

## 15-A-15

### 製品開発における SysML 適用の取り組み<sup>1</sup>

～要求の可視化～

#### 1. 製品開発への SysML<sup>2</sup>適用の背景

ネットワークに接続して使用する最近のプロジェクタなどのシステム製品開発においては、製品単体のみではなく製品間の連携による新たなユーザー価値を素早く提供することが重要となっている。プロジェクタ製品開発の例を図 15-A-15-1 に示す。

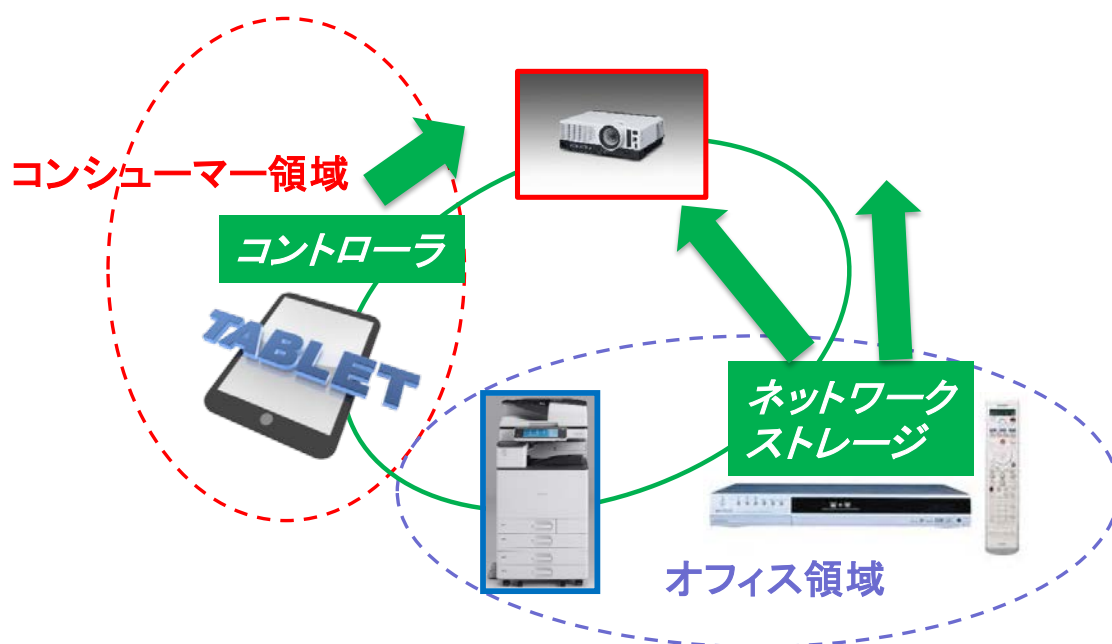


図 15-A-15-1 システム製品開発の例

また、このような製品の適用領域も、オフィス領域に閉じていた時代からコンシューマー領域にまで拡大してきている。コンシューマー領域での特徴は、ユーザーの利用シーンがパソコンだけを前提に利用していたオフィス領域とは異なり、スマートフォンやタブレットといったモバイル端末と組み合わせて使用されるなど、多様化してきていることである。また、オフィス領域においてもコンシューマー領域と同じ操作感、同じ機能（普段利用しているモバイル端末での操作）を求められることが増えてきており、領域間の境目がなくなってきた

<sup>1</sup> 事例提供: 株式会社リコー 画像システム開発本部 森 勇仁 氏、I W開発本部 馬場 洋 氏

<sup>2</sup> Systems Modeling Language

いる。

その結果、製品開発においては、全てを自社で開発するのではなく、自社開発オフィス領域製品と既存のコンシューマー領域製品を連携させることで素早く価値を提供することが必要となり、システム製品開発プロセスそのものに対する要求も多様化・複雑化してきている。

このような状況のなかで、①お客様、販売担当、商品企画担当など各ステークホルダーの要求を正確に捉えるとともに、②開発対象となるシステムの複雑化と大規模化に伴い発生するソフトウェア・ハードウェア・メカ等各技術領域間にまたがる製品開発における課題（性能や、操作性等）を適切に解決し、さらに③多様化・複雑化した仕様を漏れなく文書化し実現するための仕組みが必要となる。

本編では、これらを解決するために、システム全体を記述できるモデリング言語 SysML の要求図を中心に適用することで、①目的・要求を獲得・明確化し、②各技術領域にまたがる課題に対して目的・要求に合った解決手段の提案を行い、③製品仕様の決定を行った事例を紹介する。

## 2. 解決すべき課題と SysML を適用した狙い

### (1) システム統合における課題

市場で出回っている汎用製品の NAS<sup>3</sup> やスマートフォン／タブレット、アプリ等は無数に存在し、これらの製品同士を連携させることは以下の理由により難しい状況にある。

- 標準規格が規定されていたとしても、詳細に仕様が規定されていない部分もある。また、規格の記述に曖昧さがある場合は、実装者により実装上の解釈が異なる場合がある。
- 端末・アプリの種類により処理起動要求から処理開始までの時間などのタイミングが微妙に異なるため、これらを各端末・アプリに合わせ込むのに多大な時間を要する。
- 利用シーンも、タブレット・プロジェクタ本体それぞれで行われる操作やそれらの組み合わせ等が複雑になってきており、組み合わせ時や処理要求競合時の扱いなど、仕様の確定や検証作業も複雑化している。

### (2) 技術領域間の意思疎通における課題

近年、システム製品開発においては、開発規模増大、期間短縮、効率向上など様々な理由によりハードウェアとソフトウェアの分業が加速してきていることや、異なる技術領域(メカ設計／エレキ設計／ソフトウェア設計)毎に、成果物の形式が異なり、各技術領域の担当間で認識の齟齬が発生する恐れがある。このため、各部門で設計したものを統合すると思わぬ問題が表面化し、手戻り作業が発生することがある。

