博士学位論文

製品個体の集合に着目した
製品ライフサイクルの設計支援

松山　祐樹

2016 年 1 月

大阪大学大学院工学研究科

概要

深刻化する環境問題に対する製造業者の取り組みとして，リユースやリマニュファクチャリングや閉ループリサイクルなどによって，資源を有効利用する循環型の製品ライフサイクルを構築することが重要な課題の一つである．この課題に対応するためには，製品だけでなくそのライフサイクルフロー，つまり製造から使用やメンテナンスを経て最終処理に至るまでの処理のネットワークをも設計対象とするライフサイクル設計が効果的なアプローチである．しかし，循環型の製品ライフサイクルを設計するためには，いくつかの困難が伴う．その要因の一つとして，同じ設計解に基づいて一律に生産する製品でも，その一個体一個体は個々に異なるという点が挙げられる．つまり，ライフサイクルにわたる製品個体は，集合として多様さを示す．その結果，たとえ資源循環を意図した製品ライフサイクルを設計したとしても，多くの個体が資源循環に関する設計意図に沿わず，かえって環境性や経済性が悪化するという問題につながり得る．例えば，部品リユースを意図した製品ライフサイクルを設計したとしても，使用による劣化が激しいためにリユースされない部品個体が多く存在する場合では，リユースを想定して投入する資源（例えば，使用回数をカウントする機構やリユース部品検査設備など）により，製品ライフサイクル全体の環境負荷排出量や資源・エネルギー消費量がむしろ増大する可能性がある．そのため，ライフサイクルにわたる個体が製品ライフサイクルに対する要求を満たす範囲で多様さを示すように，製品とライフサイクルフローの設計解を決めることが重要である．なお，本研究では，設計者が設計時に決定する設計解の情報をノミナル情報と呼び，個体個別の情報を個体情報と呼ぶ.

ライフサイクル設計のための既存の設計対象モデルや計算機環境は，製品ライフサイクルを対象製品の 1 個体のライフサイクルによって代表させて表現しており，多数存在する製品個体をそれぞれ表現している訳ではない．また，ロバスト設計手法など，従来の製品設計においても，製造バラつきや使用環境の違いによって製品個体の性能にバラつきが生じることを統計的に表現した手法はある．しかしこれらの手法は，逆工程をも含めたライフサイクルにわたって個体の集合が示し得る状態の違いや量の変動，および各個体のライフサイクルにわたる履歴の違いを表現や評価の対象としているわけではない．一方，ライフサイクルにわたる個体の流れを設計段階でシミュレーションすることで，製品ライフサイクル全体の環境性や経済性を定量評価する手法として Life Cycle Simulation (LCS)が提案されている．しかし，LCS の先行研究では，LCS をあくまでシミュレーション手法として位置付けており，そのため定量評価の結果が設計要求を満たしていない場合の修正点の特定など，製品ライフサイクルの設計過程における設計サイクルの実行支援に課題がある．以上の課題に対して本研究は，多様な個体情報を設計段階で予測し，その集合が示す多様さが設計要求の範囲内に収まるような製品ライフサイクルのノミナル情報を決定する過程を支援する手法の提案を研究目的とした.

本目的を達成するために，以下 2 つの手法によって，設計支援手法を構成した．

(1) 製品ライフサイクルをノミナル情報と個体情報の両面から表現したモデル化手法
個体情報の多様さによって生じる製品ライフサイクルの問題やその原因の予測（以下，個体情報の評価）に設計段階で対応可能な作業空間として，製品ライフサイクルをノミナル情報と個体情報の両面から表現したモデル化手法を提案した．本モデル化手法を，以下 3 つの要素で構成した．第一に，先行研究が提案している製品ライフサイクルのノミナル情報を表現したモデル（製品ライフサイクルのノミナル情報モデルと呼ぶ）を用いて，個体情報に違いを与える要因（変動要因と呼ぶ）を表現する手法を提案した．第二に，個体情報を表現したモデル（個体情報モデルと呼ぶ）を提案した．第三に，製品ライフサイクルのノミナル情報モデルから個体情報モデルを作成する手法を提案した．

(2) 製品ライフサイクルの設計過程における設計サイクル実行支援手法
製品ライフサイクルのノミナル情報を決定する過程における設計サイクルを円滑に実行可能とするために，以下 2 つの設計過程を支援する手法を提案した．第一に，個体情報の評価を実行するための製品ライフサイクルのノミナル情報モデルの作成を支援する手法を提案した．第二に，個体情報の評価結果が設計要求を満たしていない場合に，修正すべきノミナル情報を特定するための支援手法を提案した．

上記 2 つの手法から成る設計支援手法を計算機実装したプロトタイプシステムとして，Entity-oriented Product Life Cycle-CAD システム（e-PLC-CAD システム）を開発した．本システムを用いてのスマートフォンを対象としたケーススタディを通じ，提案した設計支援手法の有効性を検証した．その結果，(1)のモデル化手法を用いることで，対象のノミナル情報についてのライフサイクルにわたる様々な側面や粒度から，個体情報の評価の実行が明示的に可能となった．また，(2)の設計サイクル実行支援手法によって，ノミナル情報の提案と個体情報の評価の実行サイクルにおける設計者の手間と労力を削減可能とした．以上より，提案手法が，個体が効率的に循環するような製品ライフサイクルのノミナル情報を決定する上で有効な手段であることを示した．

目次

第1章 序論

第 1 章では，本研究の背景として，ライフサイクル設計の重要性とその支援手法の構築に向けた課題を明らかにし，本研究の目的を述べる．最後に，本論文の構成を示す．

1.1. 研究背景

1.1.1. 持続可能な生産システムへの転換の必要性

産業革命以降，先進国を中心に人々は，大量生産・大量消費型の生産システムによって，多くの工業製品（以下，製品）に囲まれた社会に生き，安全と豊かさを享受してきた．その一方で，排出物の受け入れ容量や蓄えられた資源に有限性をもつ地球は，人類の生産活動を支えきれなくなり，その影響が地球規模の環境問題として顕在化している [1]．例えば，地球温暖化，オゾン層の破壊，資源やエネルギーの枯渇，大気汚染などに代表される大域的な問題が生じている．また，使用し終わった製品が大量にあふれ，廃棄場所が不足するといった局所的な問題も発生している．

世界人口は年々増加し続けており，2013 年時点でその約 82%（72 億人中 59 億人）が途上国に生活する人々である [2]．途上国の今後の経済成長を，先進国のこれまでの経済成長に鑑みると，製品の需要はますます増大する可能性がある．そのため，従来の大量生産・大量消費，さらには大量廃棄するという一方通行型の生産システムを継続していくことは，環境問題をより一層深刻化させる危険性をはらんでいる．しかし，我々の文明社会の基盤となっている生産システムを完全に捨て去ることは不可能である [3]．とはいえ，先進国が現状の生産システムを継続し，途上国の経済成長を阻害する権利はない．

この地球環境の危機に対し，資源を有効利用する循環型の生産システムを構築することの重要性が指摘されている．例えば近年の欧州では，resource efficiency roadmap [4]が打ち出されるなど，資源を有効利用することが強く求められている．また日本では，循環型社会形成推進基本法に代表されるような環境保護関連の法律規制が制定され，3R（Reduce, Reuse, Recycle）が推進されてきた．こういった社会背景を受け，製造業者を主体とした資源循環に対する取り組みが，自動車業界や電機メーカなどを中心に行われてきた．しかしその取り組みの多くは，従来の一方通行型システムにエンド・オブ・パイプ的にリサイクルを組み込んだ，いわば「大量生産・大量消費」＋「大量リサイクル」型のシステムにすぎず，宣伝されている 3R のコンセプトと現実世界での取り組みとの間には大きな乖離がある [5]．確かにリサイクルは，天然資源の消費量や廃棄物量を削減する手段として重要である．しかし，人が手を加えなくとも循環する自然界の物質とは異なり，製品という人工物を循環させるにはエネルギーやコストなどの負荷を伴う．そのため，例えばリサイクルによって無理に資源を循環させようとすると，環境負荷の排出やエネルギーの消費がかえって増大するなど，他の環境側面に悪影響を与える可能性がある．すなわち，従来の一方通行型の生産システムの延長として，3R のコンセプトを対症療法的に適用するだけでは環境問題の抜本的な解決にはつながらない．持続可能な形態への転換を目指したトップダウンアプローチによって，循環型の生産システムを構築する必要がある．

1.1.2. 循環型の製品ライフサイクルの設計支援に向けた課題

製品は，製造されてから使用や回収を経て再生処理や廃棄に至るまでの一生（以下，ライフサイクル）を通じて，資源やエネルギーを消費し，CO_2やNO_Xなどの環境負荷物質を排出する．生産システムを持続可能な形態へと転換させるためには，設計から製造という製品を作る側（順工程）だけでなく，使用した製品を回収して再度順工程へ投入したり廃棄したりする側（逆工程）に至るまでを有機的に統合し，ライフサイクル全体の環境負荷排出量や資源・エネルギー消費量を最小化することが重要である．そのための方法論の一つとして，「インバース・マニュファクチャリング」が提唱されている [6]．インバース・マニュファクチャリングは，「顧客要求を満足させ，企業利益を確保しつつ，いかに資源やエネルギーを消費しないで済ませるか」を基本コンセプトとしており，サービスの提供による付加価値の向上と安定した資源循環の実現を目的とする [6]．この目的の背景として，顧客の要求は製品自体にはなく，発現する機能に存在するという考えがあり，そのため「資源は循環，機能を更新」することが重要である [7]．このようなコンセプトをもつインバース・マニュファクチャリングを実現するための有効な手段として，「ライフサイクル設計」がある [8]．ライフサイクル設計は，製品だけでなく，ライフサイクルフローをも含めた製品ライフサイクル全体を設計対象とする [9]．なお，ライフサイクルフローとは，製造，使用，回収，リサイクルといった製品のライフサイクルにわたる処理工程（ライフサイクルプロセスと呼ぶ）のネットワークを表す [10]．ライフサイクル設計が重要な理由は，ライフサイクルを通じてどのような処理を施すかを想定した設計がなされていない製品では，実際に処理する段階で環境面や経済面に大きな負荷を与え得る点にある．例えば，製品の寿命やメンテナンス性，リユース性，リサイクル性などは，製品の設計段階で効果の大半が決まってしまう．そのため，サービス提供や資源循環の意図を設計段階で設計解に対する仕様に含めることが肝要である．

しかし，資源が効率的に循環するように製品ライフサイクルを設計するには，いくつかの困難が伴う．その要因の一つに，同じ設計解に基づいて一律に生産する製品でも，その一個体一個体のライフサイクルは個々に異なるという点が挙げられる．すなわち製品個体は，設計者が設計時に決定した設計解を共通の情報としてもつ一方で，製造時期，使用環境，廃棄時期，廃棄処理の方法といったライフサイクルにわたる履歴（以下，ライフサイクル履歴）がそれぞれ異なる（図 1-1 参照）．この設計解に対する個体個別のライフサイクル履歴の結果，製品ライフサイクルは以下 2 つの特徴を示す．

- 処理する個体の状態が個々に異なる
- 処理する個体の量が時間変化する

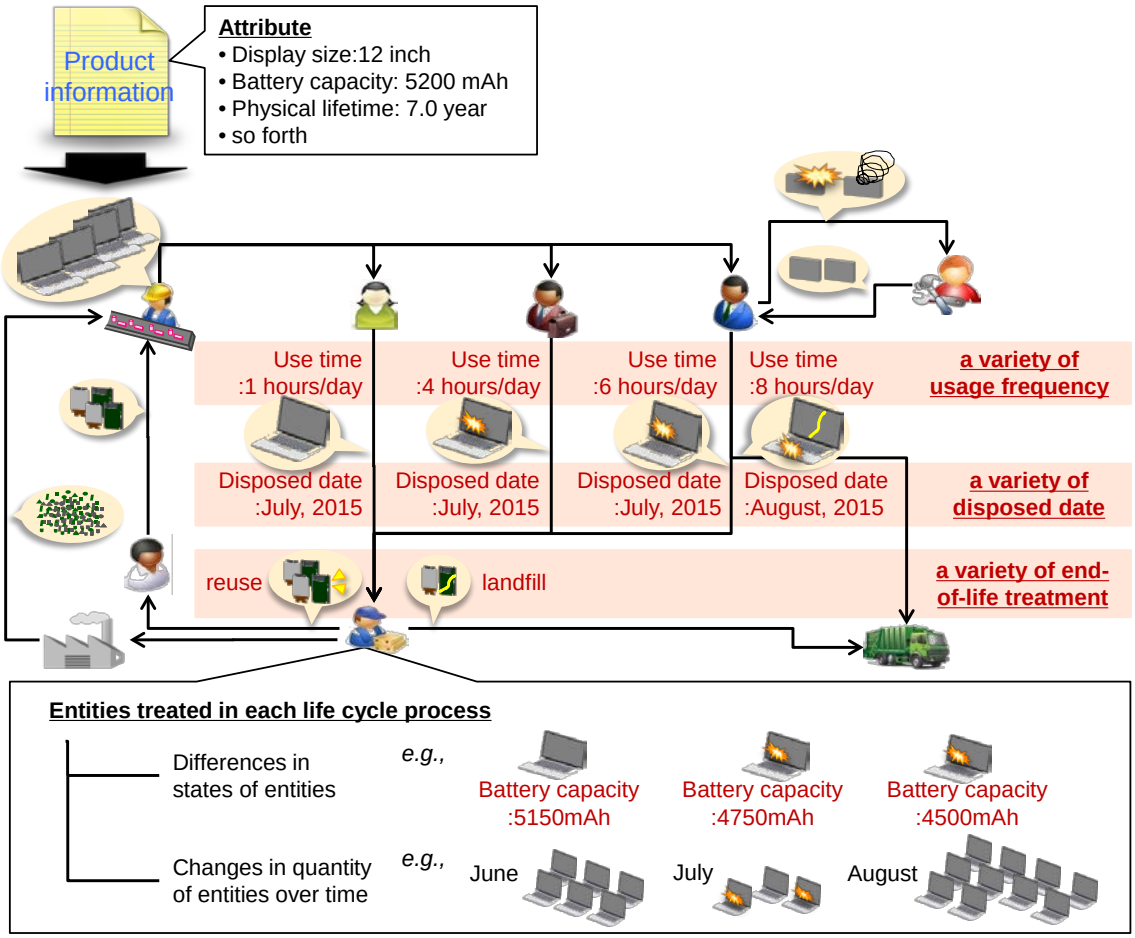


図 1-1：設計解に対する個体個別のライフサイクル履歴（例：ノートパソコン）

処理する個体の状態の違いや量の変動は，想定する製品構造やライフサイクルフローによってその有様が異なる．一方，これらの特徴に依存して，適切な製品ライフサイクルの構造が違ってくる．なぜならば，たとえ資源循環を意図した製品ライフサイクルを設計したとしても，状態に違いがあり，量が変動することによって，多くの個体が資源循環に関する設計意図に沿わず，循環の効果を十分に得ることができない可能性があるためである．例えば，部品リユースを意図した製品ライフサイクルを設計したとしても，使用による劣化が激しいためにリユースされない部品個体が多く生じる場合がある．また，リユース部品個体が需要量に対して十分かつ安定に得られない場合，安定供給の観点から資源循環が実行されない場合がある．このように多くの個体が循環に関する設計意図に沿わない場合，循環を想定して投入する資源（例えば，使用回数をカウントする機構やリユース部品検査設備など）により，製品ライフサイクル全体で消費する資源やエネルギーがかえって増大する可能性がある．

