

IoT時代のシステム品質  
～これからのシステムは品質確保がポイントとなる～

慶應義塾大学大学院  
システムデザイン・マネジメント研究科  
教授 白坂 成功

# 人は見たいモノしか見ない

**「Moonwalking Bearに気づかない」**

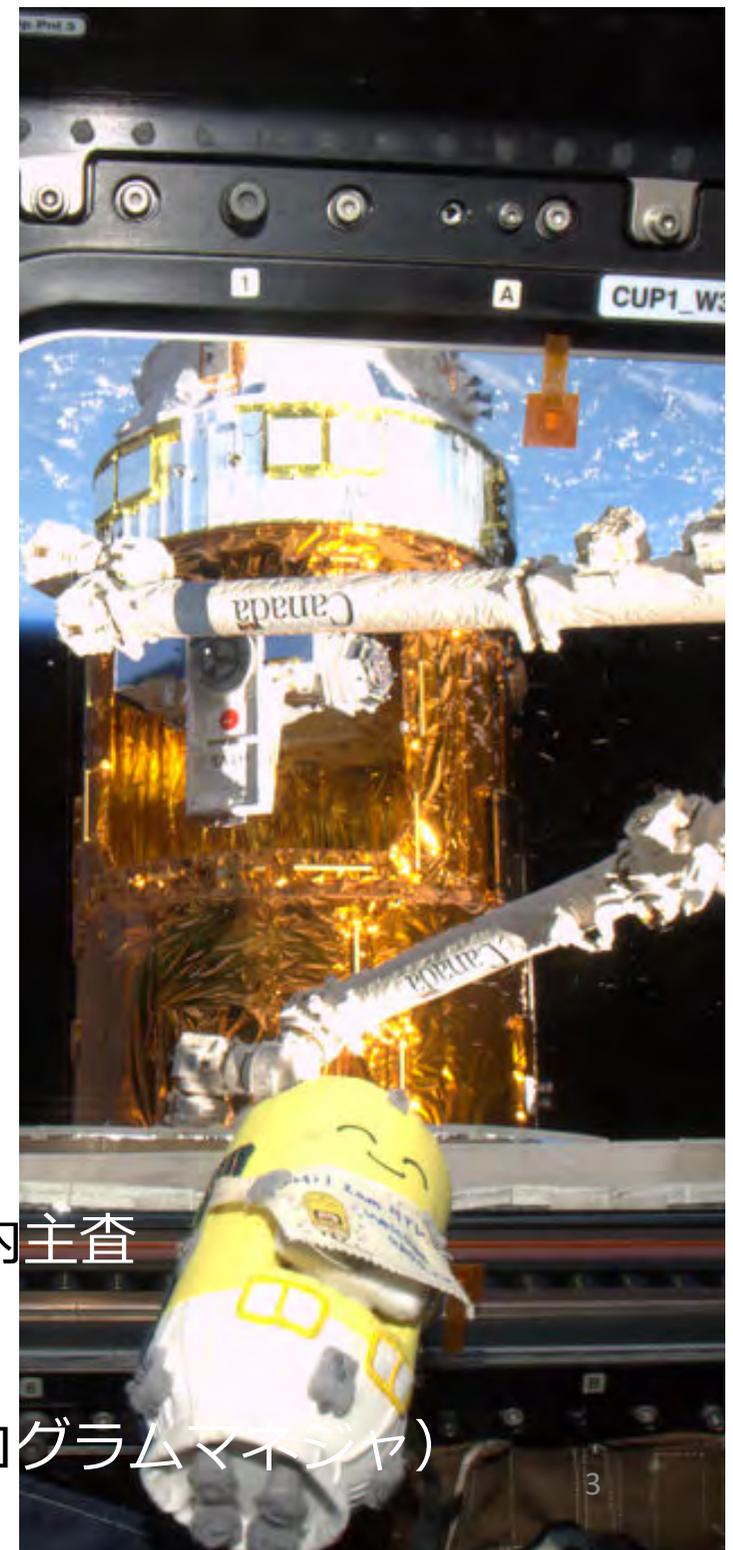
**「放射線技師の83%がゴリラを見逃した」**

- 人は無意識に専門性のバイアスにとらわれる
- 俯瞰的にものごとを捉えるのは簡単ではない
- だからこそ“道具”（=方法論）が必要

# 自己紹介

- 修士：東京大学大学院工学系研究科
- 博士：慶應義塾大学大学院SDM研究科
- 大手電機メーカーにて人工衛星開発（15年間）
  - 「おりひめひこぼし」
  - 「こうのとりのり」：技術とりまとめ
  - 「みちびき」
- 2010年4月より慶應大学専任准教授
- 途中、ドイツ 現Airbus社（宇宙部門）に駐在
- INCOSE日本支部設立メンバー
- ISO JTC1/SC7 WG42「アーキテクチャ」国内主査
- PMI日本支部 PFM/PGM WG
- 内閣府ImPACT PM（革新的研究開発推進プログラムマネージャ）

shirasaka@sdm.keio.ac.jp



# 自己紹介

- 最近の研究テーマ：方法論
  - 大規模複雑システムデザイン
  - イノベーティブデザイン
  - システム安全デザイン
  - 高信頼性システムデザイン
  - コンセプトデザイン/  
    コンセプトエンジニアリング
  - etc



# 内容

- システムの価値とシステムアーキテクチャ
- IoT時代のシステムの特徴
  - システムとしての対象の拡大
  - コンテキストの急激な変化
  - 重要なシステム特性の増加

# システムの価値

単一ではなく、複数の“要素”を組み合わせることで「価値」が生まれる

## “システム”の定義

- 「システムとは、**定義された目的を成し遂げるための、相互に作用する要素 (element) を組み合わせたものである。これにはハードウェア、ソフトウェア、ファームウェア、人、情報、技術、設備、サービスおよび他の支援要素を含む**」 (INCOSE\* SE Handbook)

INCOSE : International Council on Systems Engineering  
<http://www.incose.org>

# システムの価値

単一ではなく、複数の“要素”を組み合わせることで「価値」が生まれる

何と何をどのように組み合わせるかが  
重要となる！

何と何をどのように組み合わせるか？

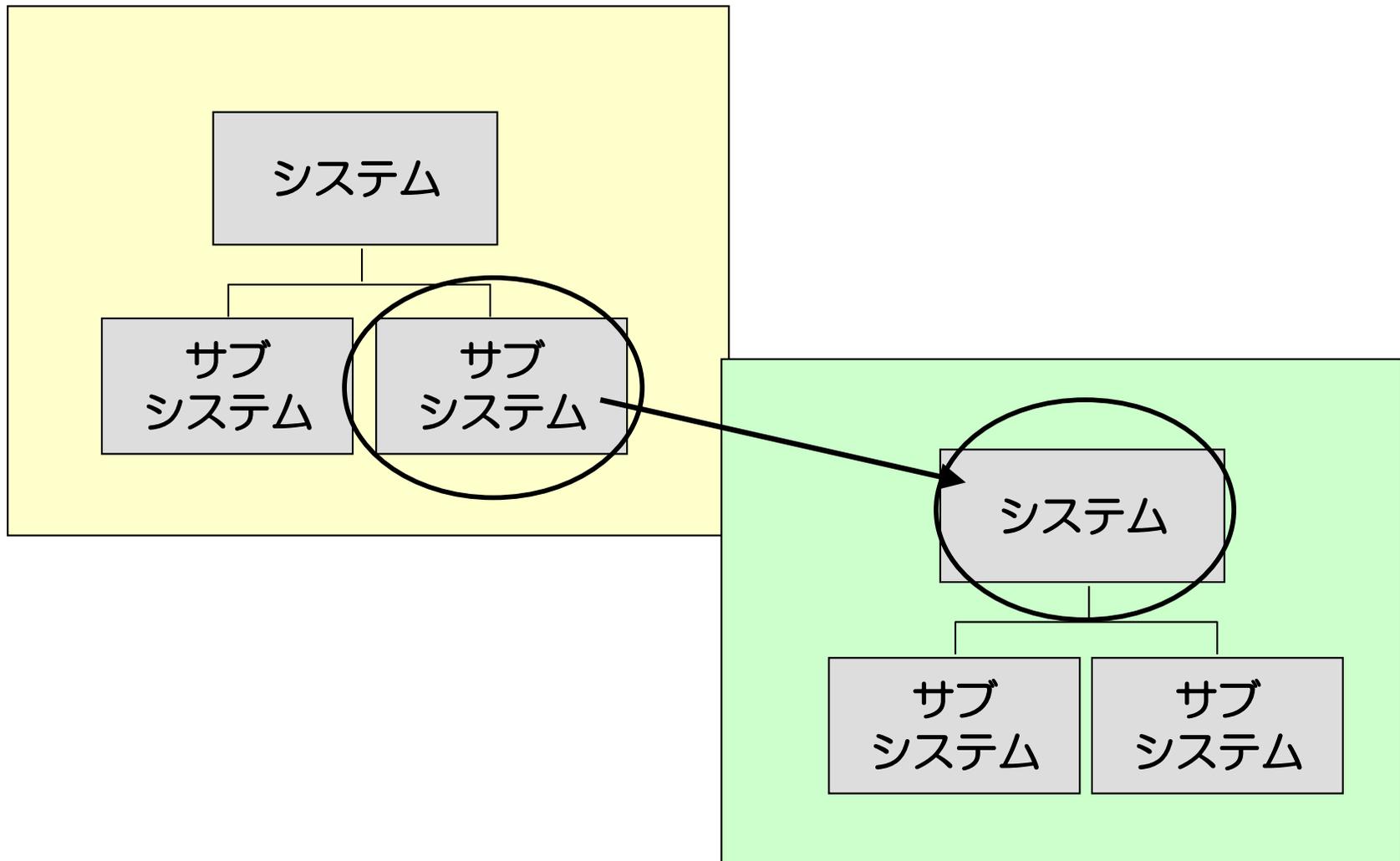
=

システムアーキテクチャ

“Connected Industries”のかけ声のもと、“システムアーキテクチャ”、“システムズエンジニアリング”人材の育成にMETIが興味を持ち始めた！

# システムの特徴

## 階層構造：Building Block

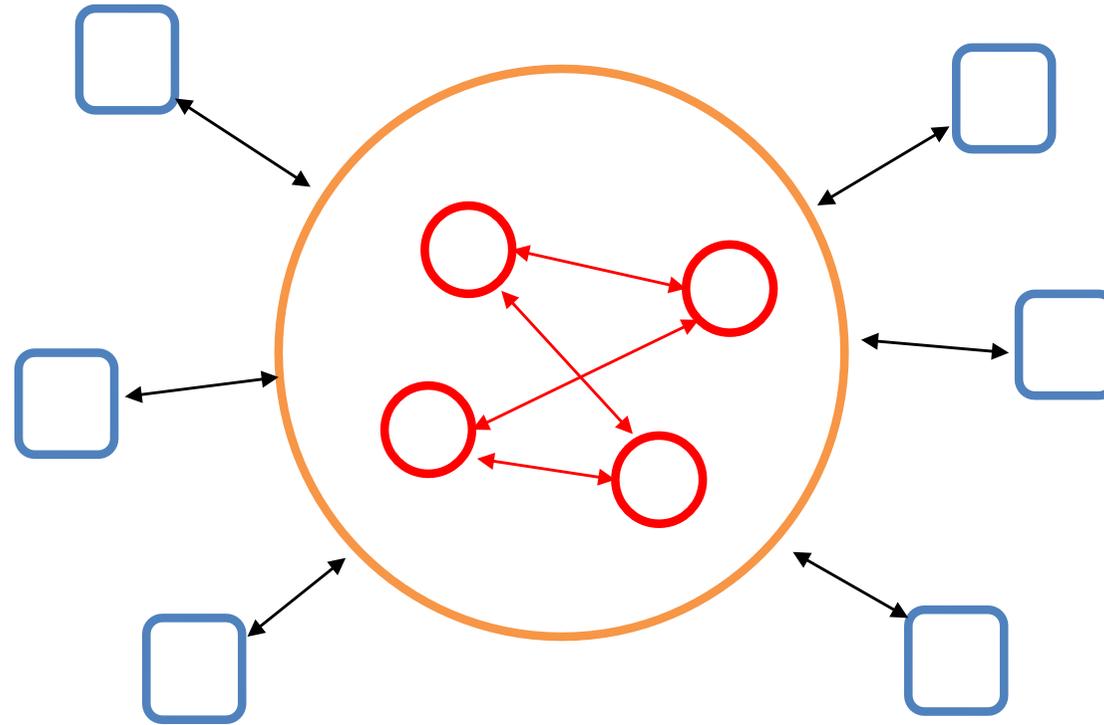


**EVERY SYSTEM HAS AN  
ARCHITECTURE.**

# アーキテクチャの定義

- 目的を最大化するような**機能と特性の配置**(Ring, 2001)
- 構成要素の設計や進化を左右するような、構成要素の**構造**、構成要素間の**関係**、そして**原理**や**指針** (IEEE STD 610.12, 1990)
- システムと**外界との関係**及びシステムを構成する**要素とその構成要素間の関係** (白坂)

# アーキテクチャの定義

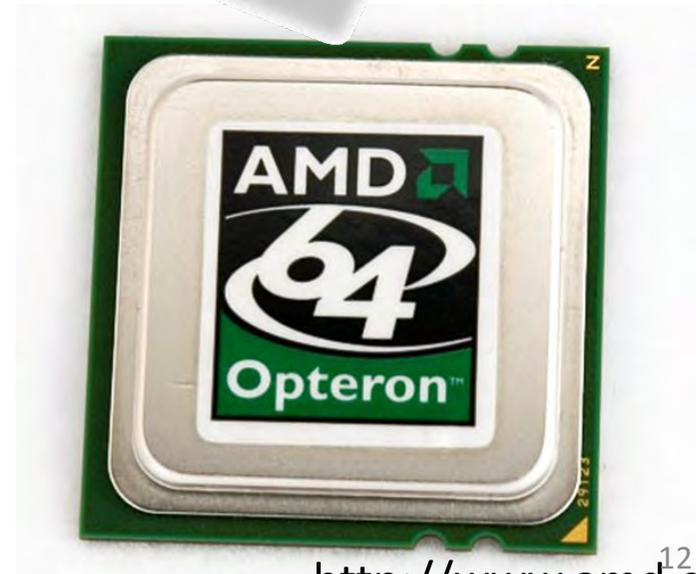


- システムと外界との関係及びシステムを構成する要素とその構成要素間の関係 (白坂)

# アーキテクチャ？



[www.intel.com](http://www.intel.com)



