

# Een tijd-(en grenze)loze manier van onderwijs: Pedagogical Patterns

Christian Köppe  
Hogeschool Utrecht  
christian.koppe@hu.nl

May 17, 2011

## 1 Introductie

Elke ervaren softwareontwikkelaar kent design patterns. In het gelijknamige boek, welke in 1995 is verschenen, heeft de zogenaamde *Gang of Four* (GoF) 23 patterns verzameld die helpen bij het oplossen van ontwerpproblemen in objectgeoriënteerde systemen [10]. Dit heeft niet alleen tot de kwaliteitsverbetering van de systemen bijgedragen, maar bijvoorbeeld ook de communicatie tussen ontwikkelaars vergemakkelijkt.

Minder bekend is dat deze manier van kennis delen – het publiceren in patterns – ook op andere disciplines dan alleen softwareontwikkeling succesvol wordt toegepast. Een van deze disciplines is pedagogiek.

Dit artikel laat zien wat pedagogical patterns zijn, hoe deze kunnen worden toegepast en welke andere soorten patterns in het onderwijs een rol kunnen spelen. Daartoe wordt in het volgende hoofdstuk toegelicht wat een pattern is en welke eigenschappen patterns hebben. Daarna wordt het Pedagogical Patterns Project gepresenteerd, wat een van de bronnen van pedagogical patterns is. Aansluitend worden een aantal voorbeelden beschreven, hierbij komt ook direct aan bod hoe deze patterns kunnen worden toegepast. Om te laten zien dat het verzamelen van patterns een voortdurend proces is, beschrijft de auteur in het volgende hoofdstuk een nieuw pattern. Tenslotte wordt gekeken naar andere mogelijkheden voor het toepassen van patterns binnen het onderwijs en uiteindelijk met een conclusie afgerondt.

## 2 Wat is een pattern?

Terwijl patterns al lang bestaan en bijna overal te vinden zijn, is het bewust gebruiken ervan begonnen door Christopher Alexander. Hij schreef in 1977 een boek<sup>1</sup> dat het begin van de pattern community vormde, namelijk 'A Pattern Language' [4]. In 1979 volgde 'A Timeless Way of Building' [3], waar hij de focus op de 'Quality Without a Name' (QWAN) en het natuurlijke formaat van patterns legde. Terwijl de boeken eigenlijk op architectuur waren gericht, heeft hij toch een manier van kennis verzamelen geïntroduceerd welke sinds de jaren '90 in verschillende disciplines wordt toegepast. Alexander beschrijft een pattern als volgt<sup>2</sup>:

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice. [4]"

De basale bouwstenen zijn dus een context, een probleem en een oplossing. Patterns kunnen op verschillende manieren worden verzameld: als *catalogus*, zoals bij het GoF boek, waarbij de patterns alleen worden opgesomd en de onderlinge relaties worden aangegeven, of als *pattern language*, zoals in het eerste boek van Alexander.

Patterns kunnen daarbij op verschillende manieren gerelateerd zijn aan elkaar. In het simpelste geval zijn er alternatieve oplossingen voor een probleem, waar bijv. alleen de context verschilt. In

<sup>1</sup>Alexander was niet de enige auteur, maar meestal wordt alleen naar hem gerefereerd.

<sup>2</sup>Deze definitie gaat verder dan de algemeen gebruikte omschrijving van een pattern als 'herhaalde vaste vorm', zoals bijv. bij gedragspatronen. In dit artikel wordt een pattern dan ook volgens de definitie van Alexander gezien.

het geval van pattern languages worden ook afhankelijkheden gedefinieerd, een voorbeeld hiervan wordt later nog gegeven.

Best practices kunnen op verschillende niveaus bestaan: van abstract en algemeen geldig tot heel concreet en specifiek. Deze verschillen zijn ook in patterns terug te vinden. Als er naar software wordt gekeken zijn bijv. software architectuur patterns zoals beschreven in [9] op een hoger niveau dan de design patterns van de GoF [10].

### 3 Het Pedagogical Patterns Project

Sinds 1993 bestaat de Hillside groep [1], een non-profit organisatie met als doel bestaande kennis in de form van patterns te verzamelen en beschikbaar te maken. Oorspronkelijk was de focus nog gericht op object-technologieën, maar inmiddels zijn patterns uit heel diverse disciplines gepubliceerd. Daarbij wordt op hoge kwaliteit van de publicaties gelet dmv. speciale vormen van reviewing en feedback geven.