しかし，ライフサイクルにわたって製品個体は，製品構造だけでなく，各ライフサイクルプロセスでの処理方法やそのネットワークといった製品ライフサイクルの様々な要因に

依存して多様となる．また，製品ライフサイクルは，性能劣化した部品個体がリユース部品個体として再び市場に投入されたり，技術進展によりユーザの使用行動が変化したりするなど，動的変化するという特徴をもつ．加えて，資源循環を含む製品ライフサイクルでは，リユースやリサイクルといった最終処理の方法が部品ごとに異なり得るため，1 つの製品に対して複数の循環経路から成る複雑なシステムとなり得る [11]．このような複雑かつ動的変化する製品ライフサイクルにおいて資源を効率的に循環させるためには，対象の製品ライフサイクルのどのような要因に影響を受けて個体が多様となり，その結果製品ライフサイクルにどのような問題が生じるかを設計段階で予測し，個々の個体がその問題解決につながるライフサイクル履歴を示すよう製品とライフサイクルフローの設計解を決めることが重要である．つまり，個別のライフサイクル履歴を示す個体が製品ライフサイクルに対する要求（例えば，環境負荷の総排出量の削減やユーザコスト削減）を満たす範囲で多様さを示すように，どのような循環経路によってどのような処理を施すか，またその処理を施すためにどのような構造や属性の製品とするかを決定することが重要である．

これまでにも，製造バラつきや使用環境の違いといった製品性能に違いを与える要因に対し，その要因による影響を受けにくい製品を設計するための手法が，ロバスト設計などの分野で多く提案されてきた．しかし，ロバスト設計に関する研究では，製品性能のバラつきを統計的手法によって表現しており，製品一個体一個体を表現しているわけではない．これに対し，製造工程における「使用部材や加工部品の状態の違い」と「加工精度や組立精度の違い」に起因して多様さを示す幾何形状を入力としたシミュレーションによって，市場に流通する製品個体のバラつきを評価する手法が提案されている（例えば，[12]）．しかし，これらの手法では，製造工程を通じて生じる製品個体の性能の違いを表現や評価の対象とするものの，逆工程をも含めたライフサイクルにわたって個体の集合が示し得る状態の違いや量の変動，および各個体のライフサイクル履歴を表現や評価の対象としていない．一方，ライフサイクル設計を支援する計算機環境として，Umeda らはライフサイクル CAD のコンセプトを提案している [10]．また國井は，ライフサイクル CAD の実現に向けて，ライフサイクル設計の設計対象である製品とライフサイクルフローを表現したモデル化手法を提案している [13]．しかし，本モデル化手法 [13]は，製品ライフサイクルを対象製品の 1 個体のライフサイクルにより代表させて表現しており，多数存在する製品個体それぞれを表現しているわけではない．すなわち，個体個別のライフサイクル履歴を設計段階で予測し，それら個体が集合として示す状態の違いや量の変動を設計要求の範囲内に収めるよう製品ライフサイクルを設計するための方法論は十分に構築されてこなかった．

1.2. 研究目的

前節で示した問題背景より，本研究では，製品ライフサイクルにおける製品に関する情報を，以下の 2 つに分類して定義する．

ノミナル情報
設計者が設計時に決定する設計解の情報

個体情報
ある時点での個体個別の情報（例えば，状態，使用環境）

本研究の最終目標は，資源が効率的に循環するような製品ライフサイクルを設計するための方法論を構築することにある．本設計方法論の構築に向けて本研究は，個体情報を設計段階で予測し，その集合が示す多様さが設計要求の範囲内に収まるような製品ライフサイクルのノミナル情報を決定する過程を支援する手法の提案を目的とする．本目的を達成するために，以下 2 つの手法を提案する．

1. 製品ライフサイクルのノミナル情報と個体情報の両面を表現したモデル化手法
   個体情報の多様さによって生じる製品ライフサイクルの問題に設計段階で対応可能な作業空間として，製品ライフサイクルをノミナル情報と個体情報の両面から表現したモデル化手法を提案する．本モデル化手法を以下 3 つの要素で構成する．第一に，製品とライフサイクルフローのノミナル情報を表現した設計対象モデル [13]を用い，個体情報に違いを与える要因を表現する手法を提案する．第二に，個体情報を表現したモデルを提案する．第三に，製品ライフサイクルのノミナル情報を表現したモデルを入力に，離散事象シミュレーション技術である Life Cycle Simulation [14]を実行することで個体情報を表すモデルを作成する手法を提案する．

2. 製品ライフサイクルの設計サイクル実行支援手法
   個体情報を予測して製品ライフサイクルのノミナル情報を決定する過程を円滑に実行可能とするために，以下 2 つの設計過程を支援する手法を提案する．第一に，個体情報の評価を実行するためのモデル作成を支援する手法を提案する．第二に，個体情報の評価結果が設計要求を満たしていない場合に，ノミナル情報の修正を支援する手法を提案する．

以上 2 つの手法から成る設計支援手法を，製品ライフサイクルのノミナル情報モデル化システム [13]を拡張して計算機上で実装する．さらに，提案した設計支援手法の有効性を検証するために，実装したシステムを用いてケーススタディを行う．

1.3. 本論文の構成

本論文は，以下の全 9 章から成る．図 1-2 は，各章の内容と関係を表す．

第1章では，本研究で対象とする研究課題の背景を述べ，取り組む研究目的を設定した．

第2章では，製品ライフサイクルの設計過程に関する既存手法や計算機環境について調査する．また，現実世界において個体の集合が示す状態の違いと量の変動を考慮した資源循環や製品ライフサイクルのマネジメントに関する先行研究を調査する．

第3章では，本研究における製品ライフサイクルの基本的な考えとその捉え方で製品ライフサイクルを設計するうえで生じる設計問題について述べる．また，本設計問題を内包する製品ライフサイクルの設計を実行するための研究課題を述べ，その課題に対する先行研究と本研究の位置づけを明確にする．

第4章では，製品ライフサイクルをノミナル情報と個体情報の両面から表現したモデル化手法を提案する．まず，製品ライフサイクルのノミナル情報の一部として，個体情報に違いを与える要因を表現する手法を提案する．次に，個体情報を表現するモデルを提案する．最後に，対象の製品ライフサイクルのノミナル情報について，個体情報を表すモデルを作成する手法を提案する．

第5章では，第 4 章で提案するモデル化手法を用い，個体情報を予測して製品ライフサイクルのノミナル情報を決定する過程を円滑に実行するための支援手法を提案する．まず，個体情報の評価の実行過程を支援する手法を提案する．次に，個体情報の評価結果が設計要求を満たしていない場合に，修正すべきノミナル情報を特定するための支援手法を提案する．

第6章では，第 4 章と第 5 章で提案する 2 つの手法から成る製品ライフサイクルの設計支援手法を計算機上で実装したプロトタイプシステムである Entity-oriented Product Life Cycle-CAD システム（e-PLC-CAD システム）について述べる．

第7章では，スマートフォンを対象としたケーススタディを通じて，第 4 章と第 5 章で提案する 2 つの手法から成る製品ライフサイクルの設計支援手法の有効性を，e-PLC-CAD システムを用いて検証する．

第8章では，本研究が提案する設計支援手法の有効性と課題について考察する．

第9章では，本研究の結論，および今後の課題と展望を述べる．

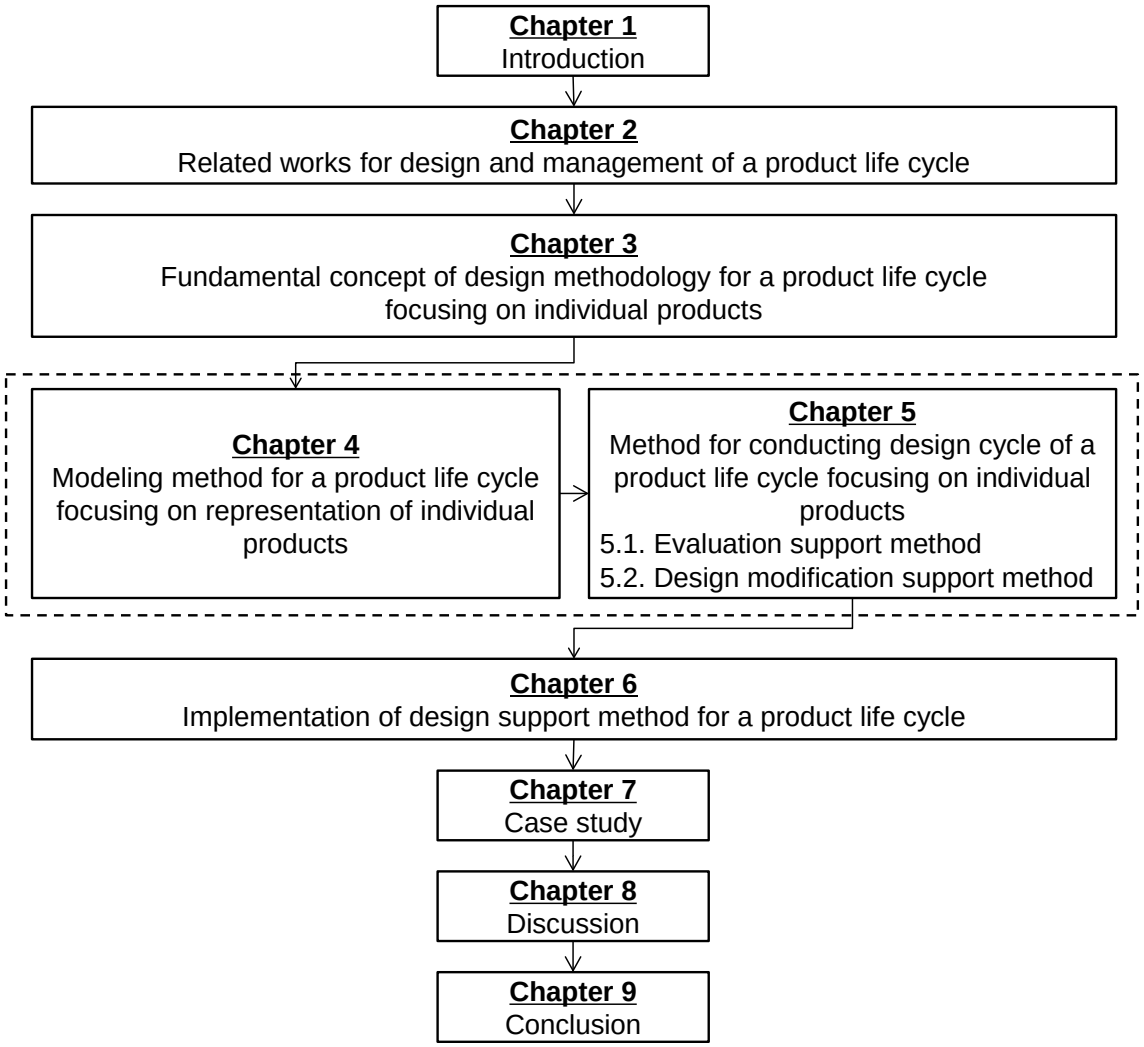

Chapter 1
Introduction
Chapter 2
Related works for design and management of a product life cycle
Chapter 3
Fundamental concept of design methodology for a product life cycle focusing on individual products
Chapter 4
Modeling method for a product life cycle focusing on representation of individual products
Chapter 5
Method for conducting design cycle of a product life cycle focusing on individual products
5.1. Evaluation support method
5.2. Design modification support method
Chapter 6
Implementation of design support method for a product life cycle
Chapter 7
Case study
Chapter 8
Discussion
Chapter 9
Conclusion

図 1-2：本論文の構成

第2章 製品ライフサイクルの設計とマネジメントの関連研究

第 2 章では，製品ライフサイクルの設計とマネジメントに関する先行研究を調査し，その課題を述べる．まず，ライフサイクル設計の基本コンセプトについて述べる．次に，ライフサイクル設計の設計過程に関する先行研究を調査する．また，現実世界における資源循環や製品ライフサイクルのマネジメントに関する先行研究を調査する．

2.1. ライフサイクル設計の基本コンセプト

ライフサイクル設計は，製品開発や製造や使用という順工程から，回収や再利用を経て廃棄するという逆工程に至るまでの製品ライフサイクル（図 2-1 参照）全体を有機的に結合して設計する作業過程である [6].

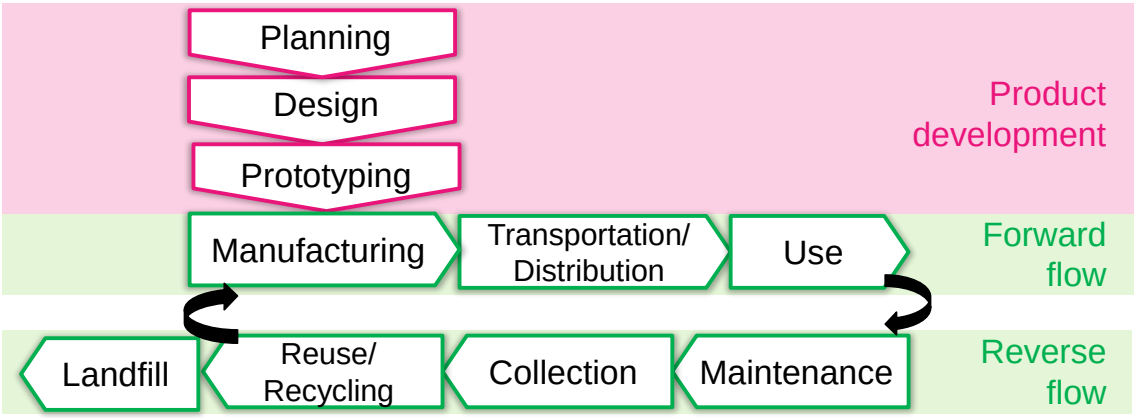


図 2-1：製品ライフサイクル（ [15]を基に作成）

すなわちライフサイクル設計は，製品だけでなく，ライフサイクルフローをも設計対象とし，この 2 つの対象から成る製品ライフサイクルのノミナル情報を決定するというコンセプトの設計である [10]．製品とライフサイクルフローのノミナル情報の両者を対象とする理由は，製品設計とライフサイクルフローの設計が密接に関係するためである [16]．製品設計がライフサイクルフローの設計に与える影響としては，対象製品の特性によって適切なライフサイクルフローが異なる点にある．一方，ライフサイクルにわたってどのような処理を施すかによって，適切な製品構造や属性は異なる．つまり，ライフサイクルフローの設計が製品設計に影響を与える．例えば，リサイクルを含むライフサイクルフローでは，破砕選別を可能とする素材構成が求められるのに対し，リユースを含む場合では，破壊を伴わない分解方法でリユース対象部品を取り出せるような製品構造が求められる [17].