<http://www.amd.com/>

© shirasaka@sdm.keio.ac.jp

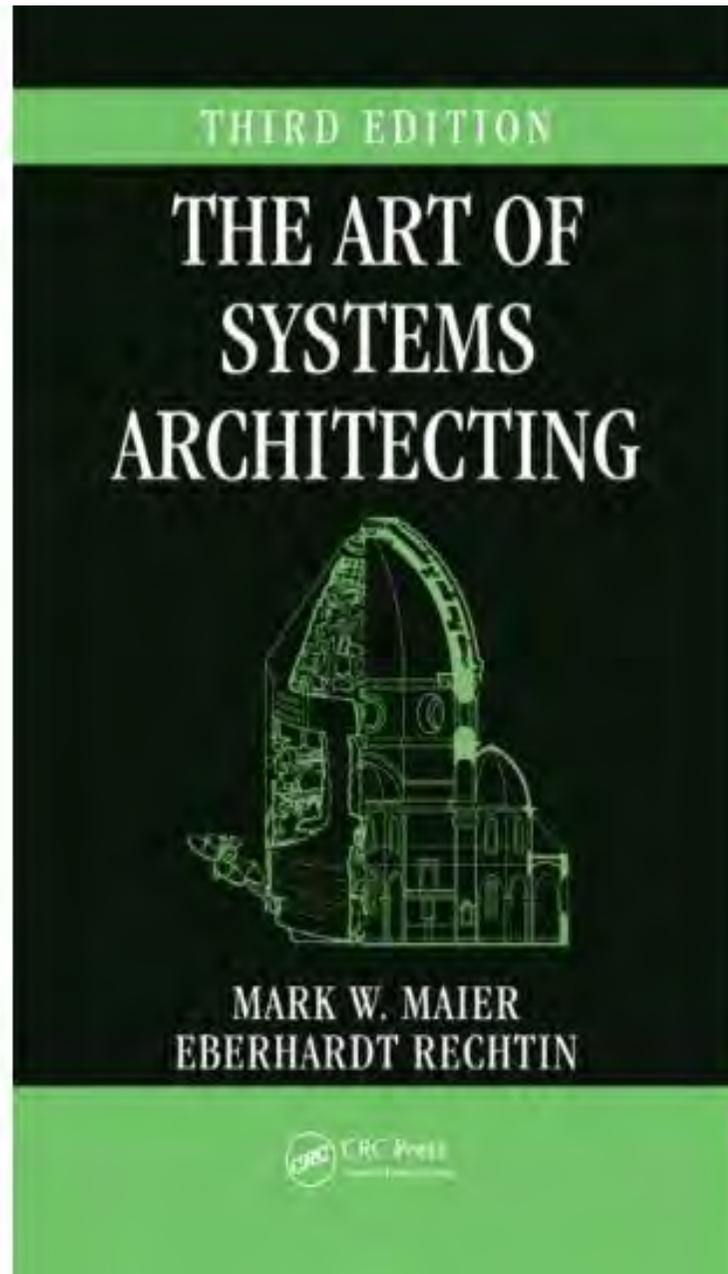
<http://slowbuddy.com/design/modern-architecture/>

アマゾンで「アーキテクチャ」を検索すると(建築以外に, , , )



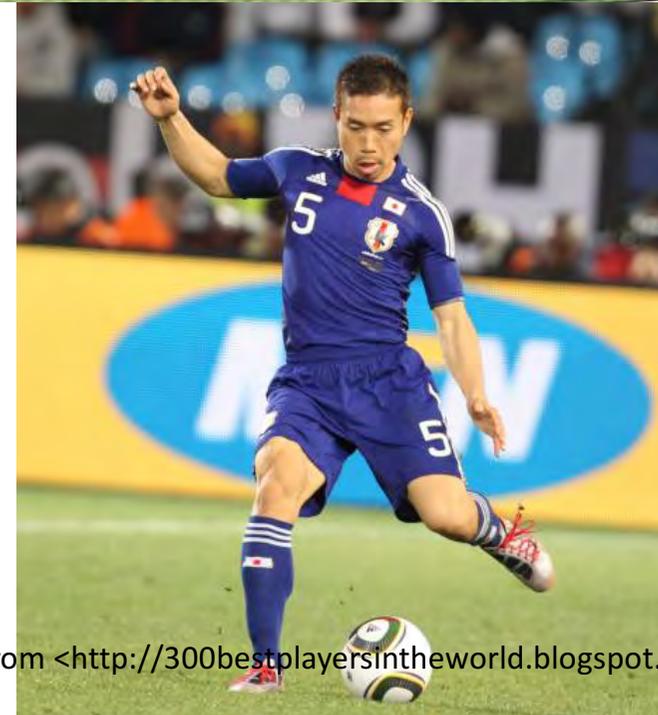
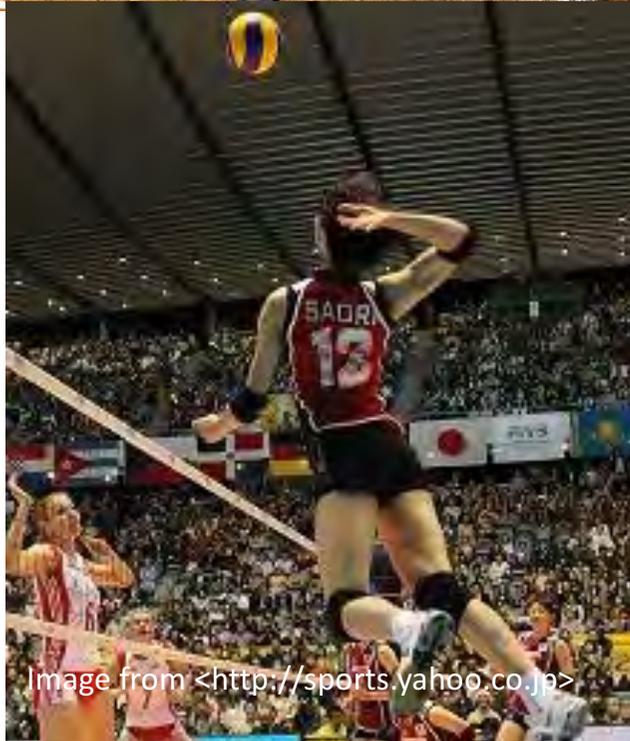
アマゾンで「アーキテクチャ」を検索すると



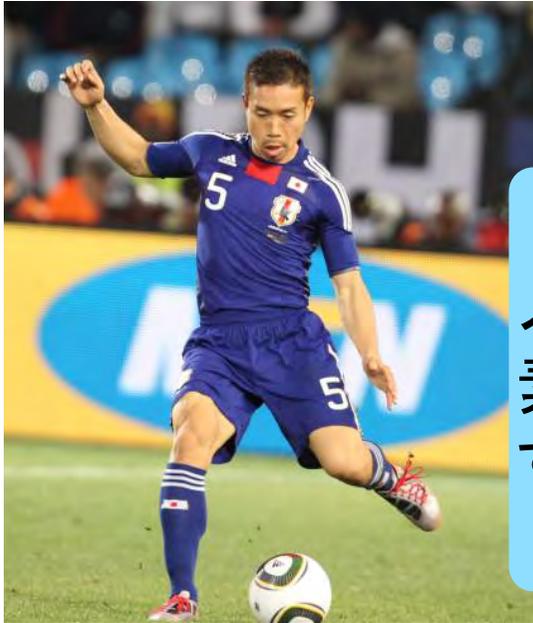


Characteristic	Architecting	A & E	Engineering
Situation/goals	Ill-structured Satisfaction	Constrained Compliance	Understood Optimization
Methods	Heuristics Synthesis	↔ ↔	Equations Analysis
Interfaces	<b>Art</b> and science Focus on "mis- fits"	Art <b>and</b> Science Critical	<b>Science</b> and Art Completeness
System integrity maintained through	"Single mind"	Clear objectives	Disciplined methodology and process
Management issues	Working for Client Conceptualization and certification Confidentiality	Working with Client Whole waterfall Conflict of interest	Working for Builder Meeting project requirements Profit vs. cost

# 次のうちアーキテクチャが 同じものはどれでしょう？



# アーキテクチャの例



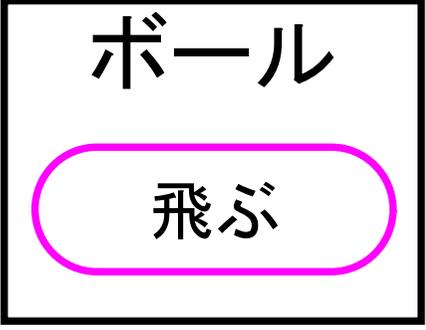
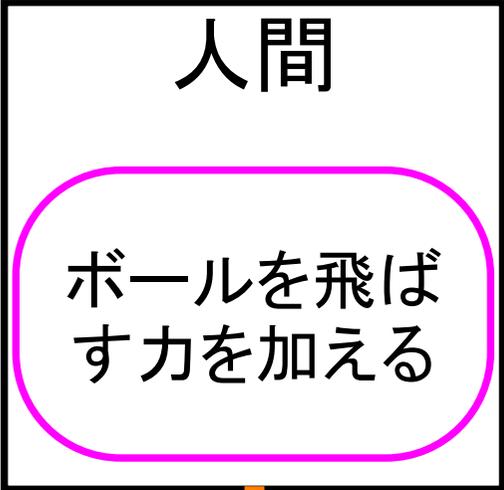
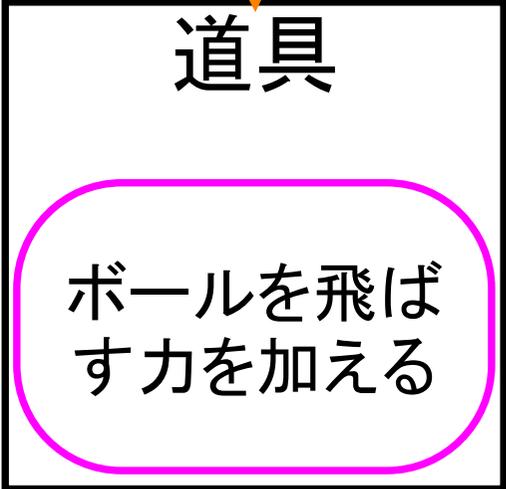
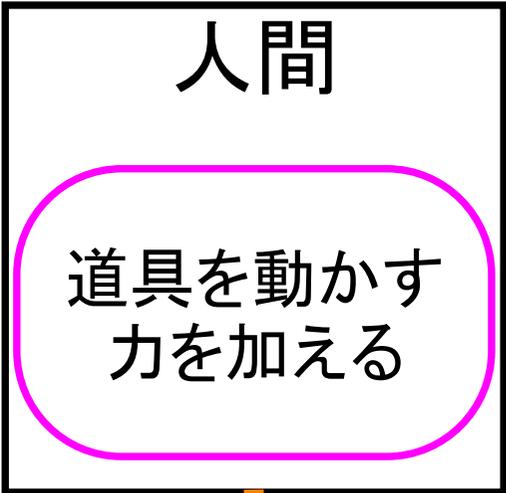
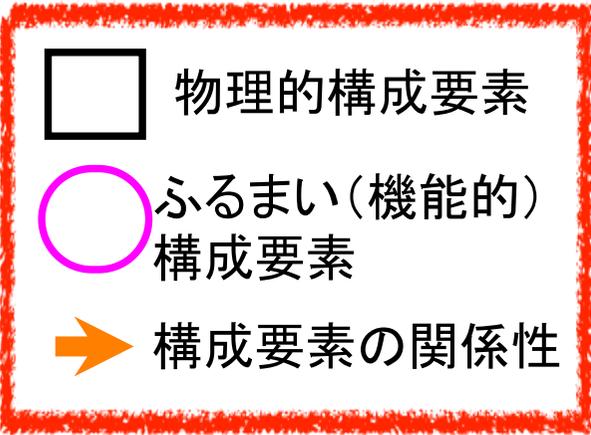
人間とボールという要素が、直接相互作用するアーキテクチャ。



人間とボールという要素が、道具を介して相互作用するアーキテクチャ。



# アーキテクチャの例



人間がボールを飛ばす力を加える。  
ボールが飛ぶ。

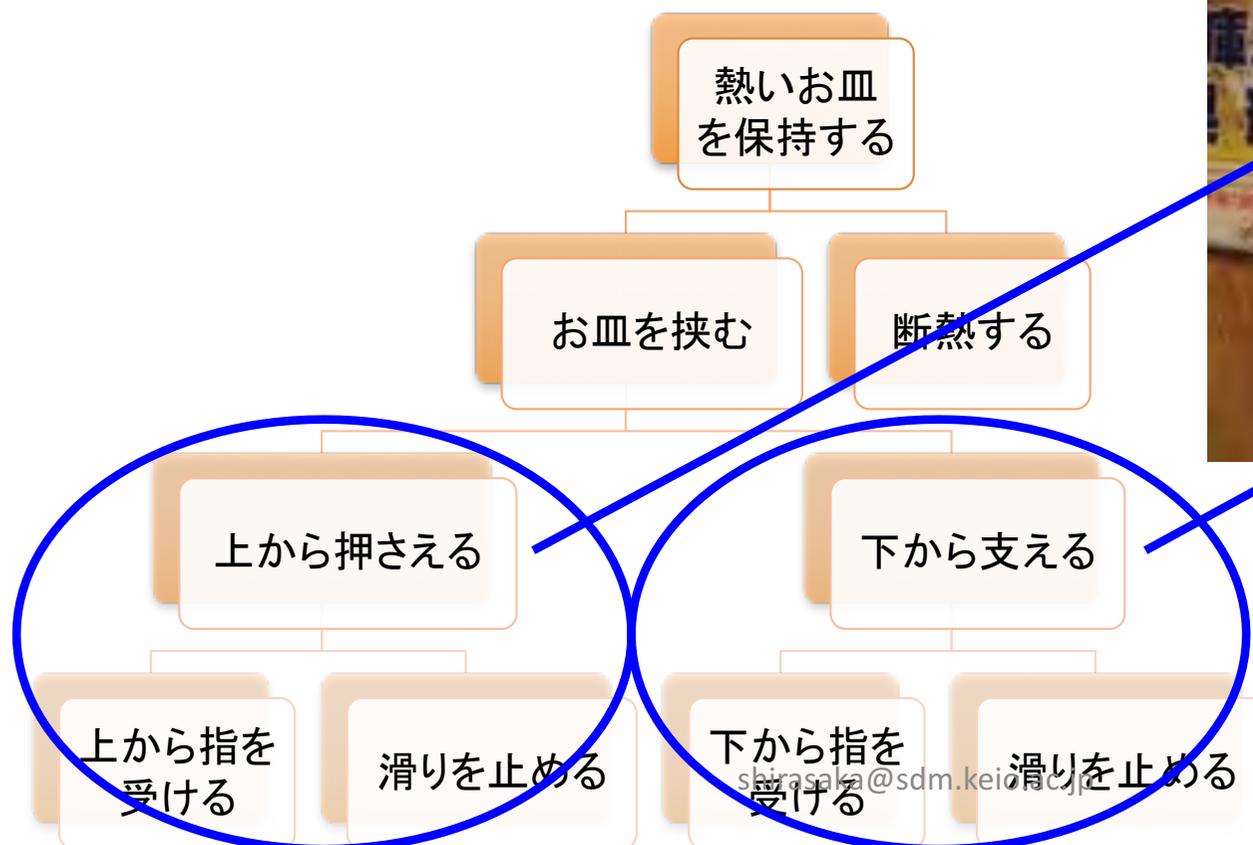
人間が道具を動かす力を加える。  
道具がボールを飛ばす力を加える。  
ボールが飛ぶ。

# アーキテクティングの定義

- アーキテクチャを作り出す行為
- 機能を要素に割り当て、要素間の関係性（インタフェース）を明確化すること（前野 2010）
- 複数の視点において、構成要素とそれらの関係性を明確化するとともに、各視点間の関係性を明確化すること（白坂）

どのような機能から構成されていて(機能の視点)、どのように実現されているか(物理の視点)を考えてみて下さい。

「パクパク皿キャッチ」の例



出典:  
[http://blog.goo.ne.jp/haru\\_hinata/e/c55459c9c1b9c306baa04289028d7676](http://blog.goo.ne.jp/haru_hinata/e/c55459c9c1b9c306baa04289028d7676)

