

# ソフトウェアパターン入門

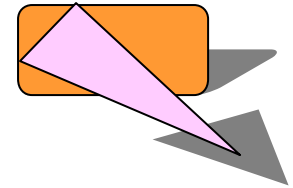
横河電機株式会社  
ソリューション研究所  
井上 健  
2003年5月23日

## 発表内容

- ・ソフトウェアパターンとは?
- ・GoFパターンや、ソフトウェアパターンの数々
- ・歴史
- ・C. Alexander
- ・パターンランゲージ
- ・課題、まとめ

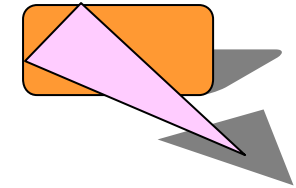
# ソフトウェアパターンとは？

## 「こういう場合はこうなさい」



- 開発における , 問題の解決方法 .
  - 繰り返し起きるような問題の解法を , 「パターン」として蓄積
  - 蓄積されたパターンを再利用し , 開発加速
  - 熟練者の「よい開発手法」を使える .
  - 正しい設計 , 正しい実装などを学べる .
- 複数の解決方法から , 適したものを選択する指針
- 暗黙知を形式知にする試み
  - 形式知を「パターン」という枠で捉え , 多くのパターンを生み出し , 洗練し , 普及させる試み
  - ソフトウェア開発のキーとなる解を追求
- パターンはソフトウェア開発技術向上のための一つのアプローチ？

<オブジェクト指向  
ソフトウェア開発技術>



簡単にすばやく,柔軟で  
品質高いソフトを開発可能

開発プロセス  
(RUP,XP,Agile...)

パターン  
(デザインパターン,  
アナリシスパターン,  
アンチパターン,  
パターンランゲージ...)

90年代のOO方法論  
(OMT,Booch,S/M...)

クラス,カプセル化,  
ポリモルフィズム,

OOは難しい

適用範囲増大

要求が大規模複雑化

OOコンサルティング

リファクタリング

コンポーネントウェア  
(EJB,サンフランシスコ,...)

UML

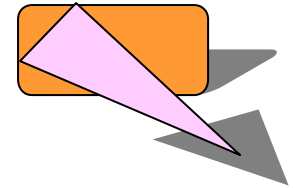
オブジェクト指向言語  
(Smalltalk, C++, Java,  
CLOS,C#...)

# GoF(Gang of Four)のデザインパターン

(Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides)

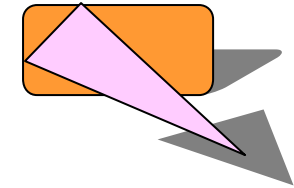
『Design Patterns 』Addison Wesley, 1994

(Elements of Reusable Object-Oriented Software)