---

<sup>3</sup> Network Attached Storage

## (3) 課題解決のために適用した技術

(1)、(2) の課題を解決するためには、システム全体を統一して記述すること、各技術領域共通で理解できることが重要と考え、モデリング言語 **SysML** を適用した。

その狙いは、以下の通りである。

- 複雑に絡み合った要求を可視化し整理することで、システムの境界と開発対象システムの要件を明確にすることを可能とする。
- 複数の技術領域で理解できる共通の言語を利用することで、担当者間の誤解をなくし効率よいシステム設計を可能とする。

本プロジェクトでは **SysML** のダイアグラムである要求図を中心に、パラメトリック図、ブロック定義図、アクティビティ図を適用することとした。

### 3. 導入のための事前準備や工夫

## (1) 活動を開始する上での準備

準備として以下を実施した。

① **SysML** に関する知識の集積

ソフトウェア設計では以前より **UML** を用いた開発を進めており、**UML** と親和性の高いモデリング開発手法である **SysML** の情報収集や知識の集積に努めた。

## ② 社外有識者からの情報収集・知識習得

**SysML** ベースに標準プロセスを提供するツールベンダーのセミナーに参加するとともに、社外コンサルティングによる知識獲得を実施した。

## ③ 表記法とプロセスの習得

ソフトウェア開発要員だけでなくエレキ設計要員を含めて、半年近く毎週 2 時間程度のワークシェアリング活動を実施し、表記法とプロセスの習得を行った。

## ④ 探求活動

我々の抱えている課題に対する効果があるかどうかを検証するため、小テーマでの実践的探究活動を実施した。

## (2) 活動するにあたっての工夫

既に制定・稼働している部署開発プロセスおよび成果物（ドキュメント）を変更するのではなく、成果物そのものの品質・精度を向上させる活動として組み込むことで、スムーズに展開することができた。

また、新しい手法の導入を成功させるには、マネージャーがその考え方を理解し、成果物に対する技術的な判断ができることを重要と考え、マネージャーに対する技術教育を実施した。

## (3) 使いこなせる人材の育成

活動前の準備として、基本的なスキルを身につけたメンバーが、他のメンバーに対してワークショップ形式の勉強会を実施した。これにより人材育成を行い、スキル保

有者を増やすと共に、教えることでさらに理解を深めることを狙った。担当者への一律の教育ではなく、テストによる客観評価と自己診断による主観評価で各担当者のスキルレベルを規定し、スキルレベルに応じた計画的なスキルアップ教育を実施した。スキルアップ教育の概要を図 15-A-15-2 に示す。

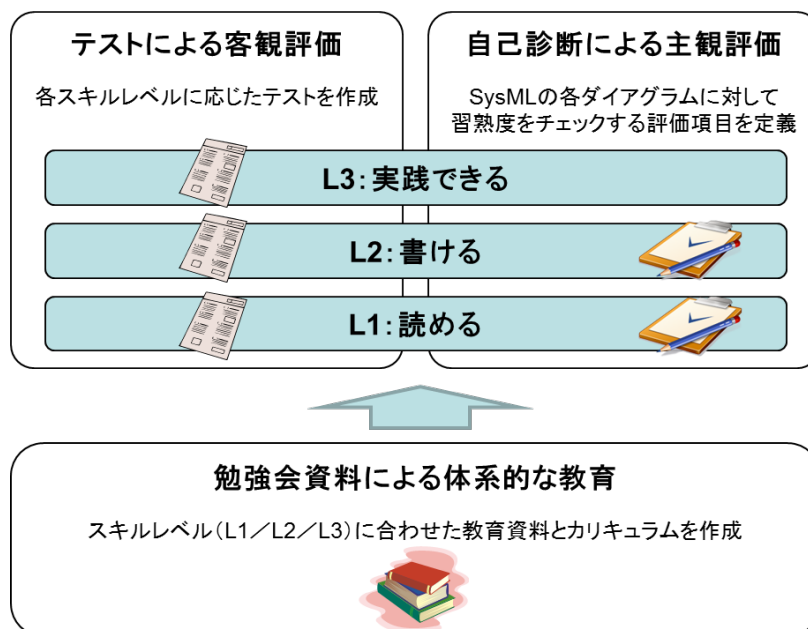


図 15-A-15-2 スキルアップ教育の概要

## 4. 製品仕様作成への SysML 要求図の適用

### 4.1. 要望の多様化・複雑化への対応

様々なステークホルダー（お客様、販売担当、商品企画担当など）からの多様化、複雑化する要望に対して、適切な製品仕様を定め実現することが、製品を開発する上で非常に重要な開発作業の一つである。

適切な製品仕様を定め、実現するために重要なポイントは以下である。

- ・ 要望を集め、その目的・背景を正確に捉えること。
- ・ 捉えた要望にそった製品仕様ならびに実現手段を技術トレンドや自社保有技術を考慮し決めること。
- ・ 性能要件を含め漏れのない製品仕様を決定すること。

上記のうち一つでも曖昧なまま開発を進めると、大きな開発手戻りの発生や望まれない製品を開発することになる可能性が高くなるため、非常に重要なポイントであると考えている。

