

モデルとアーキテクチャのマネジメントフレームワーク KS チャート の提案

嶋津 恵子[†] 小布施 聡[‡] 秋元 賢佑^{*} 中嶋大智^{**}

[†]産業技術大学院大学 〒140-0011 東京都品川区東大井 1-10-40

[‡]慶應義塾大学 〒223-8522 神奈川県横浜市港北区日吉 3-14-1

^{*}千葉工業大学 〒275-0016 千葉県習志野市津田沼 2-17-1

^{**}東京農工大学 〒184-8588 東京都小金井市中町 2-24-16

E-mail: [†] shimazu-keiko@aait.ac.jp, [‡] satoshi-obuse@keio.jp, ^{*} s1021009ZV@s.chibakoudai.jp
^{**}50013255076@st.tuat.ac.jp

あらまし 世界標準のシステムズエンジニアリングの工学フレームワークは、特に複雑性の高いシステムや、大規模なシステムを構築する際に、効果的であると知られている。一方、日本の産業界には、まだほとんど浸透していない。この原因の一つとして、国内の大学にシステムズエンジニアリングを教授する工学部系学科がほとんどないことであり、その理由は、実務経験の無い学生がこの工学フレームワークを直感的に理解するのが困難である点にあると考えた。これを受け、我々は、システムズエンジニアリングの、特にコンカレントエンジニアリングに注目し、重要な工学コンセプト6つを取り上げた。そして、これらの関係を俯瞰的・直感的に理解するためのKSチャートを開発し、実務経験の無い工学部系学生に貢献するかどうかを確認した。

キーワード Systems Engineering, Conceptual Mold, Consistency, View

A framework for managing systems Modes and Architecture — KS chart —

Keiko SHIMAZU[†] Satoshi OBUSE[‡] Yusuke AKIMOTO^{*} and Daichi NAKAJIMA^{**}

[†] Advanced Institute of Industrial Technology 1-10-40 Higashi Ohi, Shinagawa-ku, Tokyo, 104-0011 Japan

[‡] Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan

^{*} Chiba institute of technology university, 2-17-1 Tsudanuma, Narashino City, Chiba, 275-0016 Japan

^{**} Tokyo University of Agriculture and Technology, Nakamachi, Koganei City, Tokyo 184-8588, Japan

E-mail: [†] shimazu-keiko@aait.ac.jp, [‡] satoshi-obuse@keio.jp, ^{*} s1021009ZV@s.chibakoudai.jp
^{**}50013255076@st.tuat.ac.jp

Abstract When we build a high complex system or a large-scale system, the engineering framework of the systems engineering is known as an effective method. On the other hand, it yet has been hardly employed in the Japanese industry. It had few department of engineering system subjects which taught this as one of these causes in the domestic university, and the root cause thought that there was that the students whom there was not of the work experience understood this engineering framework intuitively. According to this situation, we paid attention to six important concepts for the concurrent engineering of the systems engineering, and developed KS chart.

Keywords Systems Engineering, Conceptual Mold, Consistency, View

1. システムズエンジニアリング

現在、地球上で発生している多くの問題が、同じ特徴を持っていることに注目する必要がある。すなわち、大規模化・複雑化・ネットワーク化であり、これによ

り問題が複合的になり、解決を困難にしていると言える。このことが、単一の高度な専門性を有している我々が、いくら長い時間を費やしても解決に到達できない最大の理由だと考えられる。

これに対し、INCOSE (The International Council on Systems Engineering)は、システムズエンジニアリングの導入を、世界規模で提案している。具体的には、ISO/IEC 12588 のシステムズエンジニアリング・ライフサイクル・プロセス標準に準拠した工学フレームワークの実践的利用の展開である[1]