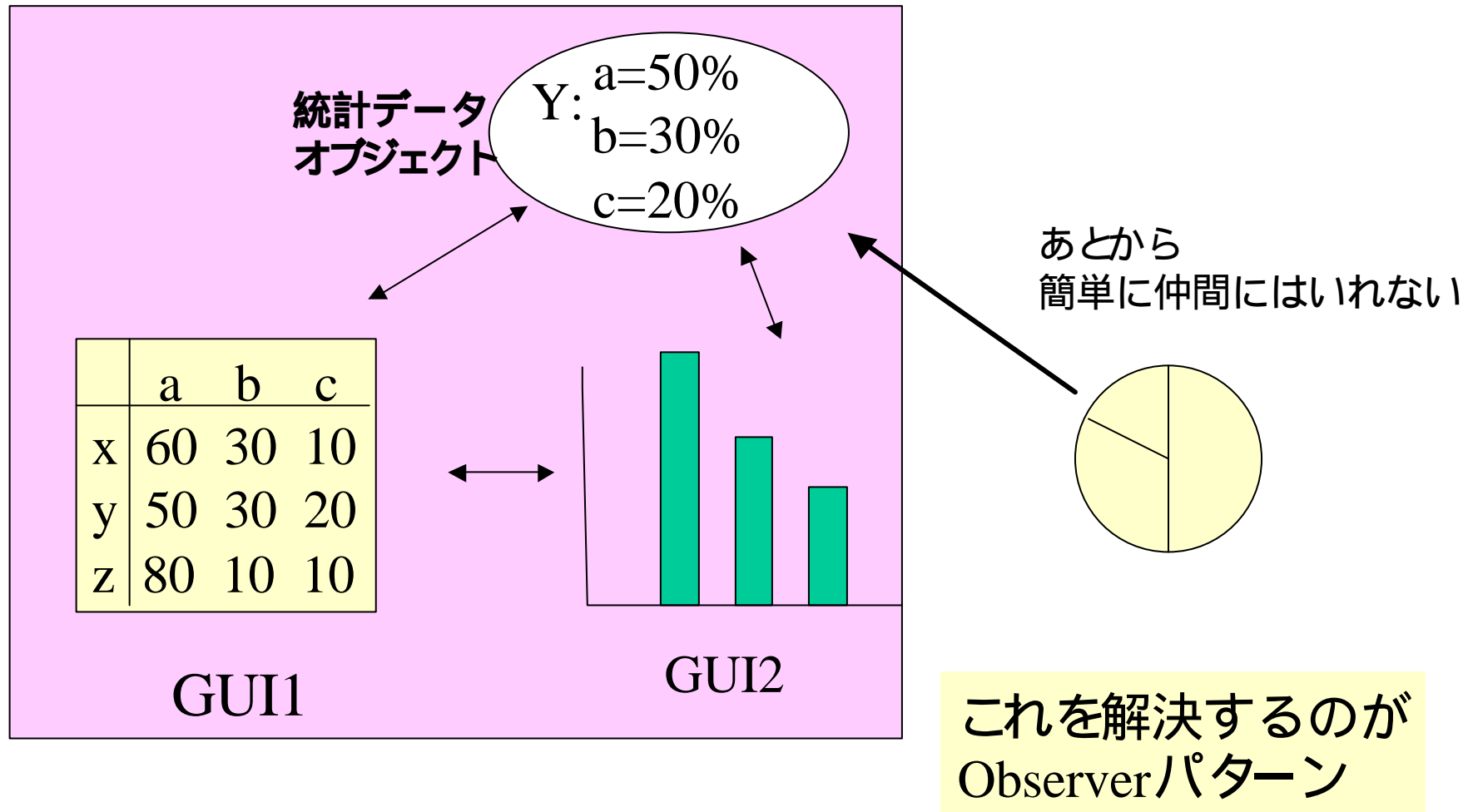


- パターンがソフトウェアの世界にブレークするきっかけ
  - すぐ使える .OO開発者に訴えるものがある
- 23のデザインパターンカタログ .
  - 生成パターン , 構造のパターン , 振る舞いのパターンに分類
  - 全パターンを , 決まった形式で記述している .
  - GoF自身が長年フレームワーク開発で直面し , 解決し , 何度も利用した , 「こなれた」実績あるパターンを提案
  - 23のパターン概要を諳んじ , 開発に利用 . 開発者間のコミュニケーションにパターン名が使われるメリットが得られている .

