

Situating Business Patterns

Allan Kelly - allan@allankelly.net

1 Introduction

This paper builds on the author's earlier work (Kelly, 2005a, Kelly, 2005b) by situating business patterns in a context that considers *pattern theory* with respect to *Wholeness* (aka *The quality without a name*), the rôle of patterns as knowledge management tools and contrasts patterns with the emerging 'story telling' movement.

2 Audience

This business patterns series is intended to codify several common business practices in a pattern language so they may be communicated and studied more clearly. Primarily this work is aimed at students and new managers.

This paper attempts to position the growing body of business patterns in the wider patterns movement and business domains. As such, this paper is directed at those in the patterns community wishing to gain a deeper understanding of the ideas behind these patterns; and, those from the business community wishing to gain a deeper understanding of patterns.

The author is interested in the applicability of the pattern form to the business domain, whether the form works, what insights it can offer and what value it offers in codifying and communicating business practice. Therefore, the author is in part the audience for his own work.

3 Positioning business patterns

Business patterns are not intrinsically different from architecture or software patterns – what is true for one pattern domain is true for the others. However, each domain will ask different, if overlapping, questions of pattern languages. Therefore it is useful to consider the framework within which patterns exist. By understanding this framework, we are better able to judge what is a pattern, what is not a pattern, what constitutes a good pattern and what we may expect of a pattern language.

Secondly, the audience for business patterns will, it is to be hoped, contain people who are new to patterns. These readers will not be familiar with the framework within which architectural and software design patterns exist. Thus it is useful to set out this framework explicitly.

3.1 *Patterns and their implicit context*

'There is a central quality which is the root criterion of life and spirit in a man, a town, a building, or a wilderness. The quality is objective, precise, but it cannot be named. ...

‘In order to define this quality ... we must begin by understanding that every place is given its character by certain patterns of events that keep happening there.’ (Alexander, 1979, p.ix)

In writing patterns, individuals attempt to codify and document the events that create the quality Alexander speaks of. The result is a pattern or set of patterns (*a pattern language*). However, these are not the patterns themselves but a human interpretation of the events and the results. As such these patterns reflect the context in which they are written: the time, political climate, technical maturity, economic environment and, significantly, the authors’ own views and understanding of the world around them.

In other words, each pattern exists within an implicit context that is not stated and may not even become apparent until later. Pattern writing is not science, although patterns may be grounded in research, theory and examples (specifically the apocryphal ‘three known uses’) they also contain conjecture and advocacy.

For example, in an age when global warming threatens the planet, where pollution in cities endangers health and public opinion dislikes concreting over swathes of greenery, what are we to make of *Ring Roads* and *Nine Percent Parking* (Alexander, 1977)? Indeed *New Scientist* reports that

‘Alexander’s philosophy has turned many cities, especially in the US, into social and ecological disaster areas, teeming with social[ly] deprived neighbourhoods whose inhabitants are forced to rely on the polluting, petrol-guzzling car to maintain the illusion of freedom.’
(Pearce, 2006)

Patterns, like other domain knowledge, are created in a context and in a period of time that brings with it its own set of assumptions, norms and standards. Patterns both document their implicit context and exist within that context. By documenting and naming events we may actually change those which are described.

One of the primary reasons for writing patterns is to communicate our understanding. Naturally, it is easier to communicate one’s own understanding to another who shares a similar environment; the greater the difference in environment the more difficult it is to communicate the idea. This is not just a matter of understanding, it is a matter of applicability.

The pattern *Nine Percent Parking* would have made little sense to any individual living much before the middle of the twentieth century. This is not because the idea was badly communicated, nor does it reflect the validity of the pattern. Before about 1950 there was simply no environment in which this pattern would exist.

It is worth considering how durable each pattern is: some patterns may well be timeless; others may last for decades or just days. While we may identify patterns that are gone and no longer used it is harder to determine exactly when a pattern ceases to be applicable.

