of evidence – that it's also true for novices, and you haven't, actually, argued against that?

**Cynthia:** But...

**Gulliver:** [interrupting] Calm down you two! I think it's time to adjourn to the pub.  
[all 3 walk off]

**Gulliver:** [musing] Perhaps this stuff about it not being simple to apply patterns explains why experts don't seem to feel threatened by patterns: they really aren't getting replaced by them.

## Act 3

[Late that same night in an Edinburgh pub. 250 malts in the background, 3 in the foreground.]

**Gulliver:** So let me see if I understand what we've agreed. You're saying, Alex, that you've maybe oversimplified in saying what patterns are useful for, and that really patterns aren't just used as recipes.

**Alex:** Yes – but actually no patterns expert would have claimed that that was all they were. The stronger thing that our discussion has pointed out is that patterns aren't even only – perhaps not even primarily – used to let someone directly reuse knowledge.

**Cynthia:** So let's focus on the ways we've agreed patterns are actually used by practitioners. Gulliver, would you summarise the last two hours' debate?

**Gulliver:** First, for direct reuse of knowledge. The idea is that you use a design pattern in your design and using the

pattern improves that design. This is the use that people usually concentrate on, but which we've agreed gets too much emphasis.

**Cynthia:** And we've agreed that it's not as straightforward to do that as people often hope.

**Gulliver:** Second, as a learning tool. Reading design patterns helps you to understand what good design is.

**Alex:** *[interrupting]* And *that's* why it's so important for patterns to include a discussion section. That's vital to help the reader learn why the pattern works, which helps her become a better designer even if she never actually uses this pattern.

**Gulliver:** But I'm not sure about this. Can I really become a better designer just by reading books?

**Alex:** It's not automatic, of course. The kind of reading I'm talking about is *reflective*. You're reading thoughtfully, relating what you read to what you've done, revising your stock of stories.

**Cynthia:** The stories you're talking about are descriptions of situations where some solution did or didn't work well, and what you're saying is that stories can be part direct experience, part vicarious experience?

**Alex:** Sounds right.

**Gulliver:** Yup. Third, as a teaching tool. Discussions about design in organisations can get sticky, especially when your colleague doesn't understand why your proposal is better than hers! You can point to a design pattern as authority. More positively, a well-written pattern does a good job of explaining the pattern, sometimes better than you could do on the board.

That's particularly important when you're talking to new designers. But experts use patterns too. Our fourth use for patterns was as a communications tool. Using pattern names as shorthand can speed up discussion of rival designs. Using pattern names in design documentation can help make the designer's intentions clear – the “why” of the design, something which Alex says is very hard to do and often neglected – by reference to a design pattern which has commonly known intentions.

**Cynthia:** A simplified version of the “why”, of course, as know-how can only be imperfectly articulated. But where were we up to?

**Gulliver:** That was four. The fifth use is to provoke thought. I liked Alex's phrase, that a pattern is an invitation to perform a thought experiment. The process of thinking about whether or not to use a pattern or something like it helps lateral thinking; it gives you a tentative solution to consider.

**Alex:** Right. There's still a lot we don't agree on and many avenues we haven't explored! But at least, we all agree that the way people often talk about pattern use is over-simplified.

As Cynthia keeps saying, everything in a pattern is necessarily abstracted and simplified, and that has serious drawbacks as well as benefits. But we've agreed that there are several ways people use patterns other than just reusing knowledge. And it's important to remember that even a little value may be enough to get an organisation ahead of its competitors.

**Gulliver:** The only thing that seems obvious to me is that you two understand this much more clearly than I do. Trouble is, you understand different things, leaving me the challenge of the differences! Let me try an analogy. Up there in bottles at the bar are 250 good solutions. I don't know which are the best, but I do know that if we'd chosen three different malts we'd be having a different experience now – and we wouldn't necessarily be enjoying ourselves any the less.

**All:** *[raising glasses]* Slainte!