資源が効率的に循環するような製品ライフサイクルを製造業者自身が構築した事例は，これまでにもいくつか報告されている．日本国内だけで見ても，レンズ付きフィルム [18]，コメットサークルに基づいた複写機の製造 [19]，リフレッシュ PC [20]が，一例として挙げられる．しかし，これらの事例は，製品特性を上手く利用して資源循環を実現した先進的な例であり，その他多くの製品ライフサイクルで実践されている資源循環はリサイクルが中心である．

また，リサイクル性設計に代表される Design for X (DfX)手法のような要素技術が数多く提案され，実用化されている．DfX 手法は，製品ライフサイクルにおける特定の視点から製品の改善（例えば，製造性，サービス性，分解性，リサイクル性の向上など）を図る手法である [21]．しかし，特定の視点からの改善が必ずしも製品ライフサイクル全体の改善

につながるわけではなく，トレードオフ問題が生じる可能性がある [22]．例えば，組立性を改善することで，分解性が改悪してしまう場合があり得る [23]．そのため，1.1.1 項で述べたように，持続可能な形態への転換を目指したトップダウンアプローチによって，製品ライフサイクルの全体構造を設計する必要がある．

2.2. ライフサイクル設計の設計過程

設計とは，顧客や市場の要求を技術的な仕様に変換し，変換した仕様を実現する設計解を探索する作業である [15]．Takeda ら [24]は，設計における設計解の探索作業を，認知的設計過程モデルとして表現している（図 2-2 参照）．認知的設計過程モデルでは，設計解の探索作業を，(i)問題提起，(ii)提案，(iii)展開，(iv)評価，(v)決定，という 5 つの設計行為で構成している．具体的には，(i)事実や仮定などの様々な情報を基に，設計で解決すべき問題を発見や指摘し，(ii)その問題を解決する複数の解候補を導出する．(iii)導出した解候補を，設計者が有する知識を用いて具体化する．(iv)具体化した解候補をある判断基準に従って評価し，(v)提案した解候補を採用するか決定する．

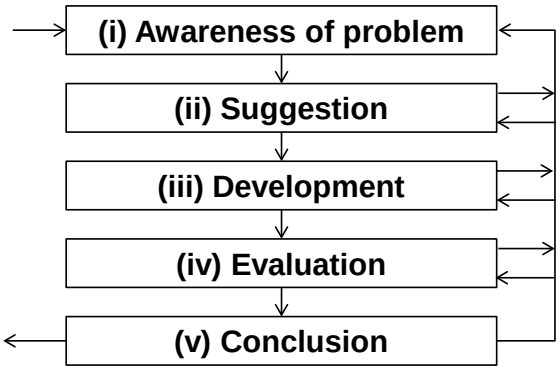


図 2-2 ：認知的設計過程モデル（ [24]を基に作成）

設計者は上記 5 つの設計行為の繰り返しによって，顧客や市場の要求を実現するノミナル情報の導出という設計問題を解く．しかし設計問題に対する解決策は，設計者にとって最初は漠としたものであり，問題を段階的に詳細化する過程で明瞭な形となる [25]．そのため，例えばドイツ流設計方法論 [26]では，設計過程全体を「役割の明確化」「概念設計」「実体設計」「詳細設計」という 4 つの設計段階に分類し，解決策を段階的に詳細化することで，最終的な設計解を決定すべきであると指摘している．

Takeda らの認知的設計過程モデルやドイツ流設計方法論に基づくと，ライフサイクル設計の設計過程は，製品ライフサイクルに対する様々な要求（以下，単に設計要求）を実現する製品とライフサイクルフローのノミナル情報を，5 つの設計行為の繰り返しにより段階的に詳細化して決定する作業である．ライフサイクル設計の設計過程として，例えば國井は， [9]を基にした図 2-3 に示すような設計過程を想定している [13]．本設計過程は，「(1) ライフサイクル戦略の策定」と「(2) 製品とライフサイクルフローの設計」という 2 つの設計段階，および各設計段階での「(3) ライフサイクル評価」という評価に関する設計行為から成る．(1)から(3)の設計行為の実行を支援する先行研究について，以下 2.2.1 項から 2.2.3 項で詳述する．また，本設計過程を支援している既存のライフサイクル設計支援環境について，2.2.4 項で述べる．

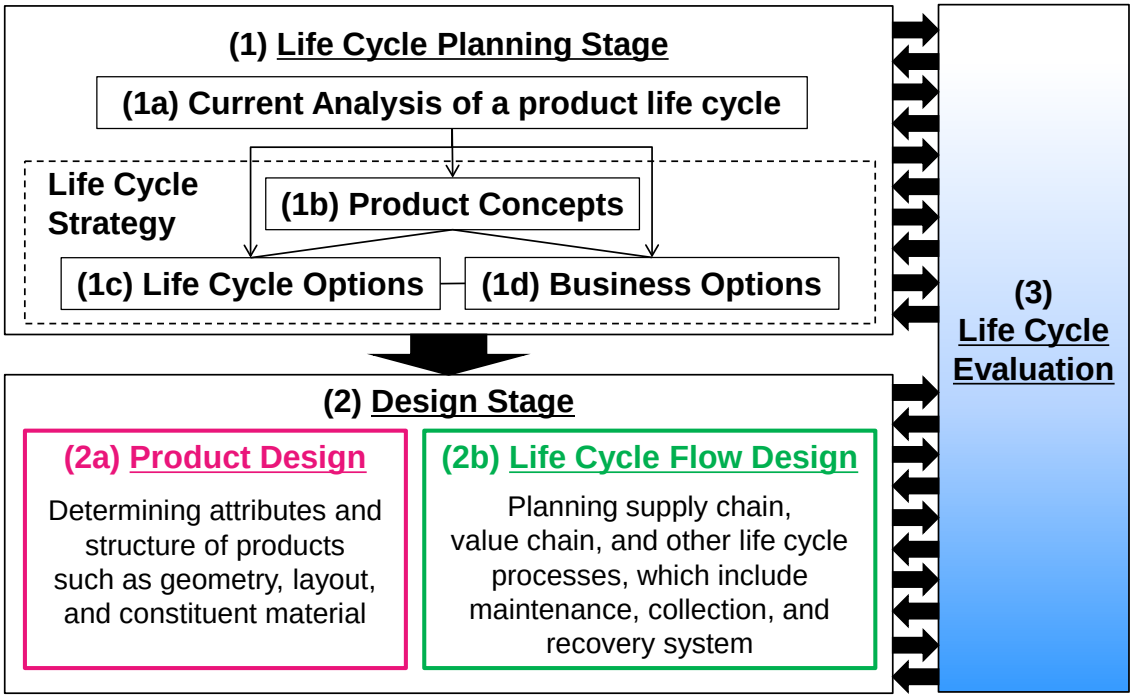


図 2-3：ライフサイクル設計の設計過程（ [13]を基に作成）

2.2.1. ライフサイクル戦略の策定

設計上流段階で設計の基本方針を決めることの重要性は，多くの研究者が指摘している．その理由のひとつは，設計上流段階で製品ライフサイクルにわたるコストの 80%が決まる点にある [15]．また，設計上流段階での意思決定が手戻りの削減につながることも挙げられる [27]．設計上流段階で基本方針を決めることの重要性は，ライフサイクル設計においても同様である．1.1.2 項で述べたように，地球環境に対する要求を設計下流段階で組み込むことが困難なためである [25] [8] [28]．

ライフサイクル設計の基本方針を決めるにあたっては，第一に，製品の特性（例えば，物理寿命，価値寿命，価格など）とライフサイクルフローの特性（例えば，販売期間，製品カテゴリの寿命，回収システムなど）を調査し，既存の製品ライフサイクルが内包する問題点を特定することが重要である．このような問題の特定作業を，本研究では「製品ライフサイクルの現状分析」と呼ぶこととする．第二に，製品ライフサイクルの現状分析によって特定した問題を解決するように，ライフサイクル設計の基本方針を策定する．Umeda らは，この基本方針を「ライフサイクル戦略」と呼び，ライフサイクル戦略を「製品コンセプト」「ライフサイクル・オプション」「ビジネス・オプション」という 3 つの要素で構成している [9]．製品コンセプトとは，製品そのものが顧客にもたらす価値や機能を表す．ライフサイクル・オプションとは，メンテナンスやアップグレード，リユースやリマニュファクチャリング，リサイクルや埋立といった，製品や部品が辿るべき循環経路を表す．ビジネス・オプションは，製品の提供方法を表す．例えば，売り切りやリースといった製品の販売形式がビジネス・オプションに該当する．ライフサイクル戦略の策定における「(1a)製品ライフサイクルの現状分析」「(1b)製品コンセプトの策定」「(1c)ライフサイクル・オプションの選択」「(1d)ビジネス・オプションの選択」に関する先行研究について，以下で述べる．

(1a) 製品ライフサイクルの現状分析

既存の製品ライフサイクルが内包する問題を特定する手法として，例えば Wimmer らは，対象製品の使用材料やリサイクル処理に関するチェックリストに答えていくことで環境負荷が集中するライフサイクルプロセスや製品情報の修正点を特定する ECODESIGN Pilot [25]を提案している．また，Rose らは，製品特性を「外部要因」「材料要因」「分解性要因」「逆工程のサプライチェーン要因」という 4 つの要因に分類しており，対象製品の特性を各要因について分析することで，ライフサイクル戦略を策定すべきだと述べている [29]．さらに，従来の製品設計で満たすべき顧客要求に加え，地球環境に対する要求を満たすよう製品開発するために，顧客要求や地球環境への要求との関連度が高い部品を特定する手法として，環境調和型品質機能展開 [30]や重み付き環境調和型設計チェックリスト [31]がある．これらの手法は設計要求を満たすために修正すべき部品の特定を支援するものの，特定した部品のノミナル情報をどのように修正すべきかは設計者自身で考える必要がある．

(1b) 製品コンセプトの策定

製品コンセプトを策定することは，ライフサイクル設計に限らず従来の製品設計でも重要な課題であり，数多くの手法が提案されている．例えば，多数の特許から抽出したアイデア発想の共通点をまとめた 40 個の法則に従って技術課題を解決する製品コンセプトの創出を支援する TRIZ [32]や，多数あるアイデアを多数の評価軸について比較評価することで基準案よりも優れたアイデアに絞り込んでいく Pugh method [33]などがある．

(1c) ライフサイクル・オプションの選択

図 2-3 に示したライフサイクル設計の設計過程は一例であるが，製品設計に先立って，製品にどのような循環経路を辿らせるか，つまりライフサイクル・オプションを決めることの重要性は，多くの研究者が指摘している（例えば， [34] [35]）．ライフサイクル・オプションは，様々な研究者による様々な定義がある．例えば，適用するライフサイクルフローの段階（Beginning-of-Life (BoL)，Middle-of-Life (MoL)，End-of-Life (EoL)）に基づく分類 [6]や 3R に基づく分類 [36]などがある．本研究では，ライフサイクルフローとの対応関係を重視して，適用段階に基づく分類を採用する．本分類を，表 2-1 に示す．

表 2-1：ライフサイクル・オプションの分類（ [6]と [36]を基に著者が作成）

Life cycle stage	Life cycle option	
Beginning-of-Life (BoL)	Reducing	
	Longlife	
Middle-of-Life (MoL)	Maintenance	
	Upgrading	
End-of-Life (EoL)	Parts Reuse	Product Installation Reuse
		Spare Parts Reuse
	Product Reuse	Remanufacturing
		Refurbishing
	Second-hand Sales	
	Recycling	Material Recycling
		Chemical Recycling
		Energy Recovery
	Disposal	Appropriate Disposal
		Incineration
		Landfill

表 2-1 のように分類されるライフサイクル・オプションを，対象製品や構成部品の特性に応じて選択することが重要である．ライフサイクル・オプションの選択支援手法として，例えば Umeda らは，廃棄要因分析表 [5]を提案している．廃棄要因分析表は，対象製品の構成部品に対する廃棄要因とその重要度を物理寿命（機能消費，故障）と価値寿命（外観，容量・サイズ，機能の陳腐化）ごとに算出し，廃棄要因を可能な限り排除する部品とライフサイクル・オプションの組を特定することで，製品の長寿命化を図る手法である．

一方でライフサイクル・オプションは，製品特性のある一側面に着目して選択するのではなく，製品ライフサイクルに対する様々な要求を満たすよう選択する必要がある．この問題に対して Ishii ら [17]は，組立やサービスや EoL の実現可能性が高くかつコストを小さくするようなライフサイクル・オプションを選択する手法を開発している．Ishii らの手法では，ライフサイクル・オプションごとに混合できない材料の組み合わせや適用できない締結方法があるという点を考慮して分解コストを評価する手法を用いている [17]．また Kobayashi は，製品ライフサイクルに対する設計要求を満たす解決策を，製品レベルから部品レベルへと段階的に展開し，最終的に部品ごとのライフサイクル・オプションを選択する手法として，Life Cycle Planning 手法 [37]を提案している．これらの手法は，各ライフサイクルプロセスでの処理方法や製品の構造や属性を決定するものではなく，あくまで設計上流段階で製品ライフサイクルのコンセプトを決定支援する手法である．またこれらの手法は，個体が集合として示す多様な個体情報を予測して，適切なライフサイクル・オプションを選択するための手法ではない．

(1d) ビジネス・オプションの選定

1.1.2 項で述べたように顧客要求は,製品そのものではなく,製品が発現する機能にある．そのため，製品を大量に作って顧客要求を満たすという量の充足性ではなく，機能の提供に主眼を置いて質の充足性を高めるビジネスモデルが注目されてきている [16] [38]．質の充足性を高めるビジネスモデルへの転換に向けた代表的なコンセプトとして，Product Service System (PSS)が提唱されている．PSS は，製品(tangible)の提供方法をサービス活動(intangible)と一体化して考えることで，大量生産・大量消費に頼らない顧客要求の実現を目的としたビジネスモデルである [39] [40] [41]．カーシェアリングやロールスロイスの power-by-the-hour [42]が PSS の代表事例として挙げられる．なお米国では，PSS と同様の概念で，Servicizing という言葉を用いている [43]．

また，サービスの提供方法を決めるためには，顧客要求の背景にあるニーズの抽出と抽出したニーズを具現化するアイデアの発想が必要である，しかし多くの企業では，地球環境に対する要求を満たすようなアイデアを発想するための知識や経験を有した専門家が少ないという問題がある [44]．この問題に対し，Kondoh ら [45]や中村ら [44]は，環境価値と経済価値の両方を増長するビジネスモデルを容易に創出や立案可能としたエコビジネスプランニング手法を提案している．

PSS の関連研究は数多くあるが，その大半は重要性の指摘に留まっている．また，PSS は製品とサービスの統合を目的としているが，使用段階にある製品にどのようなサービスを付加するべきか，もしくは旅行ビジネスのような製品を介さないサービスを対象とした研究が多い．つまり，ライフサイクル全体の環境負荷排出量や資源・エネルギー消費量を削減するようなサービスの提供方法を，製品やライフサイクルフローの両面から設計するための方法論は十分に確立していない．

2.2.2. 製品とライフサイクルフローの設計

策定したライフサイクル戦略を実現するように，製品とライフサイクルフローのノミナル情報を具体的に決定する．製品設計では，策定した製品コンセプト，ライフサイクル・オプション，ビジネス・オプションに応じて，製品を構成する部品の幾何形状や配置，構成素材，接続方法などのノミナル情報を決定する．例えば，ライフサイクル・オプションとしてメンテナンスとリサイクルを選択している場合，メンテナンス対象部品を使用段階で分解して取り出しやすいような配置や接続方法にするといったメンテナンス性設計や，破砕選別によるリサイクルが可能な素材を選択するといったリサイクル性設計を施す．ライフサイクルフローの設計では，ライフサイクルプロセスのネットワーク，つまりライフサイクルを通じて製品に施す処理の組み合わせと，各ライフサイクルプロセスにおける処理方法のノミナル情報を決定する．ここで，ライフサイクルプロセスにおける処理方法とは，生産ラインでの部品の加工方法や組立手順から，使用段階でのユーザの使用行動や廃棄行動，メンテナンスや回収システムやリサイクルシステムといった資源循環の方法，および各処理施設のレイアウトに至るまでを含む．

2.1 節で述べたように，製品設計とライフサイクルフローの設計は密接に関係するため，上記の設計行為を関連付けて行うことが重要である．以下で，製品設計とライフサイクルフローの設計に関する先行研究について述べる．

(2a) ライフサイクル設計における製品の設計

策定したライフサイクル戦略を実現するように製品を設計する手法は，DfX 手法を代表に数多く提案されている．DfX 手法は，2.1 節で述べたように，分解性やリサイクル性といった製品ライフサイクルにおける特定の視点から製品の改善を図る手法である．選択したライフサイクル・オプションを製品構造上で実現するための DfX 手法としては，以下が例に挙げられる．