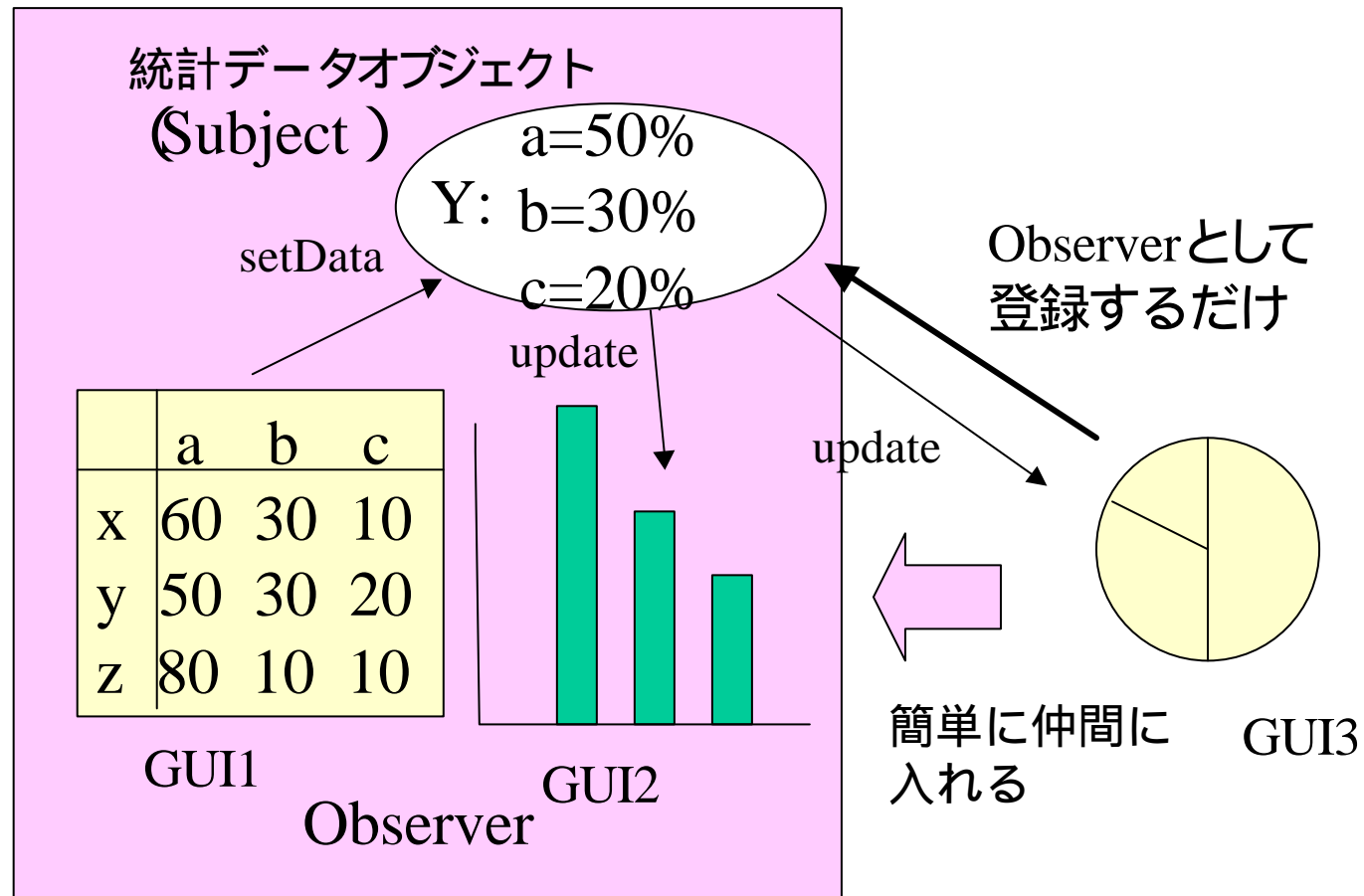
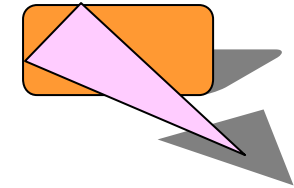
# GoFデザインパターンの例



## Observerデザインパターン(1/3)



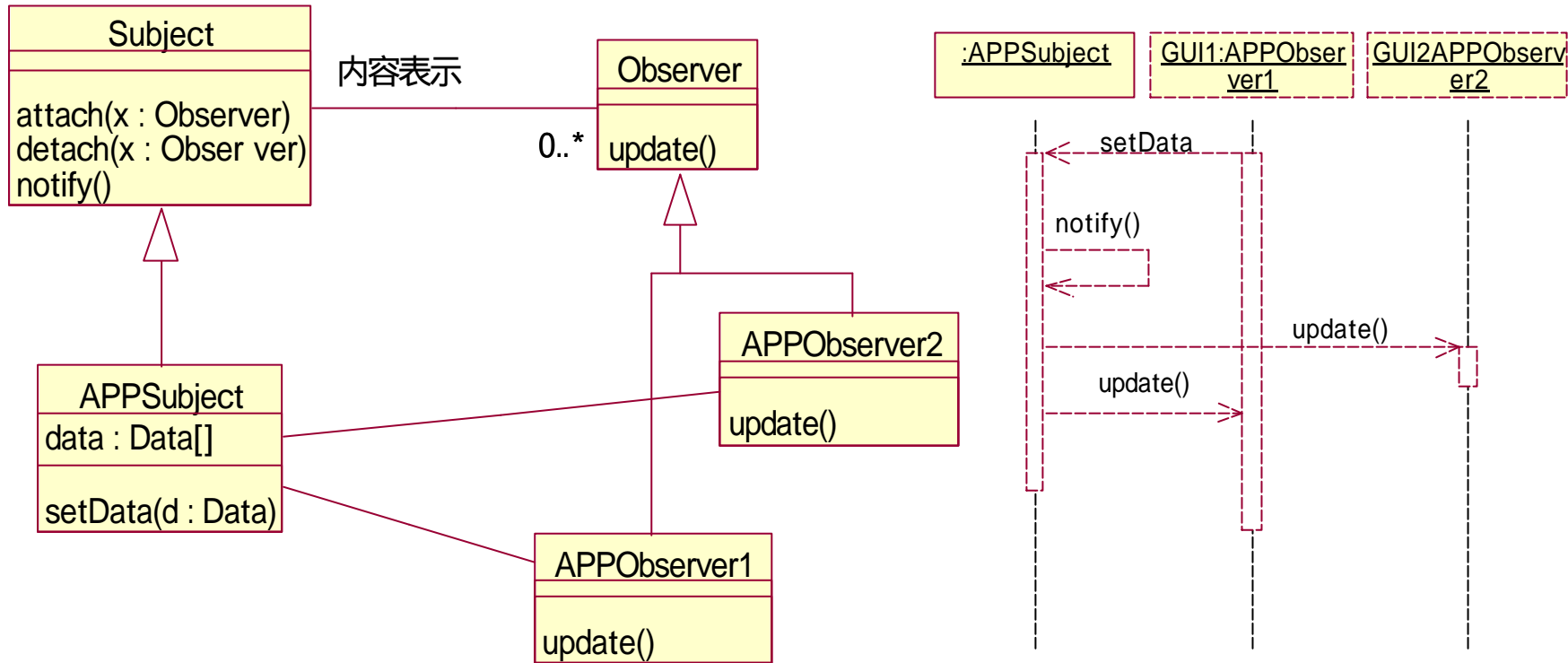
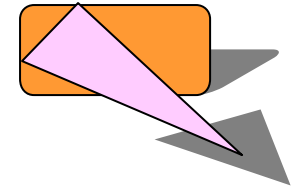
# Observerデザインパターン(2/3)