De projectleden komen uit de hele wereld, wat impliceert dat de gevonden patterns niet specifiek zijn voor een cultuur, maar toepasbaar zijn voor leeromgevingen in het algemeen. Terwijl de meeste leden van het project een ICT-achtergrond hebben, zijn de beschreven patterns ook voor andere disciplines toepasbaar. De patterns zelf zijn op een hoog niveau (zogenoemde *true invariants* of *nearly invariants*); ze bevatten alleen de essentie van de oplossingen en kunnen zo op verschillende manieren concreet worden toegepast zonder hun geldigheid te verliezen.

Door het pedagogical patterns project zijn al meerdere pattern languages gepubliceerd, waarbij elke taal gericht is op een bepaald aspect van leren. Een aantal voorbeelden hiervan zijn:

- Teaching from Different Perspectives - een taal om docenten te helpen hun cursusmateriaal van verschillende invalshoeken te bekijken [7]
- Patterns for Active Learning - didactische patterns die het actieve leren van studenten op verschillende niveaus bevorderen [6]
- Feedback Patterns - patterns die helpen bij het geven van feedback aan studenten [5]

Al deze kennis staat (waarschijnlijk) ook in andere boeken, maar is daarin niet op zo'n gestructureerde en uniforme manier opgeschreven. De basale bouwstenen - context, probleem en oplossing - maken het toepassen van de patterns makkelijk en kunnen zo vooral beginnende docenten van grote hulp zijn. Meer ervaren docenten zouden waarschijnlijk voor het grootste gedeelte de patterns herkennen, zij het niet onder dezelfde naam. Maar juist deze gezamenlijk gebruikte namen helpen om de communicatie tussen docenten te verbeteren of moduleomschrijvingen generieker te maken.

Op de website van het project [2] zijn de tot nu toe verzamelde patterns beschikbaar en er zijn nog steeds onderwijskundigen en docenten bezig met het verder vastleggen van bestaande kennis met behulp van patterns. Zo zijn bijv. bij de PLoP conferenties altijd tracks waar ook nieuwe didactische patterns worden besproken.

### 4 Voorbeelden en Toepassingen van Pedagogical Patterns

In dit hoofdstuk worden twee voorbeelden van pedagogical patterns gegeven. Deze zijn ingekort en staan in [8] volledig beschreven. Aan de hand van (een onderdeel van) de opzet van het vak 'Patterns & Frameworks' – met als inhoud de design patterns van de Gof [10] en frameworks – van de studierichting Software Engineering aan de Hogeschool Utrecht wordt uitgelegd hoe deze patterns werden toegepast. Er is daarbij bewust voor gekozen om bij de opzet van dit vak gebruik te maken van de pedagogical patterns.

#### Active Student\*\*

Het eerste gedeelte beschrijft het probleem en een stuk context en het tweede gedeelte de oplossing.

*"The deep consequences of a theory are unlikely to be obvious to one who reads about, or hears about the theory. The unexpected difficulties inherent in using the theory or applying the ideas are not likely to be apparent until the theory is actually used. However you might have grown up with the passive style of teaching only and really don't know anything else. But, readings, lectures, and multi-media demonstrations, unless interactive, leave students passive.*

\*\*\*

*Therefore: keep the students active. They should be active in class, either with questions or with exercises. They should be active out of class."*

Dit is een pattern op een hoog niveau, dus een *invariant*. Om het leereffect te verhogen moeten de studenten actief met het onderwerp bezig zijn. Dit is onafhankelijk van het onderwerp zelf. Verder is dit pattern ook een goed voorbeeld van de samenhang tussen patterns binnen een pattern language. Het toepassen van deze pattern (het actief houden van studenten) introduceert een nieuw probleem: hoe houd je de studenten actief? Hiervoor zijn vervolgens in de pattern language voor Active Learning een aantal patterns omschreven.

Dit pattern kan op verschillende manieren worden toegepast: in de vorm van practica, een interactieve opzet van de lessen of huiswerk. Hiervoor kunnen dan ook weer patterns worden gebruikt zoals bijv. *Student Design Sprint*, *Shot Gun Seminar* of *Invisible Teacher* [6]. In het vak 'Patterns & Frameworks' wordt van de net beschreven patterns gebruik gemaakt, maar ook van het volgende.

### **Peer Feedback\*\***

Dit pattern komt uit de pattern language voor Feedback, maar ondersteund ook het actieve leren. Het eerste gedeelte beschrijft het probleem en een stuk context en het tweede gedeelte de oplossing. Het volledige pattern is in [5] beschreven.

