



JANOG31 Meeting BoF 記録

デザインパターンを知っていますか？

- ・ 日時: 2013/1/25 18:50 ~ 20:15
- ・ 場所: 東京ミッドタウンホール&カンファレンス ホールA
- ・ 主催:
 - ・ 佐々木 健 (ソフィア総合研究所株式会社)
 - ・ 河野 美也 (シスコシステムズ合同会社)
- ・ URL: <http://www.janog.gr.jp/meeting/janog31/program/design.html>
- ・



事前公開アブストラクト

デザインパターンを知っていますか？

人の営みにはなんらかのパターンが存在します。そのパターンを抽出して良いパターンを見つけ活用することにより、より良い物を作り出すことができます。このアイデアは元々は建築の分野で生み出されたものですが、ソフトウェア工学の分野でも活用されています。

ネットワークも人によって作り出されたものであり、なんらかのパターンがあるはず。成功したものには良いパターンが、失敗したものには悪いパターンがあります。それを知ることによって、より良いネットワークを作れるようになるのではないでしょうか。自然の自己相似形や、音楽のハーモニーのように、美しいもの、感動するものなど、心を打つものには、たいていはある一定のパターンが存在します。一緒にそれを見つけてみませんか。

当日のオツマミは以下を考えています。オツマミの持ち込みは大歓迎です。BoFですし、みんなで気楽にガヤガヤ話をしましょう。

- デザインパターンって何？
- どうやって記述する？UMLみたいなツールはネットワークにない？
- たとえば具体的にどんなのが素敵なパターン？
- どんなのが悪いパターン？どこが悪い？
- 他の分野でのデザインパターン
- どう活用する？



BoF のファシリテーター、配置

- ・ 会場の左前方にホワイトボードを用意。
- ・ ワークショップ用に付箋紙、ペンを準備。
(協賛各社様から提供いただきました)。
- ・ プロジェクターなし。配布資料なし。マイクなし。
- ・ 参加者とモデレーターとの間の距離が近くなるように、ホワイトボードの回りに半円形に椅子を並べる。



参加者層

- ・ レイヤや業種等に特に偏りはなし。
- ・ デザインパターンに興味がある人、モデレータ(河野さん)のファン、他のBoFに比べてグダグダ感を楽しめそう、というような方が参加者ではないかと思われる。
- ・ 人数は30～40人



当日のお題目と進行

事前のアジェンダとは少し変更し、以下の順番で進行。

0. 事前前説明、自己紹介等
1. デザインパターンって何？
2. 何が嬉しいか？
3. アンチパターン
4. どうやって記述する？抽出する？
5. 他の分野のデザインパターン
6. 何のパターンを作ると嬉しい？
7. 実際にパターン抽出してみよう
8. まとめ