3.2 *The quality without a name: Wholeness*

For Alexander, patterns are not an end in themselves, they are only a means to an end; his objective is to generate that *quality without a name* referred to above:

‘To reach the quality without a name we must then build a living pattern language as a gate.’ (Alexander, 1979, p.xi)

More recently Alexander has chosen to name the *quality without a name*, its name is *Wholeness* (Alexander, 2002). As this term is relatively new, much of existing literature still refers to the *quality without a name*, or QWAN for short. In this paper, the two terms will be considered interchangeable.

Alexander is not alone in emphasizing the importance of *the quality*. The comments of Dick Gabriel apply equally to software and business patterns:

‘I believe we cannot come to grips with Alexander in the software community unless we come to grips with this concept [wholeness].’ (Gabriel, 1996, p. 34)

Therefore, if business patterns are to stand alongside architectural patterns and software patterns we must consider the role of *wholeness*, or the *quality without a name*. A full discussion can be found in Alexander (1979). For the moment we may abridge Gabriel:

‘The quality is an objective quality that things like buildings and places can possess that makes them good places or beautiful places. Buildings and towns with this quality are habitable and alive. ...

‘Alexander proposes some words to describe the quality without a name, but even though he feels they point the reader in a direction that helps comprehension, these words ultimately confuse. The words are *alive, whole, comfortable, free, exact, egoless, and external.*’ (Gabriel, 1996, p.34-36)

Gabriel questions if the writers of software patterns share the same goal:

‘When I look at software patterns and pattern languages, I don't see the quality without a name in them, either. Recall that Alexander said that both the patterns themselves and the pattern languages have the quality.’ (Gabriel, 1996, p.69)

However, even if software patterns lack the quality without a name they can still be useful and play a positive role. Gabriel outlines one of the ways in which software patterns are useful:

"Patterns provide several benefits to programmers and system developers. One is as a common language. ...

‘Another benefit is as a common base to understanding what is important in programming. ...

‘A third benefit is that with a corpus of patterns a programmer is able to solve problems more rapidly by having available a storehouse of solutions – a cookbook, if you will.’ (Gabriel, 1996, p.51-52)

We may generalize Gabriel's discussion from software patterns to encompass our consideration of business patterns. So, we may ask three things of a pattern.

- Is the pattern useful?
- Does the pattern possess *wholeness* itself?
- Does the pattern contribute to the generation of *wholeness*?

The answer to the first of these questions largely depends on the reader's point of view and experience. Answering the second and third is difficult because we do not have an objective standard for determining if a pattern actually possesses *wholeness*.

As noted above, patterns both describe and exist in a changing environment, thus a pattern possessing and generating wholeness in one environment, say 1960s America, might not be viewed in the same way in another environment, say, twenty-first century Europe. So, whilst Alexander and Gabriel claim the quality itself is objective, the changing environment makes it hard to validate this claim.

This author believes that, taken together, these arguments render the second and third questions subjective. Who is to be the final arbiter of pattern wholeness? Who is qualified to judge whether a given pattern creates wholeness?

Given these limitations, all we may ask a pattern writer is that they aspire to answer *yes* to these three questions.

3.3 *Events or things*

It is worth noting at this point that patterns are not just about physical things we build. Although we often consider patterns to describe *something* we can build; e.g. a *Marriage Bed* (Alexander, 1977, p.187) or a *Memento* (Gamma et al., 1995, p. 283) it should be recognized that Alexander actually starts by considering patterns of events:

‘We must begin by understanding that every place is given its character by certain patterns of events that keep on happening there.’
(Alexander, 1979, p.55)

The building and the events are entwined and co-dependent:

‘Those of us who are concerned with buildings tend to forget too easily that all the life and soul of a place, all of our experiences there, depend not simply on the physical environment, but on the patterns which we experience there.’ (Alexander, 1979, p.62)

Indeed, patterns about events are not hard to find, e.g. *Do Food* (Manns and Rising, 2005, p. 132) and *Self-Selecting Team* (Coplien and Harrison, 2004, p.124) So, a pattern can describe a thing or a set of events and the two are frequently related.