## Observerパターンのメリット

- GUI同士が直接やりとりせず相互依存性が低い
- 新たなGUIをここに加えるときは、Subjectに対して、「私を登録して」と伝えるだけで、既存クラスの実装は変えなくてよい

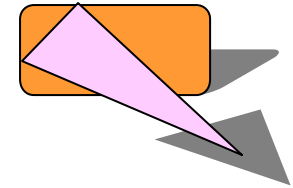
# Observerデザインパターン(3/3)



クラス構成

変更時のシーケンス

# GoFパターンの定義方法 (GoF Form)



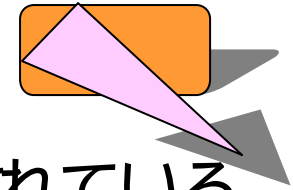
- 1)名前と分類
- 2)目的 . . .そのパターンが何をするのか
- 3)別名 (もしあれば)
- 4)動機 . . .問題を解くためのシナリオ
- 5)適用可能性 . . .どのような状況で使われうるのか
- 6)構造 . . .UMLなどを用いて ,登場クラスの関係説明
- 7)構成要素 . . .登場クラスの内容 ,責任について説明
- 8)協調関係 . . .登場クラス間のインタラクション説明
- 9)結果 . . .利点と欠点 .Mediatorクラス自身が大きな制御クラスになる危険
- 10)実装上の注意点
- 11)サンプルコード . . .C++,Java,Smalltalk
- 12)使用例 . . .過去にどこにどのように利用されたか
- 13)関連するパターン . . .似ているが目的が違うもの ,などなど .

23のパターンがすべてこの形式で記述されている

この他に ,Alexanderian, Coplien, Portlandなどformが有名



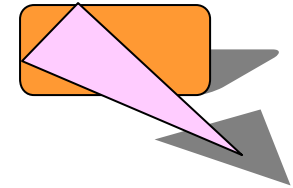
# ソフトウェアパターンの広がり



ソフトウェア開発のあらゆる場面にパターン技術が提案されている

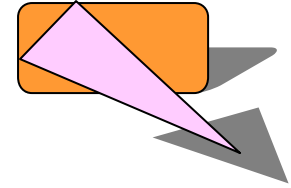
- アーキテクチャパターン(BuschmanらのPoSA, 他)
- アナリシスパターン(M. Fowler(1998)の著作が有名)
- デザインパターン
  - CORBAデザインパターン(T. Mobray 他, 1997)
  - J2EEパターン(D. Alur他,2001)
  - Network構築用(D. Schmitt ,2002)
  - その他いっぱい
- プログラミングパターン(=イディオム)
- プロセスパターン(J. Coplien, S. Ambler,...)
- アンチパターン(W. Brownらのもの ,Javaアンチパターン..)