- 組立性設計手法（Design for Assembly）：組立性向上のための設計ガイドライン（例えば，[46] [23]）など．
- 分解性設計手法（Design for Disassembly）：分解性向上のための設計ガイドライン（例えば，[23]），分解や解体が容易な構造や幾何形状をあらかじめ製品に組み込んでおく Product Embedded Disassembly 手法（例えば，[47]），破壊を伴う手分解を実現するための割れ線付加位置の決定手法 [48]，外部からのトリガーによって変形する素材を締結部品に用いる Active Disassembly 手法（例えば，[49]），分解性評価に基づく製品設計手法（例えば，[50]）など．
- アップグレード性設計手法（Design for Upgradability）：顧客要求の将来的な不確実性を予測してアップグレード可能な製品構造を策定する手法（例えば，[51] [52]）など．
- リサイクル性設計手法（Design for Recycling）：リサイクルに適した材料の組み合わ

せを選択する手法（例えば，[53]），対象製品の EoL 処理工程を考慮したリサイクル可能率の算出手法を用いて適切な製品属性を導出する手法 [54]など．

2.1 節で述べたように，DfX 手法を適用して導出したノミナル情報の間にはトレードオフ問題が生じ得る．DfX 手法間のトレードオフ問題を解消するように製品のノミナル情報を探索する手法として，例えば土井ら [55]は，選択した複数のライフサイクル・オプション（例えば，リデュースとリユース）を実現する最適な素材を，各ライフサイクルプロセス（製造，メンテナンス，分解，リサイクル）における処理コストに関する多目的最適化問題として解くことで導出する手法を提案している．

また，大半の製品は複数の部品から構成されており，各部品がそれぞれ異なる特性をもつ．そのため，部品ごとに異なるライフサイクル・オプションが選択され得る．例えば，物理寿命や価値寿命が長いためにリユース可能な部品もあれば，技術革新が速く価値寿命が短いためにリユースは不適でリサイクルに適した部品もある．一つの製品に対して多様に選択されたライフサイクル・オプションを効果的に実行するためには，同じライフサイクル・オプションが選択された部品を近い場所に配置して分解性やメンテナンス性を向上させたり，部品間の素材を統一することや混合処理できない素材の分離を容易とすることによってリサイクル性を向上させたりすることが有効な手段である．このようなアプローチでライフサイクル・オプションの実行効果を高めるための手法として，モジュール化設計手法がある [56]．モジュール化とは，設計の目的に応じて製品の機能構造と物理的な構造を構成することであり，各モジュールは本質的に独立な構造単位である [57]．これまでにも例えば，製品機能を維持しつつもライフサイクル・オプションを効果的に実行可能とするように部品をモジュール化する手法（例えば，[58] [59]），および製品の幾何形状や部品配置を導出する手法（例えば，[60] [61]）などが報告されている．また，製造から回収に至るまでに比較的長い時間幅をもつ製品ライフサイクルを設計対象とするライフサイクル設計では，多世代多品種にわたる製品を対象に設計することが重要である．そのため，製品ファミリー間で共通化する部品候補を，ライフサイクル・オプションの観点から策定する手法がいくつかの研究で提案されている（例えば，[62] [63]）．

(2b) ライフサイクルフローの設計

ライフサイクルフローの設計手法は，ライフサイクル戦略を実現するための手法という位置付けではなく，順工程や逆工程における特定のライフサイクルプロセス間での資源の流れ，および各ライフサイクルプロセスの設計やスケジューリングを支援する手法として，個別論的に取り組まれている．

順工程での資源の流れをモデル化する手法は，製品を効率よく市場に供給することを目的として，数多く報告されている．例として，製品や部品を製造するために必要な投入資源の量や時期を，部品表とサプライチェーンのリードタイムから算出する在庫管理資材所要量計画（Materials Requirements Planning，MRP）が挙げられる．また，MRP に生産能力や労働力といった生産工程全体の制約を加えて，資材調達や生産を計画する MRPⅡがある．さらに，受発注や物流や在庫などの生産管理データを，サプライチェーン全体で共有して需給バランスを調整することで, 流通の全体最適を図る Supply Chain Management (SCM)などが挙げられる．

逆工程の設計やスケジューリングに関する研究も数多くある．特に分解工程は，メンテナンス工程や EoL 処理工程に必要不可欠なライフサイクルプロセスであり，また組立工程と比較してより多くの不確実性や難点を含んでいる [64]. そのため分解工程の設計手法は，多く提案されている．例えば Wakamatsu ら [65]は，製品の幾何形状から部品間の拘束関係を特定し，特定した拘束関係をグラフ構造で表現することで，作業数が最も少ない分解手順を導出する手法を提案している．また Lee ら [66]は，製品の階層構造と部品間の接続関係を表現したモデルを用い，製品を分解するレベルと各部品の最終処理の方法を，分解工程の経済性に関する最適化問題として解くことで導出する手法を提案している．メンテナンス工程の設計を支援する手法としては，Failure Mode and Effects Analysis（FMEA）による論理構造図を用いて，潜在的な劣化や故障の発生メカニズムを分析するといった信頼性中心保全に関する手法が多く提案されている [67]．リサイクル活動をより効率的に行うためにリサイクル工程のノミナル情報を決定する手法もある．例えば Colledani ら [68]は，リサイクル処理された材料の品質が破砕や選別処理する材料構成や粒度に依存することに着目し，リサイクルシステム内での材料フロー，および破砕や選別処理で用いる機材性能をモデル化した手法を提案している．

2.2.3. ライフサイクル評価

ライフサイクル評価は，策定しているライフサイクル戦略や，製品とライフサイクルフローの設計段階で導出したノミナル情報が，設計要求を満たしているか評価する設計行為である．設計要求には，従来の製品設計の設計要求にあるような製造業者の利益向上やユーザコストの削減といった経済性に加えて，ライフサイクル全体の環境負荷排出量や資源・エネルギー消費量を削減するといった地球環境に対する要求，さらには分解コストの低減やリサイクル可能率の向上といった EoL 処理工程に対する要求が含まれる．

製品ライフサイクル全体の環境性や経済性を定量評価する代表的な手法としては，Life Cycle Assessment (LCA)と Life Cycle Costing (LCC)がある．LCA は，原料の調達から製造，流通，使用を経てリサイクルや廃棄に至るまでのライフサイクル全体を評価対象とし，各ライフサイクルプロセスで投入される資源や排出される環境負荷，および資源投入や環境負荷排出の結果が地球環境や生態系に及ぼす影響を定量評価する手法である [69] [70]. 一方で LCC は，LCA と同様の概念で定量評価するが，評価項目を製品ライフサイクル全体や各ライフサイクルプロセスで発生するコストとした手法である（例えば， [71] [72]）．

また，LCA の評価側面である環境性と LCC の評価側面である経済性は，トレードオフ関係となり得る．そのため，環境性と経済性の両面から製品ライフサイクルを評価する手法も提案されている．代表的な指標として，Eco-efficiency [73]がある．Eco-efficiency は，製品がライフサイクルにわたって生み出す環境負荷に対して得られる製品価値を表す指標である．しかし Eco-efficiency を用いた評価では，分子と分母の次元が異なることが多く見受けられる．そこで，旧製品から新製品への環境効率の改善率を表す指標として，Factor X が考案されている．Factor X は，生活の質の向上を分子に，環境への影響の低減を分母にとることで，製品価値の改善効果が環境に及ぼす影響を数値化した指標である．持続可能な社会を実現するためには，Factor 4 [74]や Factor 10 [75]を満たさなければならないという指摘がある．さらに，Social and Environmental LCA (SELCA) [76]や Social LCA (SLCA) [77]といった手法のように，社会性（例えば，労働者の安全性や健康面）を評価側面に加えた手法が提唱されている．

LCA や LCC は，静的かつ代表的な製品ライフサイクルを評価対象とする [78]. 一方で，実際の製品ライフサイクルは動的に変化する．例えば，製品や部品の機能は技術向上に伴って陳腐化し，また性能劣化した部品がリユース部品として再び市場に投入され得る．しかし LCA や LCC は，このように動的変化する製品ライフサイクルを評価対象とした手法ではない．また，1.1.2 項で述べたように，現実世界の製品ライフサイクルには個体情報が異なる多様な個体が存在するものの，LCA や LCC では多数存在する個体を表現することはできない．これらの問題に対する製品ライフサイクルの評価手法として，Life Cycle Simulation (LCS)が提案されている（例えば， [22] [14] [79] [78] [80]）．LCS は，ライフサイクルフロー上の個体（製品，部品，素材）と無形物（金銭，情報）の流れを動的にシミュレーションする技術である．LCS を実行するためには，まずライフサイクルフローを

表現したモデルを作成する必要がある．次に，作成したライフサイクルフローのモデルに製品のノミナル情報を入力する．さらに，離散事象シミュレーション技術を用いたシミュレーションによって，単位時間ステップごとに各ライフサイクルプロセスに入出力する個体や無形物の流れについての計算を繰り返す．上記の手順でシミュレーションを実行することによってLCSは，リユースやメンテナンスなどの動的なループを含む製品ライフサイクルについて，環境負荷排出量や資源・エネルギー消費量といった環境性，およびメーカ利益やユーザコストといった経済性の定量評価を可能とする．LCS のモデリングの自由度は，LCAやLCCに比べて高い．例えばKumazawaらは，LCSを用い，パソコンのリユースビジネスをモデリングしている [78]．LCAやLCCに対するLCSの，評価対象，評価手法，モデリングの自由度の違いは，表 2-2 に示す通りである．

表 2-2：LCA と LCC と LCS の違い（ [9]を基に作成）

	Output	Method	Flexibility of modeling
LCA	Environmental impact	Static evaluation based on input-output model of life cycle processes	Low
LCC	Economic impact	Static evaluation based on input-output model of life cycle processes	Low
LCS	Environmental and economic impact	Dynamic evaluation based on discrete event simulation	High

従来のLCS研究では，ライフサイクルにわたる個体の流れをシミュレーションする過程で，環境負荷排出量，資源消費量，製品回収台数，コストなどの時間変化を算出している．また，これら算出結果について最適化計算を行うことで，ライフサイクル・オプションの選択を支援するなど，定量評価の議論は多くなされてきた．しかし LCS の先行研究では，シミュレーション実行後に個体情報は存在しない．そのため，シミュレーション実行後に，解くべき設計問題や設計要求に応じて，ライフサイクルにわたる様々な側面や粒度から個体情報を表現するための支援はなされていない．またLCSの先行研究では，シミュレーションした結果が設計要求を満たしていない場合に，修正すべきノミナル情報を特定するといったような設計方法論に関する議論は十分になされていない．

LCAやLCCやLCSといった製品ライフサイクル全体の環境性や経済性を評価する手法に対し，資源循環に関する個別の評価手法や指標がある．特に，材料リサイクルの効果を評価する指標が多く存在する．例えば，国際規格団体IEC(International Electrotechnical Commission)は，リサイクル可能率の規格を定めている [81]．また日本では，リサイクルプラントに運び込まれた廃家電製品から回収される有価物の重量比を表す指標として，再商品化率 [82]が存在する．さらに，EoL処理工程を考慮したリサイクル性評価手法がいく

つか提案されている．例えば，Papakostas ら [83]は，ソリッドモデルで表現された製品の幾何形状から分解手順を特定し，特定した手順に従った分解過程で生じるコストと取り出される素材の価値によって対象製品のリサイクル性を評価する手法を提案している．資源循環の効果を評価するその他の手法や指標には，例えば製品組み込みリユースの最大可能率を算出する限界リユース率 [84]などがある．

2.2.4. ライフサイクル設計の設計過程を支援する既存の計算機環境

一般に製品設計では，ノミナル情報の探索作業を効率よく進めるために，設計対象である製品の性質を特定の視点から抽出し，選択的に記述した設計対象モデルを用いてきた[85]．設計対象モデルは，設計に関わる情報を設計者が作成するための作業空間として有用であり，また作成した情報を関係者に伝達するうえで欠かせないものである [15]．その代表例が図面である．設計対象の情報は複雑であり，ただ眺めているだけでは何と何が因果関係となっているかを把握することが困難な場合が多く，設計対象モデルはその因果関係を把握するうえで重要な役割を担う．特に昨今の製品は複雑化しており，そのため設計対象モデルの作成を計算機によって支援した，いわゆる CAD (Computer Aided Design)が必要不可欠となってきた．従来の CAD は，設計が物理世界の制約を満たす製品をつくる作業過程であることに着目し，その物理世界における製品を幾何モデルとして計算機上で表現することによって，製品設計を支援している [86].

また本節の冒頭で述べたように，設計過程は「問題提起」「提案」「展開」「評価」「決定」という設計行為の繰り返しである．そのため製品設計を支援する計算機環境は，設計対象モデルを作成可能とするだけでなく，作成した設計対象モデルを用いてのシミュレーションを実行可能とすることで，ノミナル情報の提案・展開と評価という設計行為の繰り返しを支援することが重要となる．製品設計過程で作成した設計対象モデルを用いてのシミュレーションを支援する計算機環境としては，製品性能を解析するための CAE (Computer Aided Engineering)システムが普及している．CAE システムのようなシミュレーションシステムの普及は，設計過程で導出した製品のノミナル情報の評価を，実体化に先立って可能にするという効果をもたらし，製品開発のリードタイムを向上させた．

製品設計と同様に，ライフサイクル設計の設計過程を支援する計算機環境の重要性が，木村ら [87]によって指摘されている．ライフサイクル設計の設計過程を支援する計算機環境は，これまでにいくつかの研究で報告されている．既存のライフサイクル設計支援環境は，図 2-3 のライフサイクル設計過程を，以下 4 つの側面から支援している．

① 「(1) ライフサイクル戦略の策定」から「(2) 製品とライフサイクルフローの設計」への展開支援
② 「(2) 製品とライフサイクルフローの設計」におけるモデリング支援
③ 「(2a) 製品設計」から「(3) ライフサイクル評価」の実行支援
④ 「(2b) ライフサイクルフローの設計」から「(3) ライフサイクル評価」の実行支援

① 「(1) ライフサイクル戦略の策定」から「(2) 製品とライフサイクルフローの設計」への展開支援

ライフサイクル戦略策定段階で特定した製品ライフサイクルに対する要求を満たしているか把握しながら，製品のノミナル情報を決定するための計算機環境が提案されている．例えば蔵川らは，製品ライフサイクルにわたる各側面からの要求とライフサイクルプロセスのネットワークモデルとを関連付けた web ベースの設計支援環境として，グリーンブラウザ [88]を提案している．グリーンブラウザは，製品のノミナル情報や LCA データなどを含む設計根拠を web 上で参照可能とすることで，設計者やユーザを代表とした製品ライフサイクルにわたる様々な主体間での合意形成がなされた環境配慮型の製品設計を支援する．Herrmann らは，EoL 処理工程に対する要求を製品設計要求に加えて管理することによって，設計上流段階から EoL 処理工程を考慮した製品設計を支援する FOD-ProdTect-CAD [89]を提案している．FOD-ProdTect-CAD は，「機能と階層構造という 2 つの視点から製品のノミナル情報を表現したモデルを，設計要求と対応付けて作成可能とする Functional Oriented Design (FOD)ツール」，「リサイクル率や最適な分解手順といった，製品の EoL 評価が可能な商業ソフトウェアである ProdTect」，「CAD システム」という 3 つのソフトウェアを，XML ベースのインターフェースによって統合した設計支援環境である（図 2-4 参照）．3 つのソフトウェアの統合によって FOD-ProdTect-CAD は，例えばリサイクル率やリサイクルコストといった EoL 処理工程に対する要求を満たすような製品のノミナル情報を導出し，また導出したノミナル情報を機能や製品構造や幾何形状について評価することができる．

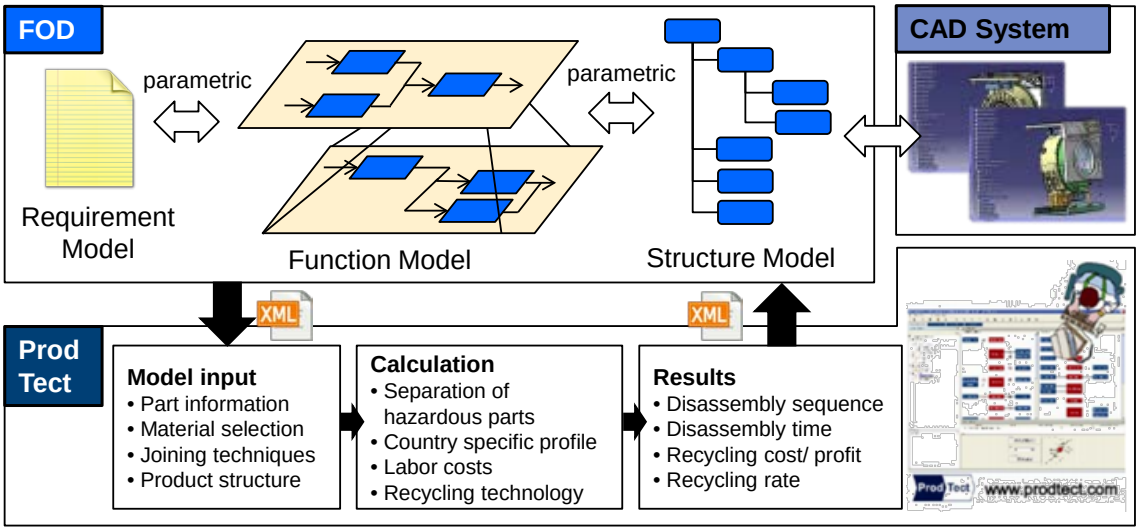


図 2-4：FOD-ProdTect-CAD（ [89]を基に作成）

② 「(2) 製品とライフサイクルフローの設計」におけるモデリング支援

製品とライフサイクルフローのノミナル情報を統合的に決定するための計算機環境として，國井は以下 3 つの要件を満たす製品ライフサイクルのモデリング支援が必要であると指摘している [13].