これら3つのポイントを実施し、適切な製品仕様を確定するための課題は以下の通りである（図 15-A-15-3）。

#### 4.1.1. 課題

- ① 要望元はお客様や販売担当者、商品企画担当者など、様々である。また、これらの人々から寄せられた要望は多様化しており、効果的に対応しないと時間もコストもかかってしまう。
- ② シーズ面では技術トレンドや自社保有技術などがあり、これらを使い多様化した要望に対して、目的・要求にそった解決手段を提案することが必要である。しかし、共通の技術認識がないと、数ある解決手段の中で、目的や要求にあった最適な解決手段を決めきれないことがある。
- ③ 製品開発については開発期間の短縮化が求められており、系統的・網羅的に検討を行わないと、仕様が不明瞭なまま開発に着手してしまうことがあった。

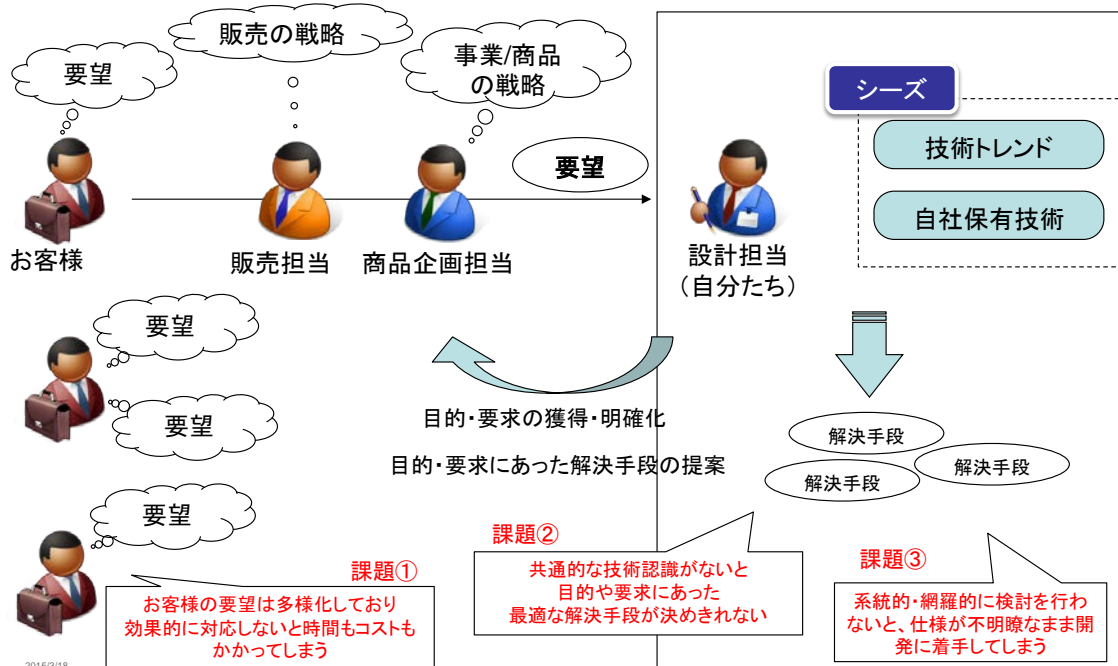


図 15-A-15-3 適切な製品仕様の確定のための課題

## 4.2. SysML の要求図の適用

4.1.1 ①～③の3つの課題に対処するために SysML の要求図を適用した例を図 15-A-15-1 に示す。

要求図の適用方法には以下の特徴がある。

- 背景、目的が理解し易いように図 15-A-15-1 の上部を目的、図 15-A-15-1 の下部を手段となるようにまとめている。
- 開発の段階にあわせて3つの Step に分けて進める。

SysML の要求図は要求間の関連を明らかにすることができ非常に有用である。しかし、自由度が高いため、作成時に図自体の目的を明確にしていなければ、何をどの程度詳細に記載

すればよいのかが不明となり、膨大な時間を費やしてしまい、かつ、膨大な作成量になりやすい。そこで3つのStepに分け、要求図を作成する目的、ステークホルダーを明確にした。