1.1. ISO/IEC 15288

ISO/IEC 15288 は、1959 年に軍事産業標準として用意され、1990 年代に一般産業用標準に拡張されたシステムのライフサイクルの全フェーズに渡るシステム構築手法である。米国国防省や欧米の航空宇宙局らが収集したベストプラクティスを、一般に利用できるモデルとフレームワークとして整理された。以降、これを基盤としたソフトウェア工学標準 (ISO/IEC 12207) や要求工学標準 (ISO/IEC 29148) など、システムを成功裏に開発するための標準が網羅的に提供されている[2,3]。特に、システム・アーキテクチャの基本的なデザイン方法となる ISO/IEC 42010 (architecture description) は、システムの基盤構造特定の際に、欧米豪の産業界で広く活用されている。

ISO/IEC 15288 が提供する工学フレームワークの狙いは、対象とする問題をシステムの構築と導入によって解決することを遺漏なく進めることにある。これを実現するために、ライフサイクルの初期から終焉まで、(部分最適化だけでなく) 全体最適化を意識して工学的作業を行うよう、コンカレントエンジニアリングの実践を前提としている(2章)。視点や粒度の異なる作業プロセス群を同時に実施しながら、それらのトレードオフの最適解を保つことで、合理的なシステム設計と実装を実現することを目指している。この視点や粒度の異なる作業を統合しようとしている点が、工学に限らず複雑な要素が絡み合う社会問題の解決に適応可能ではないかと考えられる理由である

1.2. ISO/IEC 15288 導入の困難さと KS シート

複雑性、もしくは大規模性の高い問題領域を対象としたとき、ISO/IEC15288 は、有効な工学標準とされているが、日本の産業界には、ほとんど導入されていない。“V モデル” という用語が、滝型モデル (Waterfall Model) と同様の意味で用いられている程度であり、ベストプラクティスの報告までは至っていない。

我々は、大学の工学部系にこの教育が展開されていないことが原因の一つであり、かつその理由は実務経験のない学生にこの標準を理解させることの困難さにあると考えた。

これに対し、我々は、モデル(model)とアーキテクチャ(architecture)と、さらに両者を関係づける consistency や、conceptual model を直感的に理解させることで、解決につながると推察した。

これらを達成するための KS シートを開発し、その

効果を確認した。

1.3. 本書の構成

本書は次の構成を採る。2 章にシステムズエンジニアリングの本質を、コンカレントエンジニアリングの側面と、統合工学の側面から概説する。3 章にシステムズエンジニアリング実践の困難さを、コンカレントエンジニアリング実践時の 6 つの重要フレームワークを用いて説明する。そして、4 章に困難さを軽減する KS シートを示し、5 章にインタビューから得た評価とコメントを整理する。6 章と 7 章に、それぞれ考察とまとめを述べる。

2. システムズエンジニアリングの本質

2.1. コンカレントエンジニアリングの側面

システムエンジニアリングは、その知識体系 (SEBoK: systems Engineering Body of Knowledge) においても説明されているように、領域や担当の異なる専門家たちが、それぞれの英知を最大限活用し、成功裏にシステムを構築する方法論である。そして、対象とする問題をシステムの構築と導入によって解決することを遺漏なく進めることを目的としている[4]。これを実現するために、ライフサイクルの初期から終焉まで、(部分最適化だけでなく) 全体的視点による最適化 (holistically) を意識して工学的作業を行うよう、並列作業 (concurrently) の実践を前提としている。一般に、コンカレントエンジニアリングと称する手法であるが、システムズエンジニアリングでは、これを実施するシステム構築モデルとして V 字型モデル (以降“V モデル”と表記) を採用している。これは、直感的には、滝型モデルとスパイラルアップモデルの利点を組み合わせた方法である[5]。

滝型モデルに代表されるシステム開発は、利用者や出資者の意図や期待に詳しい担当者 (所謂「上流担当者」) が、ステークホルダのニーズを取りまとめ、それを受け取った外部仕様設計者がシステムのアウトラインをデザインする。その結果を入力情報として、内部構造の担当者がこれを行う。引き続きその結果を入力情報として、より詳細な設計の担当者が作業を行う。これを繰り返すことで、最小部品までの実装方法を特定する。この方法は、マネジメントしやすい一方で、より前段階で不備が発生し、より後段階でその事実が発見されると、修復に計画を大きく超えるリソースが必要になる。そこで、これを回避するために、スパイラルアップモデルが考案された。これは、滝型モデルの全段階を 4 回繰り返す (recursion) ことで、当初から修正リソースを組み込もうとしたものである。とこ