(ア) 製品とライフサイクルフローの関係が表現可能
(イ) 製品ライフサイクルのライフサイクル評価が可能
(ウ) 製品とライフサイクルフローの関係が適したものとなっているか評価可能

製品ライフサイクルのモデリング支援に対する 3 つの要件に対して國井は，以下のアプローチをとっている．まず要件(ア)に対しては，製品とライフサイクルフローのノミナル情報それぞれを表現したモデルと 2 つのモデル間の対応関係を定義した設計対象モデルを提案している．本設計対象モデルを，製品ライフサイクルのノミナル情報モデルと呼ぶこととする．また要件(ウ)に対して，製品とライフサイクルフローのノミナル情報を表現したモデル間の整合性を管理する手法を提案している．この製品ライフサイクルのノミナル情報モデルと整合性管理手法から成るモデル化手法を実装した製品ライフサイクルのノミナル情報モデル化システム [13]は，ライフサイクル CAD [10]の実現に向けたモデリング支援環境である．しかし，1.1.2 項で述べたように，現実世界の製品ライフサイクルには個体情報が異なる多様な個体が存在するものの，本モデル化システムでは多数存在する個体を表現することはできない.

③ 「(2a) 製品設計」から「(3) ライフサイクル評価」の実行支援

製品のノミナル情報の提案と提案したノミナル情報のライフサイクル評価を円滑に実行可能とするために，市販の CAD システムと LCA を統合した計算機環境が多く提案されている．例えばダッソーシステムズ社は，Solidworks Sustainability を開発している [90]. Solidworks Sustainability は，設計者が作成したソリッドモデルに，製造地域や使用環境を入力することで，ライフサイクルにわたる環境負荷排出量を算出できる．また，どの部品が最も多く環境負荷を排出するかを特定し，製品物性や機能を保ちながらも環境負荷排出量の削減を図る代替材料の選定を支援している．しかし，ライフサイクルフローの表現が限定されているため，評価の自由度が低いという問題がある．この問題に対して井上らは，ライフサイクル・オプションや寿命などの非形状情報を付与した 3D CAD モデルを，LCA と対応付けて作成可能な計算機環境を提案している [91]. 非形状情報と 3D CAD モデルと LCA を統合することで, 製品設計者が 3D CAD モデルを用いて変更した製品のノミナル情報について，各部品のライフサイクル・オプションを考慮した環境影響評価を円滑に実行可能としている．同様に Russo らは，「CAD システム」「有限要素解析ソフトウェア」「製品構造と機能に対する要求を満たす最適な幾何形状，素材，製造方法を提示する Structural Optimizer」という 3 つのサブシステムからなる製品プロトタイピングシステムに，LCA ツールを統合した Eco-OptiCAD を提案している [92]. Eco-OptiCAD は，製品の幾何形状, 素材, 製造性に関する設計要求についての最適解の導出と, 導出した解を 3D CAD モデルを介して LCA に入力することで，ライフサイクルにわたる環境影響評価を可能とする計算機環境である（図 2-5 参照）．これら 3D CAD モデルと LCA を統合した計算機環境は，3D CAD モデルを用いて記述した製品のノミナル情報について，静的かつ代表的な製品ライフサイクルの評価を支援するものの，LCA と同様に動的な製品ライフサイクルの評価や個体情報の多様さを考慮した製品ライフサイクルの評価ができない.

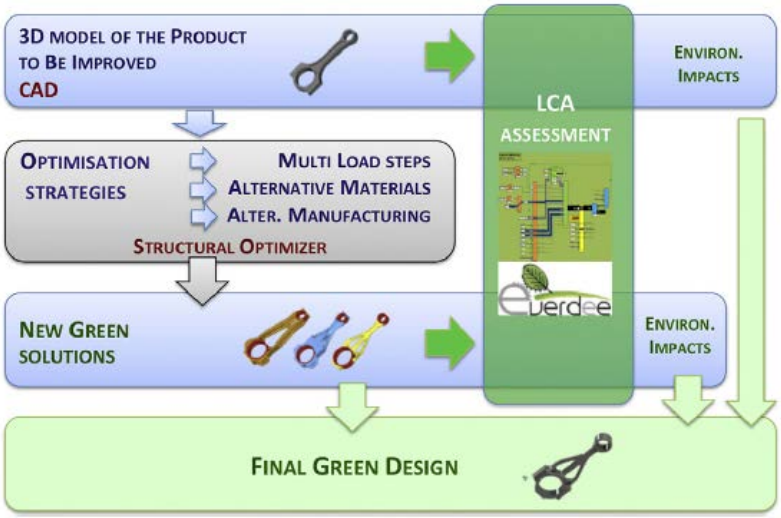


図 2-5：Eco-OptiCAD [92]

また，分解性やサービス性といった製品ライフサイクルの様々な側面からの評価を支援する計算機環境が提案されている．例えば Eubanks ら [93]は，生産や検査やメンテナンスやリサイクルといったライフサイクルプロセスにおいて，分解と組立が行われることに着目したコスト評価モデルを提案している．本評価モデルは機械システムの構造を，部品や処理を示すノードと部品に対する処理の関係を示すリンクを用いたグラフ構造によって表現している. 機械システムをグラフ表現することで, Eubanks らが開発したツール [93]は，例えばメンテナンス対象部品の交換にかかるコストを，対象部品を表すノードにアクセスするまでのリンク数によって評価可能としている．Kalyan-Seshu ら [94]は，3D CAD モデルから製品情報を取得して様々な DfX ツールに入力することで，対象製品が製品ライフサイクルに対する設計要求（例えば，組立性，サービス性，リマニュファクチャリング性など）を満たしているかの評価が可能な計算機環境を提案している．同様に Roche らが開発した DFE Workbench [95]は，3D CAD モデルから取得した製品情報を入力に LCA や DfE ツールを実行することで，ライフサイクルにわたる環境負荷排出量やリサイクル率や分解時間などを導出する．DFE Workbench は，導出した様々な評価結果について優先順位を決定することで，設計者にとって最適な製品構造を提示する計算機環境である．

④ 「(2b) ライフサイクルフローの設計」から「(3) ライフサイクル評価」の実行支援

ライフサイクルフローのモデル化やシミュレーション手法として，ペトリネットを用いてライフサイクルフローを表現（図 2-6 参照）し，ライフサイクルフローにわたる資源の流れをシミュレーションする手法が提案されている（例えば， [96] [97]）．また，2.2.3 項で示した LCS（例えば， [14]）は，ライフサイクルフローにわたる資源の流れや，その流れに伴って生じる環境負荷やコストなどをシミュレーションする手法である．さらに Komoto は，PSS のノミナル情報を設計者が体系的に創出かつ評価可能な CAD システムとして，Integrated Service CAD and Life cycle simulator（ISCL）を開発している [98]. ISCL は，顧客要求を満たす機能の提供が可能なサービスシステムの計画図を作成支援した計算機環境であるサービス CAD に LCS を組み込むことで，PSS の全体構造のモデル作成と製品ライフサイクル全体の環境性や経済性の評価を支援した計算機環境である．

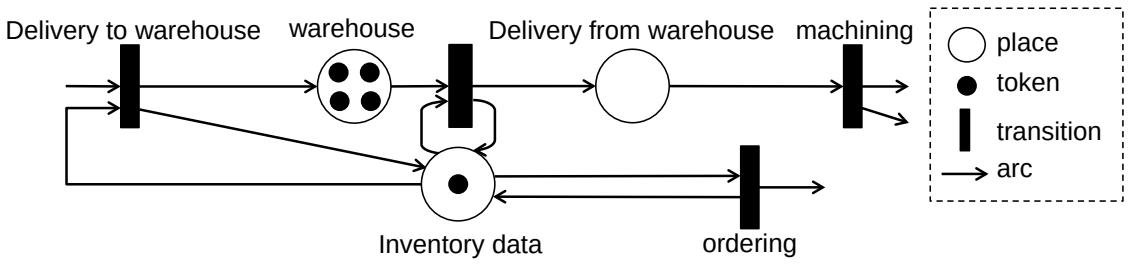


図 2-6：ペトリネットを用いたライフサイクルフローの表現 [96]

2.3. 現実世界における製品ライフサイクルのマネジメント

1.1.2 項で示したように，ライフサイクルにわたる個体の集合には状態に違いがあり，またその量が時間とともに変化する．各ライフサイクルプロセスで処理する個体の状態の違い（例えば，ネジの錆やネジ穴の変形の違い）によって，一個体一個体に対するメンテナンスの作業時間や製品販売後のサービス活動の提供時間が変動したり，部品個体の劣化度合いによって保守内容が変わったりする(例えば, 1 部品だけの交換か, オーバーホールか)．また回収量の変動に伴い，分解，検査，修理といったライフサイクルプロセスでの，日ごとの作業量は変動する．そのため，日々の作業量を許容できるよう，従業員や機材の稼働をスケジューリングすることが困難となる．特に，資源循環を含む製品ライフサイクルでは，製造業者の手を一度離れて再び戻ってくる個体（例えば，リユース部品やリサイクル材）が部品製造や製品組立のための供給の一部を担う．資源循環において供給の一部を担う個体は，いつ，だれによって，どのような状態で引き渡され，またその結果どれくらいの量を供給できるかが不安定かつ不確実である [99]．この点が，工場の生産能力や需要動向などに基づいて供給資源の状態や量を決定することができる順工程のみを対象とした生産システムを構築する場合と比較して，資源循環システムを構築することが複雑な問題となる理由のひとつである．この供給の不安定さや不確実性に加えて，循環資源の候補となる個体が新品部品として生産されてから，リユース部品やリサイクル材としての需要が決まるまでに，時間的な隔たりがある．その間に技術が発展し，生活形態が変化し得るため，資源循環における需要には不確実性が伴う [100].

以上に示した循環資源の供給の不安定さや不確実性と需要の不確実性に起因する資源循環システムの問題に対して，個体の回収量を予測することで資源の効果的な循環方法を策定する手法が，様々な分野で報告されている．例えば，使用段階の製品を循環資源のストックと捉え，ストックである市場に流入出する資源の量をモデル化した手法として，Material Flow Analysis (MFA)がある．中島ら [101]は，6 種類 11 品目の金属が市場に流入出する量を，廃棄物産業連関表を用いて分析する手法を提案している．しかし MFA は，個体に生じる確率的な挙動（例えば，故障）をモデル化した手法ではない [98]．確率的挙動を含めて製品個体が回収される時期や量をモデル化した研究としては，例えば複写機の実回収データを用いた回収量予測モデルがある [102]．また小口ら [103]は，22 品目の製品を市場調査し，その調査結果に基づいて製品特性と平均使用年数の関連性を定式化した数量モデルを提案している．小口らのモデルは，様々な電気電子機器製品の平均使用年数と使用済み台数の推計が可能である．さらに，[102]のような回収量予測モデルを用い，逆工程における資源の物流ネットワークを策定する手法や，循環資源の供給量と需要量の関係から資源の流れをマネジメントする手法が，Reverse Logistics [99]や Reverse Supply Chain Management [104]の分野で報告されている．Kara らは，Reverse Logistics に重要な要素として，分解工場の数や配置を挙げている [105]．分解工場の数や配置が重要な理由として，回収対象の製品個体が広域的に散在しており，資源循環のためには散在している

個体を製造地に戻す必要があるが，輸送に大きなコストがかかってしまい，新品部品を作る場合と比べて経済効果が小さくなり得る点が挙げられる．Reverse Logistics における分解工場の数や配置という要素に着目した研究として，例えば村山らは，回収製品やリユース可能部品の量を予測する reliability model [106]に，分解工場の地理的情報を加えたモデルを入力とし，ペトリネットによるシミュレーションを実行することで，環境性と経済性の観点から最適な分解工場の数と配置を策定する手法を提案している [107]. また Demirel ら [108]は，回収，分解，廃棄といった逆工程全体で発生するコストを最小化するような資源循環システムを構築するために，新規製造とリマニュファクチャリングする製品それぞれの最適な量と製造工場やリマニュファクチャリング工場の最適配置を策定する数理モデルを提案している. 以上のようにReverse LogisticsやReverse Supply Chain Managementの研究では，製品個体の回収量の変動を表現しているものの，ライフサイクルにわたって個体の集合が示す状態の違いを表現した手法はほとんどない．状態の違いを不確実性として分解工程を設計する手法はある（例えば， [109]）が，その状態の違いが時間変化することを前提としていない.

また，現実世界における製品個体から使用履歴や劣化進展などの情報を収集することで，メンテナンスや製品回収を効率的に実行する手法として，Product Lifecycle Management (PLM)や近年盛んに研究開発されている Internet of Things(IoT)技術が利用できる．PLM の用途は様々である．例えば，製品のノミナル情報やシミュレーション結果といった製品開発における成果物（エンジニアリング情報と呼ぶ）をメンテナンス工程や分解工程の主体に閲覧可能とすることで，メンテンスサービスの質の向上や使用済み製品の分解時間の削減に活用されている．特に，エンジニアリング情報のデータ構造を定義した研究が多く報告されている（例えば， [110] [111]）．また，製造した製品個体の情報を一元管理するためのシステムアーキテクチャを定義した研究も多い．例えば Främling ら [112]は，製品個体の使用履歴や劣化進展に関する情報を収集し，また収集した膨大な情報をライフサイクルにわたる様々な主体が個々の目的に応じて抽出できるように管理する手法を提案している．さらに，IoT 技術を用いて資源循環を支援する手法に取り組んだ研究も多くある．例えば Wang らは，製品のノミナル情報や使用履歴などをクラウド管理することで，製品のリマニュファクチャリングを支援するシステムアーキテクチャを構築している [113]. しかし，PLM や IoT 技術を用いて収集した個体情報を活用し，製品ライフサイクルを設計するための方法論は確立していない.