3つのStepは、

- ① 開発対象のコンセプト・狙いを決める
- ② 狙い(What)を実現する手段を洗い出す
- ③ システムとして実現する製品仕様を決定する

であり、詳細については4.3で記載する。

#### 開発対象のコンセプト・狙いを決める

##### (要求・目的)

背景や狙いを整理することで、優先順を見極める

##### 課題①

お客様の要望は多様化しており効果的に対応しないと時間もコストもかかってしまう

#### 狙いを実現する手段を洗い出す

##### (機能)

Whatを実現する解決策を洗い出すことで、機能を決める

##### 課題②

共通的な技術認識がないと目的や要求にあった最適な解決手段が決めきれない

#### システムとして実現する製品仕様を決定する

##### (仕様)

網羅的に検討することで、漏れのないシステム仕様を決める

##### 課題③

系統的・網羅的に検討を行わないと、仕様が不明瞭なまま開発に着手してしまう

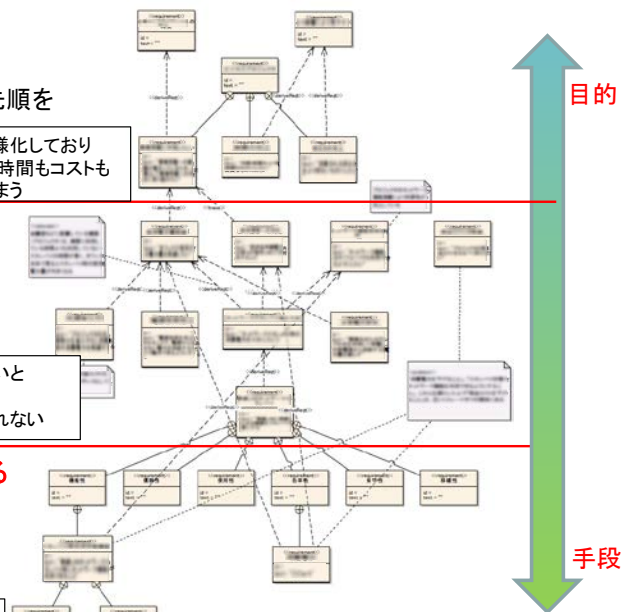


図 15-A-15-4 要求図と3つのStep

### 4.3. 各StepへのSysML要求図の適用

#### (1) 開発対象のコンセプト・狙いの決定

このStepでは、課題①の解決のため、各要求の背景・狙いを整理する目的で、要求図を利用する段階である。ステークホルダーへのヒアリングや世の中の動向、競合他社の製品分析が主な活動となる。ここでは、設計者だけでなくステークホルダー（販売担当者、企画担当者など）と共同で要求図を作成することで、ステークホルダーが気づいていなかった狙いや矛盾が見えるようになり、優先順を見極めることが可能になった(図15-A-15-5)。大事な点は、すべてを向上させるのではなくこだわるポイント（差別化）を決めて狙いを絞ることである。

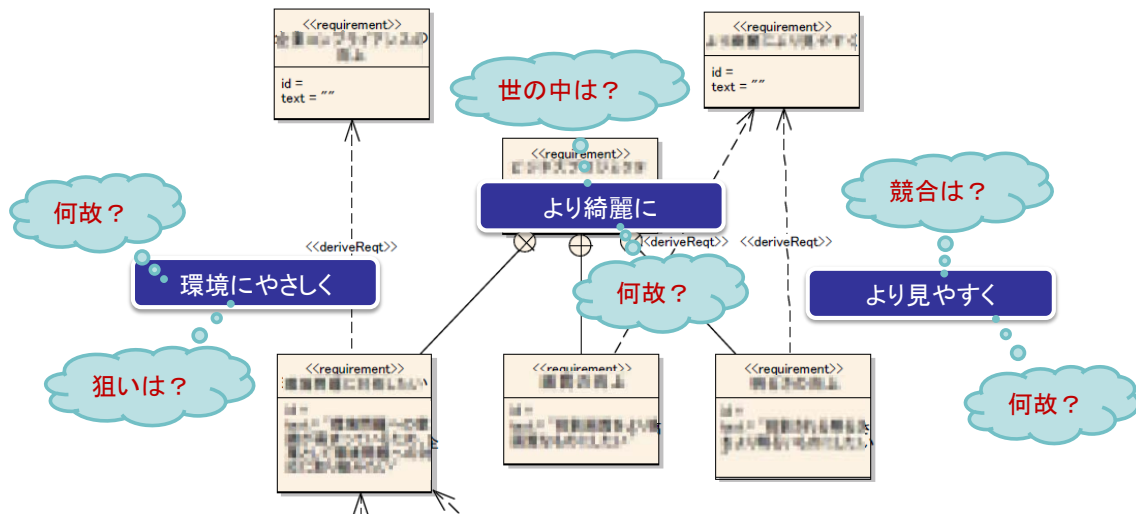


図 15-A-15-5 要求の要求図化

## (2) 実現する手段の洗い出し

この Step では、課題②の解決のため、(1) で絞りこんだ狙いに対して実現手段を複数抽出し、それぞれの実現手段の特徴を整理する目的で要求図を利用する段階である。ここではステークホルダーに対して、方式のメリット・デメリット（他方式との差異）を正確に理解してもらい、実現手段を決めることが重要である。その際、できる限りさまざまな実現手段をあげておくことで、特徴が明確になりやすいと考えている（図 15-A-15-6）。

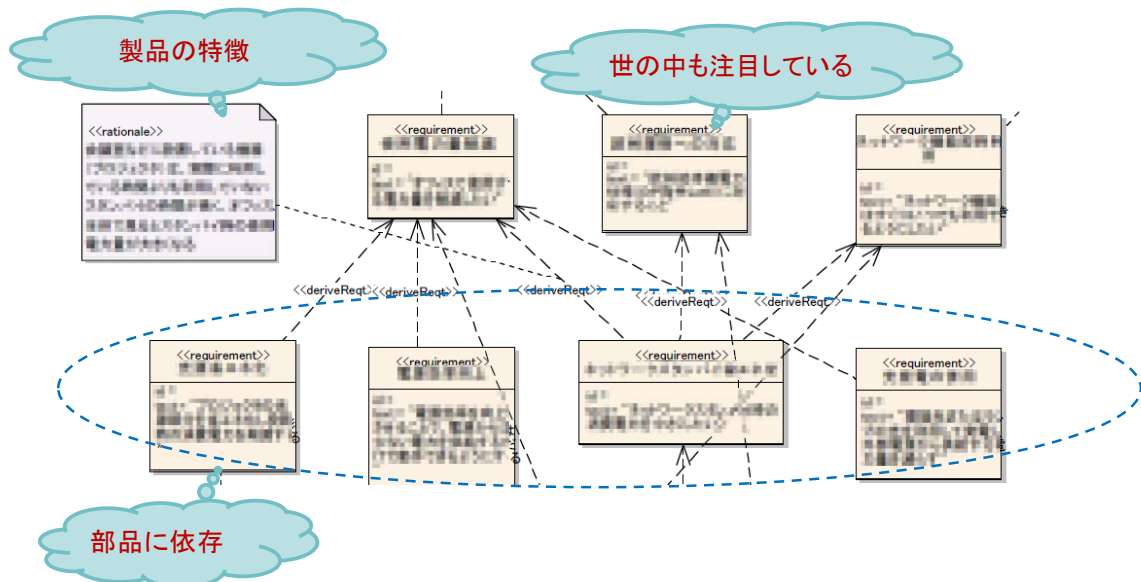


図 15-A-15-6 実現手段の洗い出し



## (3) 製品仕様の決定

この Step では、課題③の解決のため、(2) で決定した実現手段に対する製品仕様を漏れなく抽出する目的で要求図を利用する段階である。分類することにより、検討できている項目、検討不足の項目を視覚的にわかりやすくする効果がある。分類の例として、我々は ISO/IEC 9126 (JIS X 0129) <sup>4</sup>で規定されているソフトウェア品質特性 (Software Quality Characteristics) の利用や、既に社内で利用している製品仕様の分類などが考えられる。社内設計者が理解しやすい分類が好ましいと考える。(図 15-A-15-7)。