ろが、4回繰り返してもなお、全体最適化が実現されないことが多発した。

システムズエンジニアリングで導入しているVモデル型開発は、先行するこれら2つのモデルの利点を踏襲している。これにより、全体最適化を合理的・効率的に実現することを目指している。具体的には、滝型モデルで、段階的に実施していた粒度の異なる作業を並行して実施する。あらかじめ定められたレビューごとに、それぞれの出力(設計のモデル)の整合性を確認する。このレビューでの整合性合わせの焦点が、時系列に従い、外部仕様から始まり、回数を重ねるごとに最小構成物の構造の視点へと粒度が下がる。この並びがV字型になることから、モデルの名称として使われている(図1)。V字型に並ぶレビューでの作業をシステムズエンジニアリングのコア(core)と呼び、これを横断・並行して進行する各プロセス群での作業をオフ・コア(off core)と呼ぶ[6]。

V字型の左側に相当する期間の各プロセス群の作業は、対象システムの構成の特定である。より具体的には、システムを構成するのに最適な構成要素(コンポネントとインタフェースとコンポネント同士の関係)の検討と特定の作業である。この時に、物理的構成物だけでなく、その特定根拠となる、機能的構成や操作上の概念構造も特定する。さらにそれらの間で矛盾が発生しないようにすることが、最も肝要な作業である。一方、この作業が煩雑性を生む、また実務経験の無い学生に直感的な理解を難しくしている。

2.2. 統合工学の側面

一方、図5に示すV字の右側に相当する期間では、各プロセス群の作業で特定された構造に正確に従って、それらを順に統合していく。部分最適と全体最適が実現されることは、すでにV字の左側で確認されているので、右側の期間はこの決定(仕様)どおりに正しく統合することが主たる作業となる。従って、システムズエンジニアリングでは、この作業を“integration and verification”とし、徹底した検証のもとに進めるフレームワークが紹介されている。

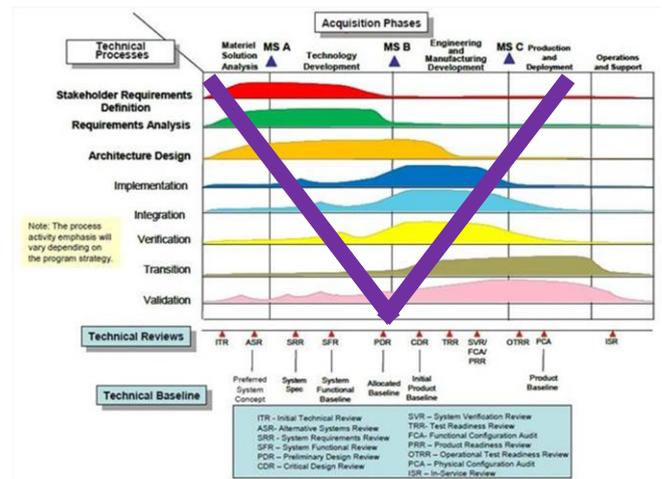


図1 Vモデル型開発：コンカレントエンジニアリングと全体最適化のためのレビュー

3. システムズエンジニアリング実践の困難さ

特にエンジニアリングの実務経験の乏しい国内の工学部系学生にとってのシステムズエンジニアリング実践の困難さは、前章に示した2.1節の直感的な理解の難しさにあると考えた。対照的に、2.2節の統合工学の側面の実践は、作成された設計書を含む全仕様に正確に準拠して部品を統合する作業であるため、国内工学部系学生にとっては得意な分野であると考えられる。

ただし、この統合作業結果であるシステムが、利用現場に導入された際、成功的な結果を生むかどうかは、2.1節の作業で、特にシステム・アーキテクチャをどのようにデザインしたかに左右される(図1のY列の見出しの上から3つ目の“Architecture Design”)。