時間切れのため、7番は実施できず。



お題目 1: デザインパターンって何？

人の営みにはなんらかのパターンが存在する。

最初にデザインパターンという概念を使い始めたのは建築の分野。

アレグザンダーが始めた。

良いパターンを抽出し、それを元にすれば良い設計ができるはず。

設計は、物を作り出すというすべての行為で行なわれること。

ソフトウェア工学の分野でもこの考え方は発展し活用されている。

アレグザンダーは、都市は階層的に構成されるツリー構造ではなく、

様々な要素が絡み合って形成されるセミラチス構造である、と言っている。

インターネットもまさにセミラチス構造になっている。

武道の達人は型を知っている。それに近い。

武道は型を学び、守り、破り、離れる、ことにより進歩している。

型を知ることによって達人も超えられるはず。



ディスカッション

デザインパターンの例として、AWSクラウドデザインパターンがある。ああいうものを作りたい、というイメージか？

--> あれは素晴らしい。エンジニアが使いやすい形でパターン化できている。

--> 実はそれは、お題目5、で少し説明しようかと思っていた。

うちの会社でも運用のパターンを作っている。ただ実装が実際には難しい。

--> パターン化する領域はフォーカスする必要がある。

--> そこは、お題目6、で説明しようかと。

ネットワークにパターン化しにくい理由があるのではないかな？

--> ソフトウェアと違って制約条件が多い。

----> 制約もパターン化できるのでは？

--> プロトコル、構造(リング、スター、メッシュ)、OSIなど、パターンはないわけではない。

SDNやクラウドにより設計の自由度が上がったことで

デザインパターンが注目されてきたのではないかな？

--> そうかもしれない。ただそうとも言えない。

--> 河野さんはずっと言っているし。ただ河野さんの言葉は難しいのでw。たとえば MTU の問題等は、ずっと繰り返されてきている。



お題目 2: 何が嬉しいか？

デザインパターンは「言語」でもある。
抽象化して名前を付けるとコミュニケーションがしやすくなる。

デザインパターンは問題発見、解決の支援をしてくれる。



お題目 3: アンチパターン

不適切な解決策や失敗を分類したもの。

単なる悪癖や悪習とは違って、リファクタリングにより直せる。

これを知ることによって失敗を防げる。失敗からのリカバーもできる。



お題目 4: どうやって記述する？抽出する？

AWSクラウドパターンでは以下のように抽出している。

- 解決したい課題
- 解決方法/パターンの説明
- 実装
- 利点
- 注意点
- その他

これはエンジニアにとって使いやすい。

ただ、以下の3項目で抽出するのが元々のアイデア。

Context (どのような状況で)

Problem (どのような問題が生じやすく)

Solution (どう解決すれば良いのか)

デザインパターンは問題発見、解決するツールとなっている。

やり方、だけでなく、ありがちな問題、解決方法、も記述される、というのが、マニュアルや手順書、との違い 10



お題目 5: 他の分野のデザインパターン

- 「オブジェクト指向における再利用のためのデザインパターン」～ GoF による原典。
- AWS クラウドデザインパターン
- 波田野さんがまとめた監視パターン

他にも様々なパターン化はされている



ディスカッション

会社の内部文書としてパターン集のようなものはある。

これは会社のノウハウで飯の種になっている。

おいそれとは共有できないんじゃないか？

--> ソフトウェアも昔はそうだった。

--> ただ今はオープンソース全盛。他で使われている実績が信頼を生む。

--> 今は内部で持っているノウハウもオープンにしたほうが良くなるはず。

--> 誰かが良いものを作って公開すれば変わるのではないか？



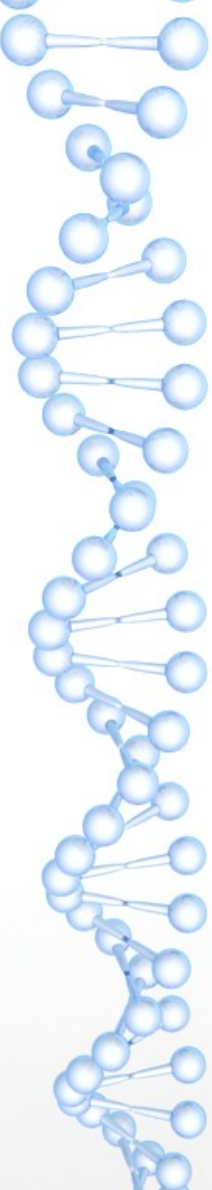
お題目 6: 何のパターンを作ると嬉しい?

ワークショップを実施。

各人が、パターン抽出したいもの、を付箋に書いて
ホワイトボードに貼っていく。

それを分類。

國武さん、田島 (taji) さんがデータ化してくれました。
(次ページ以降)