3.4 *Sustainability, living and dead patterns*

Now we understand the need to aspire to wholeness we need to understand the limits of pattern writing. Obviously, unique events with unique solutions are not patterns, but what of events that recur and produce recurring solutions but which do not contribute to wholeness? For example, the smog creating highways that blight cities such as Los Angeles?

Alexander draws a distinction:

‘These patterns of events are always interlocked ... The specific patterns out of which a building or a town is made may be alive or dead. To the extent they are alive, they let out inner forces loose, and set us free; but when they are dead, they keep us locked in inner conflict.’ (Alexander, 1979, p.x)

Simply, some patterns, dead patterns, perpetuate the conflict. This is not to say all ‘alive’ patterns resolve all forces, far from it. After the application of a living pattern, some forces are resolved, some mitigated, some unresolved and some new ones created. This creates the environment for the next pattern, this is a living process where each solution flows from the next.

Conversely, a dead pattern fails to resolve forces, enhances others and creates new ones in ways that restricts change and resolution. Our ability to create wholeness is reduced.

For example, we might see

- architectural patterns describing the creation of slum housing,
- software patterns describing the creation of a highly coupled monolithic application.
- business patterns describing Enron-style accounting.

We could write patterns about these subjects, but the resulting patterns would not contribute to wholeness; they would be dead patterns in Alexander’s terms.

Things become clearer if we substitute *sustainability* for *wholeness*. Living patterns contribute to the sustainability of architecture, software applications and businesses; conversely, dead patterns reduce sustainability. In fact, Alexander comes close to saying this himself:

‘In short, saying these patterns are alive is more or less the same as saying they are stable.’ (Alexander, 1979, p.118)

So, a business run according to the *Enron Pattern Language* might well grow, employ thousands of people, increase shareholder value and reward the business leaders but eventually it would prove to be unstable and unsustainable.

3.5 *Classifying patterns: Good, Bad, Dead and Alive*

It is useful to explore the subject of *alive* and *dead* patterns a little further. In fact, Alexander initially describes *good patterns* and *bad patterns*:

'A pattern which prevents us from resolving our conflicting forces, leaves us almost perpetually in a state of tension.'

'This stress is then no longer functional at all. It becomes a huge drain on the system ...

'And so 'bad' patterns ... stress us, undermine us, affect us continuously.

'... on the other hand, the corresponding 'good' patterns, when they are correctly made, help us to be alive, because they resolve our conflicts and ourselves.. (Alexander, 1979, p.114-115)

In other words bad patterns don't really solve our problems, at best they may displace the problem from one place to another, they solve the immediate problem but create a problem in another place or at a later date.

Some in the patterns community refer to this type of pattern as *dysfunctional*. We may consider a dysfunctional pattern as one that appears to offer a solution to a problem but does not actually solve the problem in hand, or appears to solve the problem while actually displacing it.

Others have used the term *Anti-Patterns* (Brown *et al.*, 1998). One problem with some so-called anti-patterns is that they describe a problem that reoccurs again and again – while interesting this isn't particularly useful. Some anti-patterns go further and describe the solution to the problem. In effect, now we have an *anti-pattern* that describes a problem, in a context and a solution, with the net-effect that the *anti-pattern* is little different to an ordinary *pattern*.

Alexander himself quickly advances the discussion from *good* and *bad* to *living* and *dead*:

'Each bad pattern in our environment constantly reduces us, cuts us down, reduces our ability to meet new challenges, reduces our capacity to live, and helps to make us dead ...

'While, on the other hand, the corresponding 'good' patterns, when they are correctly made, help us to be alive, because they allow us to resolve our conflicts ourselves.' (Alexander, 1979, p.115)

And he continues:

'Patterns are not merely instruments which help us live: they are themselves alive and dead.' (Alexander, 1979, p. 115)

One could argue with the term *dead pattern* on the grounds that something that is dead does not actually exert any force, by definition, a dead thing is inactive and inert. In this way *dead* is not the opposite of *alive*; that which is dead is not destructive, indeed, a *dead thing* is highly stable and changes little. Conversely, something that is living changes and may never reach stability. On the whole it might be simpler to stick with the terms *good* and *bad*.