*"Typically people assume that learning involves receiving feedback, but this is a rather reactive way of learning and ignores the fact that students can learn a lot by giving feedback. Students are knowledgeable and are able to give helpful feedback, but often they are not confident about the relevance of their experience and are unsure about the value of their own knowledge.*

\*\*\*

*Therefore, invite students to evaluate the artifacts of their peers. The students will provide feedback to their peers by drawing on their own experience and because each student will also have produced the artifact for himself or herself, their experience and knowledge will be explicitly relevant."*

Ook dit pattern is toegepast binnen een project in het kader van het vak 'Patterns & Frameworks'. De studenten kregen een opdracht en hadden twee weken de tijd om een eerste versie van een design te maken, waarvoor ook design patterns een rol speelden. Welke design patterns ze konden gebruiken was niet van tevoren verteld.

Alle groepen presenteerden dan hun eerste versies, waarbij ook een onderbouwing van hun gemaakte keuzes gevraagd was. Verder konden de studenten onderling feedback geven en vragen stellen. De verbeteringen in de volgende opgeleverde versies waren bij alle groepen opmerkelijk.

## **5 Beschrijven van nieuwe patterns**

Nieuwe patterns ontstaan niet uit het niets! Zoals eerder beschreven leggen ze eigenlijk alleen vast wat al aan oplossingen voor een bepaald probleem in een deterministische context bestaat. En dit kan op verschillende niveaus gebeuren, van heel generiek tot heel speciaal.

Een bekend probleem is bijvoorbeeld wat men *typisch studentengedrag* zou kunnen noemen. Als studenten een opdracht krijgen die een lange periode beslaat, dan gaan ze, zonder externe

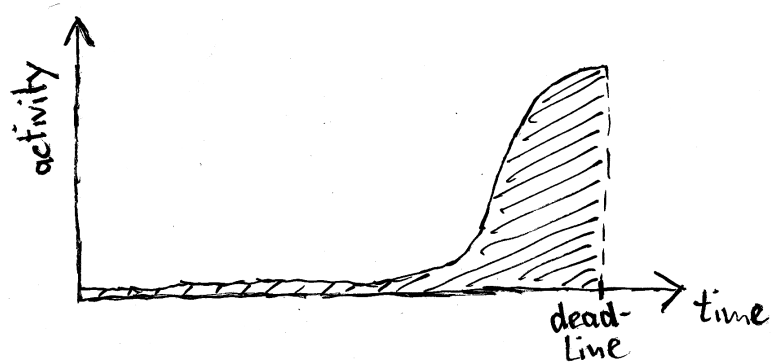


Figure 1: Typical activity-distribution from students

sturing, meestal pas kort voor deadline aan de slag. Figuur 1 laat de verspreiding van activiteit tijdens zo'n project zien<sup>3</sup>.

Dit levert een aantal problemen op. Enerzijds lukt het studenten op deze manier vaak niet het project volledig af te ronden. Anderzijds is de kwaliteit van de opgeleverde producten vaak van lager niveau omdat de studenten niet meer de tijd hebben om op alle aspecten te letten.

De oplossing voor dit probleem is niet nieuw en wordt ook al op verschillende manieren toegepast. Het is dus een kwestie van het isoleren van de essentie van de oplossingen, het gezamenlijke en generieke aspect. In dit geval worden in alle oplossingen continue oplevermomenten gebruikt. Dit houdt de studenten bezig en je kan vaker formatief feedback geven. Figuur 2 laat de nieuwe verspreiding zien.

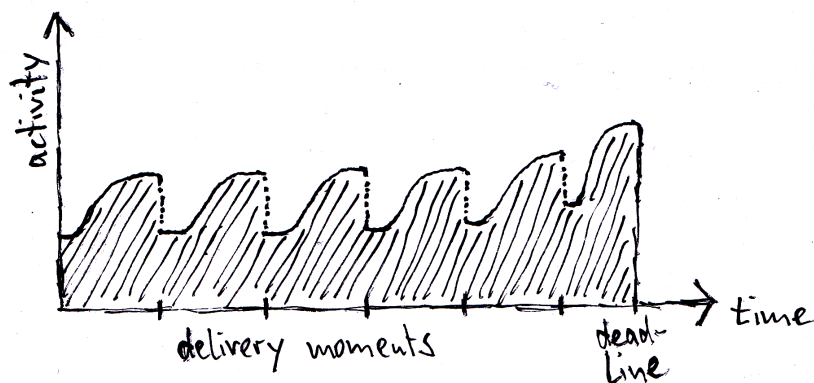


Figure 2: Activity-distribution with delivery moments

Het totaal aan activiteit is toegenomen, het is dus nu waarschijnlijker dat de studenten de opdrachten succesvol af kunnen ronden. Verder heeft de docent een beter overzicht over de status van de studentenprojecten en kan zo gericht feedback geven, wat ook de kwaliteit van de producten en het leerproces van de studenten ten goede komt.