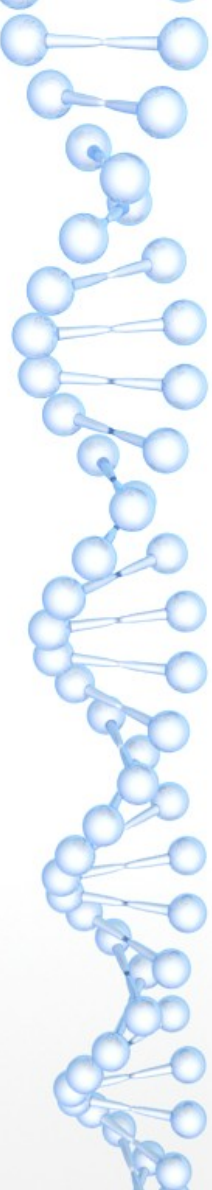


パターン抽出したいもの その 1

「ネットワーク設計」

- * トラフィック分散時のアンチパターン(ECMP)
- * POP or iDCの増やし方
- * NW構築時のパターン
- * NWトポロジー
- * DC間を接続したインタークラウドの設計パターン
- * 品質
- * 長距離バックボーン回線でのLAG
- * 輻輳・迂回設計のパターン
 - * mazのプログラムそのままだけ
- * トラフィック規模の設計方法
 - * 1万人規模ISP
 - * 10万人
 - * 100万人くらい

- * ネットワークの階層構成
- * スケーラビリティ実現パターン
- * 冗長化のパターン分類
- * 止まらないネットワークの作り方
- * 冗長
- * DC内の物理ファブリックの構成
- * 検証環境のIPアドレスアドレッシング
 - * 組み替えしているとすごくぐちゃぐちゃに
- * アドレスの割り当て
- * 仮想ネットワークと物理ネットワークのマッピングパターン
- * 企業向けシステムの作り方

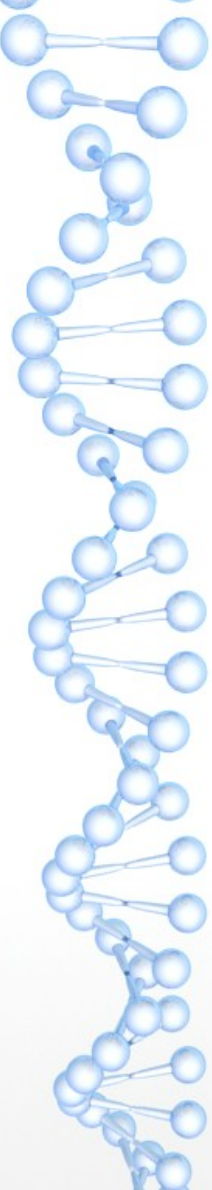


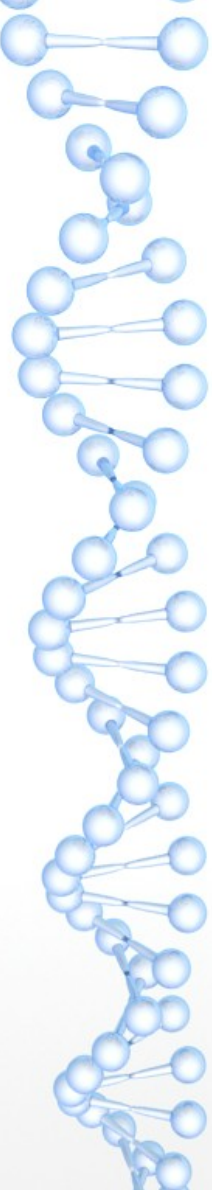
パターン抽出したいもの その2 「プロトコル」

- * ダメなプロトコルの見分け方 (STP)
- * ポート番号埋め込まれプロトコル (ex. FTP)
- * プロトコルのアンチパターン
 - * IPv6 は
 - * 4byteAS は
 - * MTU/MSS は
- * エラーコードの統一
- * プロトコルのシンプルさを求めた時になにをもってシンプルとするか？
- * 標準化の進め方