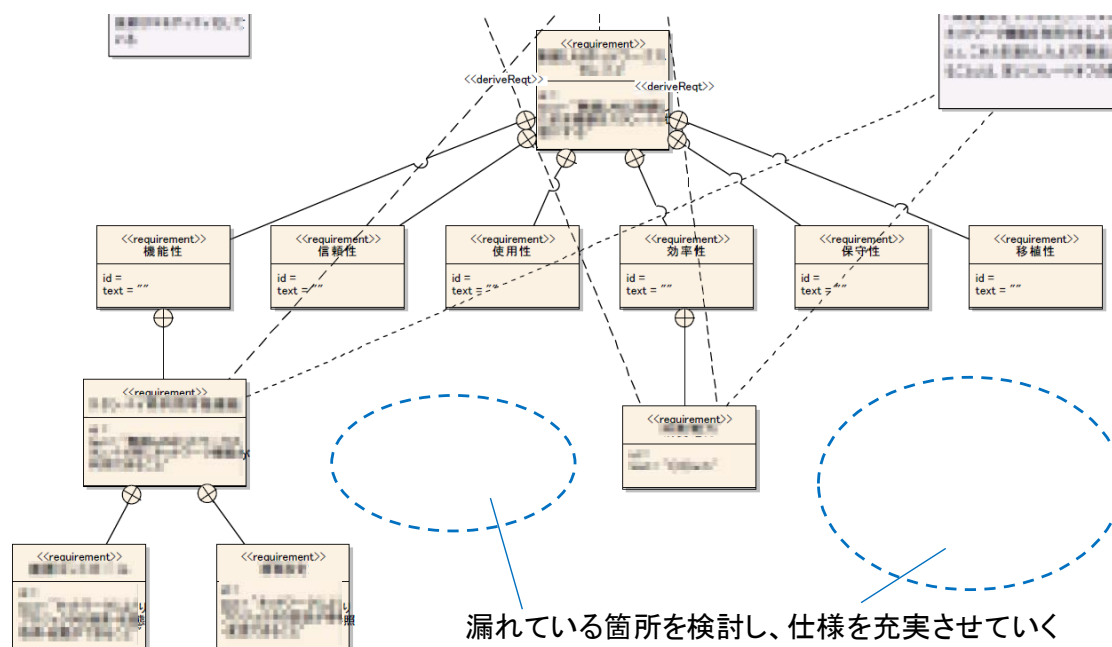


図 15-A-15-7 製品仕様の決定

## 5. ソフトウェアとハードウェアの協調設計への SysML ブロック定義図・アクティビティ図の適用

### (1) ソフトウェアとハードウェアの個別開発における課題

製品の開発においては常に、開発期間の短縮化、開発作業の効率化、技術の進歩に対応していくための専門分野化が求められてくる。さらに既存製品の開発を経験している場合は、既存製品をベースとした設計によりソフトウェアとハードウェアが個別に開発を進めてしまいがちになる。

その結果、各部門で設計したものを統合すると思わぬ問題が表面化し、手戻り作業が発生することがあった。

個別開発による具体的な課題としては、

<sup>4</sup> 再編され、ISO/IEC 25000 (SQuaRE) シリーズとして整備されている。



- 既存製品での責務分担を元に、新機能の責務分担を各自が想像して進めてしまうことにより認識のズレが発生する。
- 既存製品をベースにした設計から大きく変わることがないため、製品仕様（特に性能要件）に対して、達成できないことが起こりうる。または、気づいたときには手遅れとなる場合がある。

などが挙げられる。

## (2) SysML のブロック定義図・アクティビティ図の適用

上記課題を解決するために、製品仕様の決定後に、SysML のブロック定義図・アクティビティ図を利用することで、製品の振る舞いを定義し、全体の構造を分解して、ハードウェア仕様とソフトウェア仕様を決定することを徹底した。その結果、各設計部門が協調してシステム全体を検証することができ、要求仕様・製品仕様を満たした製品開発を実施した。

ソフトウェア設計者とハードウェア設計者が共同で活動するためには、双方の担当者が理解できる共通の言語が必要であり、その点で SysML は有効であった。

## (3) ネットワーク通信の例

省エネ性を考慮したネットワーク通信を設計した際に発生した問題を紹介する。

ネットワークを介して動作制御できる製品では、常にネットワークからの制御信号を受け付けるための電力が必要となる。そのため、利用しないときの消費電力を極力小さくしたいというのは重要な要求である。

ソフトウェア設計者とハードウェア設計者が個別に検討した場合、それぞれの認識がズレることで最適な性能を出すことができないことがある。図 15-A-15-8 は、パケット解析処理のための電力を、エレキ担当者が考慮できていなかったため、ソフトウェア担当者と認識の相違が生まれた例を示している。