# 同じ機能を違う物理で実現してみよう！



n.co.jp

出典：amazon.co.jp

# 同じ機能を違う物理で実現してみよう！



# 同じ機能を違う物理で実現してみよう！



上から

上から指を  
受ける

amazon.co.jp【Bahagia】鍋蓋つかみ

出典：amazon.co.jp【Bahagia】鍋蓋つかみ

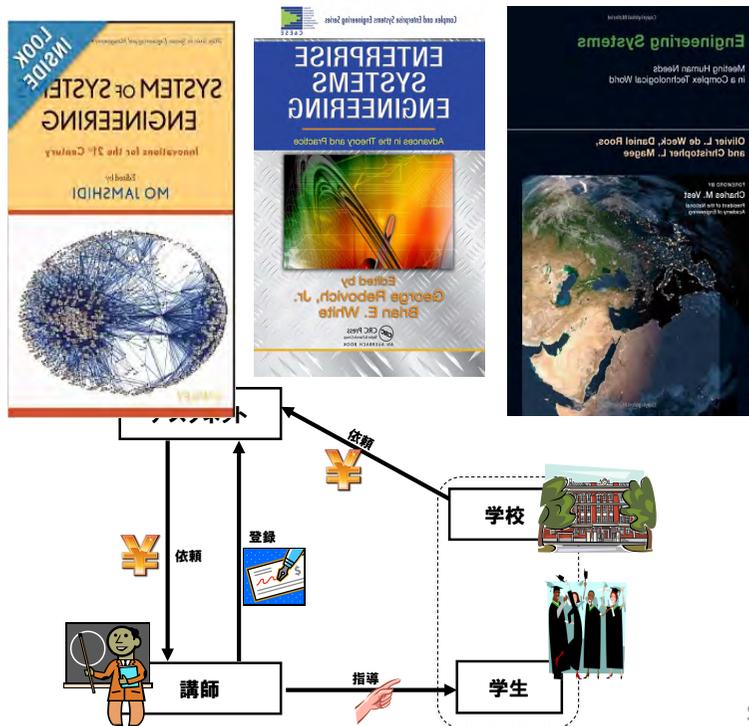
# 内容

- システムの価値とシステムアーキテクチャ
- IoT時代のシステムの特徴
  - システムとしての対象の拡大
  - コンテキストの急激な変化
  - 重要なシステム特性の増加

# IoT時代のシステムの特徴

- システムとしての対象の拡大
- コンテキストの急激な変化
- 重要なシステム特性の増加

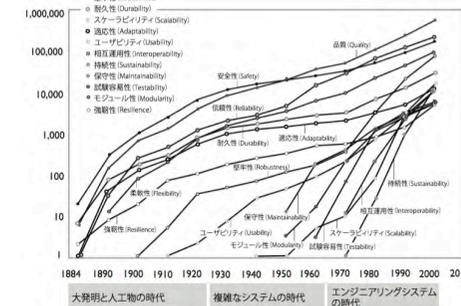
## 対象の拡大



## コンテキストの急速な変化



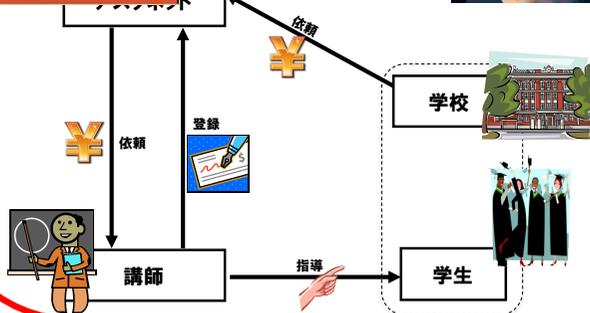
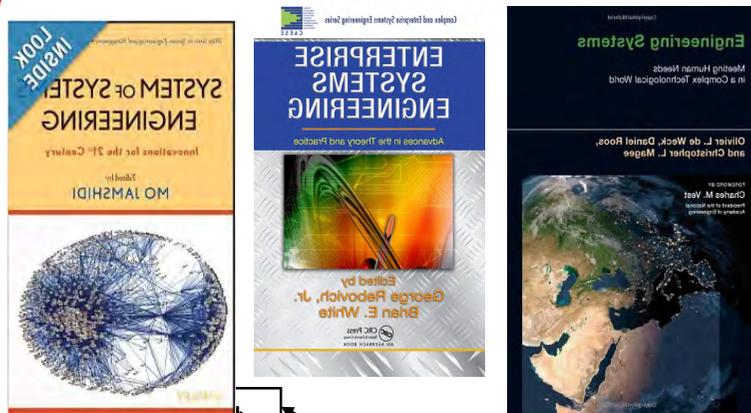
## 重要なシステム特性の増加



# IoT時代のシステムの特徴

- システムとしての対象の拡大
- コンテキストの急激な変化
- 重要なシステム特性の増加

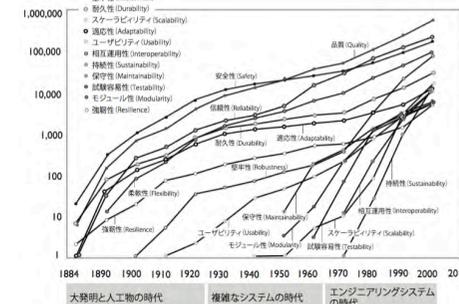
## 対象の拡大



## コンテキストの急速な変化



## 重要なシステム特性の増加



# システムの価値

単一ではなく、複数の“要素”を組み合わせることで「価値」が生まれる

## “システム”の定義

- 「システムとは、**定義された目的を成し遂げるための、相互に作用する要素（element）を組み合わせたものである。これにはハードウェア、ソフトウェア、ファームウェア、人、情報、技術、設備、サービスおよび他の支援要素を含む**」（INCOSE\* SE

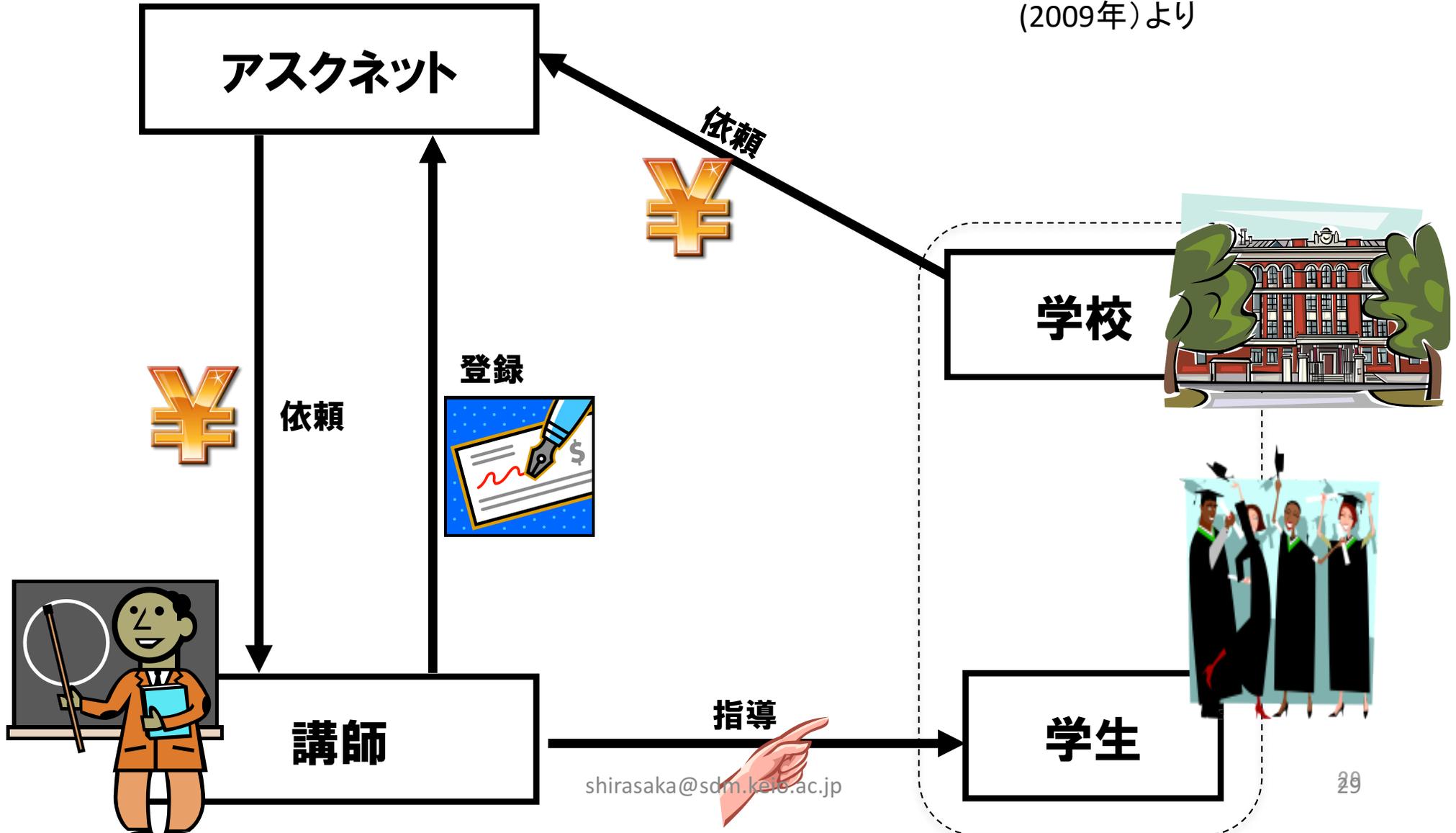
システムは単なるプロダクト  
(H/W, S/W) だけではない

IN  
htt

# ビジネスのアーキテクチャ 価値連鎖分析

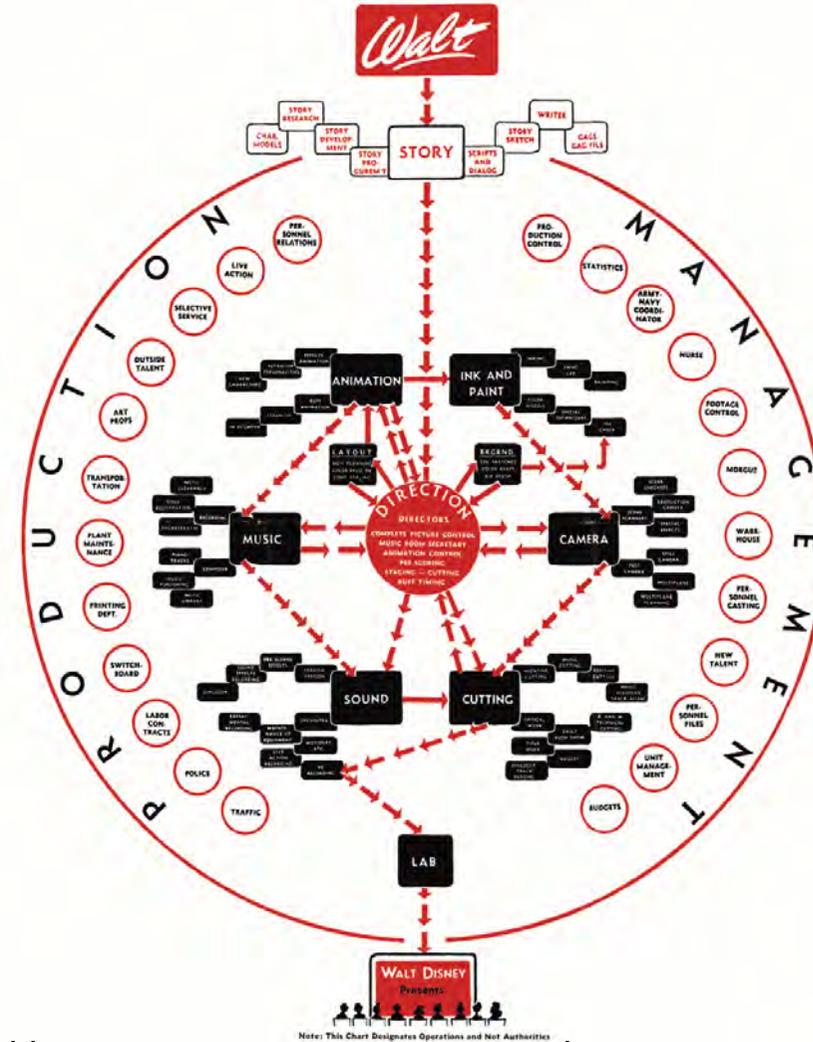
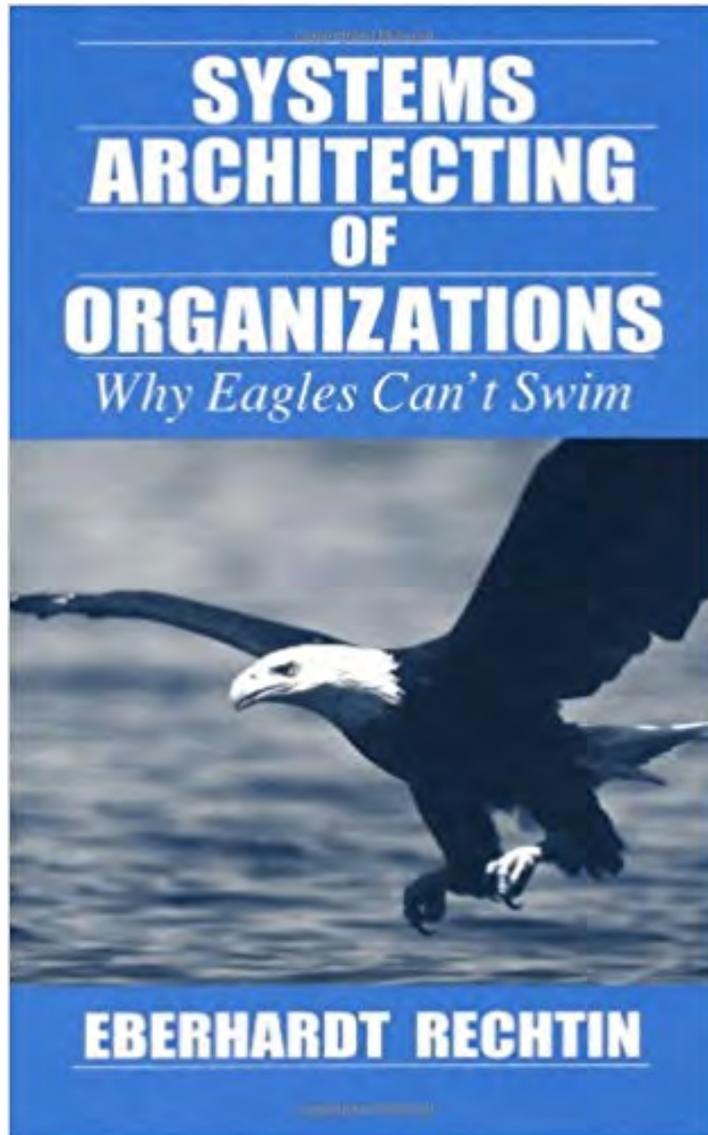
## NPO法人 アスクネット

