ISSEP '2005

Professional Analysis of the "Abakusz" Software Development Competition

Erzsébet Angster
Dennis Gabor College
angster@szamalk.hu

Contents

1. The "Abakusz" Competition
2. "Abakusz" documentations and pattern use
3. "Abakusz" surveys
4. Why software pattern?
5. Conclusion, plans

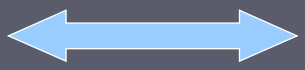
1.
The "Abakusz" Competition

The Abakusz Software Development Competition

- searches for talented students
- examines the student's SD practices (method and technique)
- examines whether students get enough help
- examines whether there are enough material (theory and practice)

http://ww.gdf.hu/abakusz

Level of Competition



They strengthen each other

Level of Education

The following things help	In Abakusz?
Works are publicly available	YES
Prizes to students and **teachers**	YES
Good guidance and materials	TOO FEW so far
Continuing education of students and **teachers**	YES for students (first steps) NO for teachers so far

Developer's category

- 3 tasks in every year:
  information process, game, strategy
- Students have to give
  - Technical documentation
  - User's guide
  - Source code
  - Running code with test data

Basic concepts: OPENNESS, FAIRNESS, QUALITY

- Open documentation
- Open sources
- Open review considerations
- Open review process
- Open review of winner works

”Our goal is that the presented works **serve education** and the judgment is fair.”

Review Considerations

Review consideration	*Score limits*
Technical documentation	10
User documentation	7
Deployment, launching	3
Correctness	10
User-friendship	10
Efficiency	5
Source code	5
Test data	3
Ideas	7
Total:	*60*

Secondary School Students

In spite of the high requirements, **two-thirds** of the competitors and half of the winners come from **secondary schools**.

2. "Abakusz" documentations and pattern use

Documentations are poor!

Some WINNING work's review details:

- ”...Adequate design details are missing.”
- ”The design contains only a data model, and the association representations are not correct.”
- ”The program's logic is missing, so the system is not understandable and implementable.”
- ”Diagrams are not correct and are too abstract.”
- A little bit half made and ”disintegrated”.
- ”... concepts are not clear. The design is too simple, not too understandable.”
- ”Technical documentation is too small.”
- ”The logic is not clear enough.”

Where are the patterns?

- None of the Abakusz competitors used software patterns so far
- Is it a special Hungarian problem???
- Is it a problem at all?

3.
"Abakusz" Surveys

Survey summarizing

Students

- think, that documentation is important, but they have no guidance.
- know nearly nothing about software patterns
- are not satisfied with the teaching materials
- are not satisfied with the help at all
- do not know and do not use complete SDP-s

4.
Why Software Patterns?

What is Software Pattern?

The software pattern is the documented solution of a frequently occurring problem in a given software environment, which **has been tried out more than once by more than one person**.

The pattern contains **everything** which is necessary to understand the problem and the solution, and to solve another, similar problem.

What is Software Pattern? (2)

"... a pattern is a repeating best practice of what works – in any domain"
(Larman, 2002)

"Good programmers know what to write. Great ones know what to rewrite (and reuse)."
(Raymond, 1998)

What is Software Pattern? (3)

The software patterns criteria (1998)

- Rule of three
- Pattern name
- Documentation
- Teaching
- Review
- **Solid solution:** the solution answers the "what?" and "how?" questions ... **to new software practitioners** or experts not practicing this solution. ... **There is an abstract description as well as a specific example.**
- Template
- Practical feedback
- Popular acceptance

- Analysis pattern:
  guidance, how to build up the domain model of an address book, a card play, a drawing program etc.
- Architectural pattern:
  how to organize the software into big blocks. MVC is a base pattern.
- Design pattern
- GUI pattern
- etc.

Problems with software patterns

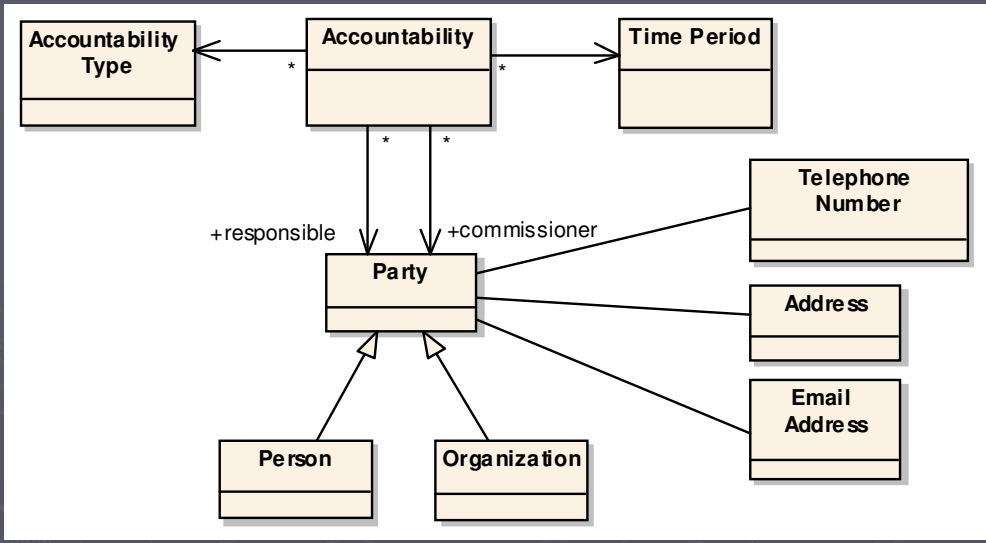
- Pattern use do not diffuse in general [Manns, 2002]
- There are no good and public pattern repositories with concrete and complete examples, especially for teaching purposes

Why patterns are important?