パターン抽出したいもの その3

「トラブル対応」

- 
- * お客様への謝り方
 - * 障害に対する顧客説明や対策立案のパターン
 - * トラフィック傾向からの障害解析
 - * <BADパターン> 増えすぎて管理できなくなったVM群
 - * テーブル溢れ
 - * Ex. ARP, NAT, 経路
 - * トラフィックとかが単一のところでサチる
 - * Ex. ラックスイッチ
 - * フロントが軽くてバックエンドがささる
 - * Ex. Webサービス
 - * 警察にログ出してって言われた時のパターン
 - * トラフィックが 100倍になった時の対象方法？
 - * なにを Traffic異常とするか
 - * オペミスのパターン
 - * 障害発生時でもお客様に文句いわれないパターン
 - * 嫁さんとのケンカのパターン
 - * 半落ちルータの Isolation or Troubleshooting
 - * 障害発生時の対処（迂回方法）とか
 - * 障害発生パターン
 - * 一週間問題、解決できない時にどうするか？（オペレーションリサーチか？）
 - * アラートアグリゲーション
 - * たくさんのアラートメールをまとめる
 - * 要求のはねのけ方パターン
 - * こういうお客様ならこうする。
 - * NWトラブルのデジャブのパターン
 - * NWトラブルのジャメブのパターン



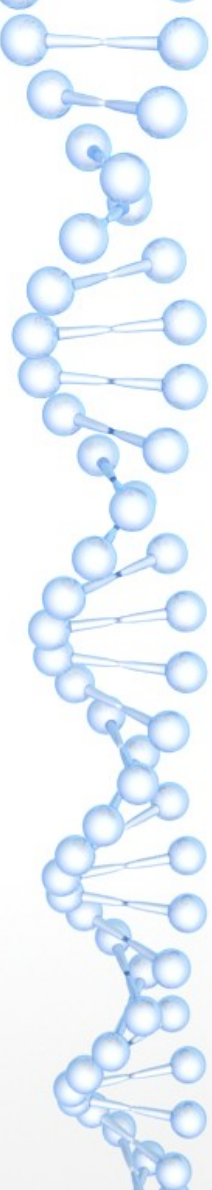
パターン抽出したいもの その4 「運用」

- * 少人数でも運用が楽にできるパターン
- * コンフィグバージョンング 設定ファイルはバージョン管理ツールに入れる
- * テストパターン (擬似障害とか)
- * チケットシステムとDBの作り方
- * ネットワークの移設
- * 幸せな運用体制パターン
- * リソース監視とその利用
- * 適切な監視システムパターン
- * ユーザの動向(使い方)
- * 東日本大震災クラスのシミュレーションを毎年やる方法？
- * 設定のパターン
- * メンテナンスの進行管理



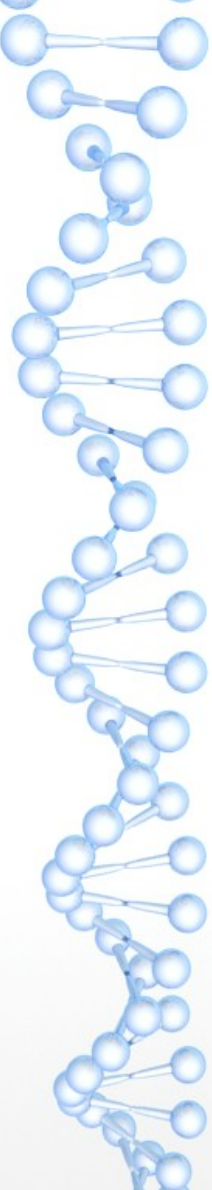
パターン抽出したいもの その5 「組織」

- * エンジニアが元気な会社のパターン
- * 一連の上司の絵？ 資料の好み？
- * JANOG Meeting に参加するために上司を説得する方法
- * 海外出張に行くための上司のおとしかた
- * 活発なコミュニティ活動のパターン
- * 女性技術者を増やすパターン