2.4. 第 2 章のまとめ

本章では，ライフサイクル設計支援に関する先行研究と現実世界における資源循環や製品ライフサイクルのマネジメントに関する先行研究について述べた．

まず，ライフサイクル設計の基本コンセプトを示した．次に，ライフサイクル設計の設計過程を，一般的な製品設計の設計過程に基づいて論じ，本設計過程に関する先行研究について調査した．ライフサイクル戦略策定に関する既存手法は，あくまで製品ライフサイクルのコンセプトの決定支援を目的とするため，各ライフサイクルプロセスでの処理方法や製品の構造や属性を決定する手法ではない．また，製品設計とライフサイクルフローの設計は密接な関係にあるが，2.2.2 項に示した製品設計やライフサイクルフローの設計に関する先行研究のうち，その両者を併せて設計するための手法はない．つまり，製品とライフサイクルフローの設計を独立に行うことを前提とした研究が大半である．この問題に対し，製品ライフサイクルのノミナル情報を表現する設計対象モデルのモデリングを支援する計算機環境，および製品設計とそのライフサイクル評価の実行を支援する計算機環境が，2.2.4 項で述べたようにいくつかの研究で提案されている．しかし，既存のライフサイクル設計支援環境の大半は，個体情報の異なる多数の個体が存在する製品ライフサイクルを，対象製品の 1 個体のライフサイクルにより代表させて表現しており，製品個体それぞれを表現している訳ではない．一方で，2.2.3 項に挙げたように，ライフサイクルにわたる個体の流れをシミュレーションする手法として LCS がある．しかし，LCS の先行研究では，多数存在する製品個体の個体情報を製品ライフサイクルのノミナル情報の決定に活かすための設計方法論に関する議論が十分になされていない．

さらに，現実世界における資源循環や製品ライフサイクルのマネジメントに関する先行研究を調査した．Reverse Logistics や Reverse Supply Chain Management の研究では，製品個体の回収量の変動を表現するものの，ライフサイクルにわたって個体の集合が示す状態の違いを表現した手法はほとんどない．また，PLM や IoT 技術など，現実世界における個体情報を収集するための研究が報告されているものの，収集した個体情報を活用して製品ライフサイクルを設計するための方法論は確立していない．

第3章 製品個体の集合に着目した製品ライフサイクルの設計方法論の基本コンセプト

第 3 章では，製品個体の集合に着目した製品ライフサイクルの設計方法論に関する基本的な考えを示す．まず製品ライフサイクルを 3 つの階層に分類し，この 3 つの階層から製品ライフサイクルを捉えた場合の設計問題について述べる．次に，本設計問題を含む製品ライフサイクルの設計を実行するために方法論が満たすべき要件を示す．また，本要件に関連する先行研究として，製品ライフサイクルのノミナル情報のモデル化手法 [13]と Life Cycle Simulation (LCS) [14]について詳述する．最後に，本研究の位置付けを示す．

3.1. 製品ライフサイクルの基本的な考え方

本研究では製品ライフサイクルを，以下 3 つの階層に分類して捉える（図 3-1 参照）.

階層 1：製品とライフサイクルフローのノミナル情報
ライフサイクル設計の設計対象であり，設計者が設計時に決定可能な世界
階層 2：仮想的な個体の集合
階層 1 と階層 3 の中間．階層 1 についての階層 3 を仮想した世界
階層 3：現実世界における個体の集合
ライフサイクルにわたる個体が集合として状態の違いや量の変動を示す現実世界

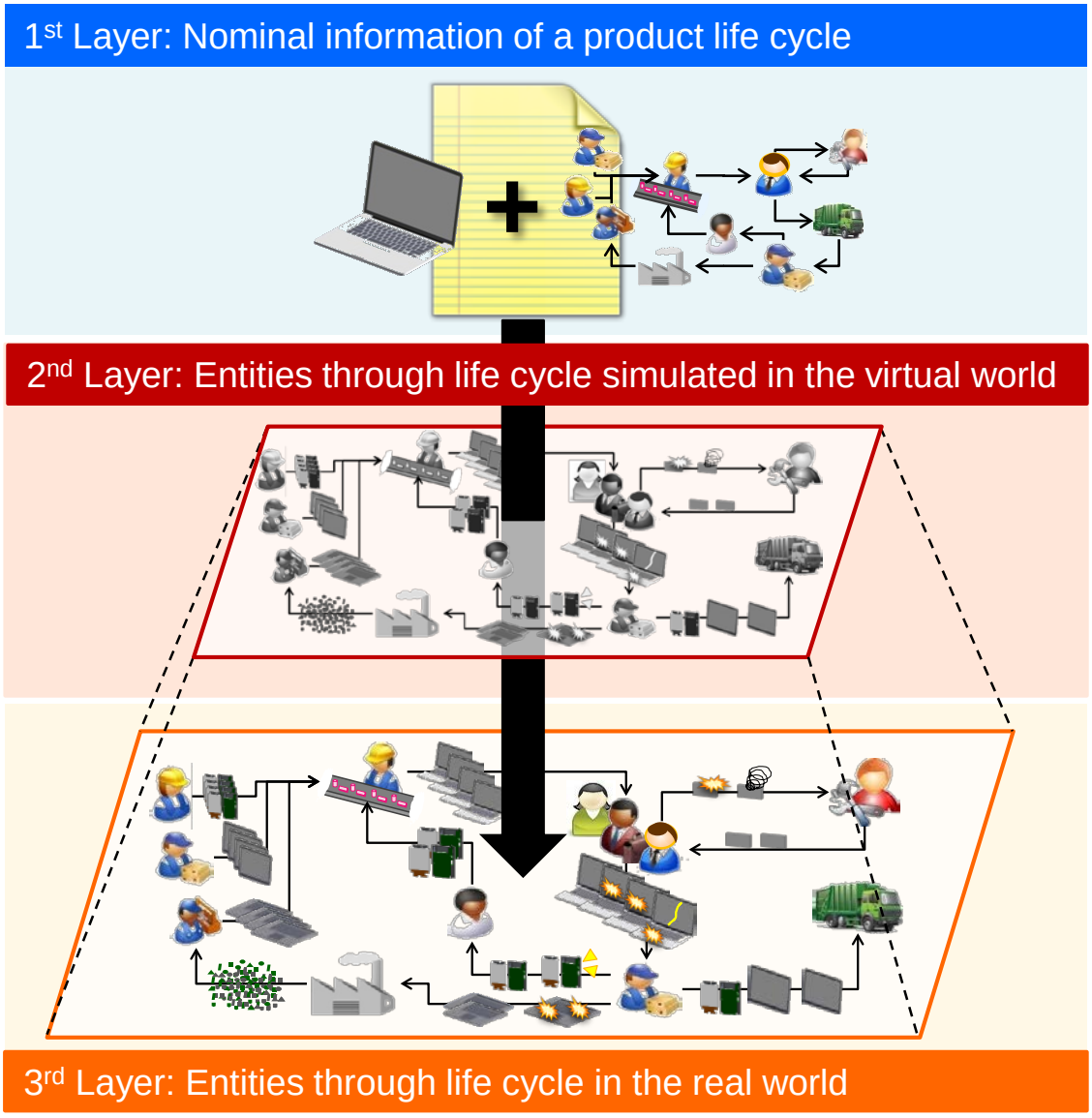


図 3-1：製品ライフサイクルの 3 つの階層

このうち，階層 3 の現実世界において個体の集合が示す「(a)状態の違い」と「(b)量の変動」という製品ライフサイクルに関する 2 つの特徴と，これらの特徴を引き起こす要因（変動要因と呼ぶ）について，以下で詳述する．

(a) 個体の状態の違いとその変動要因

製品個体は同じノミナル情報に基づいて一律に生産された場合であっても，ライフサイクルにわたって個々に異なる状態を示す．つまり，製品個体の状態はライフサイクルを通じて変化するが，その状態変化は個々の個体で異なり得る．特に，一度使用されて循環してくる個体は，バージン材や新品部品個体に比べて同質さに欠けている [99]．

個体の状態に違いが生じることは，従来の製品開発においても重要視されてきた．市場に流通した製品個体に違いがあれば，不具合発生の原因となる可能性があるためである．市場に流通する製品個体に違いが生じる最大の要因は，製造段階に存在する [114]．市場に流通する製品個体の違いは具体的に，加工と組立という 2 つの処理工程における「(a1)個体の物理的な性質の違い」と「(a2)人や機械が施す処理の違い」という 2 種類の変動要因に起因して生じる [12]．図 3-2 が示すように，加工工程では「(a1)使用部材の純度の違い」と「(a2)加工者や加工機械の能力に依存した加工精度の違い」に起因し，加工した部品の状態（例えば，表面粗さなどの幾何形状）に違いが生じる．また組立工程では，「(a1)加工に起因して異なる状態を示す部品個体の組み合わせ」と「(a2)組立作業者や組立機械による部品個体の組立精度の違い」に起因して，仕上がり製品個体の状態に違いが生じる．

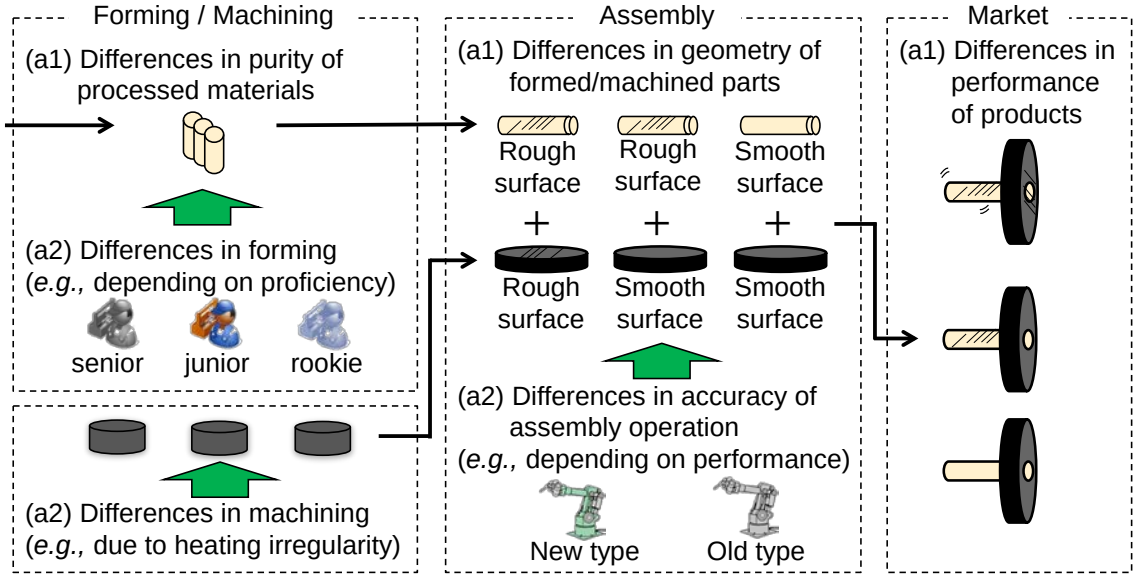


図 3-2：市場に流通する個体の状態の違いとその変動要因（ [12]を基に作成）

個体の物理的な性質や施す処理に違いが生じることは，避けられない現象である．製品ライフサイクル全体に視野を広げると，製造業者による加工や組立に加えて，ユーザによる使用や修理業者によるメンテナンスといった，より多くのライフサイクルプロセスにおいて，様々な主体が様々なレベルで様々な処理を施す．具体的には，あるライフサイクルプロセスで同じ処理を施す製品個体であっても，製造工程での変動要因(a1)と同様に，ライフサイクルを通じて蓄積した物理的な性質の違いによって異なる挙動を示す．また，ユーザによって使用頻度が異なったり，メンテナンス業者によって修理能力が異なったりするなど，変動要因(a2)に該当する処理の違いによって個体の状態に差が生じる．特に資源循環を含む製品ライフサイクルでは，個別の処理が施された個体がリユース部品やリサイクル材として再度製品個体に利用され，再び市場に投入される．その結果，個体それぞれの状態は分散が拡大し，より多様となる．

(b) 個体の量の変動とその変動要因

各ライフサイクルプロセスで処理する個体の量は時間変化する．そもそも処理する個体の量は，ライフサイクルプロセスごとに異なる．なぜならば，同じノミナル情報に基づいて生産された個体であっても，「(b1)辿るライフサイクルが個々に異なる」ためである．例えば，腐食の進展が小さい部品個体はリサイクルされ得るが，腐食の進展が大きい部品個体はリサイクルされずに廃棄され得る．この各ライフサイクルプロセスで処理する個体の量が，上流のライフサイクルプロセスで施される以下 2 種類の処理の違いに起因して時間変化する．第一に，各個体に対して施される処理のタイミングが「(b2)主体による処理の違い」に起因して異なる．第二に，外部環境の変化に伴って「(b3)個体に対する処理自体が時間変化」する．例えば製品個体が市場から廃棄されるタイミングは，劣化と陳腐化に起因する [104]．劣化による廃棄は，各ユーザの使用頻度に依存して製品個体そのものの変化の程度が個々に異なることで，そのタイミングに違いが生じ得る（(b2)に該当）．陳腐化による廃棄は，同じ使用環境下の同じ主体が使用している製品個体であっても，技術進展によって相対的な価値が低下することで発生する（(b3)に該当）．

3.2. 製品ライフサイクルの設計問題

本節では，3.1 節で示した 3 つの階層から製品ライフサイクルを捉えた場合の，ライフサイクル設計の設計問題について示す.

現実世界において個体が効率的に循環するような製品ライフサイクルを構築するためには，設計段階で個体情報を模擬し，その仮想的な個体情報が製品ライフサイクルに対する様々な設計要求を満たす範囲で多様さを示すように，設計者にとって操作可能な製品とライフサイクルフローのノミナル情報を介して制御することが重要である．対象の製品ライフサイクルが設計要求を満たすかどうかを多様な個体情報について評価する（以下，個体情報の評価）には，様々な側面や粒度が考えられる．例えば，多様な個体に対して処理を施す過程で生じる環境負荷排出量や資源消費量といった環境性，およびユーザコストやメーカ利益といった経済性など，製品ライフサイクル全体の評価が考えられる．また，限界リユース率 [84]のような循環資源の需要量と供給量のバランスの評価，およびメンテナンス工程のスケジューリングや逆工程のライン設計をする過程で，処理する個体の状態のバラつきが許容可能か否か評価するといった，ライフサイクルプロセスレベルでの評価が考えられる．さらには，リユース部品個体の性能を厳密に評価するために個々の個体を劣化シミュレーションするといった，個別のライフサイクルプロセスにおける個体レベルでの評価が考えられる．このようなライフサイクルにわたる様々な側面や粒度からの個体情報の評価結果が設計要求を満たすように，製品およびそのライフサイクルフローのノミナル情報を探索することが設計問題となる．具体的な設計問題を，以下に示す.