Dit pattern heeft de naam *Continuous Activity* gekregen:

*"If students get an assignment and a deadline, they mostly start too late to work on the assignment. They often are not able to finish the assignment in acceptable quality and on time.*

\* \* \*

<sup>3</sup>In de Alexandriaanse form wordt vaak ook met sketches gewerkt. Alexander schrijft: "If you can't draw it, it isn't a pattern" [3]

*Therefore: include regular delivery moments of appropriate artefacts to force the students to be active over the whole time of the assignment. These artefacts should be of value for the students.”*

## 6 Andere onderwijsrelevante patterns

Naast de pedagogische patterns zijn er nog andere patterns welke voor het onderwijs relevant (kunnen) zijn. Hieronder staan een aantal voorbeelden.

### Learning patterns

Aan de Keio Universiteit in Tokyo werd een pattern language van learning patterns verzameld. Deze werd in de vorm van een boek aan de studenten gegeven<sup>4</sup>. Het voordeel hiervan is dat het concept *pattern* vooral voor ICT-studenten makkelijk toegankelijk is en deze patterns dus een relatief makkelijke ondersteuning tijdens het studeren kunnen bieden. Learning patterns kunnen ook worden toegepast in het kader van Studieloopbaanontwikkeling (SLO).

Het boek wordt op dit moment (status: 7 januari 2011) vertaald naar het Engels.

### E-learning patterns

Omdat e-learning steeds vaker wordt toegepast in het onderwijs is er langzamerhand ook meer kennis over beschikbaar. Zo zijn er bijvoorbeeld patterns voor online discussions, welke een belangrijk onderdeel (kunnen) uitmaken van e-learning [11]. De auteur van deze patterns werkt op dit moment aan verdere e-learning patterns.

## 7 Conclusie

Pedagogical Patterns kunnen een belangrijke rol spelen in het verbeteren van onderwijs in het algemeen en, vanwege de al bestaande affiniteit met patterns, met ICT-onderwijs in het bijzonder. Ze bevorderen de communicatie en kunnen helpen bij het opzetten van nieuwe modules of het verbeteren van bestaande modules. Verder bieden pedagogical patterns een goede mogelijkheid om bestaande kennis op een gestructureerde en herbruikbare manier te verzamelen. Beginnende docenten kunnen ze als handvat dienen en meer ervaren docenten zijn misschien toch nog soms verbaasd over hoe collega's bepaalde problemen hebben aangepakt. Een voordeel is hierbij dat het concept *pattern* al bekend is bij ICT'ers.

Een uitwisseling over ervaringen met de toepassing van pedagogical patterns, maar ook het verder verzamelen van de bestaande kennis in de vorm van nieuwe patterns is een goede en waardevolle toevoeging aan grenzeloos ICT-onderwijs. De auteur kan hierover worden gecontacteerd.

## Dankbetuiging

Mijn dank gaat aan Nini Salet, Jeroen Weber en Henk Plessius voor hun constructieve feedback tijdens onze writers workshop en aan Esther van der Stappen voor haar correcties.

## References

- [1] Hillside group homepage. <http://hillside.net/>, 2010. Retrieved January 04, 2011.
- [2] Pedagogical patterns project. <http://www.pedagogicalpatterns.org/>, 2010. Retrieved June 08, 2010.
- [3] Christopher Alexander. *The Timeless Way of Building*. Oxford University Press, New York, later printing edition, 1979.

---

<sup>4</sup>3600 studenten van twee faculteiten

- [4] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series)*. Oxford University Press, later printing edition, August 1977.
- [5] Joseph Bergin, Jutta Eckstein, Mary Lynn Manns, and Helen Sharp. Feedback patterns. <http://www.pedagogicalpatterns.org/>. Retrieved June 06, 2010.
- [6] Joseph Bergin, Jutta Eckstein, Mary Lynn Manns, and Helen Sharp. Patterns for active learning. <http://www.pedagogicalpatterns.org/current/activelearning.pdf>. Retrieved June 06, 2010.
- [7] Joseph Bergin, Jutta Eckstein, Mary Lynn Manns, Helen Sharp, and Marianna Sipos. Teaching from different perspectives. <http://www.pedagogicalpatterns.org/>. Retrieved June 06, 2010.
- [8] Joseph Bergin, Jutta Eckstein, Mary Lynn Manns, and Eugene Wallingford. Patterns for gaining different perspectives. <http://www.pedagogicalpatterns.org/>. Retrieved June 06, 2010.
- [9] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-oriented Software Architecture - A System of Patterns*. John Wiley & Sons, Chichester, 1996.
- [10] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley, Boston, MA, 1995.
- [11] Jill O'Sullivan and Raul Zevallos. Patterns for online discussions. <http://hillside.net/plop/2010/papers/osullivan.pdf>. Retrieved January 04, 2011.