# パターン利用の落とし穴



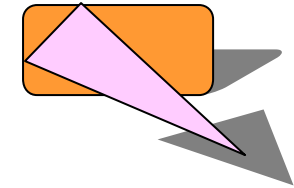
- 「データ部と表示部がある」 「それは当然Observerだ」
- 「入れ物の階層になってる」 「勿論Compositeを使うべき」
- 知ってるパターンを片っ端から使うのは逆効果
- パターンの中身, 性質, 利点欠点を把握して使うこと
- パターン利用の正しい手順
  - 1)問題を正確に捉え, 正しい解決方法を考える (自然なモデリング)
  - 2)問題解決に要求される項目は何か(force)を調べる
  - 3)forceに合致する, 使えるパターンがあるかどうかを調べ, 使う

# パターンの「フォース(Force)とは？」



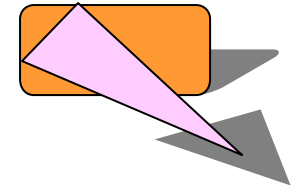
- そのパターンを利用することによって、利用者が得たいと考える利益。
  - 「パターン」は、単なる解法ではない。
  - 開発物の特性の優先順位、トレードオフを考えて、パターンを選択例)
    - あるクラスの拡張性を優先したい
    - 外乱に強い造りにしたい
    - パフォーマンスのよい実装にしたい
  - GoFのパターンに明記されていないが、パターン記述には必要な物
  - 語源は、パターンの父、Christopher Alexander
    - 建築物の構成要素パターンどうしがうまく力のバランスをとりながら美しく住みやすい建物が生まれる」

# ソフトウェアパターン技術の歴史



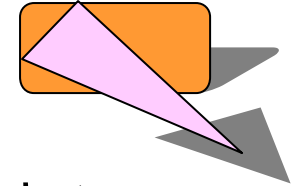
- 1979 C. Alexanderによる建築のためのパターンランゲージ
- 1987 OOPSLA87で ,K. Beckらがパターンランゲージのソフトウェアへの利用を提唱
- 1990 E. Gammaがチューリヒ大博士論文で「デザインパターン」の概念を記述(1991)
- OOPSLAで ,パターンワークショップ(91,92,93)  
4人組が出会ったり ,P. Coadによるパターン記述WS開催など
- 1995 Design Pattern本の発表(1994)  
PLoP開催開始(1994)  
(パターン関連書籍次々に発刊 )
- 2000 Japan PLoP結成(1999)
- 情報処理学会パターンWG設立(2003)

# Christopher Alexanderについて



- カリフォルニア大学建築科教授
- 建築設計のための「パターン言語」を提案
  - 著書『Pattern Language』(1976)
    - 建築のための「パターンランゲージ」
    - 地域, 街, 田園, 住宅, ライフサイクル, コミュニティ, 正面玄関, 庭の造り, 階段, 遊び場, 動物...など, 253のパターンからなる.
  - 著書『Timeless way of Building』(1979)
    - パターンの考え方, パターンランゲージの定義,
    - 人が生き生きと活気をもち, 美しく, 自然に生きられるためには, 無名の質を知る必要がある.
- 建築の世界では, 一つの流派らしい
  - オレゴン大学キャンパス
  - 入間市の私立東野高校のキャンパスデザイン
    - <http://www.plato-net.or.jp/higashino/>
  - 神奈川県真鶴町の「美の条例」
- 中埜博さん (まちづくりカンパニー) の活動
  - パターンランゲージを駆使した, まちづくり, 建築