第一は，個体の状態の問題である．1.1.2 項で述べたように，資源循環を含む製品ライフサイクルを構築しようとしても，個体の状態に違いがあることで，多くの個体が循環に関する設計意図に沿わず，かえって環境負荷排出量やコストが増加する可能性がある．このような問題に対し，従来の製品開発では，目標とする状態から大きく外れた個体が生じてしまい，使用段階での不具合発生につながることがないよう，製品や製造処理方法に関するノミナル情報を決めることが求められてきた [115]．例えば，加工による仕上がり精度にバラつきが出にくい素材の選択や加工方法の適用が，対応方法のひとつである．一方で，ライフサイクルにわたる環境負荷排出量や資源・エネルギー消費量を最小化するためには，個体の状態の違いを小さくすることが必ずしも是ではない．例えば，部品個体の状態の違いは，多様な顧客要求を活用することで資源循環を成立させ，環境性を向上させる可能性があるためである．具体的には，性能低下が小さい部品個体はスペアパーツ部品としてリユースし，性能低下が大きい部品個体は低い性能でも安価な製品を求めるユーザに部品リユースするよう設計できる可能性がある（図 3-3 参照）．しかし，この部品リユースを実現するための設計によって，製品ライフサイクル全体の環境性や経済性が改悪する可能性もある．そのため，上記の例では，「個体の状態の違いを小さくして，単一の方法で部品リユースする」か「個体の状態の違いをそのまま活用して，図 3-3 の 2 通りの方法で部品リユースする」か「部品リユースを棄却する」かを，設計要求を満たすかどうかによって判断

し，その循環方法を実現する製品ライフサイクルを設計することが考えられる．このように，個体の状態の違いを小さくして循環に活用するかそのまま活用するかを，設計要求を満たすかどうかによって判断し，解決策となる製品やライフサイクルフローのノミナル情報を探索することが設計問題となる．

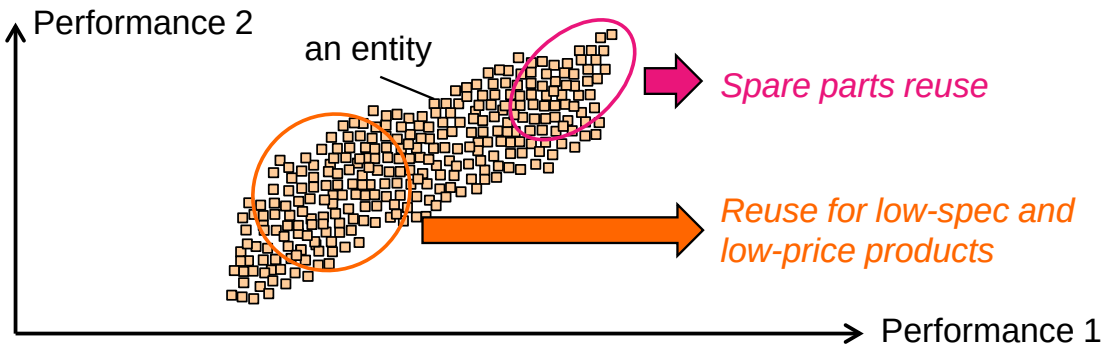


図 3-3：状態の違いを活用した資源循環の例

第二は，個体の量の問題である．循環資源としての製品や部品個体の需要と供給の間には，発生時期と発生量のミスマッチが生じ得る．つまり，限界リユース率 [84]として議論されているように，循環資源の需要である製品や部品の製造時期と供給である製品の回収時期との重なりが，循環可能な資源の最大量となる．例えば，レンズ付きフィルムのような寿命の短い製品では同一世代での限界リユース率が高い（図 3-4(a)参照）．レンズ付きフィルムは実際に，同一世代での部品リユースが成立している [18]．一方で，回収された製品個体が同一機種製品にリユース可能な残余寿命をもつ部品個体を含むとしても，回収時期には次世代機種製品の製造へと移行している場合が想定される．例えば，複写機のような中程度の寿命の製品では，同一世代の製品製造期間に対して製造されてから回収に至るまでの期間が長い．この場合，同一世代ではなく次世代機種に対して部品リユースするよう設計することで，循環可能な資源を最大限に活かせる可能性が生じる（図 3-4(b)参照）．このとき，次世代機種への部品リユースが製品構造的に実現可能か，また世代間で部品の共通化・共有化をすることで環境性や経済性が改悪しないかどうかによって，量の変動の活用方法を決定し，その活用方法を実現する製品ライフサイクルを設計することが考えられる．または，各ライフサイクルプロセスに流れてくる個体の量を多くするか少なくするか，また流れてくるタイミングを早くするか遅くするかを，製品やライフサイクルフローのノミナル情報を操作することで実現可能かどうかによって判断するなど，量の変動を制御して資源循環を実現するよう製品ライフサイクルを設計することが考えられる．このような量の変動への対応を，設計要求を満たすかどうかによって判断し，その解決策となるノミナル情報を探索することが設計問題となる．

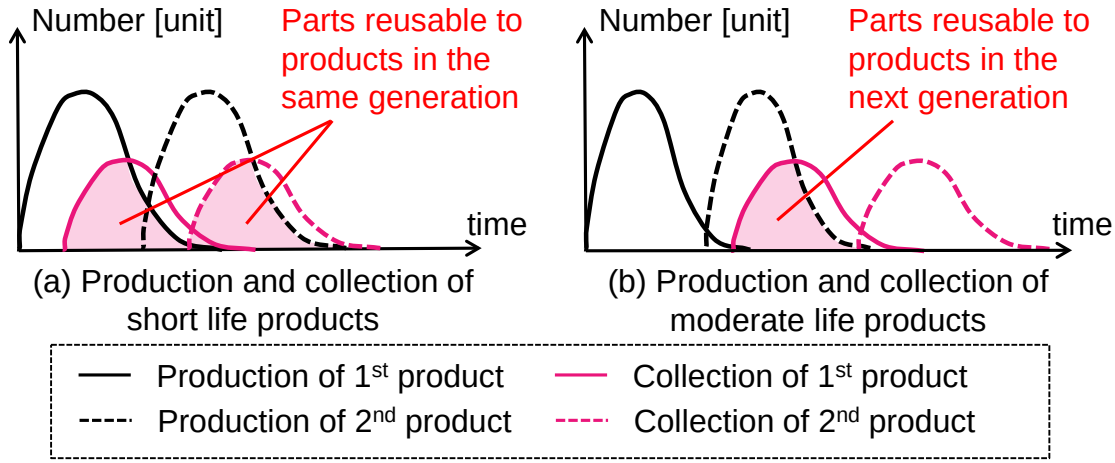


図 3-4：量の変動を活用した資源循環の例

3.3. 製品ライフサイクルの設計方法論に対する要件

3.2 節で示した設計問題を含む製品ライフサイクルの設計を実行するために方法論が満たすべき要件として，以下 3 つを挙げる（図 3-5 参照）.

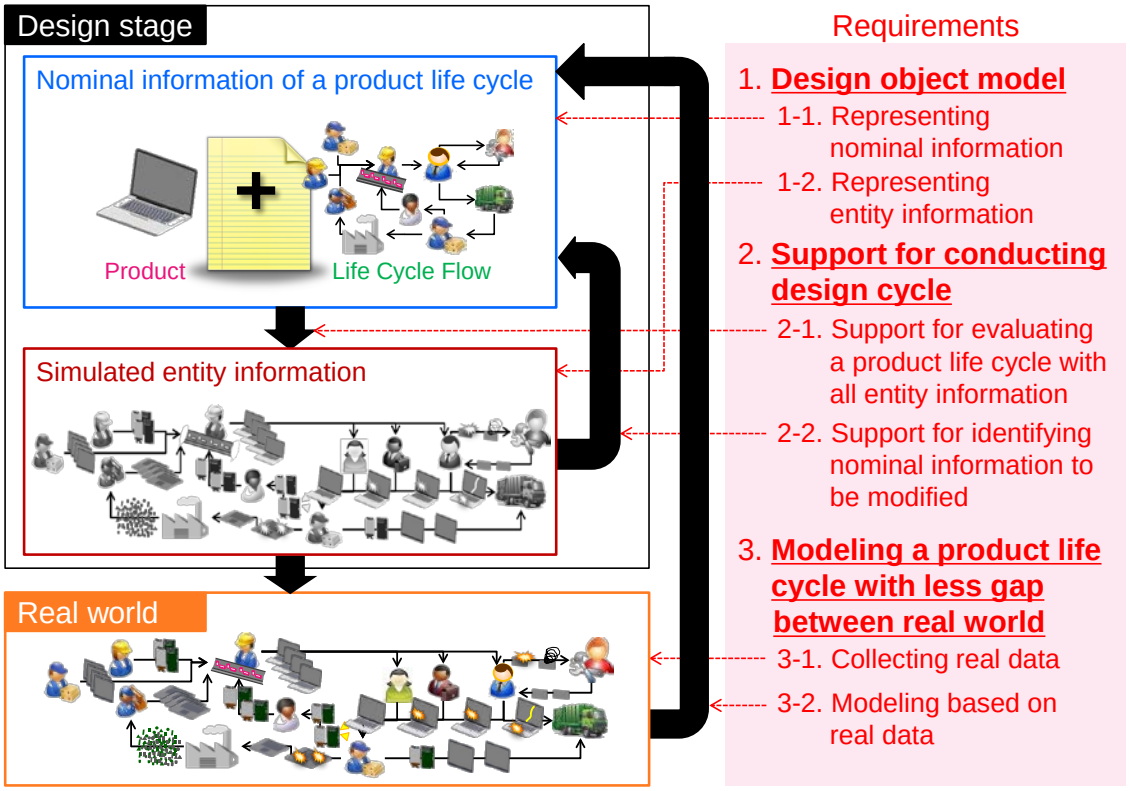


図 3-5：製品ライフサイクルの設計方法論に対する要件

1. 製品ライフサイクルをノミナル情報と個体情報の両面から表現可能とすること

3.2 節で述べた個体情報の多様さに起因する製品ライフサイクルの設計問題に対応するための作業空間として，以下 2 つを表現する対象モデルが必要である．

1-1. 製品ライフサイクルのノミナル情報の表現

ライフサイクル設計の設計対象である製品とライフサイクルフローの両方のノミナル情報を表現可能とする必要がある．また，資源は世代や品種を跨いで循環し得るため，多世代多品種にわたる製品ライフサイクルのノミナル情報を表現可能とする必要がある．

1-2. 個体情報の表現

個体情報の多様さに起因する製品ライフサイクルの設計問題とその問題の原因となる個体がどのようなライフサイクル履歴を示すかの予測（個体情報の評価）を設計段階で可能とするために，ライフサイクルにわたって個体が示す個体情報を明示的に表現可能とする必要がある．特に，3.2 節で示したようなライフサイクルにわたる様々な側面や粒度での個体情報の評価を，設計要求や解くべき設計問題に応じて実行可能とするモデルの存在が不可欠である．LCS はライフサイクルにわたる個体の流れをシミュレーションする手法であるが,既存の LCS ではシミュレーション実行後に個体情報は消えてしまう．シミュレーション実行後も個体情報を仮想的かつ継続的に存在させ，製品一個体一個体の集合として製品ライフサイクルを表現し，上記の評価を実行可能とするモデルが必要である．

2. 製品ライフサイクルの設計過程における設計サイクルを円滑に実行可能とすること

製品だけでなくライフサイクルフローのノミナル情報の決定をも含む製品ライフサイクルの設計過程においては，従来の製品開発プロセスに極力支障をきたさないよう設計サイクルを円滑に実行することが求められる．本研究では，製品ライフサイクルの設計過程における特定の設計段階での設計サイクルを，2.2 節で述べた一般的な製品設計の設計過程[24]に基づいて，以下 4 つの設計行為の繰り返しとして捉える（図 3-6 参照）．

(i) 問題提起：設定した設計要求を満たすために，対象の製品ライフサイクルのノミナル情報について解決すべき問題を提起する．
(ii) 提案・展開：提起した問題に対して，製品とライフサイクルフローのノミナル情報を導出する．
(iii) 評価：導出したノミナル情報について，個体情報の評価を実行する．
(iv) 決定：個体情報の評価結果から，導出したノミナル情報を採用するか否か決定する．

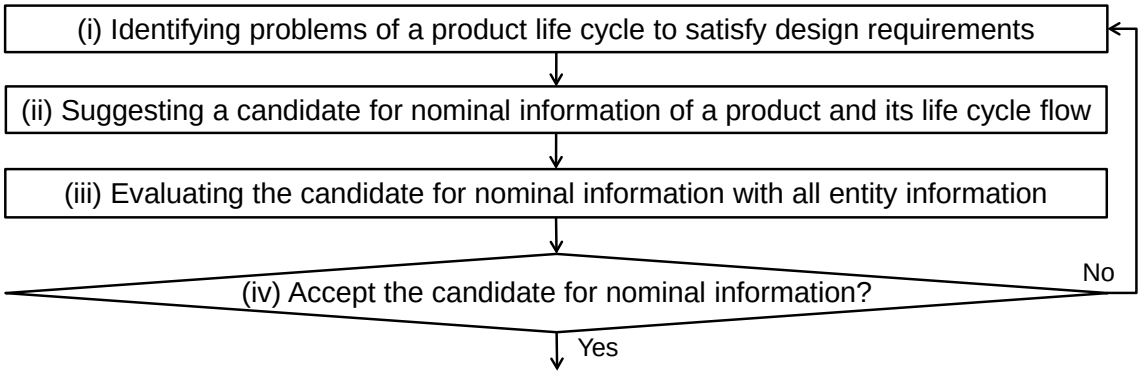


図 3-6：製品ライフサイクルの設計過程における設計サイクル

図 3-6 の設計サイクルを円滑に実行するためには，以下 2 つの設計過程の支援が必要である．

2-1.　個体情報の評価の実行

個体情報は，製品構造や属性といった製品のノミナル情報だけでなく，各ライフサイクルプロセスでの処理方法やそのネットワークといったライフサイクルフローのノミナル情報を構成する膨大なパラメータやパラメータ間の複雑な関係に依存して多様となる．例えば，製品個体の劣化は，使用頻度や設置環境や廃棄時期だけでなく，製品構造を含めた製品ライフサイクルの様々な変動要因が関係し合って決まる．個体情報の評価の実行のために膨大かつ複雑な関係のパラメータを表現するには大きな手間と労力を要するため，この表現の過程を支援する必要がある．

2-2.　ノミナル情報の修正

導出した製品ライフサイクルのノミナル情報について，個体情報の評価を実行した結果が設計要求を満たしていない場合，設計者はその問題を解決し得る修正点を探索する必要がある．しかし，上述の通り，製品ライフサイクルのノミナル情報を構成するパラメータは膨大かつ複雑な関係をもつ．設計サイクルを円滑に実行可能とするためには，修正すべき製品およびライフサイクルフローのノミナル情報を，構成するパラメータやパラメータ間の関係から特定するための支援が必要である．

3. 現実世界との乖離が小さい製品ライフサイクルのモデリング支援を可能とすること

高い精度で評価を行い，適切なノミナル情報を決定可能とするためには，設計段階で模擬する個体情報と現実世界における個体情報との乖離を小さくすることが重要となる．そのための課題として，以下の 2 つが挙げられる．

3-1. 実データの収集と管理

現実世界の製品個体は，大量生産品である場合，個体数が膨大であり，かつ散在している．また，個々の製品個体の実データ（例えば，使用履歴，劣化進展）を，製造から廃棄に至るまでの長い期間にわたって収集する場合，そのデータ量は膨大となる．この空間的かつ時間的に膨大な製品個体の実データを収集するための技術開発と，収集した実データを管理するためのデータベースが必要である．

3-2. 実データを利用した製品ライフサイクルのモデリング

現実世界から収集可能な実データは，すでに設計が終了している前世代以前の製品の実データとなる．そのため，収集した実データを分析し，次世代の設計に利用する方法を検討しなければならない．例えば，使用頻度などライフサイクルプロセスでの処理に依存した実データは，次世代製品を同じ環境下で同じように処理すると想定できる場合，変動要因を表すパラメータにそのまま利用できる．一方で，劣化進展など，ライフサイクルプロセスでの処理だけでなく製品構造にも依存する実データもある．製品とライフサイクルプロセスの両方に依存する実データは，次世代製品の構造や属性が変わる場合，実データから製品構造に起因した挙動を除外してライフサイクルプロセスでの処理に起因する挙動だけを抽出する，もしくは実データを用いることで生じ得る現実世界との乖離を許容するといった，データの分析や解釈が必要となる．このようなライフサイクルにわたる様々な実データについて，製品ライフサイクルのモデリングに利用する方法を決定するための支援が必要である．