It is useful to make one further distinction in classifying patterns: that between patterns that are currently in use, and patterns that have fallen from us. For example, *Bathing Room* states:

‘Concentrate the bathing room, toilets, showers and basins of the house in a single tiled area. Locate the bathing room beside the couple’s realm - with private access – in a position half-way between the private secluded parts of the house and the common areas; if possible, give it access to the outdoors; perhaps a tiny balcony or walled garden.

‘Put in a large bath – large enough for at least two people to get completely immersed in water; an efficiency shower and basins for the actual business of cleaning; and two or three racks for huge towels, one by the shower, one by the sink.’ (Alexander, 1977, p.685)

While *Bathing Room* sounds like a wonderful pattern - and this author would certainly like to live in such a house - it is absent from most, if not all, modern constructions. Observation in Europe and North America shows little evidence of large baths being installed. A more current pattern might state:

‘Provide several small bathrooms, one *en suite* for parents, one shared between the children and another *en suite* for the guest room. Only the master bathroom need contain a bath.’

This is not to say *Bathing Room* is necessarily wrong just that it is seldom used. The context (be it explicit or implicit) that created and sustained *Bathing Room* no longer exists. It may well be that with time the context will return and *Bathing Room* will become common again.

Using the two desiderata of good/bad and used/unused we may classify and name patterns types as shown in Table 1.

Table 1 - Classifying and naming patterns

Current	Pattern (Good pattern, Living pattern) <i>Builds towards wholeness</i> <i>Sustainable</i>	Dysfunctional pattern <i>Detracts from wholeness</i> <i>Unsustainable</i>
Unused	(Unused) Pattern <i>Could build towards wholeness if applicable</i>	– <i>Does not detract from wholeness because it is unused.</i>
	Good (Alive)	Bad (Dead)

3.6 *Worse is better: path dependency?*

If our goal is to address all forces and create wholeness we must indeed address all forces: technical, social and economic. In some cases it may simply not be possible to address all forces; in other cases a *good enough* solution may suffice to resolve some of the forces, or at least mitigate those that remain.

Gabriel's *Worse is Better* (-, 1996, 1990) argument suggests that all forces are not equal; the forces of economics and the market can overwhelm technical forces. Similar arguments are made in the pattern *Big Ball of Mud* (Foote and Yoder, 1999). This describes how shanty towns and monolithic software applications can come to be when the immediate forces of architecture and software design are overwhelmed by the environment we find on the ground.

On the one hand, *Big Ball of Mud* cannot contribute to *wholeness* because it describes how we break every rule of good architecture. On the other hand, both the monolithic application and shantytown continue to live; the monolithic application continues to be run and the shantytown continues to provide homes for people; both continue to change and evolve as living entities.

The monolithic application stands in contrast to the software application that, no matter how beautifully engineered, has been abandoned and no longer lives. The shantytown contrasts with the rural village that died after the population moved to work in a city and live in the shantytown. Beauty is less important than life, the monolith may be ugly but it lives, the shantytown may offend our senses but it lives.

Given the right context the *Big Ball of Mud* makes sense. Sometimes the forces will balance in ways we do not expect; sometimes the virtues of a design may not be apparent until the context changes. For example, shantytowns model many features we now consider virtuous:

“These shanties meet many of the ideals of eco-city designers. They are high-density but low-rise; their lanes and alleys are largely pedestrianised; and many of their inhabitants recycle waste materials from the wider city.” (Pearce, 2006)

We return to the issue of sustainability. The monolithic application and the shantytown are sustainable for a period of time but maybe not indefinitely. Eventually the application will fail or become unmaintainable; the shantytowns may fail because of natural or social phenomenon such as earthquake, landslides, social breakdown and riots.

Given the disadvantages, how can the *Big Balls of Mud* survive for so long? Part of the answer may be that they are highly adapted to their environment; i.e. the context they live in. Another part of the answer may be that, like living organisms, they exist to survive, and the odds of survival increase once they survive childhood.