# パターンランゲージとは？



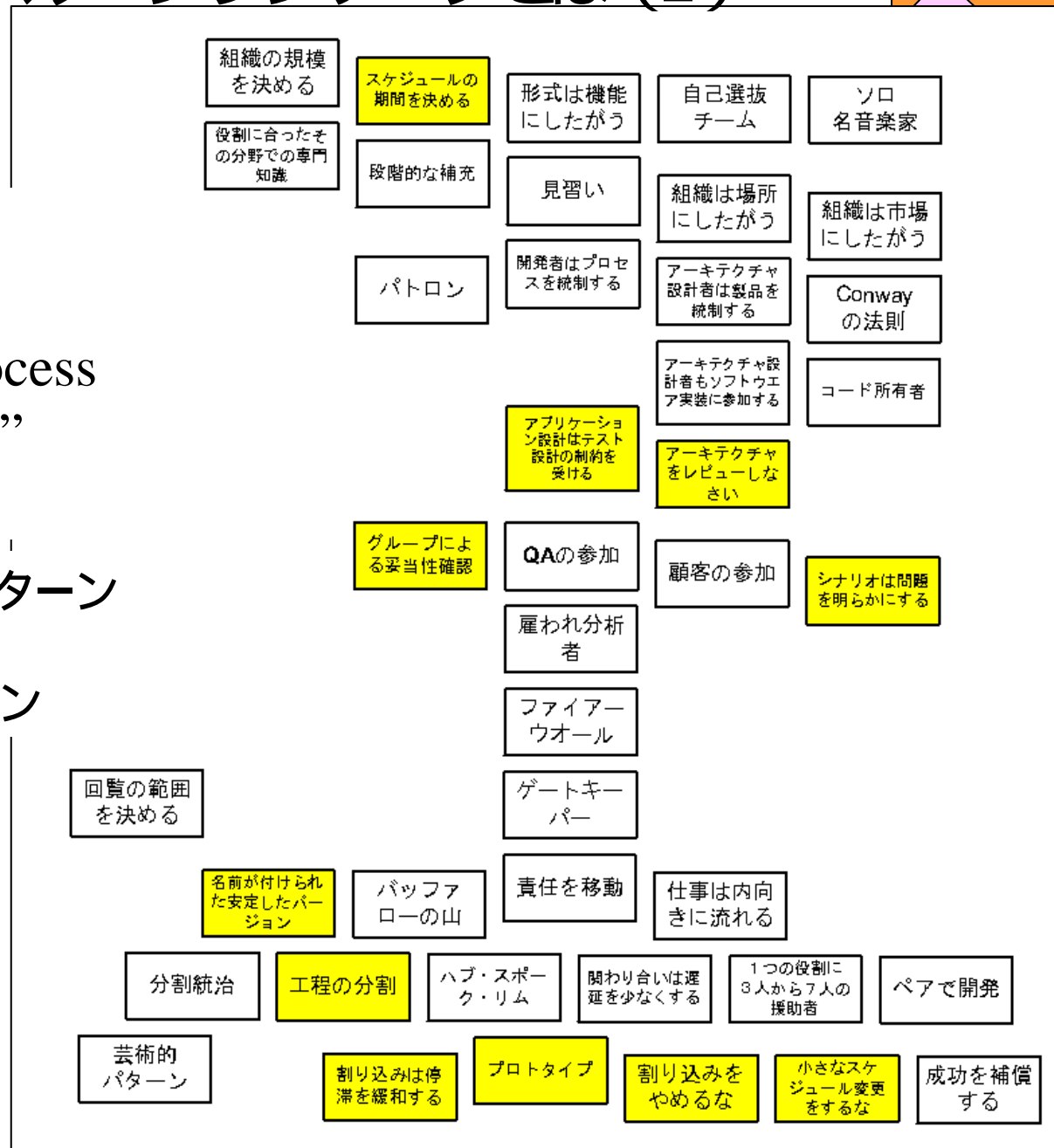
- ある目的のために用意された ,パターン群
  - パターン群の中のパターンを選択 駆使して目的をすべて達成する .
  - 建築家のChristopher Alexanderが「Timeless way of Building」(1979)の中で記述した概念 .
  - 個々のパターンが語彙となって ,言語を構成する
    - いわゆる ,文法ルールがあり ,構文解析可能な「ランゲージ」とは違う .
    - 言語を使うことにより ,意味のある解を構築する .
- パターンランゲージ例---PLoPDより
  - OO開発における ,Requirements-Analysis-Processパターンランゲージ
  - 汎用開発プロセスのためのパターンランゲージ
  - リアクティブシステムの能力を向上させるパターンランゲージ
  - パターン記述のためのパターンランゲージ
- パターンランゲージの広がり
  - あらゆる分野に対して ,パターンランゲージを作ろうとする動き
    - パターン評価 ,開発 ,テスト ,プロジェクト管理 ,旅程 ,イベント企画...
- PLoP活動と深いつながり

# パターンランゲージとは (2)

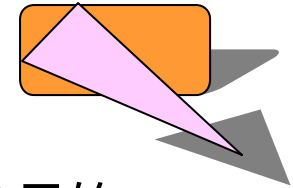
Jim Coplienの  
 “Generative  
 Development-Process  
 Pattern Language”  
 (現在も進化中)