これを実践するには、少なくとも次の工学的概念を取り扱うことが必要になる。

- ① モデル(model)：設計対象となっているシステム(system-of interest)を表現する特徴点を特定し、その情報だけで表現する方法。換言すると単純化操作であると言える。目的に応じて表現する特徴点異なるため、ビューごとにアブストラクション・レベルを調整し、モデルは複数用意される。
- ② アブストラクション・レベル(abstraction level)：モデル作成時に、特定した表現する特徴点が明確に表現できる程度。特にソフトウェアエンジニアリングでは、抽象度や粒度と呼ぶことがある。
- ③ ビュー(view)：モデルを作成する際の、(アブストラクション・レベルとは別に)視点を変えて、必要なモデルを用意するための複数の見方。システムの種類に関わらず、最低限、functional viewとphysical viewとoperational viewが必要だとされる。
- ④ コンセプト(concept)：それぞれのviewで、検討作

業を行っているステークホルダ間で、最も端的に説明できるモデル (図 2)。

- ⑤ トレーサビリティ (traceability) : 同一ビュー内で、アブストラクション・レベル毎に整備されたモデル間で、互いの要素が漏れなく・余分も無く、完璧に対応づけられること。システムズエンジニアリングでは、モデル作成時に、必須の要件。
- ⑥ アーキテクチャ (architecture) : ビュー毎に用意されたそれぞれの複数のモデル間で、ビューを横断して整合性 (consistency) がとれるモデル群。構成するモデルは、いずれもコンポーネント群とそれらをどのように統合するかインタフェースと、さらにインタフェースで統合された場合のコンポーネント間の関係が明示される。

以上のようにシステム・アーキテクチャは、一つのモデルで表現させるのではなく、サブモデル群で表現される。

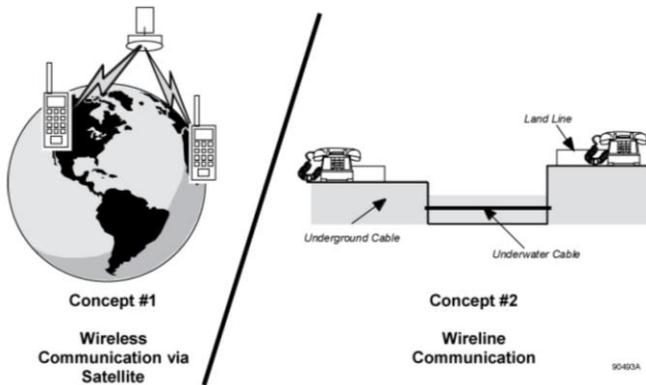


図 2 コンセプトの例 [6]

4. KS シート

前章に述べた 5 つのシステムズエンジニアリングの概念と、それらの関係を直感的に把握し、議論・検討している焦点の明確化と、ステークホルダ間での共有化を実現するために KS チャートを開発した (図 3)。

X 軸にシステム・アーキテクチャをデザインするのに必要なビューを配置する。システムの特徴や利用ドメインに関わらず、最低 3 種類、つまり、機能 (Function)、物理 (Physical)、そして利用 (Operation) のビューが必要だとされる。ビュー同士の関連性はなく、それぞれで作成されたモデルは、他のビューのいずれのモデルとも直接的な依存関係は無い。ただし、アーキテクチャを構成するモデルに関しては、ビュー間で整合性が取られる (後述)。

Y 軸はシステム・アーキテクチャをデザインする際のアブストラクション・レベルのメジャメントとなる。アブストラクション・レベル異なるモデル間で、(同一ビュー内では) 完全なトレーサビリティ (traceability) を保

つ必要がある。

つまり、ビュー毎に、トレーサビリティを保った複数のモデルが、それぞれ複数用意されることになる。これらの中で、ビューを横断して整合性 (consistency) が担保されるモデル群が、システム・アーキテクチャとなる。