- A student can build his software more easily, if he has a repository of given components, guidance and ideas.
- A software pattern is a unit of SD guidance which has a name and a form so that it can be easily understood.
- A pattern language is a collection of patterns which covers a given domain.
- Everybody needs a pattern repository on his level of knowledge. Also **secondary school students**, who develop software!

An example

Party and Accountability Analysis Patterns



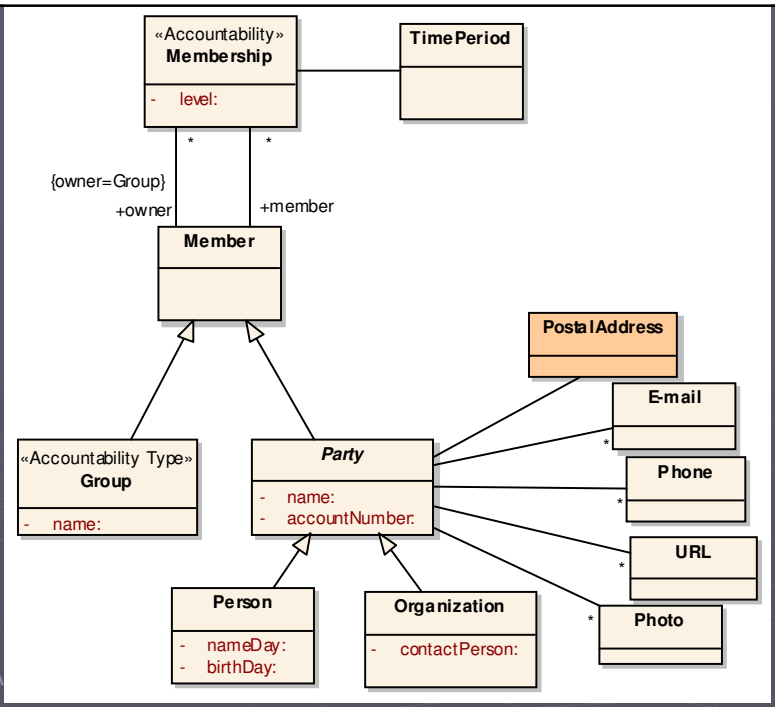
Problems

- This pattern is too abstract. Developers (not only secondary school students) cannot (or do not) apply it to a real problem.
- This is the base pattern for an Address Book.
- Can we expect a secondary school student to develop a non trivial Address Book???
  The reality: they want to develop it!
  The try and do it!!!

A new Abakusz task: Remote Member Storage

- Compulsory elements in the task:
  - Person and Organization
  - Memberships in any groups
  - Handling old members
  - Group in group in any deepness
- Working time: 3 weeks
- Deadline: 4th April, 2005. (next Monday)
- **Several secondary school students are working on it!** (total: 21, secondary: 12)

Possible solution



5.
Conclusion, plans

Survey in the near future

- Is it easy and natural to create the concrete model?
- Is it easy and natural to carry out the implementation?
- How easy is it for a student, a teacher or a developer?

A Possible Solution: a Website with Examples

"Many people know many things but nobody knows everything"
(Latin wisdom)

„I provide you, so that you can provide me!"
(Bismarck)