□ : プロセスパターン

■ : 組織パターン



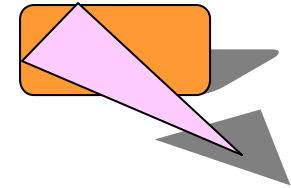
# PLoP(Pattern Languages of Programming)の活動



- より多くのパターンやパターンランゲージを集め ,洗練 ,公開する目的
- 1994年からワークショップを毎年開催
  - Pattern提案者とレビューにより ,提案パターンを議論し ,洗練させる . (否定的な批判は基本的に行わない)
  - PLoPワークショップについて
    - 8月または9月 ,シカゴ郊外のイリノイ大学敷地内 ,Allerton Houseで開催
    - 毎回100人程度の参加 (半数はアメリカから).
    - Big Namesの参加
    - 数トラックのパターンワークショップ以外に ,BoF ,ゲーム (互いによく知り合うため) ,キーノートスピーチ ,記念グッズ販売 ,ジャズ演奏 ,
  - 成果をホームページ(<http://www.hillside.net/>)で公開
  - 成果の中で優れたものは ,PLoPD(Pattern Languages of Program Design)として出版 .これまで4冊発行されている .
- メーリングリストによる普段からのパターン議論
- 世界への影響
  - ChilliPLoP, EuroPLoP, KoalaPLoP, Sugarloaf PLoP, JPLoP などなど活動 .

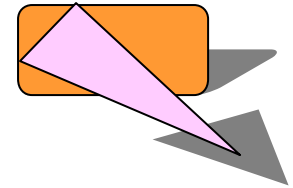


# Japan PLoP(=JPLoP)の活動(1999-2001)

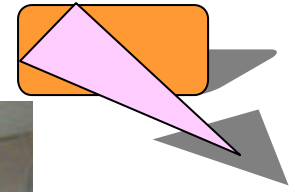


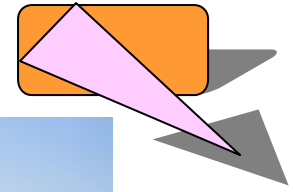
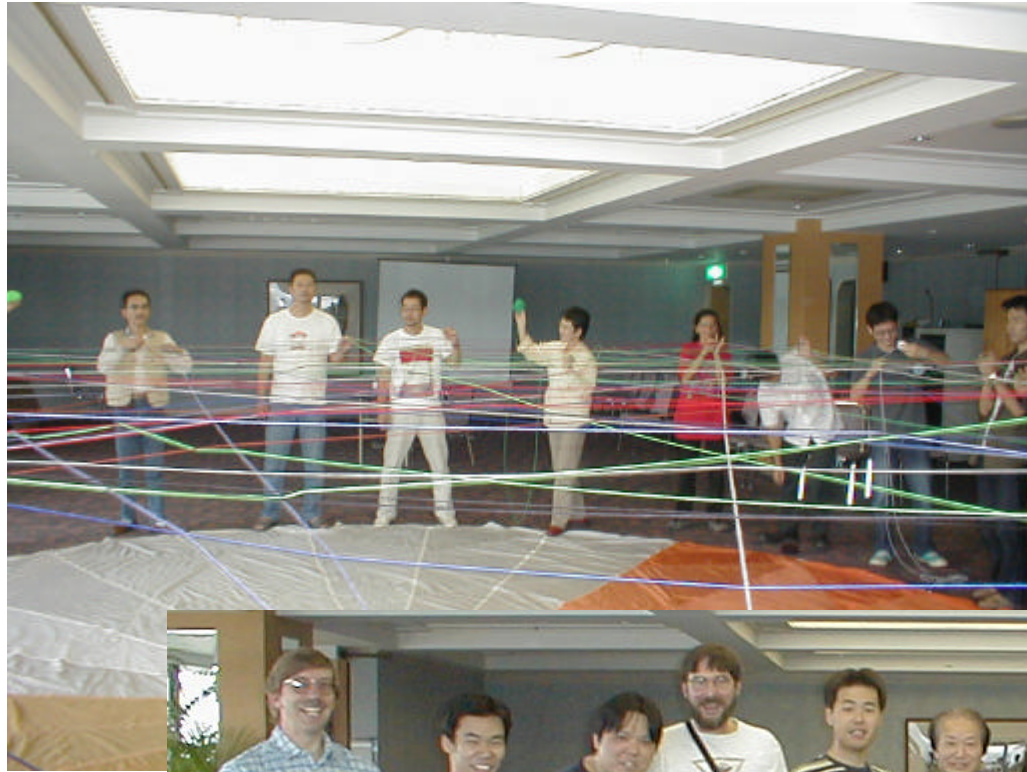
- PLoPの活動に刺激され , 設立 .
  - 現在会員数は , 200人程度 . 日本のOO著名人も多く参加
- ソフトウェアパターンの概念を世の中に普及させたい .
- 主な活動
  - ワークショップ (まる一日) .
    - パターンレビュー , ライターズワークショップ , ライティングワークショップ , グループポエトリ , 基調講演 , ゲーム
  - 隔月のパターン研究会
    - パターンレビュー
    - パターン関連のチュートリアルや最新情報紹介
  - PLoPDのパターン翻訳とホームページでの紹介
  - パターンに関するメーリングリスト
  - 2001年11月に , 国際PLoPである , MensorePLoPを開催