この方法で、つまり KS シートを参照しながらステークホルダ間で議論と検討を行うことで、自然にコンカレントエンジニアリングを実施しながら、モデルの整備とアーキテクチャのデザインが可能になると考えた。

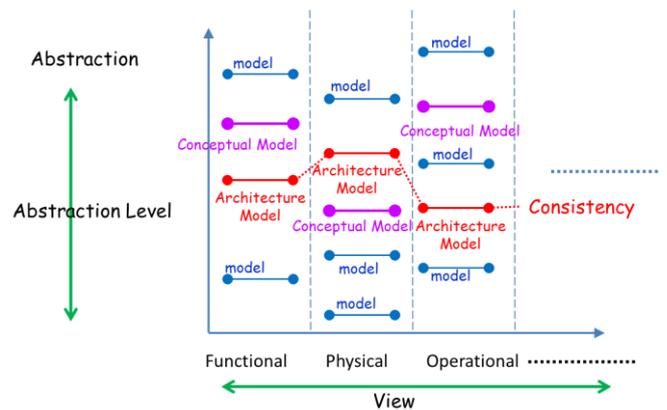


図 3 KS シート

5. KS チャートの評価

開発した KS チャートの効果を次の方法で確認した。エンジニアリング実務経験の無い (もしくはほとんど無い) 国内の大学の工学部系学部に通学する学部生および大学院生に対し、無償自由参加型のシステムズエンジニアリング・ゼミナールを開催した。このゼミナールで、3 か月 (毎月 2 回の開催、1 回 90 分) かけて、座学方式でシステムズエンジニアリングの解説を行った。主に、3 章に示した 6 つの工学概念の説明と演習である。この後、4 か月目に KS シートの解説と使い方を説明した。受講者に、KS シート導入前と導入後の、3 章に示した 6 つの工学概念に関する理解度に関し、インタビューを行った。

インタビューに応じた受講者は 11 名であり、うち 8 名が KS シートはシステムズエンジニアリングの主要な工学概念の理解に大変役立つと回答し、残る 3 名はある程度役立つとした。さらに、より具体的な感想や評価を求めたところ、次ページの表に示した内容を入力した。

表 KS チャートに対する意見 -インタビュー結果-

複数回答を含む・原文のまま掲載

1	modelのイメージがあれば、よりわかりやすくなると思います。
2	役立つと思います。横軸は単にviewを表している、粒度は関係ない(?)と思うのですが、国・県・市などと言われると混乱してしまうかと思いました。
3	とても分かりやすかったです。アブストラクション・レベルの高さの違いを自転車などの具体的な例で説明して頂ければより分かりやすかったです。
4	国境、河川の具体例は、粒度を変えただけで、"view"は同じではないか?と勘違いしやすいと思った。
5	今まで Model と Architecture をごちゃごちゃに考えているところが有ったので、具体例を混ぜた KS シートの説明によって理解することが出来て、良かったです。
6	KS シートで分析し、整合性を取れた Architecture は何向けのものかを明確にしたほうが良いと思います。今回は View の国、川、建物の軸で分析しましたが、それを整合性を取れたことでどんなものが作れるかが不明瞭です。
7	システムズエンジニアリングのライフサイクルの中で、どこで重点的に使うのに役立つのかを知りたい。
8	view ごとそれぞれで見たときのアブストラクション・レベルは直感的に理解できるのですが、別々の view 同士においてどちらがアブストラクション・レベルが高いか迷うことがあるかと思いました。
9	それぞれの View の Abstraction level が Consistency が取れた時にのみ、それぞれの線が横 1 列になるという機能があれば、より分かりやすかったです。(すいません、その方法は思いつきませんでした。)
10	整合性が取れる線が直線である必要性が無い事が初見では分からない。
11	何かしら説明を手助けるような例があると良いと思う。
12	毎回説明していただきましたことを別のサンプル図として作成して一緒に提供したらもっとわかりやすくなると思います。
13	KS シートの課題ではないと思いますが、ビューの設定を確実にやる必要があると感じました。
14	言葉を聞いてその意味だけを知っても具体的にどういうものなのかが曖昧になってしまい、アーキテクチャーを考えるのが困難ですが、KS シートを用いて考えるとそれが考えやすくなったと思った。
15	コンセプトモデル、アーキテクチャモデル、view、Abstraction レベルの概念が一枚に整理され、大変理解しやすいと思います。前回、前々回よりもさらに分かりやすくなりました。
16	システムズエンジニアリングの初心者に対して、model・view などの概念をわかりやすく説明できる。
17	VIEW 間を横断して俯瞰する時に、ARCHTECTURE MODEL で CONSISTENCY が取れているか? SYSTEM DESIGN MODEL で TRASABILITY が取れているか、を理解しやすい。
18	architecture 設計の大切さがわかりました。
19	状況が整理しやすい
20	粒度の違いを図に示すことで、よりそれぞれのモデルの違いが理解し易くなった。
21	アーキテクチャと言うものが視覚的に理解出来る

