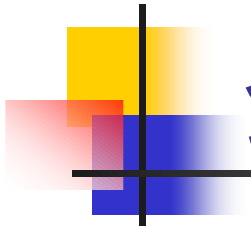
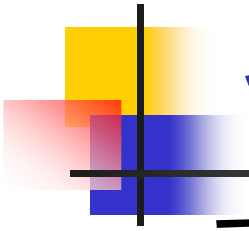
EA(エンタープライズアーキテクチャ)と SOA(サービス指向アーキテクチャ)



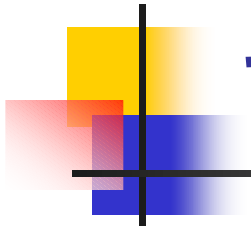
発表の流れ

- 背景となる技術的要素について
  - アーキテクチャ
  - 階層構造
  - コンピュータシステムの本質
- エンタープライズアーキテクチャについて
  - 概要
  - 行政における取りくみについて
- サービス指向アーキテクチャについて
  - 概要



背景となる技術的要素について

- アーキテクチャ
- 階層構造
- コンピュータシステムの本質



アーキテクチャとは？

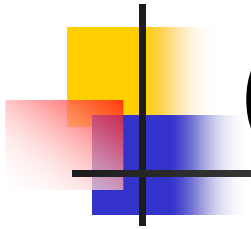
- 辞書的には「建築（学）」や「構造」
- ハードウェア等の「基本設計」、という意味で使われる。
- 様々な文脈で様々な物を指す。

アーキテクチャの例 (CPUアーキテクチャ)

- データ幅(bit数)
- CISC、RISC
- データと命令を一緒に扱う？分離する？
- レジスタの数
- 命令の対称性
- キャッシュ
- I/O

- ※「命令セットの仕様」のことのみを指す場合もある
  - インテルアーキテクチャ、MIPSアーキテクチャなど

アーキテクチャの例 (ハードウェアアーキテクチャ)