We may consider life itself to be a force. Once a living thing is brought into being it is no longer neutral and will exert forces itself. One of the most powerful is the survival force, the force to continue living. The desire to continue living is not unique to biological forms and de Geus (1997) has suggested this exists within corporations too.

Consider a situation were a bad pattern, one that does not contribute to wholeness and has limited sustainability becomes the dominant solution to a problem. Once in existence, the more established it becomes the harder it is

to remove. Such a position is analogous to ‘vendor lock-in’ and the economic theory of *path dependency*.

According to the path dependency theory, the QWERTY keyboard and VHS video recorders achieved market dominance not because they were technically superior but because they were first to market. Following this line of argument, the *Big Ball of Mud* exists because, once it has been brought into being, the cost of replacing it not justified.

A fuller discussion of this theory and the analogy behind it is beyond the scope of this paper. For the moment it is sufficient to say that we may draw on the analogy and that the theory of *path dependency* is disputed (e.g. Margolis and Liebowitz, 1998).

3.7 *Knowledge management and Patterns*

Patterns are written with the objective of communicating some piece of knowledge from the author to the reader. A long list of pattern publications demonstrates some degree of success here. Actually, a second objective, less often mentioned, is to allow the authors themselves to understand the pattern better.

The idea that patterns are a form of knowledge management is not new.

‘A relatively new approach known as patterns has created new possibilities for those who wish to manage knowledge in a more efficient and effective way.’ (Manns, 2001)

‘The really big opportunity for patterns and pattern languages, an opportunity for exponential growth, comes to light when we perceive patterns as an important part of a larger challenge - that of valuing, exploiting, and managing the knowledge available all around us, in both tacit and explicit forms, ripe for the picking.’ (May and Taylor, 2003)

Patterns both embody knowledge and describe its practical use. Each pattern describes a solution to a problem with supporting evidence of usage and applicability. There is an implied chronology to the pattern that gives it a somewhat story-like nature, for example:

‘We had a problem, there were these forces... therefore, we did X, and as a result we got here. And now we know of some other folks who did similar things.’

There is an emphasis in pattern writing on what you *do* rather than what you *know*, as such, the pattern is a call to action (Alexander’s events). The pattern narrative calls for the pattern to be used – the search for *wholeness* directs us to create and use patterns that live, patterns that are not only knowledge but action too. The concern with action as well as knowledge gives them a role in closing the so-called *Knowing-Doing Gap* (Pfeffer and Sutton, 2000).

A pattern is a form of structured story. Like poetry the structure may take many forms; indeed authors are free to devise their own structures.

However, there must be a recognisable structure, and it must help express the

core elements of a pattern: context, problem, problem forces, solution and solution consequences.

Patterns are not just structured stories. In describing a set of actions and choices, they outline a design. People must consciously choose to build that which is described in the pattern and modify the pattern to their specific environment. Implementing the pattern is a conscious design-oriented action by one or more individuals.

Without the structure, core elements and design imperative, a problem-solution pair may occur again and again and, in common parlance, it may be called *a pattern*, but it is not a pattern in the terms of Alexander, Gabriel and the wider pattern community.

The structured form of the pattern, the emphasis on design and construction, the description of actions, the core elements and tradition of review within the pattern writing community all serve to help capture in depth knowledge. In particular, as May and Taylor point out, it is important to capture the less obvious tacit knowledge within the pattern in addition to the better known explicit knowledge. The structured form challenges us to capture the tacit knowledge.

Patterns in the Alexandrian tradition do not exist in isolation. True, anyone may pick up a book and read some patterns but the creation of patterns is seldom done in isolation. A ‘community’ exists around most writers to review the work, guide the author and socialize the pattern knowledge.

The community helps the written form and more importantly helps authors to capture the nuances and intricacies of the pattern, that is, helps the author capture and document much of the tacit knowledge that hides inside the problem-solution.

The pattern community also serves to keep the writer honest: a writer who ignores *wholeness* will eventually be questioned and his dead patterns exposed.