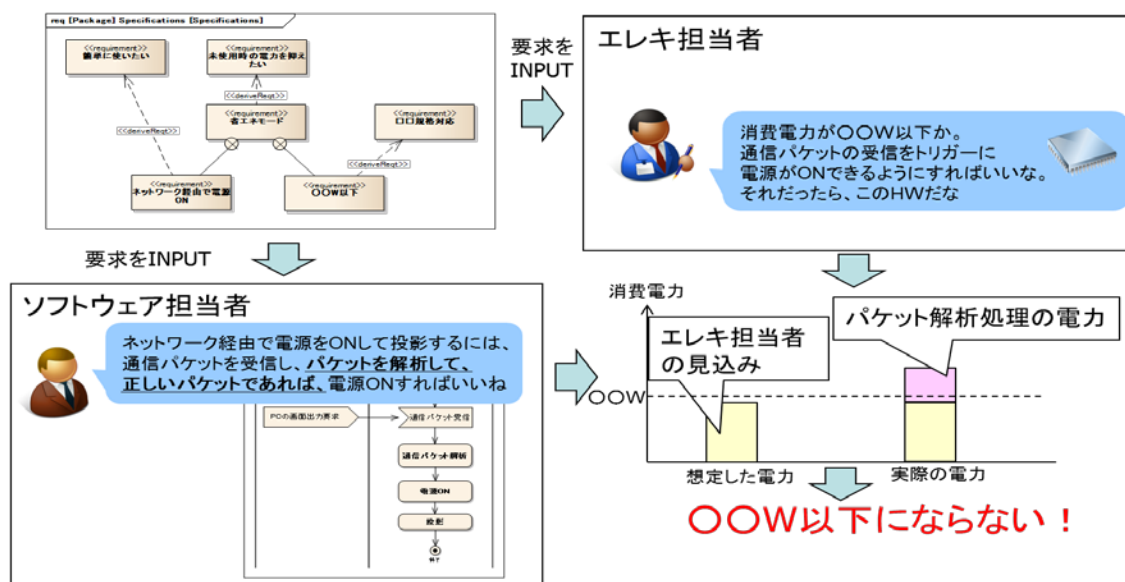


図 15-A-15-8 個別設計時の問題事例

ここでは、製品の振る舞いを共同で定義し、最適な構造を決定するフェーズを設けることで対策を取ることができた。図 15-A-15-9 は、省エネ時でも動作する部分（電源管理、無線 LAN、通信処理）と電源 ON 時のみ動作する部分（投影装置、映像処理）を、ソフト・エレキの担当者が共同で洗い出し、そのためのハードウェアを分離した例である。この結果、省エネ時に動作するハードウェアの消費電力は減少し、パケット解析処理のための電力を考慮しても目標の消費電力を実現できた。

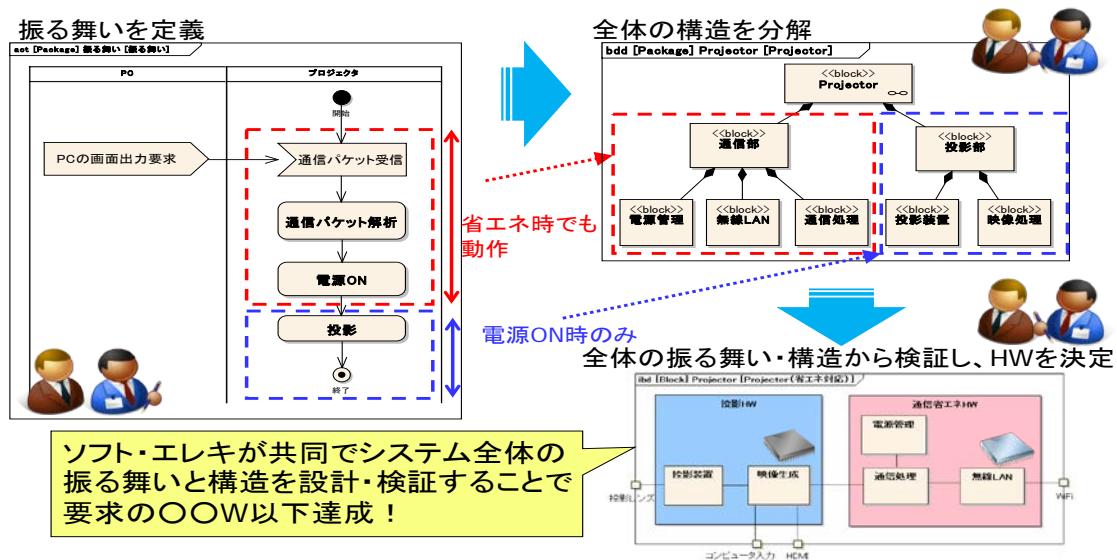


図 15-A-15-9 協調設計の例

## 6. 性能設計への SysML 要求図・パラメトリック図の適用

### 6.1. 性能とコストのバランス

性能に関する要求のすべてを満たせばベストな製品になるが、現実的には性能とコストのトレードオフが必ず発生する。ここでの課題は、

「限られた資源の中で最適なバランスを見つけ出す」

ことである。

解決策として着目したポイントは、狙い・用途（利用シーン）によって重要視する性能が違ふということである。そこで、狙い・用途（利用シーン）を明らかにするための要求図の利用と、性能決定要素間の関連を明確にするパラメトリック図を活用することを実施した。

### 6.2. SysML の要求図を活用した重要視する性能の明確化

最初に求められた要求は具体的な機能であることが多い。そこで要求図を利用することで目的・狙いを明らかにし、目的・狙いに沿った手段を導くことで、目的・狙いごとに重要視

する性能が違うことに気づくことができる。

背景・目的を明確にし、目指す性能目標を決めていく事例を図 15-A-15-10 に紹介する。

ネットワーク経由（無線）で PC の画面をプロジェクタに対して投写する機能を要求された場合の事例である。その際、画質、操作感、滑らかさが性能要件として挙げられる。しかし、全ての性能要件を満たすにはコスト（高性能な部品を利用すればコストが高くなる）があわない、あるいは、技術的な限界（画質、操作感、滑らかさ）が想定される（図 15-A-15-10 の①）。そのため、各性能要件とコストの最適なバランスを検討していく。