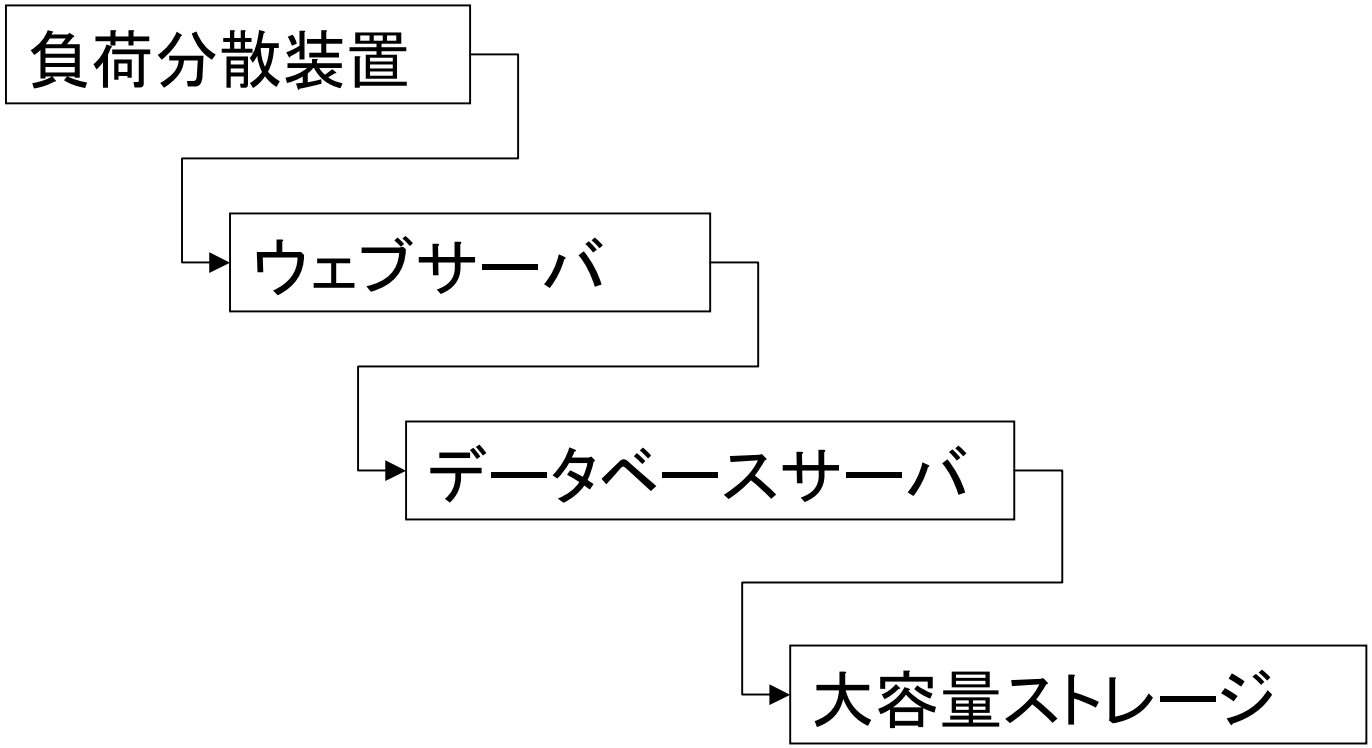


- CPUには何を使う？
- バスに何を使う？
  - バス毎にもアーキテクチャがある
- メモリ
- ストレージ(ハードディスク)
- 外部インターフェイス
- グラフィックカード

- ※典型的なアーキテクチャには名前が付いている
  - PC互換機アーキテクチャ、SPARC システム

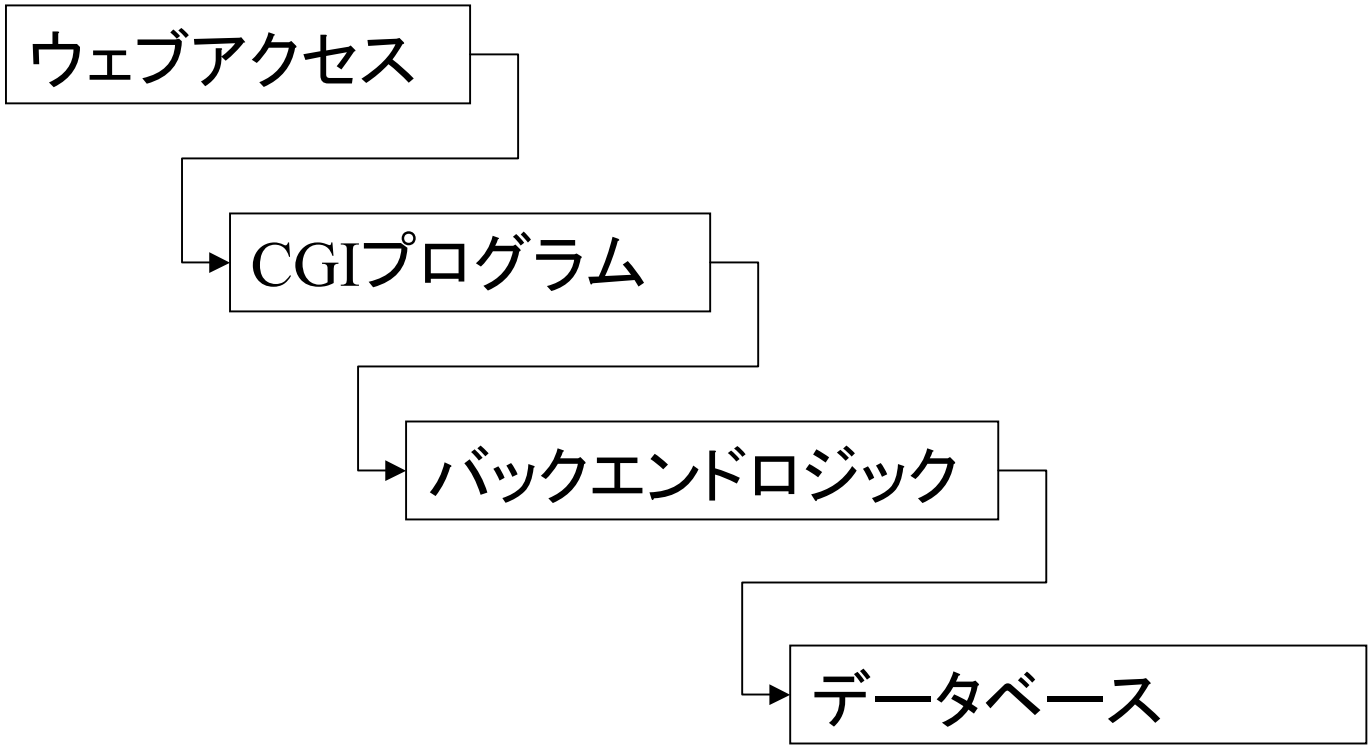
アーキテクチャの例 (システムアーキテクチャ)