3.8 *Story telling and Patterns*

Recently, another form of knowledge management has appeared that also seeks to combined action with knowledge: *Story Telling* (Denning, 2001, Brown *et al.*, 2005).

‘Stories are one of the ways knowledge is transmitted, especially social knowledge.’ (Prusak in Brown *et al.*, 2005, p.47)

‘Once we understand what knowledge is, where it resides, and how knowledge is communicated, we discover that narrative plays an unexpectedly large role. ...

‘There are a lot of ways to capture knowledge that kill it stone dead, and it’s very hard to spread knowledge when it’s dead.’ (Brown *et al.*, 2005, p.53-54)

Reading Denning and Brown’s works, it is hard not to draw parallels with the pattern community: values and concerns about change, community, abstraction (or *compression* for those subscribing to Gabriel’s (1996)

definition) and the importance of practice recur. Brown's call to capturing *living knowledge* echo's Alexander's call for *living patterns*.

There are also differences between the two techniques, for example Denning emphasis the role of verbal story telling while patterns emphasis the written form; stories are spread by socialization where patterns (outside of shepherding and conferences) are spread *via* books and the web.

Denning (2001, p.197) provides an appendix describing 'Elements for Developing a Springboard Story' and it is interesting to contrast these elements with accepted a pattern norms – Table 2.

The approach taken by Denning and others to knowledge management through storytelling seems parallel to the approach taken by pattern writers but the two are not the same. This author hopes that the two communities will be able to learn from one another.

Table 2 – Parallels and differences between Denning's story elements and pattern writing

Denning's Characteristic	Denning's explanation	Parallel in patterns
The explicit story should be relatively brief and textureless	The story can be brief and should have only enough texture and detail for the audience to understand it. ...	Patterns should be short and concise. Patterns should not include needless information.
The story must be intelligible to the specific audience	The audience needs to understand enough about the protagonist and the initial incident for them to be <i>hooked</i> ...	Writers should understand who their audience is. Pattern must pose a real, non-trivial, problem, writers should build tension to be resolved later in the pattern.
The story should be inherently interesting	To capture interest, the <i>actions</i> described might be difficult, with a <i>predicament</i> that cannot be handled in a routine manner, and some tension between characters in the story; or <i>unexpected events</i> ...	Patterns should describe a real problem with multiple forces creating tension in the problem statement. The solution should be non-obvious, at least at first, and should describe implementation action.
The story should spring the listener to a new level of understanding	... it must epitomise or embody the change idea, almost like a premonition of what the future will be like. ...	The reader should learn something from the pattern. By naming the pattern we move to a new level of abstraction.
The story must have a "happy ending"	The story needs to spring the listener out of the typical negative, questioning, sceptical frame of mind ... the listener	The solution is an essential element of the pattern, reading it should give the reader an " <i>aha!</i> moment".

	must [achieve] a positive <i>aha!</i>	
The story should embody the change message	Stories can help persuade people to change ... which the audience can discover on its own and make into its own change message.	Patterns do not necessarily embody a change message. The solution may imply some degree of change but often the solution describes a technical fix.
The change message should be implicit	... one lets the listener discover the implicit change message so that it becomes the audience's own idea and meaningful to them.	Patterns contain explicit messages that should be clearly communicated. Readers are encouraged to relate the pattern to but everything is described clearly.
The listener should be encouraged to identify with the protagonist	Much of the power of the springboard story derives from the fact that the audience identifies with the protagonist ...	No direct parallel. Forces and consequences encourage the writer and reader to consider the other protagonists/stakeholders effected by the problem or solution.
The story should deal with a specific individual or organization	People are more likely to project onto single individuals ...	Examples are often, but not always, used to build the pattern through a single case.
The protagonist should be prototypical of the organization's main business	The individual should be a central figure in the business of the organization ...	No direct parallel. Patterns are written for a specific audience.
Other things being equal, true is better than invented	Our confidence in the veracity of a supposedly true story is usually not on very solid ground.	Patterns should not be works of theory or conjecture. Patterns should be observed in use, in the real world before documentation. Multiple known uses are valued.
Test, test, test	... A story can be tested on individuals or small groups before being tried out on a large group in a high-risk setting.	Patterns are not directly tested themselves but are expected to document known (and therefore tested) solutions. Shepherding, writers workshops and pre-publication review may provide some degree of pattern testing.