「経済産業省選定 ソーシャルビジネス55選」  
(2009年)より



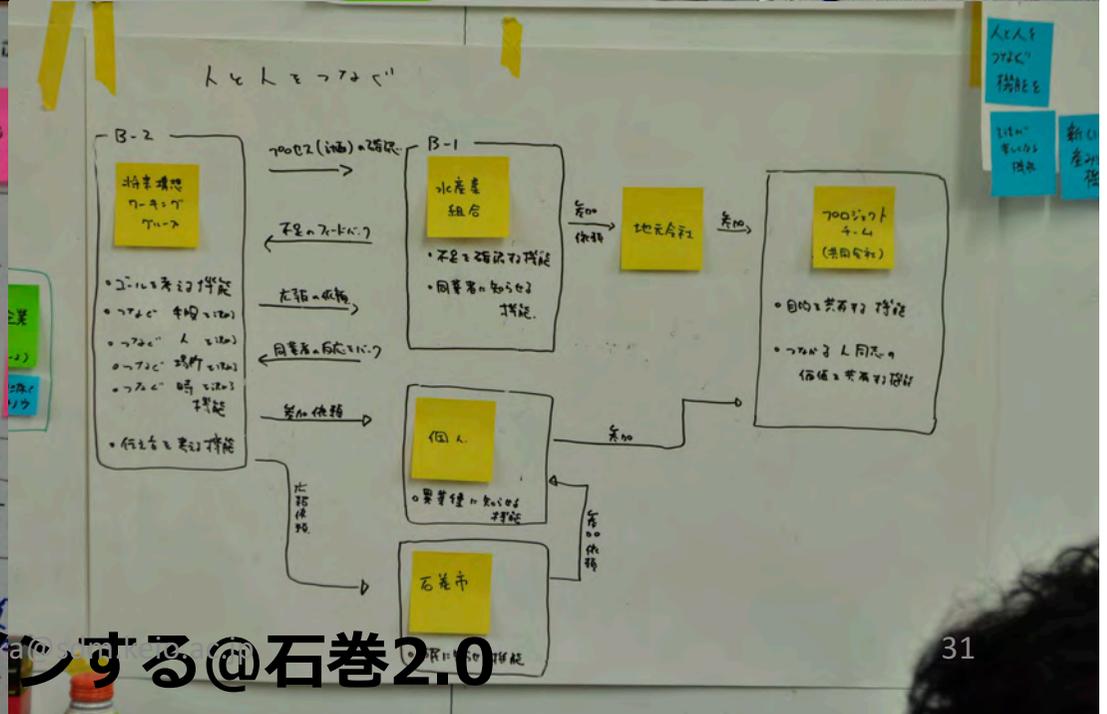
# 組織のアーキテクチャ

ウォルトディズニーの組織アーキテクチャ



<http://www.atissuejournal.com/wp-content/uploads/2009/08/disneyorgchart1.jpg>

# 街のアーキテクチャ



MACHIZUKURI 2.0  
石巻の街をシステムとしてデザインする @石巻2.0

# 慶應SDM

## SDM序論

慶應SDMのM1学生への必修科目

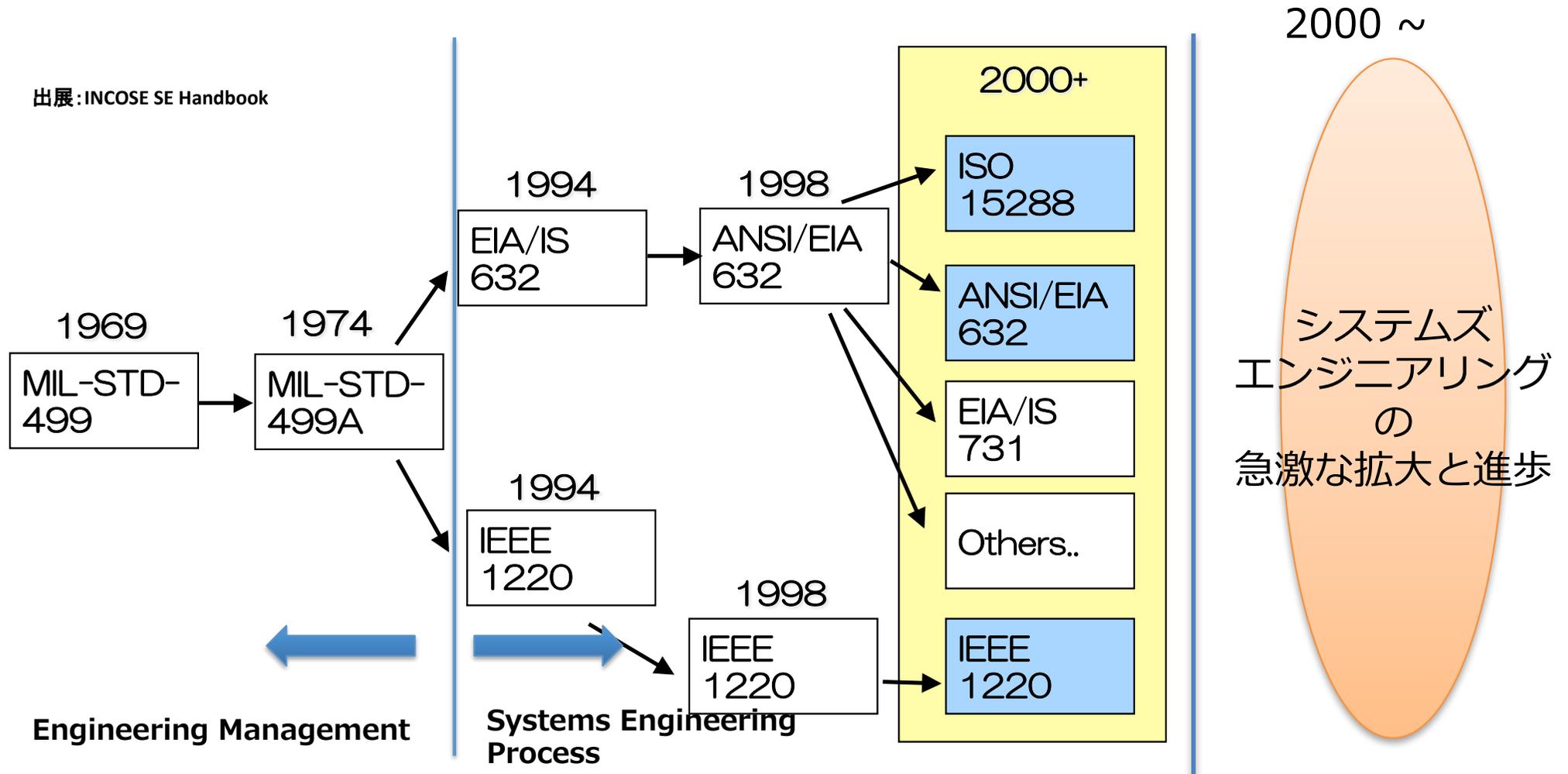
システムエンジニアリングの基本を理解し、デザイン思考・プロジェクトマネジメント（別の必修科目）との関係を理解することを目的とする

全く同じシステム設計のプロセス/手法を使って、以下をデザインする

- ・ カレーライス  
 特殊な環境で食べるカレーライス
- ・ 写真ビジネス  
 写真店のビジネスを設計→ライフログサービスビジネスへの進化
- ・ 100円ショップ会社の組織  
 100円ショップ会社組織の設計→M&A
- ・ 街の防災/減災施策

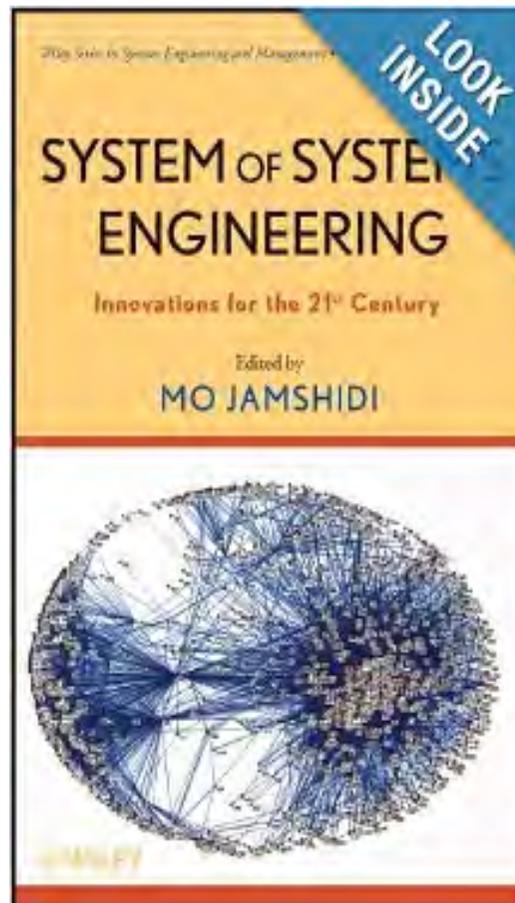
# システムエンジニアリング

出展: INCOSE SE Handbook

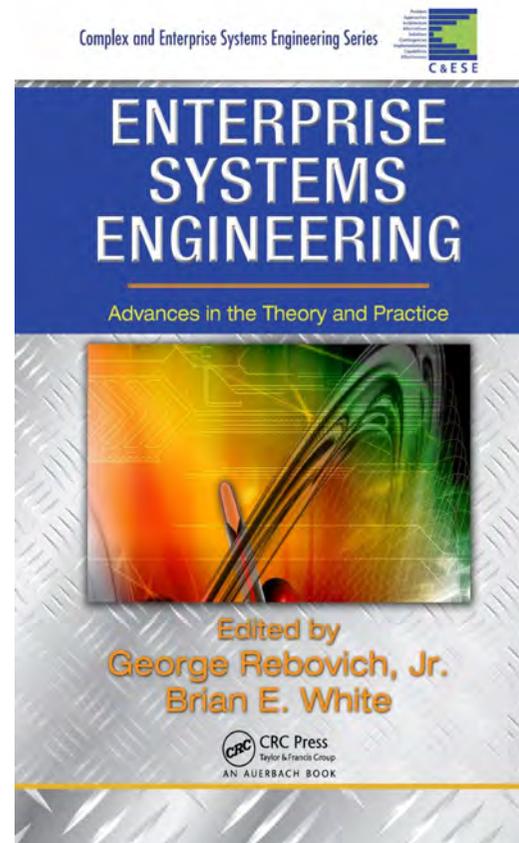


# 世界の動向

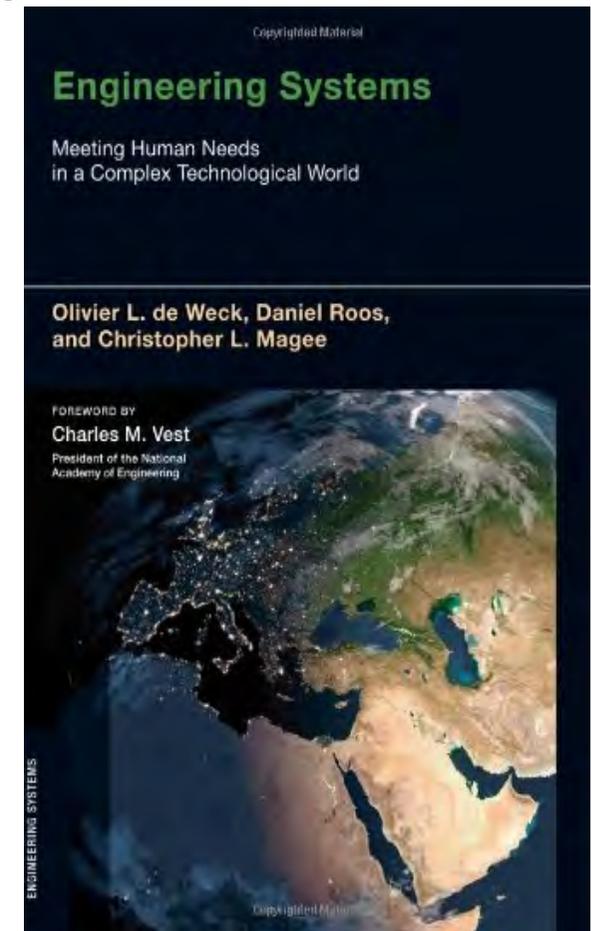
System of Systems Engineering  
(2008)



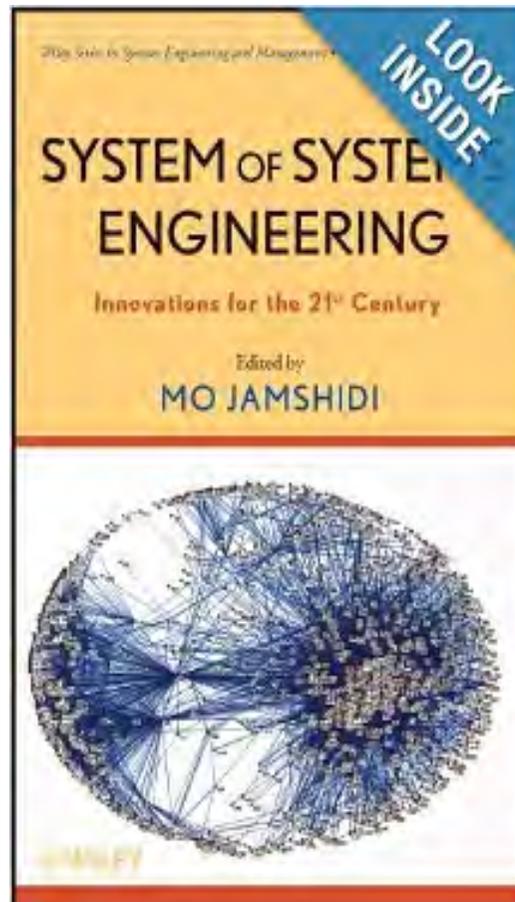
Enterprise Systems Engineering  
(2010)



Engineering Systems  
(2011)