- 典型的なウェブデータベースシステム



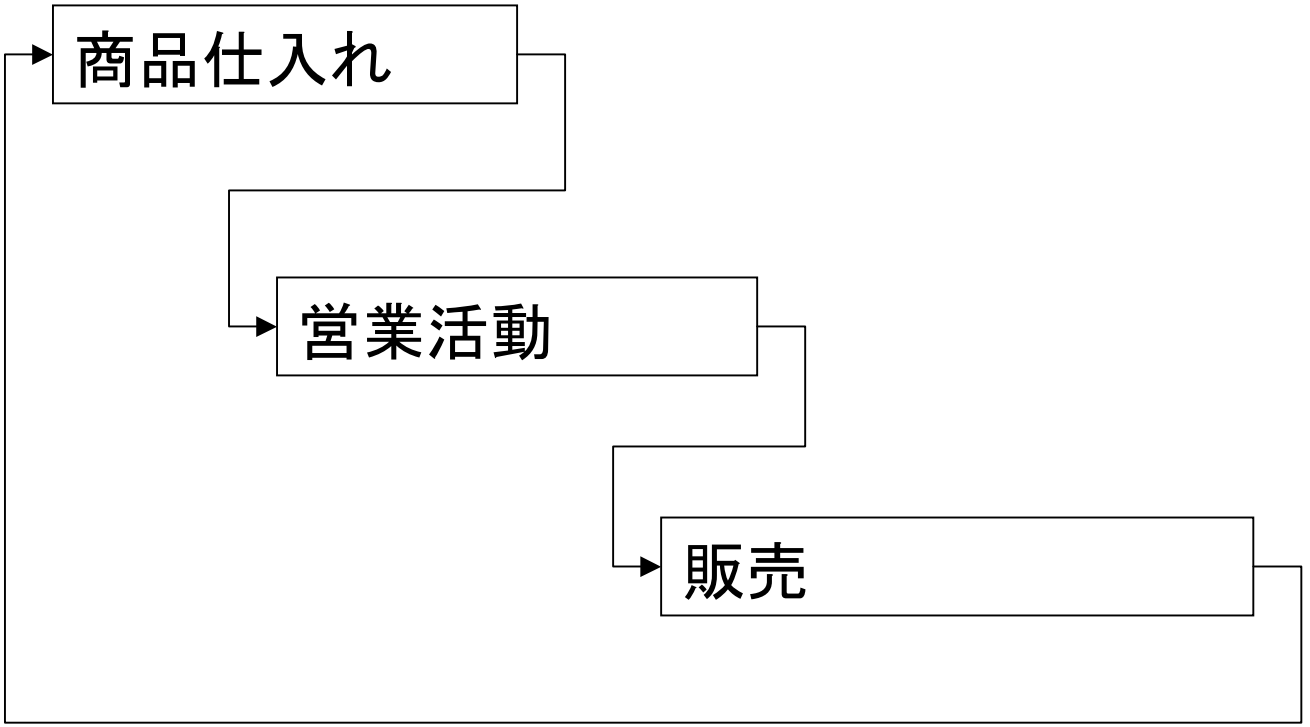
アーキテクチャの例 (ソフトウェアアーキテクチャ)

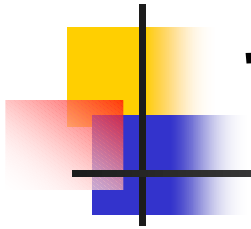
- 典型的なウェブデータベースシステム



アーキテクチャの例 (業務アーキテクチャ)

- 流通業モデル

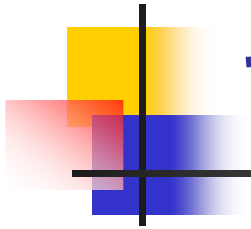




アーキテクチャの例(その他)

- クライアント・サーバ・アーキテクチャ
- .NETアーキテクチャ
- TCP/IPアーキテクチャ

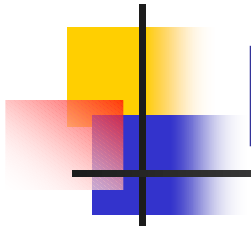
- 企業内のテクノロジー標準の意味で使う
- アプリケーションの共通基本設計の意味で使う
- デザインパターンの意味で使う



アーキテクチャのまとめ

- アーキテクチャとは、「システムの組織的な構造」という意味で使われる。
- アーキテクチャ、という言葉は様々な文脈で様々な物について使用される。
- どの文脈で使っているかを正しく理解しないと思わぬ誤解を生むことになる。