6. 考察

前章に記したインタビューの回答結果から、全員が KS チャートの効果を認めていると言える。また、11人のインタビューに対し、21件の具体的な感想を入手できたことは、KS シートに対する興味と期待の高さを示しているとも考えられる。

一方、入手した感想の内容を見ると、導入の効果の高さを認める一方で、(1)システム開発のライフサイクル中のどの段階するとよいのか、(2)異なるビュー間のモデルのアブストラクション・レベルを調整し、整合性を取る方法の不明確さ、また特に(3)実際のアーキテクチャ・デザインに利用した事例の提示など、具体的なエンジニアリング作業中での使い方を求める声が際立っている。より実践的・具体的な教授法を求めていると判断できる。

7. まとめ

システムズエンジニアリングの工学フレームワークは、特に複雑性の高いシステムや、大規模なシステムを構築する際に、効果的な方法であると知られている。一方、日本の産業界には、まだほとんど浸透していない。これらの原因の一つとして、国内の大学にこれを教授する工学部系学科がほとんどなく、その根理由は、実務経験の無い人材がこの工学フレームワークを直感的に理解するのが困難である点にあると考えた。これを受け、我々は、システムズエンジニアリングのコンカレントエンジニアリングの重要な工学コンセプト6つに注目した。これらの関係性を理解し、操作できるようになることが、システムズエンジニアリングを理解することに直結すると考えたことがその理由である。

そこで、これらを一覧にして、現在、議論・検討しているのがどの位置に相当するのかを把握するための KS チャートを開発した。

実務経験の無い、学生を中心にインタビューをおこなったところ、KS チャートはシステムズエンジニアリングの理解促進に貢献することがわかった。その一方で、より多くの事例を用意し、深い理解を期待することも入手できた。

これらを反映し、次のテーマとして、引き続き取り込む予定である。

謝辞

KS チャートの利用に関し、システムズエンジニアリングツールの開発に取り組んでいらっしゃる鳥取大学工学研究科 機械宇宙工学専攻 助教 三浦政司先生と議論させていただき、有益なコメントをいただきました。

文 献

- [1] INTERNATIONAL STANDARD, ISO/IEC/IEEE 15288, 2015-01-15, Systems and software engineering – System life cycle processes, 2015
- [2] INTERNATIONAL STANDARD, ISO/IEC/IEEE 12207, 2008-02-11, Systems and software engineering – Software life cycle processes, 2008
- [3] INTERNATIONAL STANDARD, ISO/IEC/IEEE 29148, 2011-12-01, Systems and software engineering – Requirement engineering, 2011
- [4] [http://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_\(SEBoK\)](http://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK))
2015年12月1日参照
- [5] K. Forsberg etc., Visualizing Project Management, Charts and Frameworks for Mastering Complex Systems, Third Edition, pp. 245-246, John Wiley & Sons. 2005.
- [6] Cecilia Haskins, SYSTEMS ENGINEERING HANDBOOK , A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES, INCOSE -TP -2003 -002 -03.2.2, October 2011 pp. 100, 2011