# System of Systems Engineering (2008)



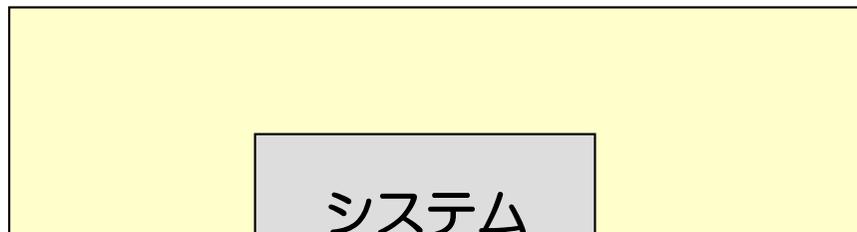
IoT時代のシステムの特徴である  
SoS

出典 : Amazon.com

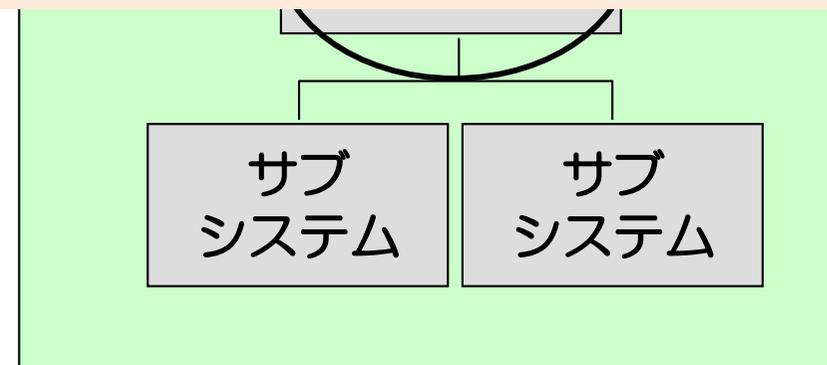
shirasaka@sdm.keio.ac.jp

# システムの特徴

階層構造：Building Block



定義によると、SystemとSystemを組み合わせるとSystemである  
では、System of Systemsとは？



# System of Systems

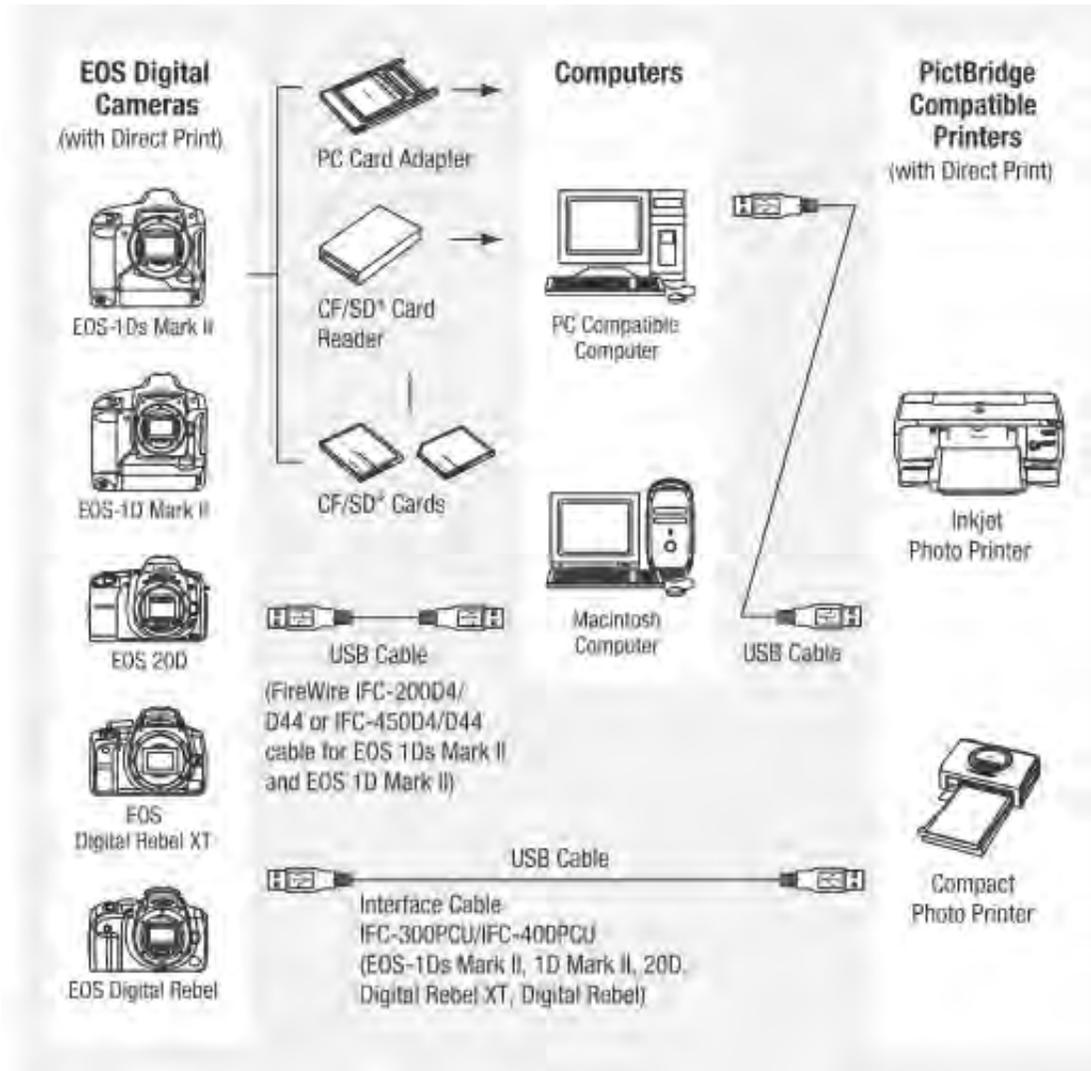
以下の5つの特徴をもったシステムをSystem of Systems (SoS)と呼んでいる。

1. 運用の独立性：SoSの構成システムは、個別に運用。
2. 管理の独立性：構成システムは別々に調達され、インテグレート。しかし、運用中の構成システムはそのまま運用。
3. 進化的開発：機能や目的が追加／削除されたり、途中で変更されるなど、開発とシステムが進化的。
4. 創発的振舞い：構成システム単独では実現できない目的を、SoSとして実現。
5. 地理的な分散：構成システムが離れており、構成システム間では、質量やエネルギーの物理量ではなく、情報を交換。

SoSでないSystemを“Monolithic System”と呼ぶ。

出典：“Architecting Principles for Systems-of-Systems”, Mark W. Maier (1998)