  - ※他のコンピュータ用語でも同じような例がある。
    - ウェブ、サービス など



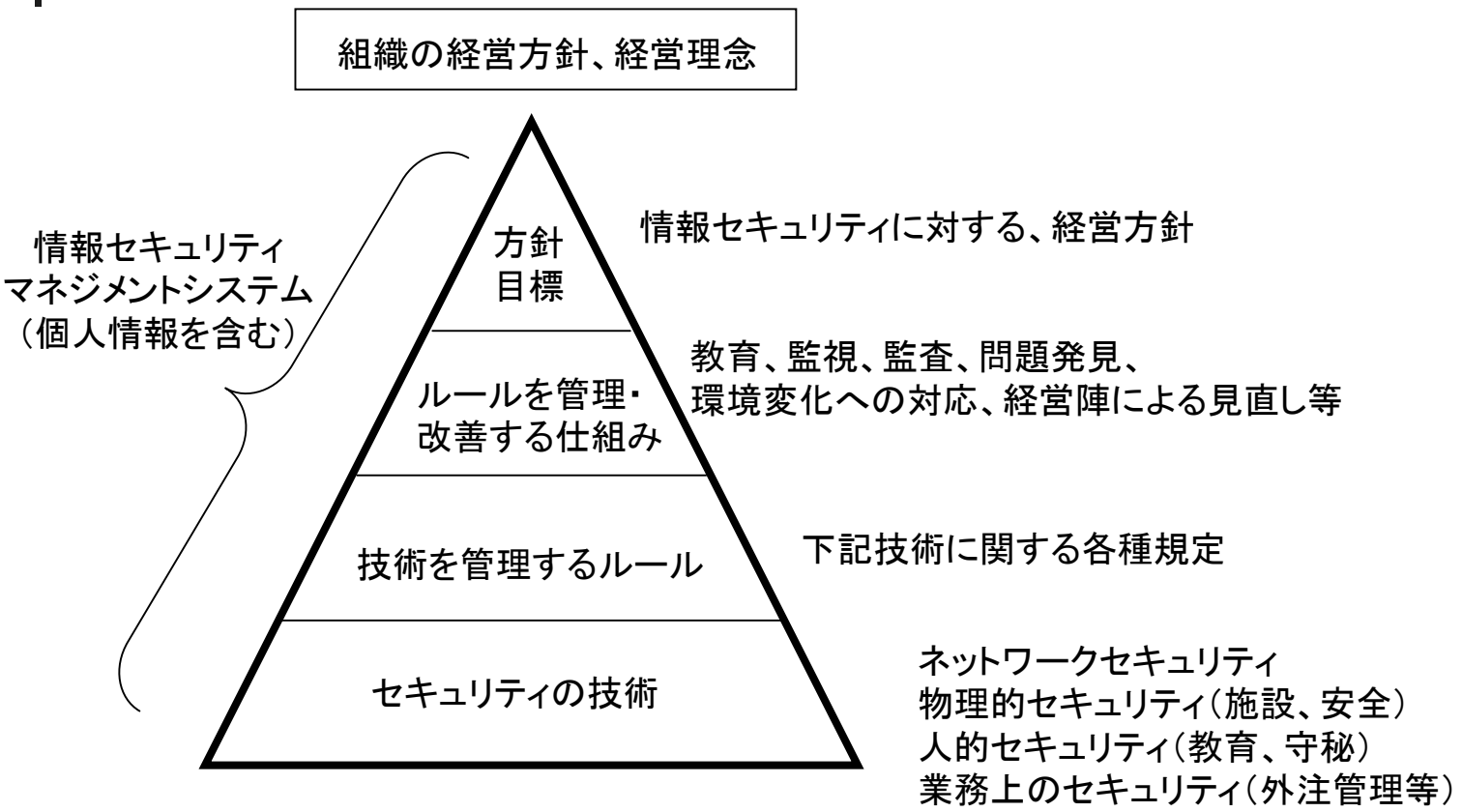
階層構造モデル

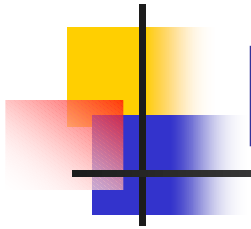
- システムを階層構造で捉えることにより構造的に考えることができるようになる。
- ネットワークにおけるOSIの7層モデル等で有効性が広く認知された。

階層構造の例
(ネットワークアーキテクチャ)

アプリケーション層
プレゼンテーション層
セッション層
トランスポート層
ネットワーク層
MAC層
物理層

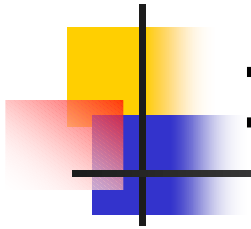
階層構造の例
(マネージメントシステム)





階層化のメリット

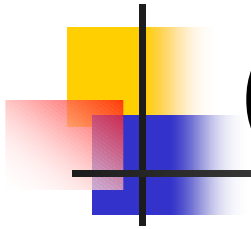
- 階層毎に扱う情報を限定できる。
- 階層毎に扱う処理を限定できる。
  - →実装が容易になる
- 全体では高度な処理が可能
- 各レイヤを自由に交換可能。
  - インターネットにおいては、アプリケーションのプロトコルを変えるだけで様々な処理を実現
  - インターネットでは1、2層目を高速化することにより20年以上も同じプロトコルを利用できている。