まず始めに、要求の背景・目的を要求図により整理する（「4.3 (1) 開発対象のコンセプト・狙いの決定」を参照）ことで狙い・利用シーンを明確にする。何のために画面を投写するかを明らかにすることで複数の狙い・利用シーンが浮かび上がってくる。この時点で狙いを一つに絞ればベストだが、機能レベルの要求を分析している場合、絞りきれないことも多い（図 15-A-15-10 の②、③）。狙い・利用シーンの特徴を分析すると重要視する性能を抽出できる。例えばプレゼンテーションに利用する場合、文字をはっきりと見せることが重要となるため、高画質がより求められる（図 15-A-15-10 の④）。

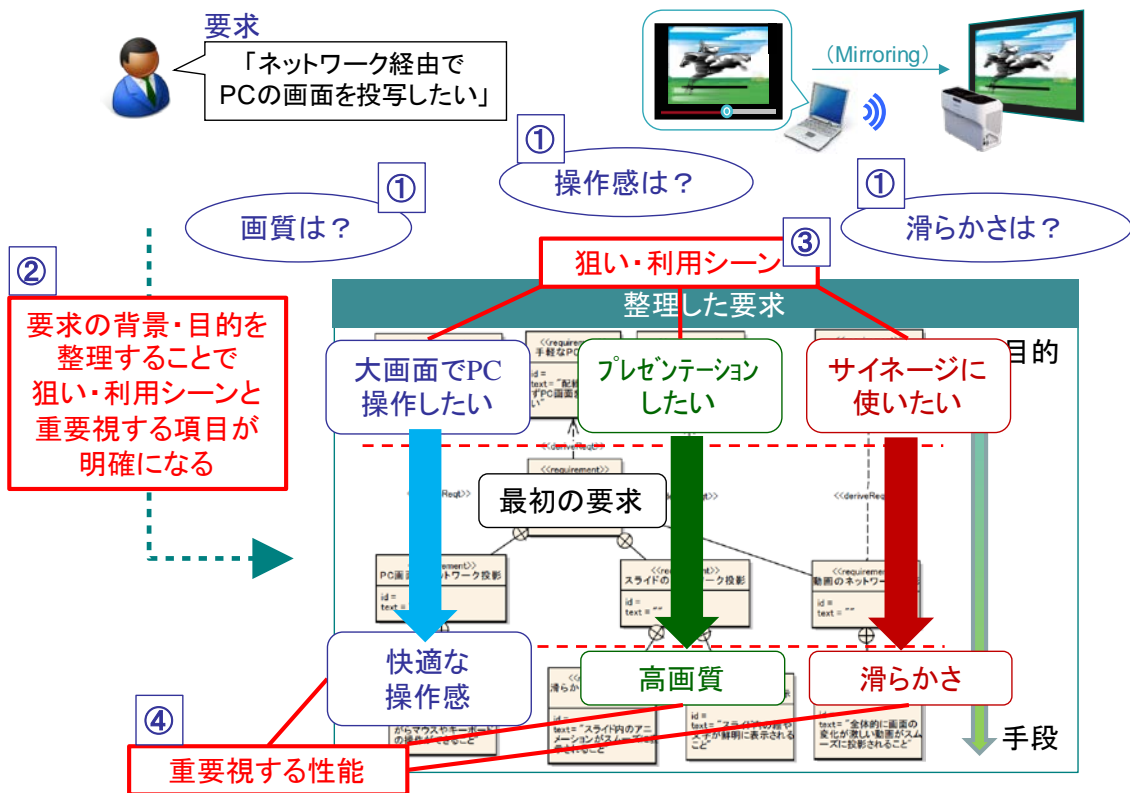


図 15-A-15-10 要求把握と性能設計

### 6.3. SysML のパラメトリック図の適用

性能は、様々な設計因子間の関係により決定されるため、重要視する性能をできる限り高くにするためには、設計因子間の関係を明らかにし、シミュレーションすることが必要である。詳細設計を着手する前の段階でステークホルダーとシミュレーション結果を共有することで、トレードオフの関係を正確に共有でき、開発終盤でのステークホルダーとの認識のズレによる開発手戻りをなくすることができる。

設計因子間の関係を明らかにし、シミュレーションを実施するための課題は、

- 設計因子の洗い出し
- 設計因子間の関係を数式化
- 複数の数式を組み合わせ、システム全体の設計因子間の関係をモデル化

となる。

#### (1) 設計因子の洗い出しと数式化 (SysML のブロック定義図の利用)

性能はいろいろな要素により決まるが、その全てを最初から網羅することは熟練の技術者でなければ難しい。PC 画面の投写の場合、

- 画質を良くすると、画像データのサイズが大きくなる
- データサイズが大きくなると、通信時間が長くなる

といった要素・関係が考えられるが、全てを洗い出すためには、少しずつ分割していくことがポイントとなる。

最初の分解は、PC とプロジェクトに分けるレベルからでよい。その場合の画面投写の処理時間は以下に分解される。

- PC での処理時間
- ネットワークを利用したデータ通信にかかる時間
- プロジェクトでの処理時間

次に、それぞれの処理時間に関わる設計因子を洗い出し、計算式を出す。ここで計算式が出せない場合は、計算式が出せる粒度になるまで、ブロック定義図を利用しシステムを分解することが重要である。

#### (2) 複数の数式を組み合わせ、システム全体の設計因子間の関係をモデル化 (SysML のパラメトリック図の利用)

(1) の数式を出した段階では、計算式が複数でできている状態であるため、組み合わせることで、システム全体レベルでの計算式をモデル化することができ、重要視する性能に対して、どの設計因子がどのように作用するかをシミュレーションすることができるようになる。