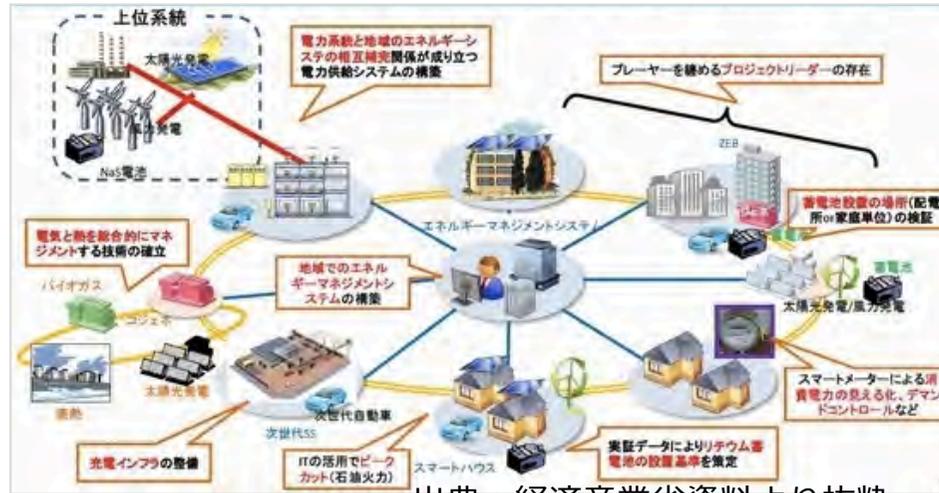
# System of Systems



出典 : INCOSE Systems Engineering Handbook

# System of Systems

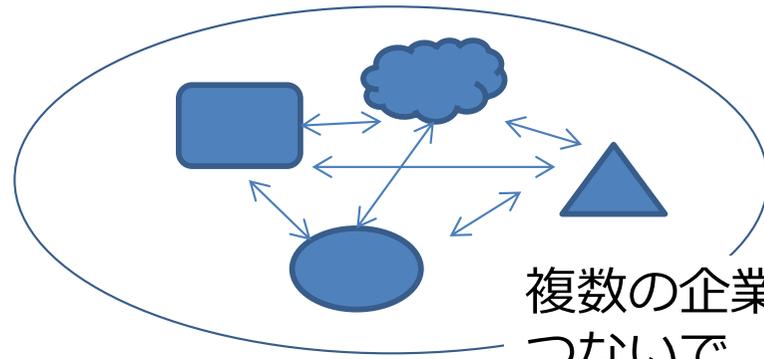
## 世の中の動き：つながりははじめたシステム



出典：テキサスインスツルメント資料より抜粋

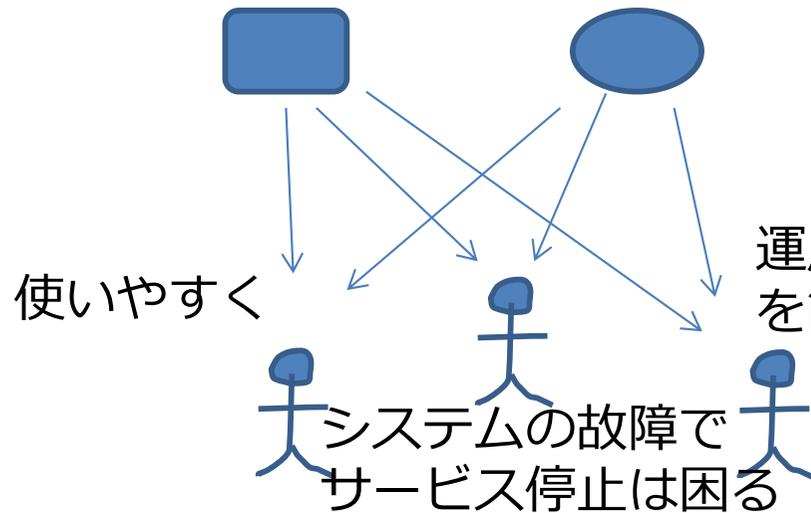


# 品質保証が困難

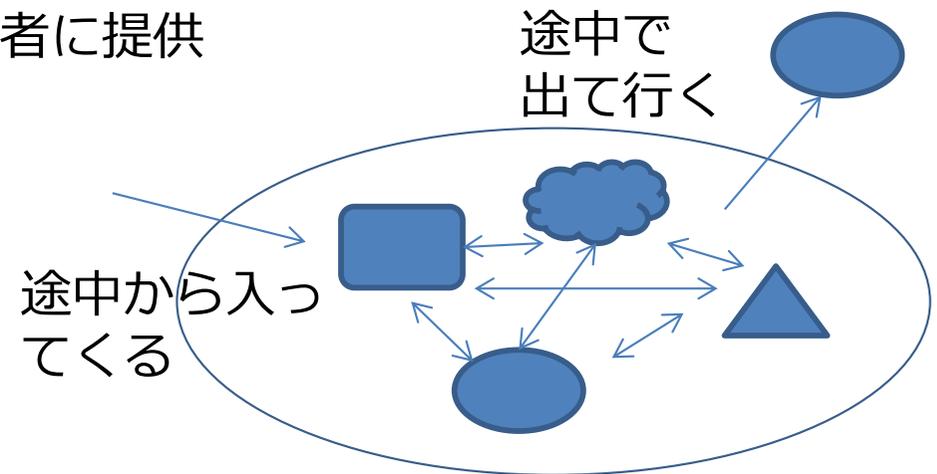


複数の企業の製品・サービスをつないで、利用者に提供

**「つながるシステムの拡大」**

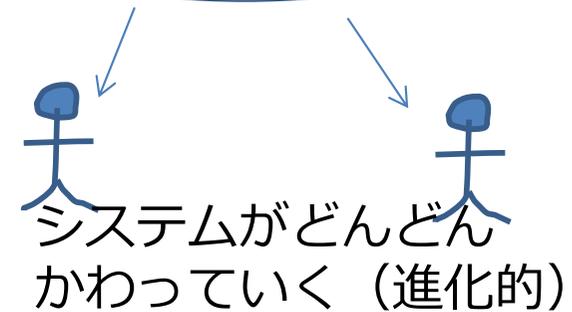


**「多岐にわたるステークホルダ」**



途中から入ってくる

途中で出て行く



**「運用中の予定外のシステムの変化」**

# Systems of Systemsの4つのタイプ

## Directed

- 中央集権的
- 各要素の独立性はあるが、一つの目的

## Acknowledged

- 承認された
- それぞれが目的を持つが、方針は共通

## Collaborate

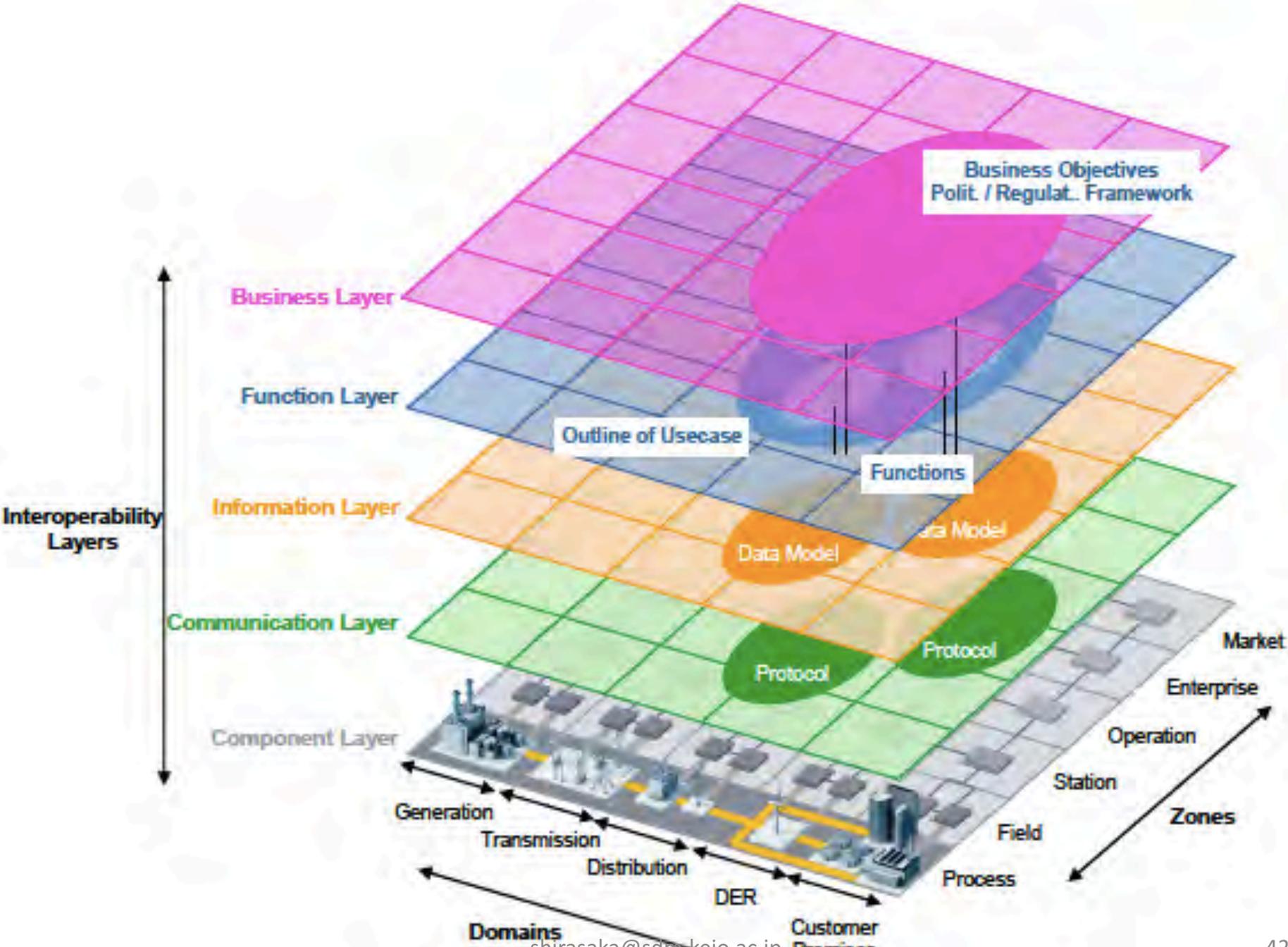
- 協調的
- 共通の目的のために自発的に協力

## Virtual

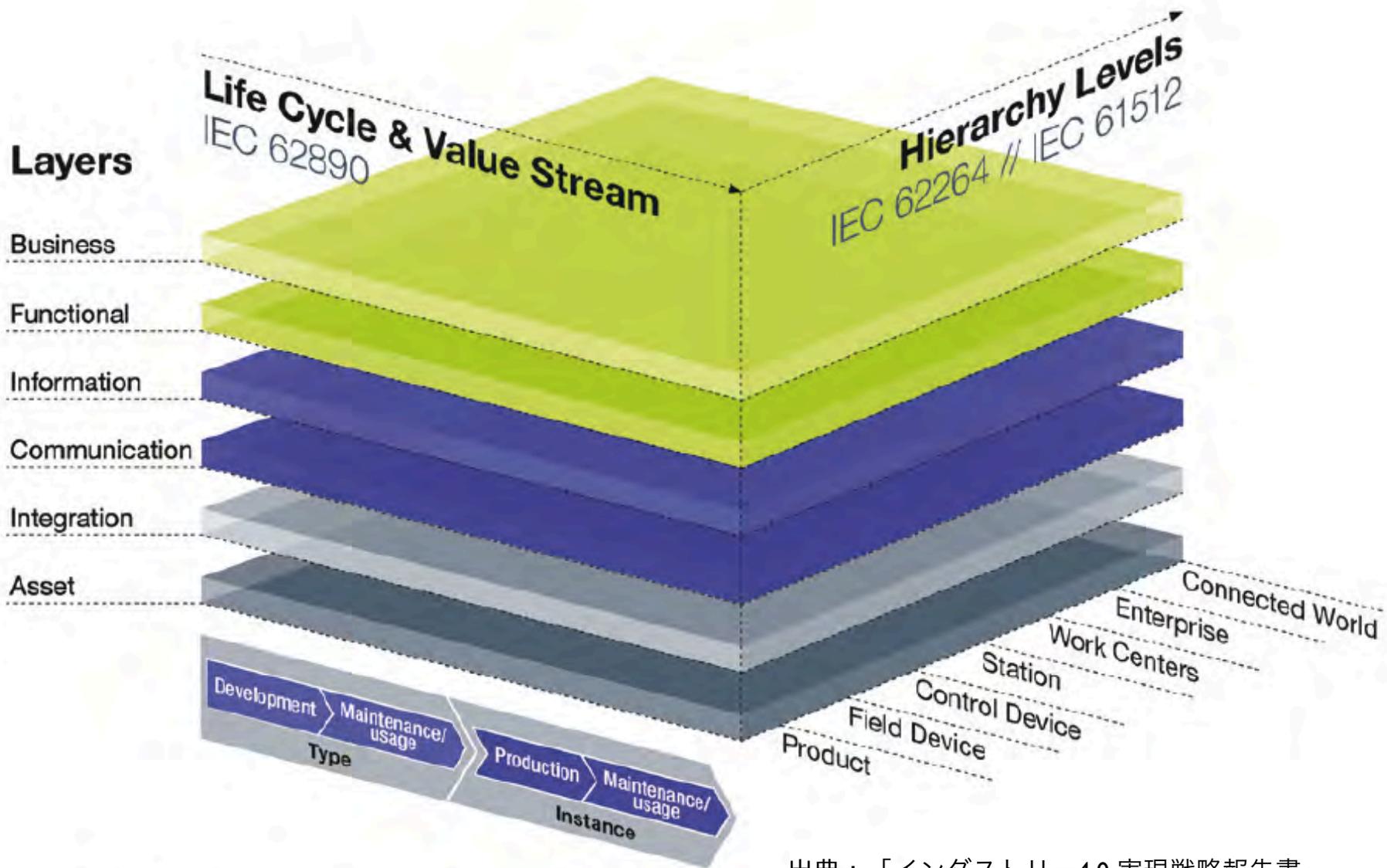
- 仮想的
- 目的の認識がなく、創発的に目的が達成

出典: Systems Engineering Guide  
for Systems of Systems, 2008

# Smart Grid Reference Architecture



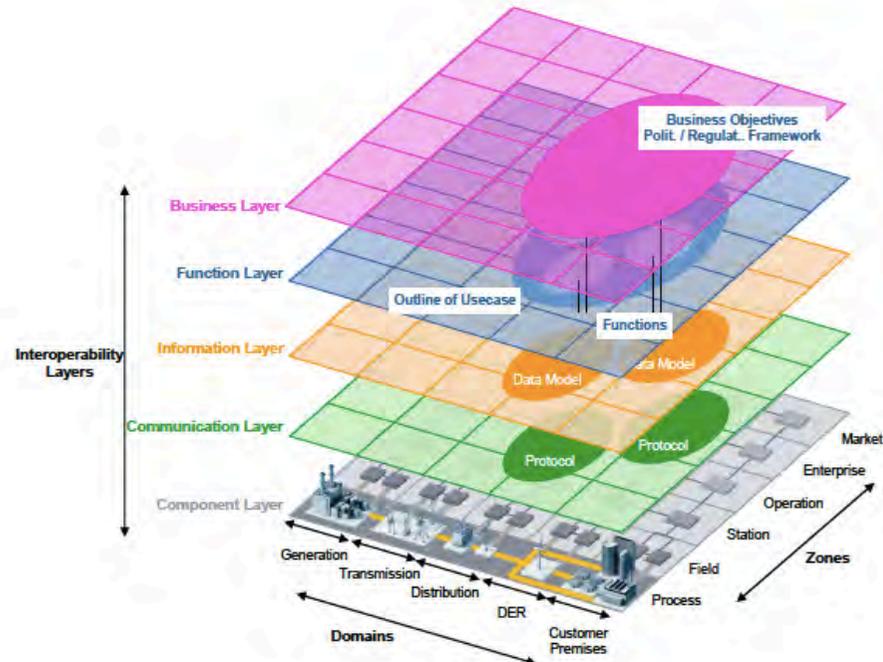
# インダストリー4.0リファレンスアーキテクチャモデル RAMI4.0 : Reference Architecture Model Industrie 4.0



出典：「インダストリー4.0 実現戦略報告書」

# Smart Grid Reference Architecture

# RAMI4.0



出典: Smart Grid Reference Architecture

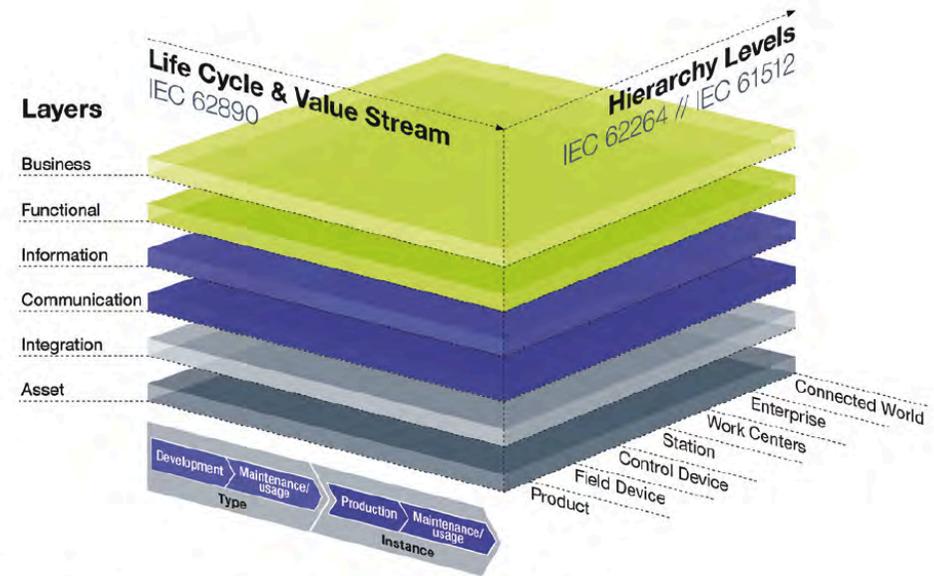


Abbildung 15: Referenzarchitekturmodell / Reference Architecture Model Industrie 4.0 (RAMI 4.0)

出典: 「インダストリー4.0 実現戦略報告書」

- 範囲を限定することによって、ArchitectureのReference Modelをつくることができる
- どこをプラットフォームにしてクローズにし、どこを標準としてオープンにするかをこの上で検討
- 品質、安全性、相互接続性をこのモデル上に位置づけながら担保するための方針
- Architecture Reference Modeは、「時間軸x空間軸x意味軸」で構成

# Systems of Systemsの4つのタイプ

## Directed

- 中央集権的
- 各要素の独立性はあるが、一つの目的

## Acknowledged

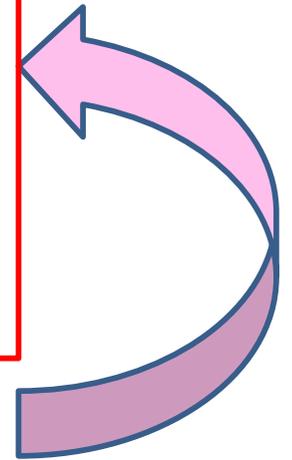
- 承認された
- それぞれが目的を持つが、方針は共通

## Collaborate

- 協調的
- 共通の目的のために自発的に協力

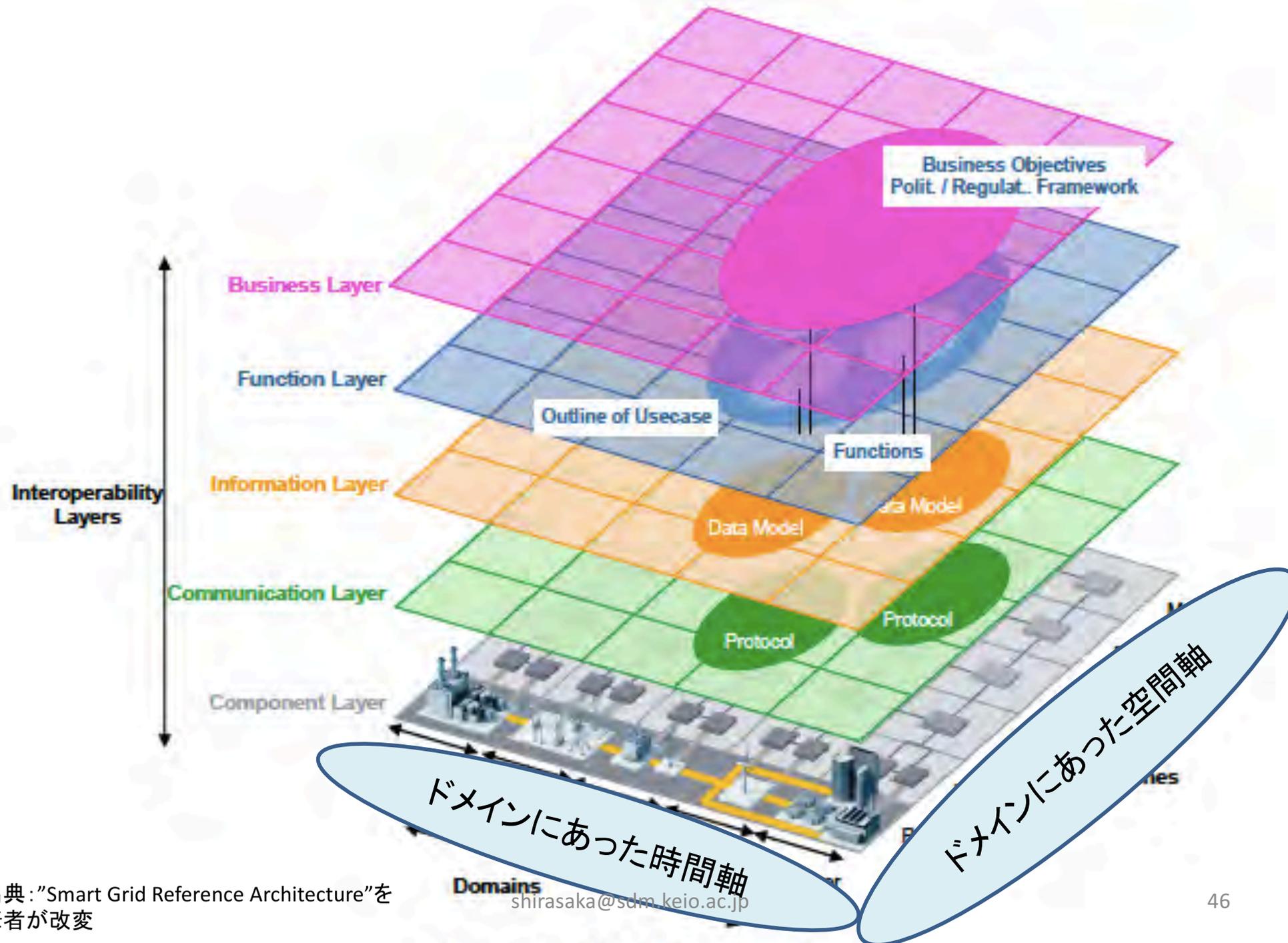
## Virtual

- 仮想的
- 目的の認識がなく、創発的に目的が達成



出典: Systems Engineering Guide  
for Systems of Systems, 2008

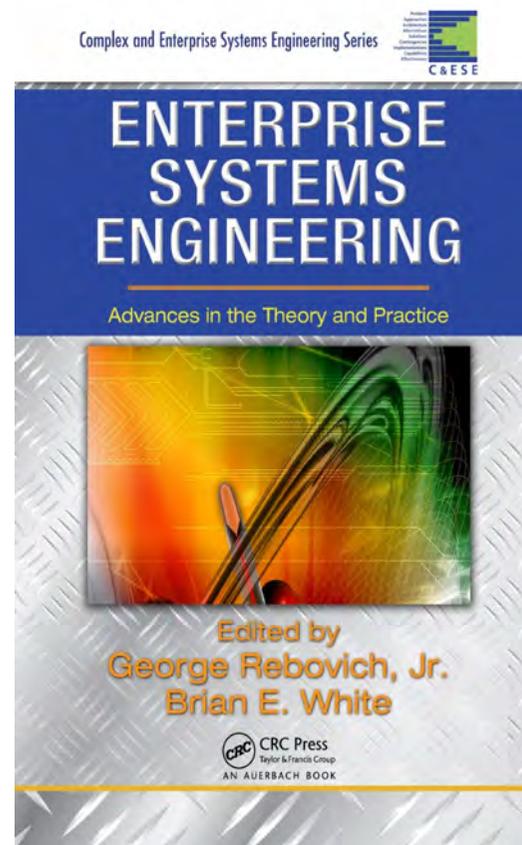
# 対象となるドメインのためのReference Architectureは？



出典：“Smart Grid Reference Architecture”を  
筆者が改変

shirasaka@sdm.keio.ac.jp

# Enterprise Systems Engineering (2010)



# Enterprise Systems Engineering

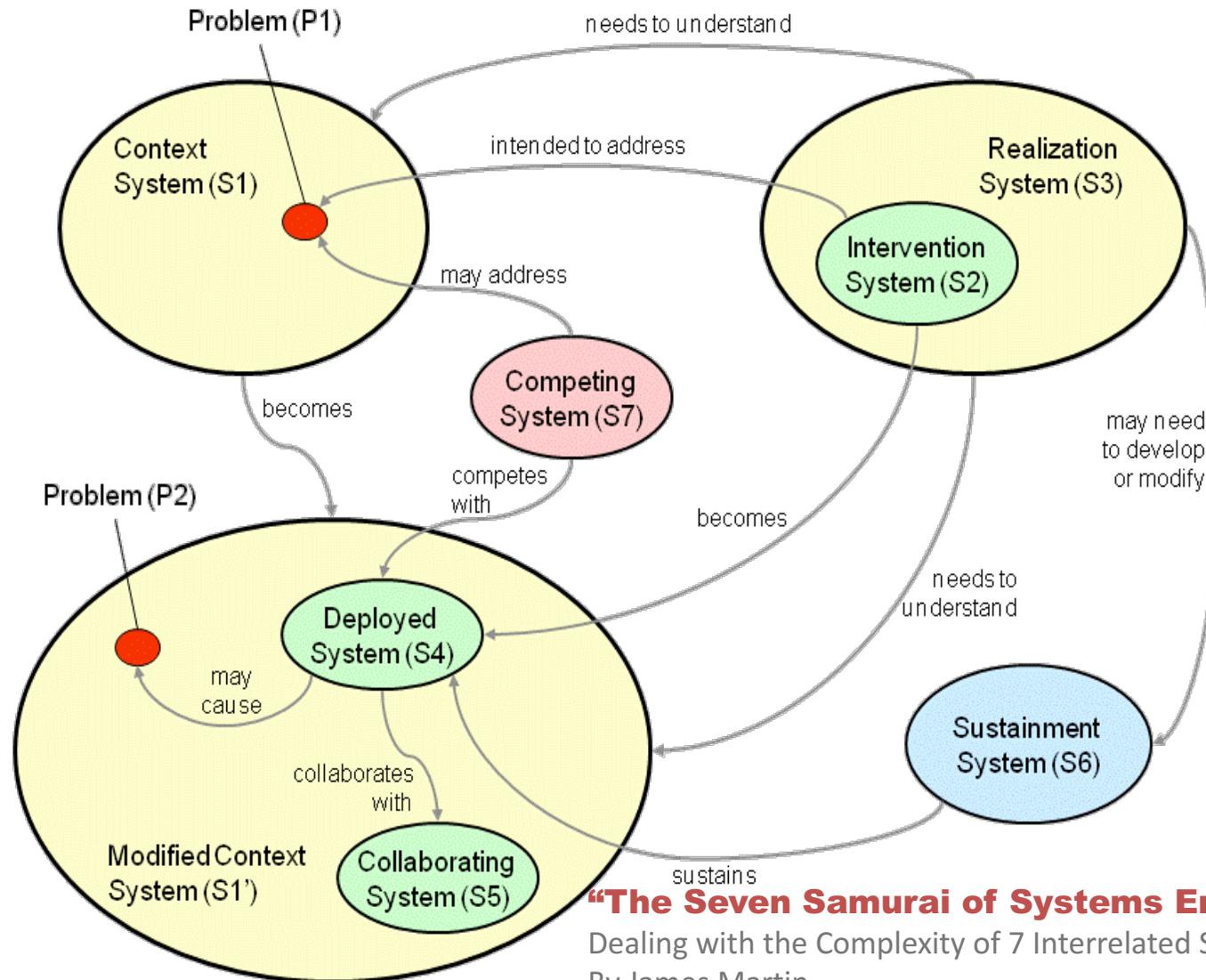
## The Seven Samurai of Systems Engineering