SDP-city

for developers, teachers, students and mamagers

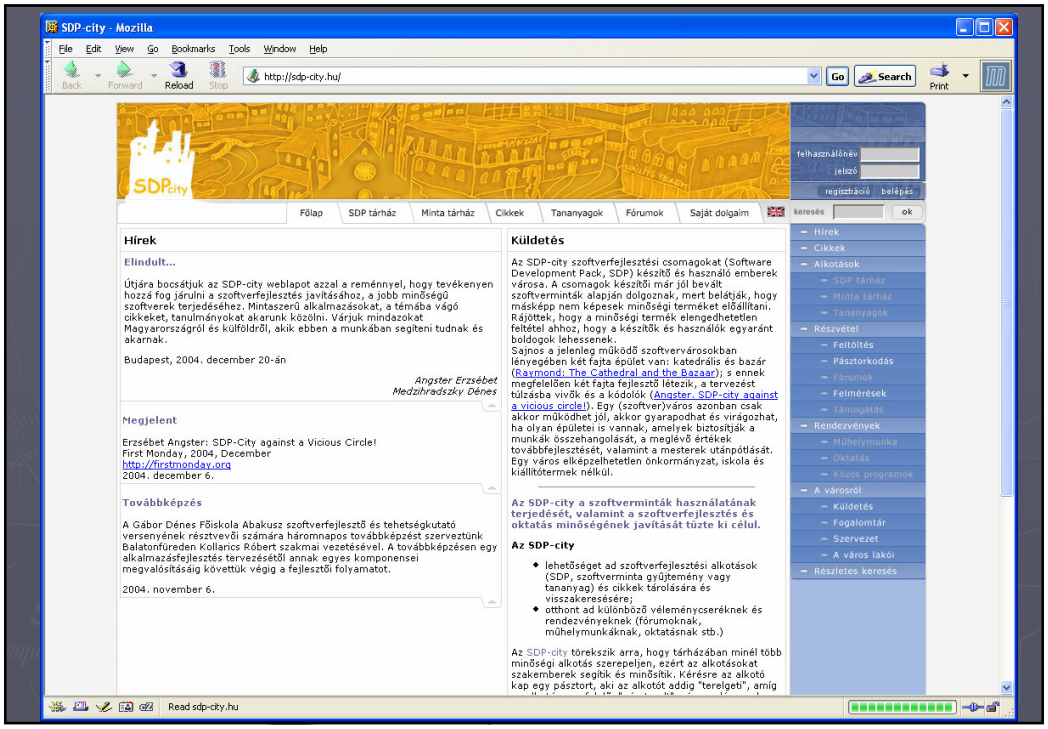
Software patterns

Analysis patterns
Architectural patterns
Design patterns
...

Software Development Packages

Vision
Solution (technical documentation, user's guide, model file, source code, application etc.)
Development history
Patterns contained
Teaching material
...

http://sdp-city.hu


SDP-city - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://sdp-city.hu/
Go Search Print
SDP city
felhasználónév
jelszó
regisztráció belépés
Főlap SDP tárház Minta tárház Cikkek Tananyagok Fórumok Saját dolgaim
keresés ok
Hírek
Elindult...
Útjára bocsátjuk az SDP-city weblapot azzal a reménnyel, hogy tevékenyen hozzá fog járulni a szoftverfejlesztés javításához, a jobb minőségű szoftverek terjedéséhez. Mintaszerű alkalmazásokat, a témába vágó cikkeket, tanulmányokat akarunk közölni. Várjuk mindazokat Magyarországról és külföldről, akik ebben a munkában segíteni tudnak és akarnak.
Budapest, 2004. december 20-án
Angster Erzsébet
Medzihradszky Dénes
Megjelent
Erzsébet Angster: SDP-City against a Vicious Circle!
First Monday, 2004, December
http://firstmonday.org
2004. december 6.
Továbbképzés
A Gábor Dénes Főiskola Abakusz szoftverfejlesztő és tehetségkutató versenyének résztvevői számára háromnapos továbbképzést szerveztünk Balatonfüreden Kollarics Róbert szakmai vezetésével. A továbbképzésen egy alkalmazásfejlesztés tervezésétől annak egyes komponensei megvalósításáig követtük végig a fejlesztői folyamatot.
2004. november 6.
Küldetés
Az SDP-city szoftverfejlesztési csomagokat (Software Development Pack, SDP) készítő és használó emberek városa. A csomagok készítői már jól bevált szoftverminták alapján dolgoznak, mert belátják, hogy másképp nem képesek minőségi terméket előállítani. Rájöttek, hogy a minőségi termék elengedhetetlen feltétel ahhoz, hogy a készítők és használók egyaránt boldogok lehessenek.
Sajnos a jelenleg működő szoftvervárosokban lényegében két fajta épület van: katedrális és bazár (Raymond: The Cathedral and the Bazaar); s ennek megfelelően két fajta fejlesztő létezik, a tervezést túlzásba vivők és a kódolók (Angster, SDP-city against a vicious circle!). Egy (szoftver)város azonban csak akkor működhet jól, akkor gyarapodhat és virágozhat, ha olyan épületei is vannak, amelyek biztosítják a munkák összehangolását, a meglévő értékek továbbfejlesztését, valamint a mesterek utánpótlását. Egy város elképzelhetetlen önkormányzat, iskola és kiállítóterem nélkül.
Az SDP-city a szoftverminták használatának terjedését, valamint a szoftverfejlesztés és oktatás minőségének javítását tűzte ki célul.
Az SDP-city
lehetőséget ad szoftverfejlesztési alkotások (SDP, szoftverminta gyűjtemény vagy tananyag) és cikkek tárolására és visszakeresésére;
otthont ad különböző véleménycseréknek és rendezvényeknek (fórumoknak, műhelymunkáknak, oktatásnak stb.)
Az SDP-city törekszik arra, hogy tárházában minél több minőségi alkotás szerepeljen, ezért az alkotásokat szakemberek segítik és minősítik. Kérésre az alkotó kap egy pásztort, aki az alkotót addig "terelgeti", amíg
Hírek
Cikkek
Alkotások
SDP tárház
Minta tárház
Tananyagok
Részvétel
Feltöltés
Pásztorkodás
Fórumok
Felmérések
Támogatás
Rendezvények
Műhelymunka
Oktatás
Közös programok
A városról
Küldetés
Fogalomtár
Szervezet
A város lakói
Részletes keresés
Read sdp-city.hu