ここで、コストとのトレードオフが明確になり、狙い・利用シーンに合わせた最適なバランスについて、ステークホルダーと認識をあわせることができるようになる。

パラメトリック図を適用した例を図 15-A-15-11 に示す。図 15-A-15-11 の左側は PC 画面共有機能の構造と性能項目を具体化したブロック図で、右側が設計因子間の関係

を表したパラメトリック図である。ここでの設計因子は、PC 処理時間、プロジェクトの処理時間、ネットワーク転送時間である。複数の設計因子間の関係から最適な設計を決め、用途に合わせた最適な性能を達成することが可能になる。

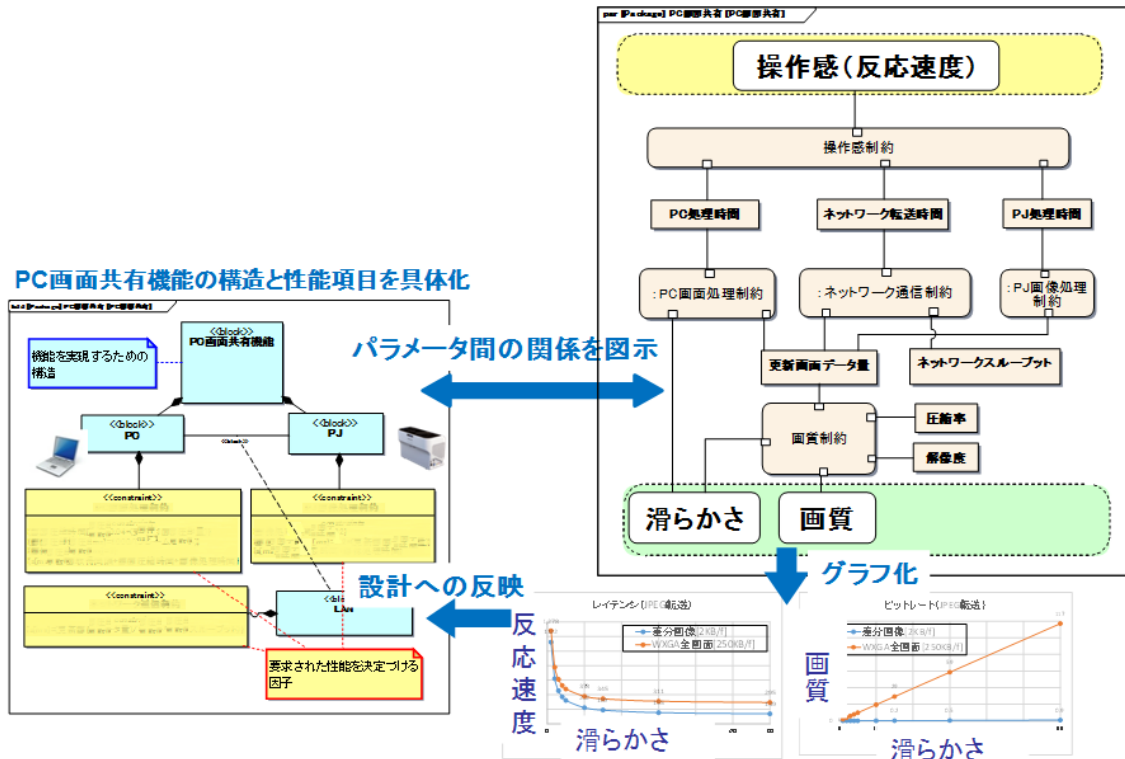


図 15-A-15-11 パラメトリック図適用例

## 7. 考察

### (1) 品質・生産性について

開発中のバグ件数の削減という定性的な目標でプロジェクトを開始したが、今回の SysML 適用により、間違いが少なく、効率的なシステム製品の開発が可能となる見通しが得られた。なお、品質・生産性については現在、定量評価中である。

### (2) 検証工程での効果・期待

SysML の要求図を使い、要求や狙いが明確になった。このため、この要求を検証する運用テストでも実施方法を明確にできた。また、運用テストだけでなくその後の運用・評価のフェーズにおいても、システム開発が狙い通りであったかを確認できることになる。さらに製品が完成し、市場に出た後、当初の製品の要求が市場で受け入れられたかどうかの確認（検証）を行うことができると期待している。

### (3) 今後の展開

今回は、SysML の要求図を中心に、パラメトリック図、ブロック定義図、アクティビティ図を適用することで、システムの複雑化時における要求の可視化方法と、上流

工程での性能設計方法を定義・実践した。今後、要求図、パラメトリック図、ブロック定義図、アクティビティ図以外の SysML ダイアグラム（例：内部ブロック図など）についても、開発の短縮化や開発の効率化に効果があるか検証していく。また、部内の他製品にも展開を行っており、さらに普及促進と、手法精緻化のため、社内で研究会の立ち上げを進めている。

## 8. まとめ

SysML の採用により、複雑に絡み合った要求を可視化し整理することができるとともに、システムの境界と開発対象システムの要件を明確にすることが可能となった。また、ソフトウェア開発、エレキ開発、メカ開発といった複数の技術領域間で理解できる共通の言語を利用することにより、効率よくシステム設計ができるようになった。さらに、開発中の障害発生件数の削減も達成した。今回の事例は、全面的に SysML を適用するのではなく、一部の機能について適用したが、このような、ノウハウや知見を蓄積することにより、SysML の適用を他の機能や製品に拡大することが期待される。

掲載されている会社名・製品名などは、各社の登録商標または商標です。

独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター (IPA/SEC)