4 Acknowledgements

Many thanks to Klaus Marquardt for shepherding this paper to EuroPLoP 2006. Klaus was good enough to shepherd my VikingPLoP 2005 submission too; why he came back for more I don't know!

Further thanks to the participants of Workshop C at EuroPLoP for many useful comments, praise and suggestions for further reading and work, and in particular to Ian Graham for proof reading the revised paper.

History

October 2006	Minor revisions and edits to post conference version.
July 2006	Post conference revisions: original paper split into two, this one contain <i>theory</i> and a second paper containing the two patterns (Kelly, 2006).
May 2006	Interim version posted on web
February 2006	Submitted to EuroPLoP 2006, shepherding begins.
January 2006	First draft

Bibliography

Alexander, C. 1979 *The Timeless Way of Building*, Oxford University Press, New York.

Alexander, C. 2002 *The nature of order : an essay on the art of building and the nature of the universe*, Center for Environmental Structure, Berkeley.

Alexander, C., et al. 1977 *A pattern language*, Oxford University Press.

Brown, J. B., Malveau, R. C., McCormick, H. W. and Mowbray, T. J. 1998 *Anti-Patterns*, Wiley, New York, NY.

Brown, J. S., Denning, S., Katalina, G. and Prusak, L. 2005 *Storytelling in Organizations*, Elsevier Butterworth-Heinemann.

Coplien, J. O. and Harrison, N. B. 2004 *Organizational Patterns of Agile Software Development*, Pearson Prentice Hall, Upper Saddle River, NJ.

de Geus, A. P. 1997 *The Living Company*, Nicholas Brealey Publishing.

Denning, S. 2001 *The Springboard: How story telling Ignites Action in Knowledge-Era Organizations*, Butterworth-Heinemann.

Foote, B. and Yoder, J. 1999 *Pattern Languages of Program Design volume 4* (Eds, Harrison, N., Foote, B. and Rohnert, H.) Addison Wesley.

Gabriel, R. P., 1990, *Worse is Better*, EuroPAL, Cambridge,

Gabriel, R. P. 1996 *Patterns of Software: Tales from the Software Community*, Oxford University Press.

Gabriel, R. P. - <http://www.dreamsongs.com/WorseIsBetter.html>,

- Gamma, E., Helm, R., Johnson, R. and Vlissides, J. 1995 *Design Patterns - Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA.
- Kelly, A., 2005a, *Business strategy patterns for knowledge based products and services*, VikingPLOP 2005, Espoo, Finland,
- Kelly, A., 2005b, *A few more business patterns*, EuroPLOP 2005, Irsee, Germany,
- Kelly, A., 2006, *Patterns for Technology Companies*, EuroPLOP, Irsee, Germany,
- Manns, M. K. and Rising, L. 2005 *Fearless Change - Patterns for Introducing New Ideas*, Addison Wesley, Boston, MA.
- Manns, M. L., 2001, *Patterns: A Promising Approach to Knowledge Management*, ABIT (Association of Business and Information Technology), Monroeville, Pennsylvania,
<http://www.eberly.iup.edu/ABIT/proceedings%5CPatternsAPromisingApproach.pdf>.
- Margolis, S. E. and Liebowitz, S. J. 1998 *Look, I understand too little too late*,
<http://www.utdallas.edu/~liebowit/palgrave/palpd.html>,
- May, D. and Taylor, P. 2003 Knowledge Management with Patterns, *Communications of the ACM*, 46, <http://www.thedanielmay.com/publications/p94-may.pdf>.
- Pearce, F., 2006, *Ecopolis now*, New Scientist, 16 June 2006
- Pfeffer, J. and Sutton, R. 2000 *The Knowing-Doing Gap*, Harvard Business School Press.