Dealing with the Complexity  
of 7 Interrelated Systems

**“The Seven Samurai of Systems Engineering  
Dealing with the Complexity of 7 Interrelated Systems”**

By James Martin

# Enterprise Systems Engineering: Seven Samurai



**"The Seven Samurai of Systems Engineering**

Dealing with the Complexity of 7 Interrelated Systems"

By James Martin

shirasaka@sdm.keio.ac.jp

# “Classical” Systems Engineering

1. Context System (S1) contains a Problem (P1)
2. Intervention System (S2) is intended to address P1
3. Realization System (S3) brings S2 into being
4. S2 is a constituent of S3
5. S3 needs to understand S1
6. S3 needs to understand the Modified Context System (S1')
7. S3 may need to develop or modify the Sustainment System (S6)
8. Intervention System (S2) becomes Deployed System (S4)
9. S1 becomes the Modified Context System (S1')
10. S4 is contained in (and interacts with) S1'
11. S4 collaborates with one or more Collaborating Systems (S5)
12. S4 is sustained by Sustainment System (S6)
13. S4 may cause new Problem (P2)
14. Competing System(s) (S7) may address the original Problem (P1)
15. S7 competes with S4 for resources and for user/operator attention

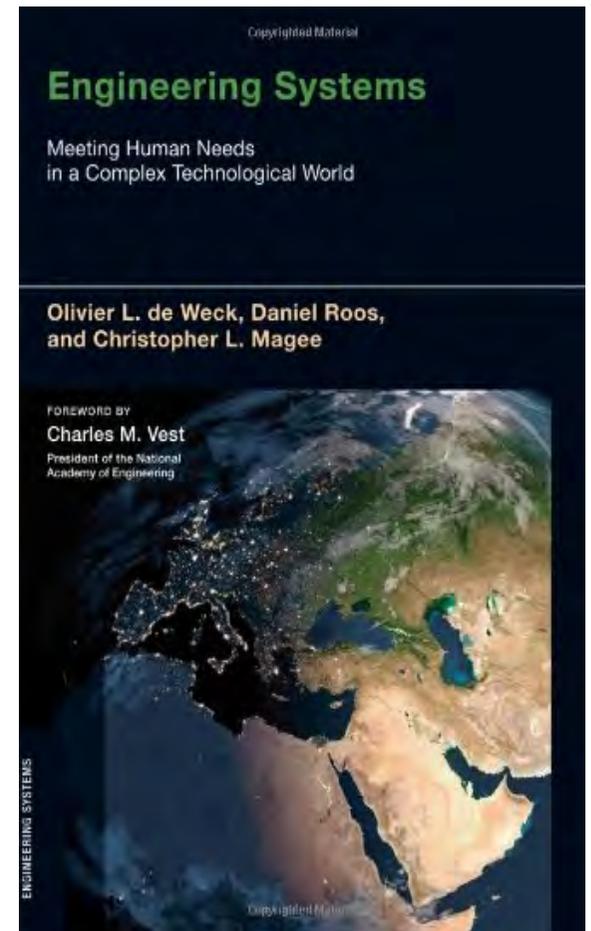
**System  
Requirements**

**Interface  
Requirements**

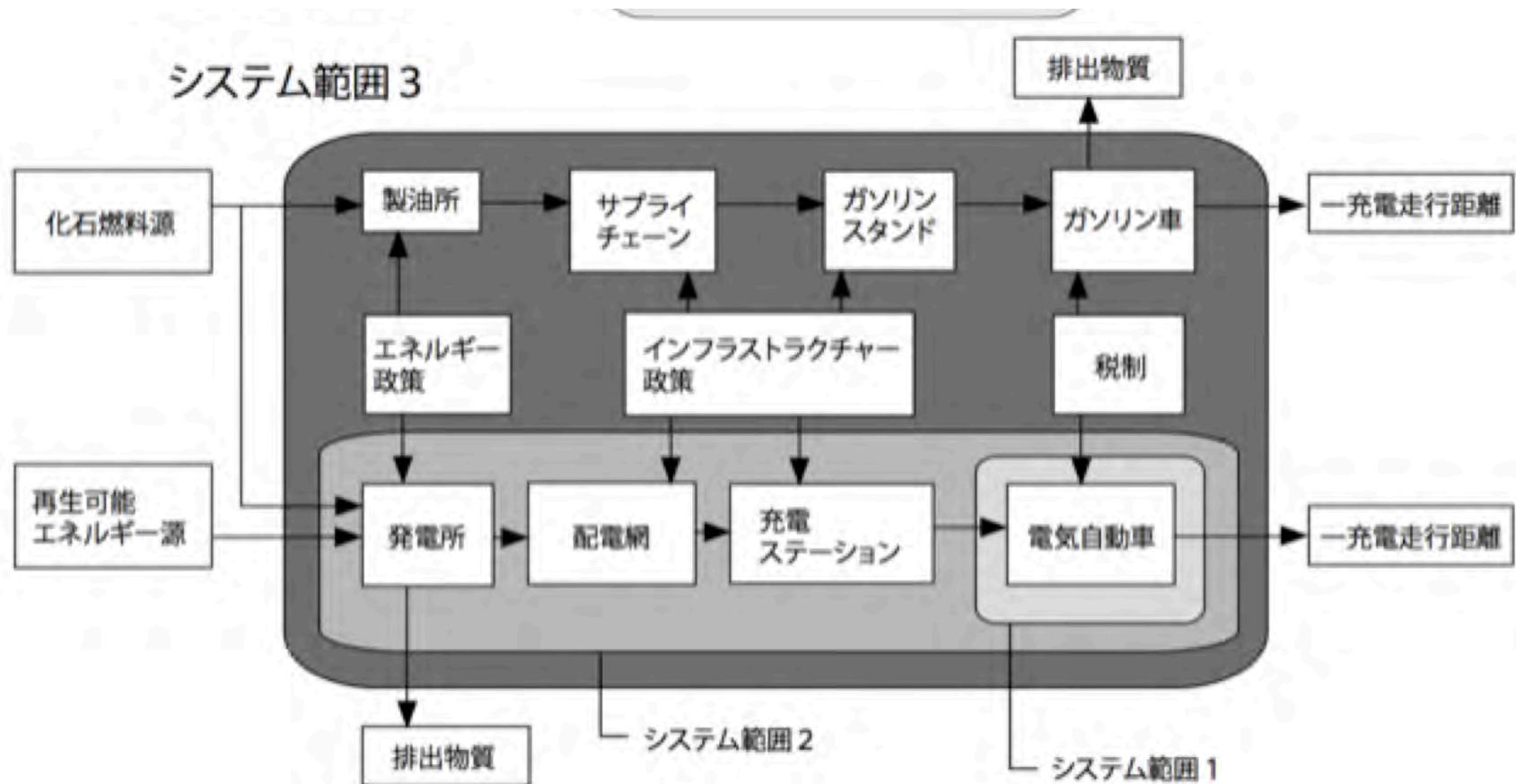
# “Enterprise” Systems Engineering

1. Context System (S1) contains a Problem (P1)
2. Intervention System (S2) is intended to address P1
3. Realization System (S3) brings S2 into being
4. S2 is a constituent of S3
5. S3 needs to understand S1
6. S3 needs to understand the Modified Context System (S1')
7. S3 may need to develop or modify the Sustainment System (S6)
8. Intervention System (S2) becomes Deployed System (S4)
9. S1 becomes the Modified Context System (S1')
10. S4 is contained in (and interacts with) S1'
11. S4 collaborates with one or more Collaborating Systems (S5)
12. S4 is sustained by Sustainment System (S6)
13. S4 may cause new Problem (P2)
14. Competing System(s) (S7) may address the original Problem (P1)
15. S7 competes with S4 for resources and for user/operator attention

# Engineering Systems (2011)



# Engineering Systems

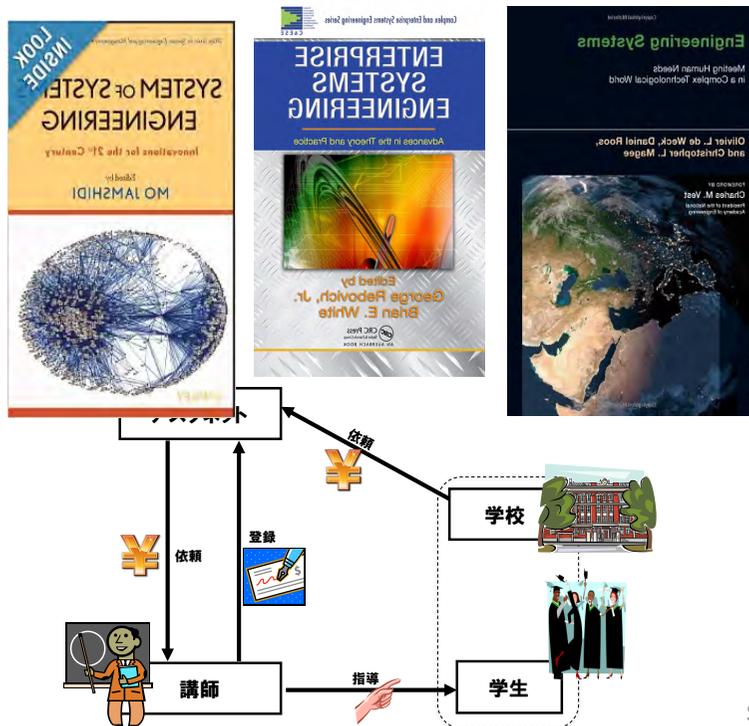


出典: Engineering Systems

# IoT時代のシステムの特徴

- システムとしての対象の拡大
- コンテキストの急激な変化
- 重要なシステム特性の増加

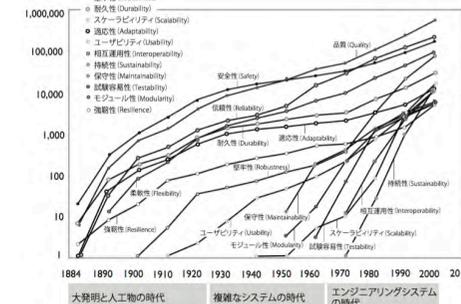
## 対象の拡大



## コンテキストの急速な変化



## 重要なシステム特性の増加



# VUCAワールド時代

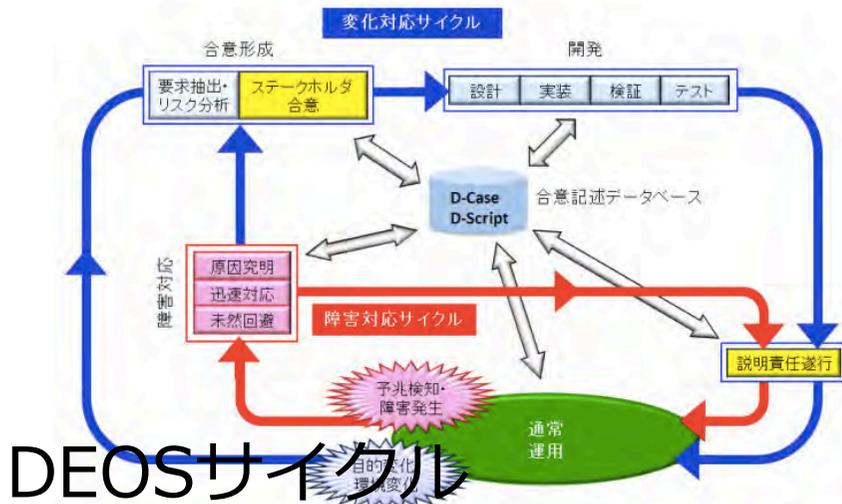
Volatility:変動, Uncertainty:不確実, Complexity:複雑, Ambiguity 曖昧

いくら考えて計画を立て、その通りに実行したとしても、周り（コンテクスト）が予想しない状態になる。

# コンテキストの急激な変化

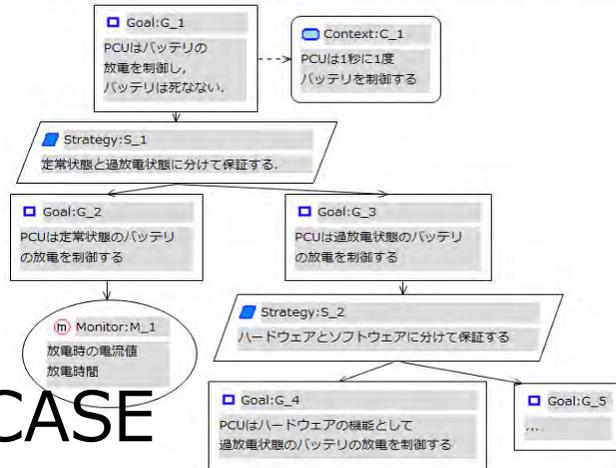
変化を捉え・対応する

変化に対応しやすくする



DEOSサイクル

D-CASE



高い抽象度（10個以内のブロック）で、自動運転全体の実現アプローチをデザイン  
論理的に実現性を確認  
安全性等の追加

環境の相違などに対して、高い抽象度レベルは固定、低い抽象度レベルは環境に依存

抽象度を上げる

抽象度を下げる  
実現手段には至らないで機能レベル

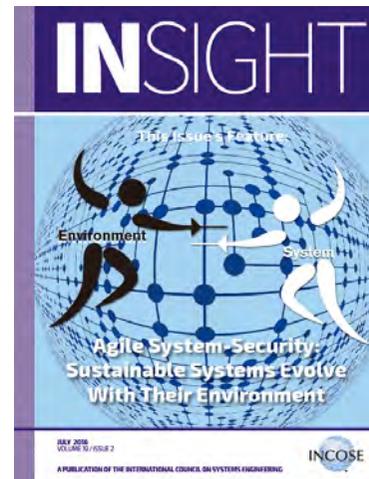
現在、実現している方法

実現手段をきめる（物理レベル）

技術の進化などに対して、機能レベルは固定、物理レベルは技術進化に依存

INCOSE System Security

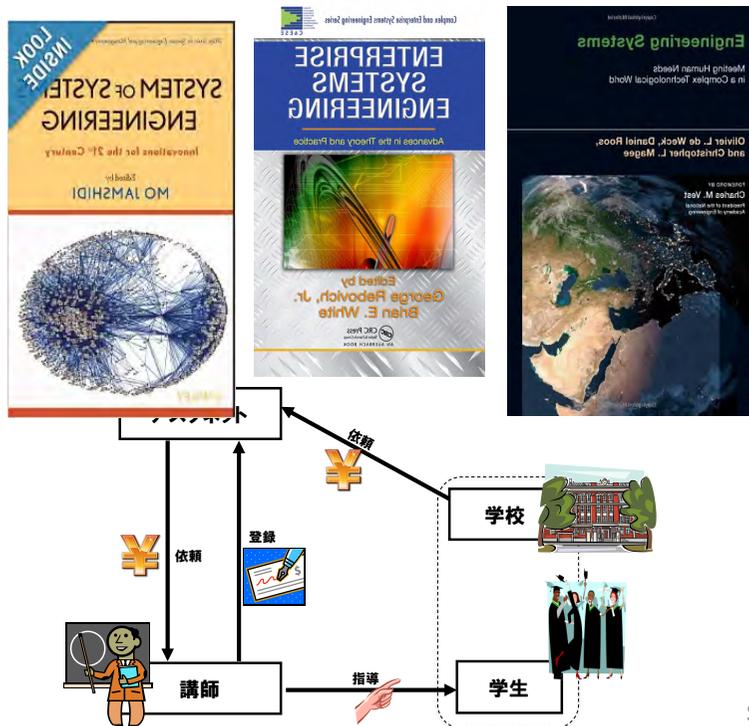
内閣府サイバーセキュリティセンタ (NISC) サイバーセキュリティ研究開発戦略