# MensorePLoPの紹介



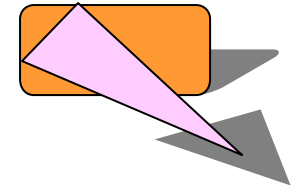
- 2001/11/9-14 , 沖縄本島ムーンビーチ
- 23名参加(J. Coplien, N. Harrison, JPLoPメンバ , 琉球大学の先生と学生 , その他企業の方々 )
- パターンに関するModeration , 講演 , ゲーム , 小旅行
- Moderation
  - 2つのチームに分かれ , 10人ずつで議論 .
  - 一つのパターンに関し , 約 1時間 (資料は事前にWebで公開済み)
- ゲーム : 互いに親しむためのもの .
  - パターンはコミュニティである .
  - 議論の疲れを癒す





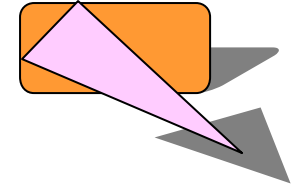
All Rights Reserved: Takeshi Inoue, 2003

# パターンとパターンランゲージ



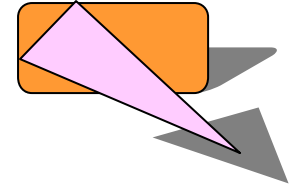
- 個々の問題に対処するのがパターン
  - 単なるパターン集がパターンランゲージとはならない
- パターンを語彙として, 包括的に解法を作りだすのがパターンランゲージ
  - 個々のパターンが互いに力の均衡を保ちながら協力して解を作りだす.
  - 個々の要求仕様を最大限満たし, 全体として機能性の高いシステム
  - 個々の問題を解決しながらプロジェクトを成功に導く
    - Agile Process ,eXtreme Programmingなどはパターンランゲージの視点から捉えられる.
  - 利用しながらパターンランゲージ自身も成長しつづける

# パターン関連の活動, 課題



- パターンは「コミュニティ」という認識を普及させる
  - 多くの経験, 知識がよりよいパターンを発見, 形成する.
- パターンを発見, 記述, 洗練, 普及させること
  - どうすればよりよい解法を普及できるか
  - 多種多様なパターンを把握しきれない
    - Pattern Almanac200(L. Rising)の試み
    - すべて有用なパターンなのか?
- パターン技術はパターンランゲージに向かうべきか?
  - 現状はパターンランゲージ以前に, 「個々のパターンを発見する段階」というのは本当か?
  - パターンランゲージは本当にソフトウェア開発を救えるのか?
    - Alexanderの言っていることはすべて正しいのか?
- パターン技術はもっと可能性があるのではないか?
  - 個々の問題を, 個々のパターンで解決していて十分か?
  - それとも, 十分すぎるくらい頑張ってしまった, もうない?

# まとめ



- 「パターン」という言葉がソフトウェアの世界に根付いた
  - 新たなパターンが提案され続けている
- 課題は多い
  - 新しいパターンは認知度も低く、整理統合されていない。
  - 誤った認識の広がり
  - 「パターン」概念そのものの掘り下げ
  - 真に役立つパターン、パターンランゲージとは
- パターンWGの設立は、パターンの課題掘り下げと、普及に貢献させるべき。
  - 多くの参加者のコミュニティ意識から解が生まれ、世界を変えるはず