コンピュータシステムの本質

- コンピュータシステムとは何物なのか？
- コンピュータシステム発達について

コンピュータシステムとは何か(論理的背景)？

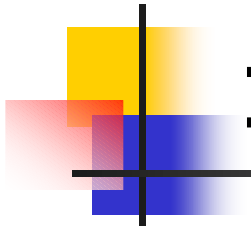


- 数学(論理学)から誕生したもの
- アルゴリズムを実行する情報機械
- 何かをインプットし、何かをアウトプットする
- 学校でやった「関数」のイメージに近い

- 現在のシステムは、そのような生の姿を隠している場合が多い。
  - 見方によっては、この世の万物すべてを情報機械ととらえることも実はできる。

コンピュータシステムとは何か(技術的背景)

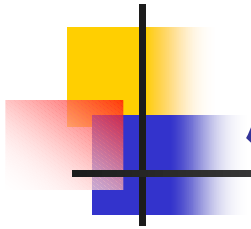
- 論理学の成果を、数学、物理学(量子力学、電気学)、工学(通信工学、生産技術)、社会学(教育、認知心理学)、などの様々な技術や学問により実装したもの。
- 幅広い技術から成りたっている。



コンピュータシステムの限界

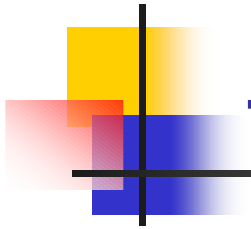
- 数学的限界：計算できないものは実行不能
- 物理学的限界：光の速さは超えられない
- 社会学的限界：人が使う部分がボトルネックになる
- リソース的限界：資源、時間は有限

- 万能ではない。
- 万能ではないことを認識した上で利用する。



歴史的背景

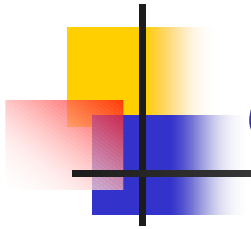
- コンピュータの捉え方は時代ごとに変わってきた。
- 利用ターゲット、開発手法などにより捉え方が決定されてきた。
- 人間は近視眼的に物を見る傾向がある。
  - 道具が思考を限定してしまう



手続型言語の時代

- Fortran、COBOLなどの手続型言語が開発の主力となっていた時代。
- 情報の処理手順を記述する。
- 流されるデータが決まっていて、処理したいことも決まっている。
- 扱うデータは少なく、処理も単純

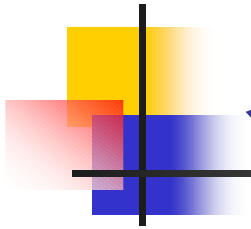
- コンピュータはデータを処理する機械である、と捉えられていた。



オブジェクト指向の時代 あるいはGUIの時代

- オブジェクトという概念が一般化した時代。Smalltalkなどが切り開き、JavaやVBなどの開発言語により概念が広まった。
- 情報をオブジェクトと捉え、処理はオブジェクトの性質として記述される。
- 実社会を比喩的に表現することができ、人間が感覚的に利用するためのインターフェースを実現。した。

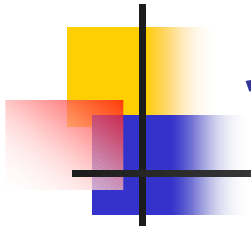
- コンピュータは文房具のような道具の高性能なもの、と捉えられていた。



ネットワークの時代

- インターネットが一般化し、HTMLや電子メールが使われるようになった時代。
- 情報は通信により得られるものと捉え、処理は通信の一部として記述される。
- 情報を流通させる基盤が共通化されることにより、コミュニケーションが加速。必要な情報へすぐに到達できることによる進化の加速。

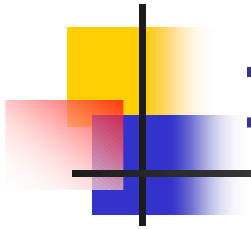
- コンピュータはコミュニケーションツール、情報収集ツールと捉えられている。



セマンティックWebの時代

- ネットワークによる流通される情報を、システムが収集加工し、人にとってより便利な情報を集めることができるた時代。
- システムどおしがネットワークを介在し積極的に情報交換を行なう。人間が仲介しないため劇的にスピードアップ。
- 人間が直接目に触れる1台のシステムの裏側では多数の別のシステムが協調し動作を行なう。