# IoT時代のシステムの特徴

- システムとしての対象の拡大
- コンテキストの急激な変化
- 重要なシステム特性の増加

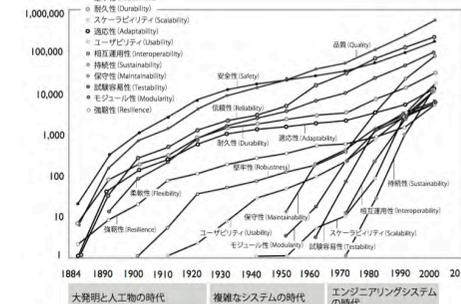
## 対象の拡大



## コンテキストの急速な変化



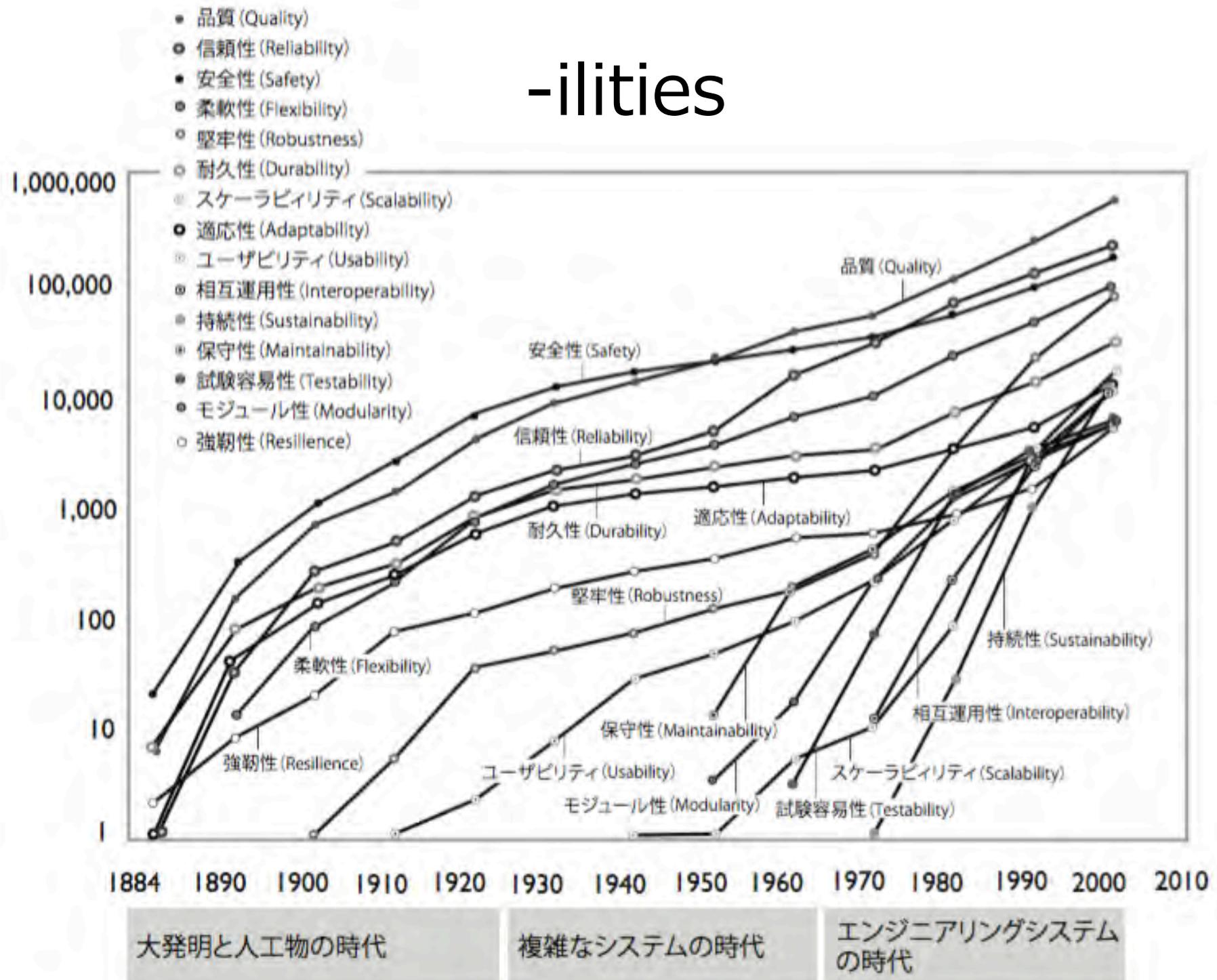
## 重要なシステム特性の増加



# -ilities

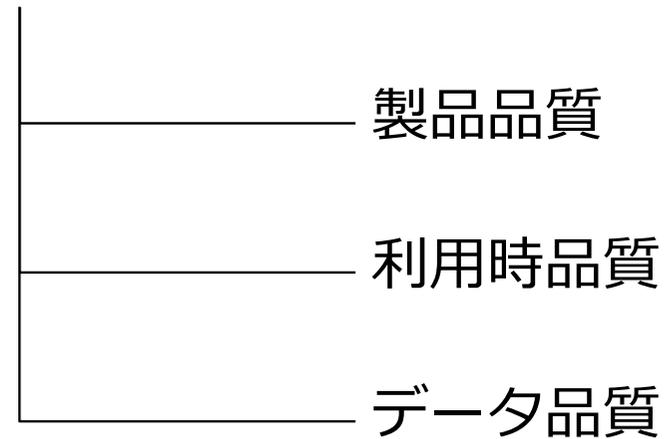
- “-ilities”はシステムズエンジニアリングのRelated Disciplineである
- より正確には“Specialty Engineering”（専門工学活動）と以前から呼ばれていた
- システムズエンジニアリングは、-ilitiesをIntegrateする必要がある
- SEBOK Part6に記載
  - Reliability, Availability, and Maintainability
  - Human Systems Integration
  - Safety Engineering 例え、STAMPはここにあたる
  - Security Engineering
  - System Assurance
  - Electromagnetic Interference/Electromagnetic Compatibility
  - Resilience Engineering
  - Manufacturability and Producibility
  - Affordability
  - Environmental Engineering

# -ilities



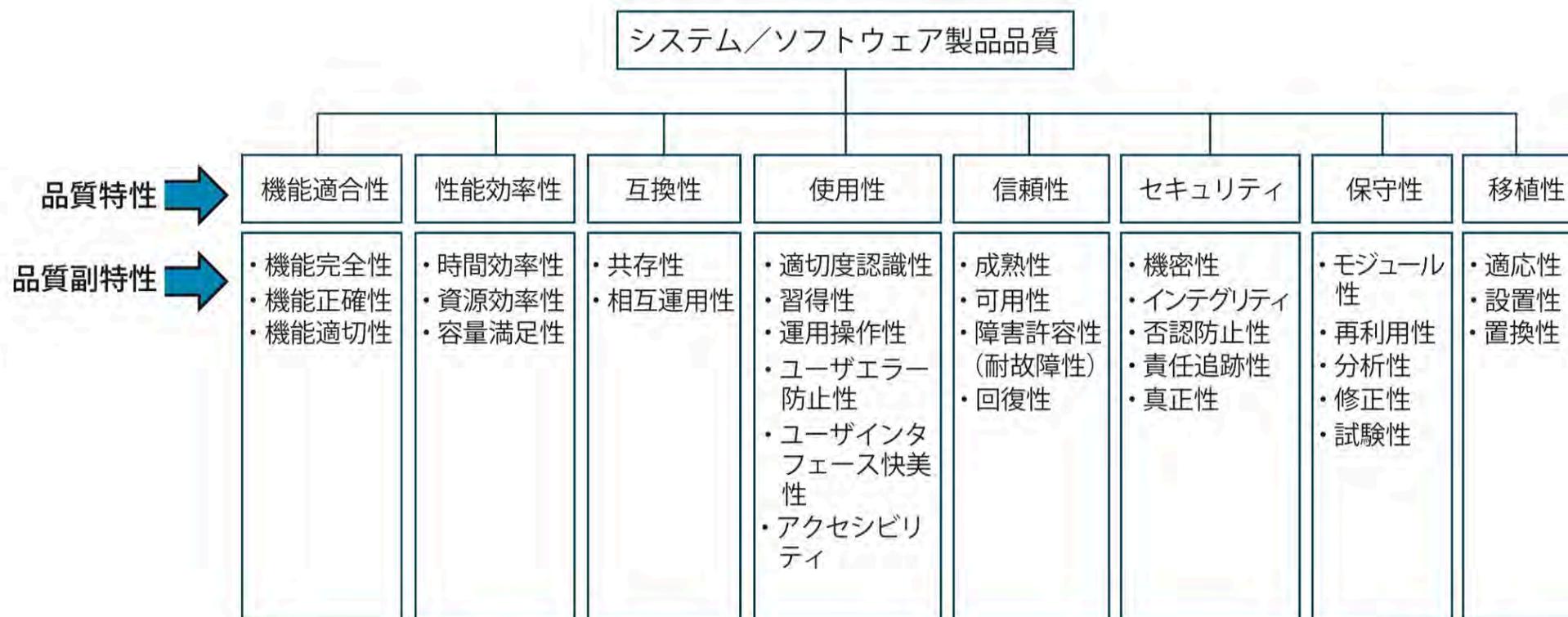
# システムの品質

## システムとソフトウェアの品質



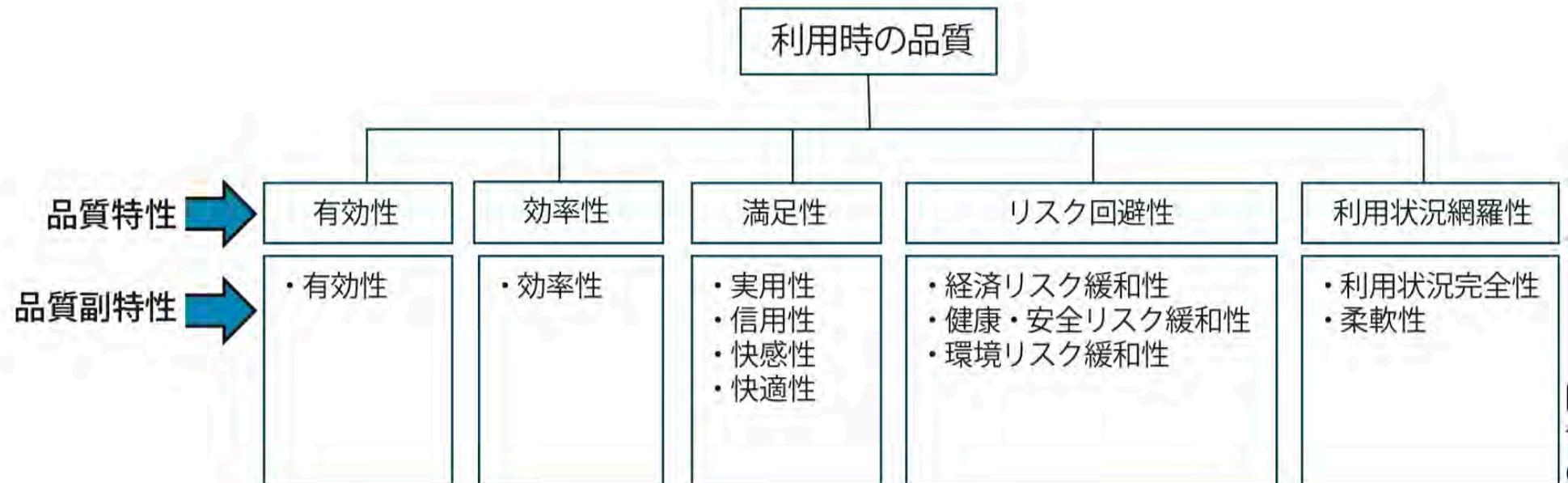
ISO/IEC 25000 SQuaRE (Systems and Software Quality Requirements and Evaluation)

# システムの品質



ISO/IEC 25000 SQuaRE (Systems and Software Quality Requirements and Evaluation)

# システムの品質



ISO/IEC 25000 SQuaRE (Systems and Software Quality Requirements and Evaluation)

# 世界と日本のアプローチの違い

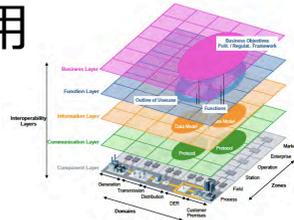
さらにその根本である思考法で勝つ！

方法論

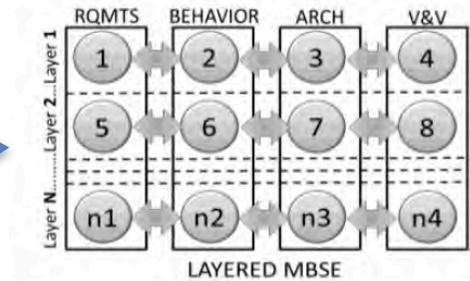
標準  
フレーム  
ワーク

単一の  
技術

ドメイン・  
ディシプリン  
専用



より汎用的

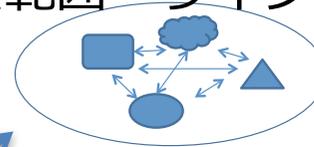


ドイツではIndustrie4.0の下で、MBSE  
研究の大学間ネットワークを構築中

通信標準等

システム標準  
機能安全など

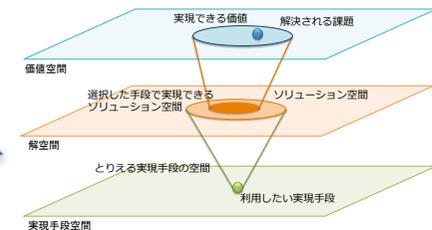
広範囲・ライフサイクル



レベル

(価値から技術まで)

単一の技術等



# まとめ

- システムの価値とシステムアーキテクチャ
- IoT時代のシステムの特徴
  - システムとしての対象の拡大
  - コンテクストの急激な変化
  - 重要なシステム特性の増加

システムとして、品質確保のために考えることが増大。これに対応できるかどうかがこのからのシステムの鍵となる！

# Design the future!

公開講座:SDMの基礎

第1回目(4回1セット)

- 9月23日(土)、24日(日)、30日(土)、  
10月1日(日)

第2回目(3回1セット)

- 11月5日(日)、12日(日)、19日(日)

詳細は、SDMのWebサイトあるいは  
白坂のブログにて。

日吉駅前 協生館