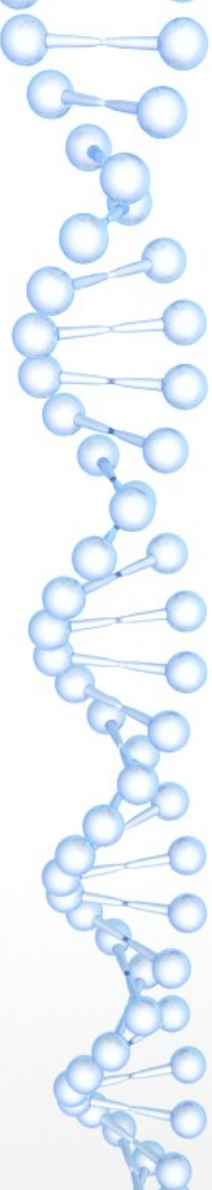
パターン抽出したいもの その6 「抽象化」

- * SDN/ クラウド上の NW の
Abstraction(L2/L3 とか) パターン
- * 違いに気づく
- * 勘に頼る



パターン抽出したいもの その7 「マーケティング」

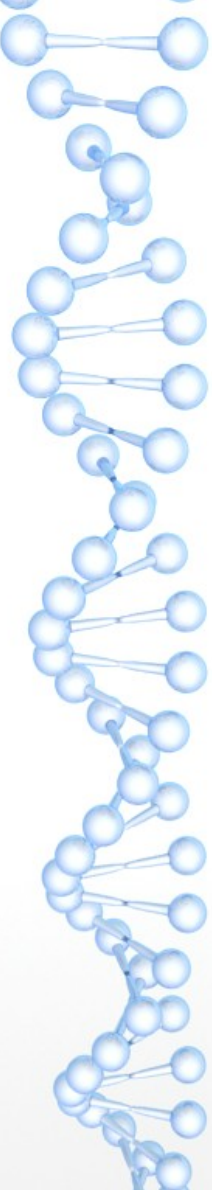
- * buzzword 発見パターン
- * buzzword の定義のされ方



パターン抽出したいもの その 8

「その他」

- * 社内のファイルサーバ 整理パターン
- * PC、タブレット、ケータイ等 エンド端末の進化と分類
- * コスト削減 対処方法
- * うけるセッションのパターン
- * JANOG のセッションの立て方
- * 河野さんの思考パターン



お題目 7: 実際にパターン抽出してみよう

※ ワークショップの予定だったが時間切れのため実施せず。



お題目 8: まとめ

モデレータは楽しい時間を過ごせた。なにかあったら、メールをくれたり、つぶやいたりしてくださいまし。

Twitter のハッシュタグは以下にしました。

#janog_dp

これからどうするか、はちょっと考えてみます。



参考資料 (関連 URL)

井庭先生の講義資料:

「パターンパターン・ランゲージによる経験のマイニングと共有」

<http://www.slideshare.net/takashiiiba/ss-12990656>

「オブジェクト指向プログラミングのためのデザインパターン」

<http://www.techscore.com/tech/DesignPattern/index.html/>

デザインパターン_(ソフトウェア) - wikipedia

http://ja.wikipedia.org/wiki/%E3%83%87%E3%82%B6%E3%82%A4%E3%83%B3%E3%83%91%E3%82%BF%E3%83%BC%E3%83%B3_%28%E3%82%BD%E3%83%95%E3%83%88%E3%82%A6%E3%82%A7%E3%82%A2%29

アンチパターン - wikipedia

<http://ja.wikipedia.org/wiki/%E3%82%A2%E3%83%B3%E3%83%81%E3%83%91%E3%82%BF%E3%83%BC%E3%83%B3>

クリストファー・アレグザンダ - wikipedia

<http://ja.wikipedia.org/wiki/%E3%82%AF%E3%83%AA%E3%82%B9%E3%83%88%E3%83%95%E3%82%A1%E3%83%BC%E3%83%BB%E3%82%A2%E3%83%AC%E3%82%B0%E3%82%B6%E3%83%B3%E3%83%80%E3%83%BC>

AWSクラウドデザインパターン

<http://www.publickey1.jp/blog/12/amazon45.html>

<http://aws.cloudesignpattern.org/index.php/%E3%83%A1%E3%82%A4%E3%83%B3%E3%83%9A%E3%83%BC%E3%82%B8>

波田野さんがまとめた障害監視のデザインパターン

<http://rsh.csh.sh/kasf/design-pattern.html>