- コンピュータは生活のために必要な社会インフラとなる。



コンピュータシステムのまとめ

- 実社会におけるコンピュータシステムの適用範囲は広がっている。
- 限定された用途から、社会インフラに成長

- しかしコンピュータの限界がなくなるわけではない。
- 本質は変わらない。
  - 情報処理機械
  - インプットがあり、アウトプットがある

- 今後さらに重要になってくること
  - インプットできるオリジナル情報。
  - オリジナル情報が資産になる。

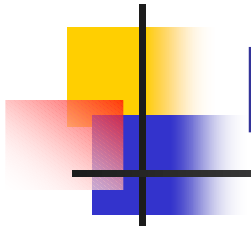
EA
エンタープライズアーキテクチャ

- ようやく本題

- EAとはなにか
- EAが必要な理由
- EAの具体例
  - 政府における取り組み

EAとは何か

- 組織、業務的視点からシステムを分析・再構築する手法。
- 組織（enterprise）の構造と機能をある一定の考え方・方法で包括的に体系化
- 全体と構成要素の相互関係を明確化
- 構造の背景にある基本理念・設計思想を含めて企業活動の全体最適を実現を指向
- あるべきモデル（To Be）を目指して、長期的かつ体系的に行われる



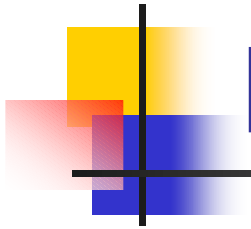
EAの誕生

- 1987年にジョン・A・ザックマン（John A. Zachman）がIBM System Journalに発表した「A Framework for Information Systems Architecture」という論文。
- ここに示された枠組みが「ザックマン・フレームワーク」
- 情報システムを対象としていたが、1992年に組織そのものを対象とするように拡張。
- 米国においてはEAフレームワークのデファクト・スタンダーになっている

EAの考え方

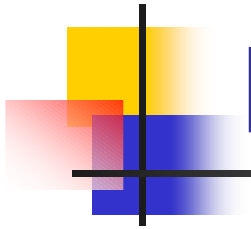
- 4層の視点から業務を分析し整理を行なう

政策・業務体系
データ体系
適用処理体系
技術体系



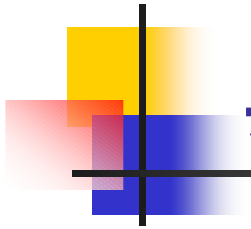
EAが必要な理由

- 部門毎にシステムが存在し、機能が重複していることによる投資の無駄をなくしたい
- システムにかかわる技術や用語を統一することで混乱を避けたい
- トップダウンによるシステム、業務の見直しを行なうことにより戦略を実現したい
- 部分最適ではなく全体最適を実現するためには統一的な枠組みが必要



EAのポイント

- EAは組織戦略そのもの
- 全体最適を重要視
- 実現テクノロジーはそれほど重要ではない
- 無駄を省き、投資効率の最適化をはかる
- 経営とシステムを結合させるための手段
- 現状のシステムの検証だけでなく、よりよいシステムへの改善を指向している

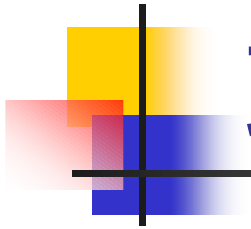


具体的なEAの導入方法

- フレームワークを導入するのが一般的
  - 枠組み／ガイドラインを用いて実際の組織の構成要素をあてはめていく
- 具体的な進め方
  - 経産省(ITアソシエイト協議会)資料参照

政府における取り組み

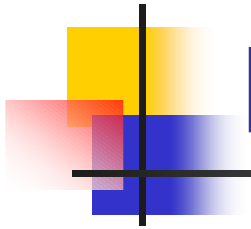
- 電子政府の実現のため「ITアソシエイト協議会」を設置
- EAの導入を推進することを決定。
- 成果物(一部配布)
  - http://www.meti.go.jp/policy/it_policy/itassociate/it.associate.htm



政府によるEAの定義

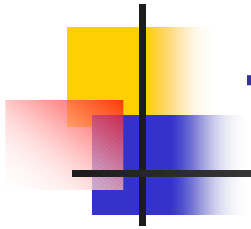
- 電子政府の構築にかかわる関係者がその政策・業務ごとに必ず共有しなければならない知識を参照しやすく整理したもの

- 知識共有を促進するために必要なもの
  - 制度化
  - 参照知識の整備
  - Web化
  - 研修
  - コーチング



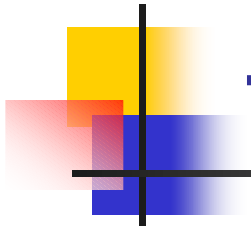
EAについてのまとめ

- EAは哲学。
- コンピュータを支える様々な考え方の1つとして押さえておくべきもの
- 具体化についてはまだ発展途上



SOA
サービス指向アーキテクチャ

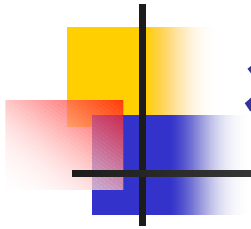
- SOAとはなにか。
- SOAの背景。



サービスとは

- リクエストに対して提供される何らかの機能。
- 通信の考え方がベースになっている。

- (例)ブラウザからウェブを見る場合
  - ブラウザが特定のディレクトリの情報を送るようにウェブサーバにリクエスト
  - ウェブサーバはブラウザに情報を送信

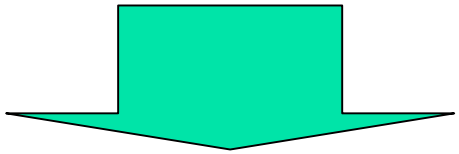


従来型サービスの例

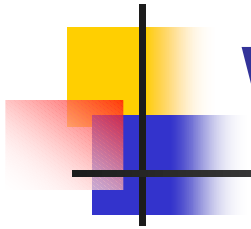
- メール送受信
  - メールクライアントがメールサーバにrequest
- 時刻同期
  - 時刻を同期したいPCがサーバに時刻を聞く
- データベースリクエスト
  - データベースにある特定の情報をデータベースサーバにrequest

従来型サービスの問題点

- Request を出す側、受ける側、ともに専用の実装がされている
- Requestの手順やサービス手段はサービスによって異なる



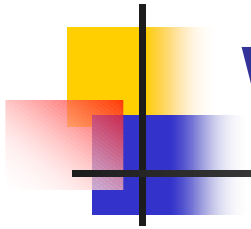
- 統一フレームワークがあると便利
  - 「Webサービス」フレームワーク
    - 通信の一部分を共通化 → 大きなメリット



Webサービスの技術的ポイント

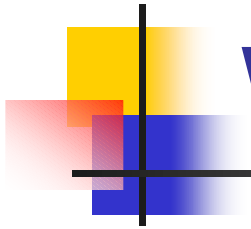
- 既存の技術を活用
- 通信はWeb技術がベース
  - TCP/IP および HTTPを利用
- データ形式はXMLを利用
  - XMLによるデータのカプセル化を行なう

- データの中身については各アプリケーションが実装を行なう。
  - ある程度のデータ型はW3Cで定義が行なわれている



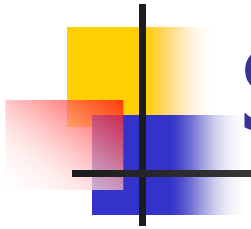
Webサービスのメリット

- システム間での連携を取ることができる
  - RPCなどのプロセス間通信と考え方は一緒
- ウェブ技術を使っているためインターネットを介しての利用が可能
- 広範囲な用途に利用可能
- 標準化された技術を元にしており、サービスを部品化が可能



Webサービスの例

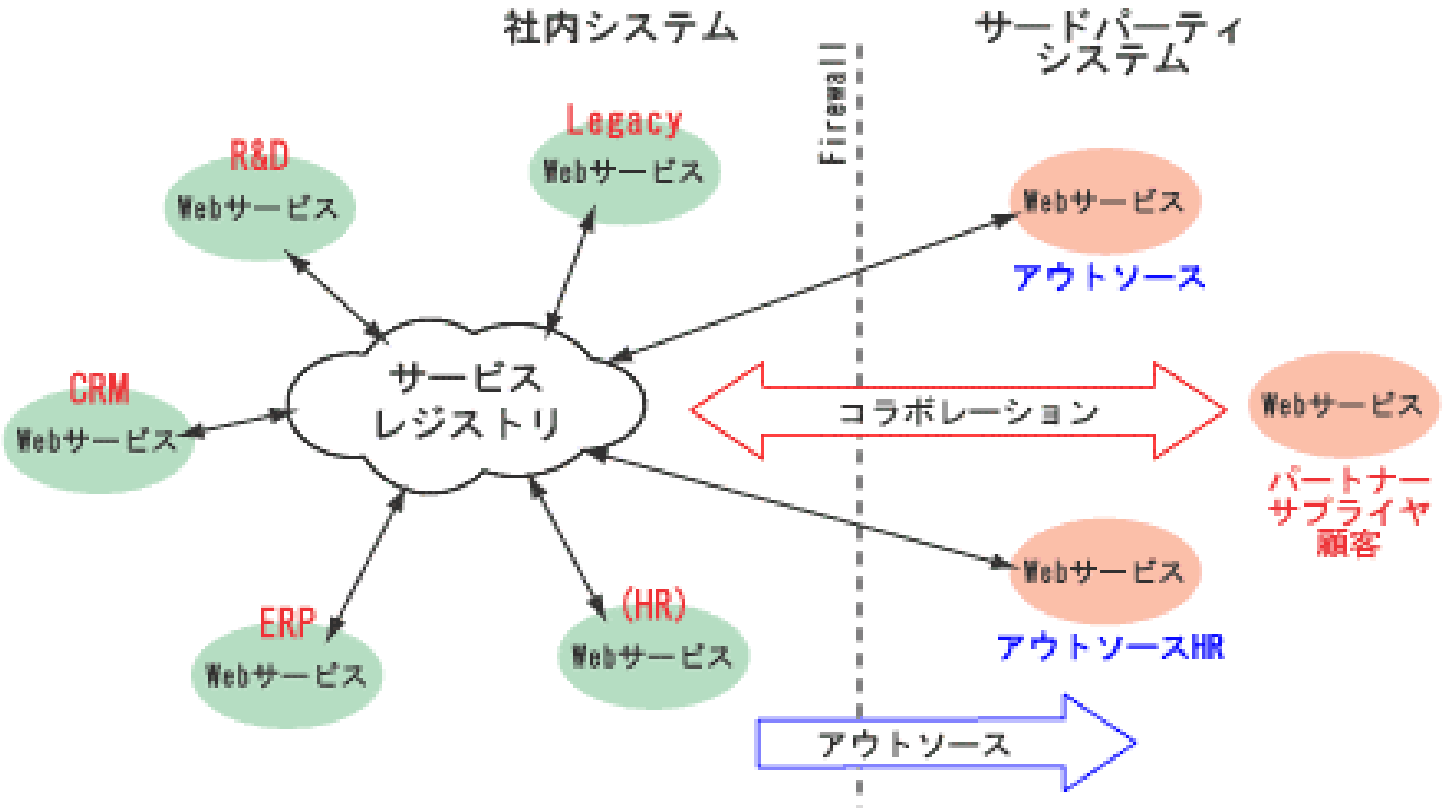
- Amazon
  - Amazonの機能を外部から利用可能
- Google
  - Googleの検索エンジンの様々な機能を利用可能
- .NETテクノロジー
  - システム間の連携をWebサービスで簡単に構築可能
- SalesForce.com
  - ASPと外部システムとをWebサービスで連携

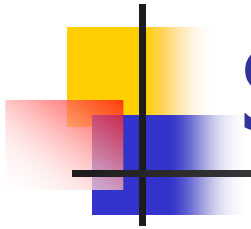


SOAとWebサービス

- SOAはWebサービスとセットで捉えるのが一般的
  - ただし概念的には別
- 個々のシステムは、requestに対してなんらかのサービスを提供するもの、と考える
  - Input：request / Output：サービス
- 業務プロセスは個々のシステムがWebサービスにより連携を行ない実現される。

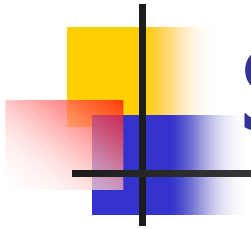
SOAの概念図


社内システム
サードパーティ
システム
Firewall
Legacy
Webサービス
R&D
Webサービス
Webサービス
アウトソース
CRM
Webサービス
サービス
レジストリ
コラボレーション
Webサービス
パートナー
サプライヤ
顧客
Webサービス
アウトソースHR
ERP
Webサービス
(HR)
Webサービス
アウトソース



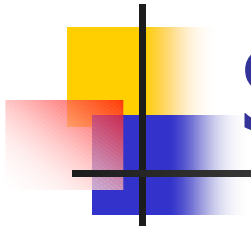
SOAのメリット

- システムが実現するビジネス・プロセスを「サービス」と見なすことにより、ビジネス・プロセスの構築を非常に柔軟にできる
- サービスは必ずしも自社でインストール＆カスタマイズする必要はない
- 利用可能なサービスが存在する場合には、インターフェイスのみ用意すれば良いので短期間、低コスト、低リスクでシステム追加が可能



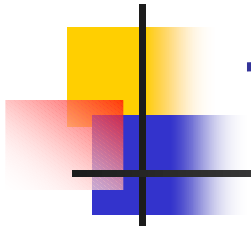
SOAのビジネス的インパクト

- オフショア・アウトソーシングと並ぶプログラマーへの脅威として位置付けられている
- SOAはアウトソーシングの活用を促すITアーキテクチャである



SOAについてのまとめ

- SOAは業務をサービスのまとまりとして捉える
- SOAの考え方で構築されたシステムは柔軟性が非常に高い



まとめ

- EAとSOAという新しい考え方について説明を行なった
- アーキテクチャという単語は同じだが視点は違う
- コンピュータシステムは捉え方により様々な見方ができる
- システム、業務に関して、柔軟な視点を常に持ち、変革に対処し続けなければいけない