3.4. 製品ライフサイクルの設計方法論の要件に対する先行研究

本節では，前節で挙げた製品ライフサイクルの設計方法論の要件に対する先行研究について述べる．まず，要件 1-1 のノミナル情報の表現に対応した手法として，製品とライフサイクルフローのノミナル情報を対応付けてモデル化する手法（製品ライフサイクルのノミナル情報のモデル化手法と呼ぶ） [13]について述べる．次に，要件 1-2 に関連し，ライフサイクルにわたる個体の流れをシミュレーションする手法である Life Cycle Simulation (LCS)について述べる．

3.4.1. 製品ライフサイクルのノミナル情報のモデル化手法

製品ライフサイクルのノミナル情報のモデル化手法 [13]は，製品とライフサイクルフローのノミナル情報を対応付けて表現したモデル（製品ライフサイクルのノミナル情報モデルと呼ぶ）とその間の整合性を管理する手法から成る．製品ライフサイクルのノミナル情報モデルは，製品のノミナル情報を表す「(A)製品階層構造モデル」，ライフサイクルフローのノミナル情報を表す「(B)ライフサイクルフローモデル」，2 つのモデル間の「(C)対応関係」，という 3 つの要素から成る．製品ライフサイクルのノミナル情報モデルを構成するこれら 3 つの要素と「(D)整合性管理手法」について，それぞれ以下で詳述する．

(A) 製品階層構造モデル

製品階層構造モデルは，製品・モジュール・部品・素材といった実体を表す実体ノードと，実体間の関係を表すリンクから成る．その一例として図 3-7 は，液晶テレビ（Liquid Crystal Display-Television, LCD−TV）の製品階層構造モデルを表す．

実体ノードenは，実体の名前，品番，属性を保持する．なお，$en \in EN$であり，ENは製品階層構造モデルを構成する実体ノードの集合を表す．実体ノードenの属性$a_\alpha \in A_{en}$は，項目（例えば，構成素材，重量，性能，幾何形状）と属性値から成る．属性の一つである幾何形状を表すため，実体ノードはソリッドモデルを値として保持する．

リンクは，階層リンク，接続リンク，変換リンクの 3 種類から成る．階層リンクは，実体間の階層関係を表す．階層リンクで関係づいた実体ノードは，一方が構成要素を，他方がその集合を表す．接続リンクは，実体間の接続関係を表す．接続リンクで関係づいた実体ノードは，固定，運動拘束，運動伝達のいずれかの関係にあることを表す．変換リンクは，精錬や破砕といった実体間の変換関係を表す．

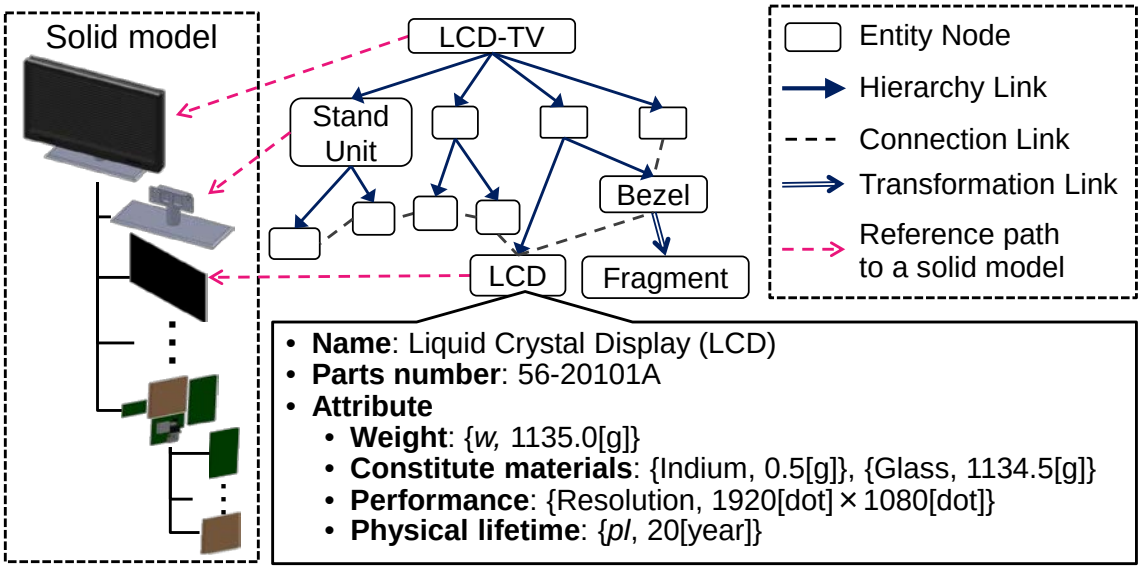


図 3-7：製品階層構造モデル（対象例：液晶テレビ）

(B) ライフサイクルフローモデル

ライフサイクルフローモデルは，ライフサイクルプロセスを表すライフサイクルプロセスノードと，ライフサイクルプロセス間の実体（製品，モジュール，部品，素材）や無形物（金銭，情報）の流れを表すフローリンクから成る．その一例として図 3-8 は，LCD-TV のライフサイクルフローモデルを表す．

ライフサイクルプロセスノードは，Given parameter，Input parameter，Output parameter，Procedure という 4 つの要素から成る．本研究では，ライフサイクルプロセスノードlcpを構成する各パラメータを，$GP_{lcp} = \{gp_{lcp,i} | i = 1,2,\dots I\}$，$IP_{lcp} = \{ip_{lcp,j} | j = 1,2,\dots J\}$，$OP_{lcp} = \{op_{lcp,k} | k = 1,2,\dots K\}$と置き，Procedure を$PR_{lcp}$と置く．なお，$GP_{lcp} \in GP$，$IP_{lcp} \in IP$，$OP_{lcp} \in OP$である．$GP$，$IP$，$OP$はそれぞれ，ライフサイクルフローモデルを構成する Given parameter，Input parameter，Output parameter の集合を表す．

Given parameter $gp_{lcp,i}$は，各ライフサイクルプロセスの処理能力や主体の要求といった定数属性を表す．例えば，回収率やリユース部品の検査基準が$gp_{lcp,i}$に該当する．Input parameter と Output parameter は，ライフサイクルプロセスに入力および出力する実体や無形物を表す．Input parameter $ip_{lcp,j}$は，個体の集合か外部からの信号（例えば，需要量）を表す．Output parameter $op_{lcp,k}$は，個体の集合か評価値（例えば，環境負荷排出量，資源消費量，コスト）を表す．ライフサイクルフローモデルはライフサイクルプロセス間の実体と無形物の流れを表すため，ライフサイクルプロセスノードの Output parameter の値は，下流のライフサイクルプロセスノードの Input parameter の値になる．

Procedure PR_{lcp}は，Given parameter，Input parameter，Output parameter 間の関係を定義することで，ライフサイクルプロセスの振る舞いを表す．例えば，「回収された LCD-TV を部品に分解する」といった分解処理の内容が，Procedure の記述対象となる．

本研究では，PR_{lcp}の処理の最小単位を，各行$pr_{lcp,l}$（lは自然数）と置く．$pr_{lcp,l}$によって$op_{lcp,k}$の値は変化し得るが，$gp_{lcp,i}$や$ip_{lcp,j}$の値は変化しない．$op_{lcp,k}$の値変化に関する処理を行う$pr_{lcp,l}$は，lcpを構成する他のパラメータを変数とした以下の式で表す．

$$op_{lcp,k} = pr_{lcp,l}(\{gp_{lcp,i'}\},\{ip_{lcp,j'}\},\{op_{lcp,k'}\}) \tag{3.1}$$

ただし，$gp_{lcp,i'} \in GP_{lcp}$，$ip_{lcp,j'} \in IP_{lcp}$，$op_{lcp,k'} \in OP_{lcp}$である．

本モデル化手法では，フローリンク上で実体の状態が変化したり，またライフサイクルプロセスノードを経ずにフローリンクが分岐や合流したりしてはならず，実体の状態変化や分岐や合流は必ずライフサイクルプロセスノード上で表現しなければならないと規定している．

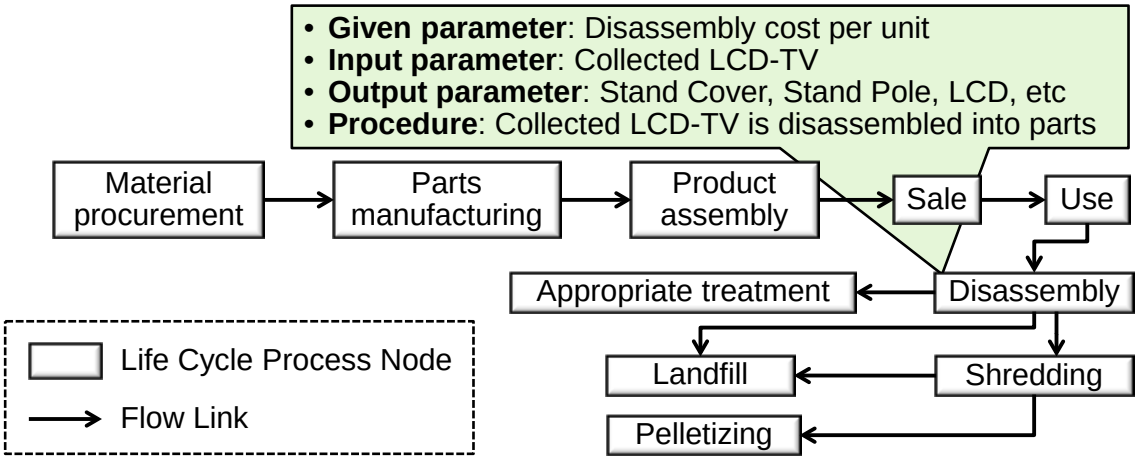


図 3-8：ライフサイクルフローモデル（対象例：液晶テレビ）

(C) 製品階層構造モデルとライフサイクルフローモデル間の対応関係

製品とライフサイクルフローのノミナル情報を統合的に決定するために，製品ライフサイクルの対象表現モデルが表現すべき側面として以下の 3 点が挙げられる [13].

- 実体ごとのライフサイクル

製品を構成するモジュール，部品，素材といった実体は，それぞれの特性に基づいて，辿るライフサイクルが異なる．例えば，LCD-TV の構成部品のうち，ABS 樹脂のみを構成素材とする Stand Cover はリサイクル可能であるが，冷陰極蛍光管を構成素材とする Backlight はリサイクル不可であり，適正処理する必要がある．このとき，Stand Cover はリサイクル工程を辿るように，Backlight は適正処理工程を辿るように，各部品のライフサイクルを同一モデル上で表現可能とすべきである．

● ライフサイクルを通じた実体の状態変化

実体はライフサイクルを通じて状態が変化する．例えば，組立や分解，またシュレッダ処理を経て部品が破砕片になる．各ライフサイクルプロセスでの実体の状態変化を表現し，表現した実体の状態変化が製造性やメンテナンス性やリサイクル性といった様々な側面から適切であるかを評価可能とすべきである．

● ライフサイクルにわたる主体による認識の仕方の違い

製品はライフサイクルにわたり，様々な主体によって異なる目的で処理される．例えば，組み立てる主体もいれば，分解する主体もいる．処理する目的が異なれば，製品の階層構造，つまり製品の認識の仕方は異なり得る．具体的には，製品を製造する主体であるメーカは，組立性の観点から製品を認識する．これに対し，製品をリサイクルする主体であるリサイクル業者は，貴金属を含む部品を取り出すというリサイクル性の観点から製品を認識する．図 3-9 は，主体による製品の認識の違いを，異なる階層構造によって示している．最終的に決定する製品のノミナル情報は 1 つであるため，ライフサイクルにわたって複数存在する階層構造を 1 つのノミナル情報として表現可能とすべきである．

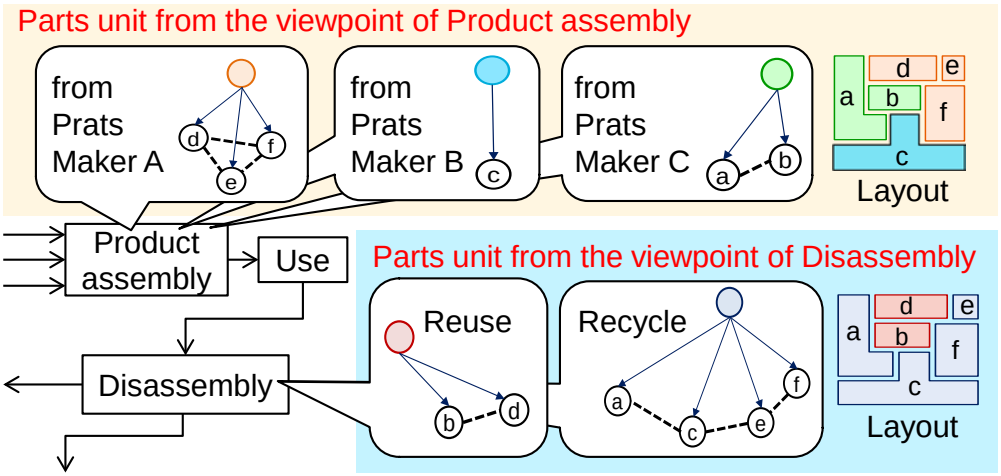


図 3-9：主体による認識の仕方の違い

上記 3 つの側面から製品ライフサイクルを表現可能とするために，國井は製品階層構造モデルとライフサイクルフローモデルの対応関係を定義している [13]．具体的には，製品階層構造モデルの実体ノードとライフサイクルフローモデルのライフサイクルプロセスノードが保持する Input や Output parameter との間に双方向リンク（対応リンクと呼ぶ）を張ることで，「実体ごとのライフサイクル」「ライフサイクルを通じた実体の状態変化」「ライフサイクルにわたる主体による認識の仕方の違い」を表現している．例えば図 3-10 の設計対象モデルは，Disassembly 工程において，LCD-TV が Stand Unit などの部品に分解されることを表現している．

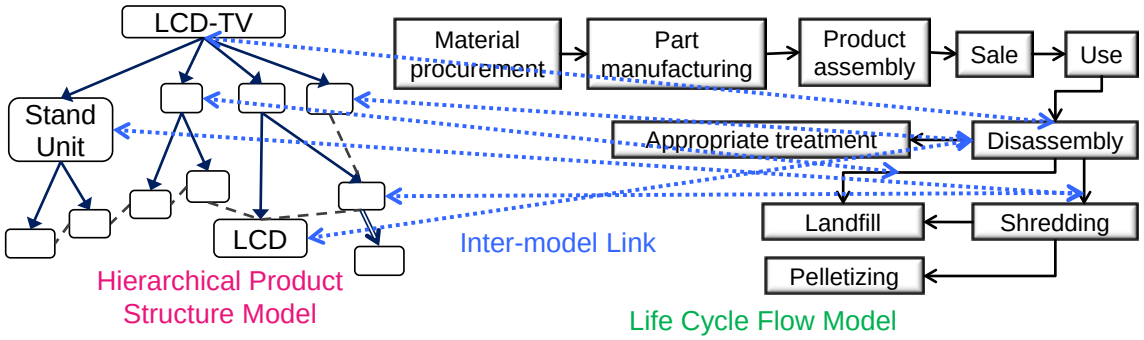


図 3-10：製品階層構造モデルとライフサイクルフローモデルの対応関係

(D) 製品とライフサイクルフローの整合性管理手法

國井は，製品とライフサイクルフローの間の整合性を，「トポロジーレベル」「ジオメトリレベル」「テクノロジレベル」という 3 つのレベルから管理すべきであると指摘している [13]．各レベルの判断基準は以下のとおりである．

- ✧ トポロジーレベル：製品とライフサイクルフローを表す 2 つの設計対象モデル間の対応関係に矛盾がないか否か．
- ✧ ジオメトリレベル：幾何形状や機構といった製品構造が，各ライフサイクルプロセスでの処理能力に適しているかどうか．例えば，破砕機に投入する製品について，その幾何形状が破砕機の投入口の寸法より小さいかどうかが判断基準となる．
- ✧ テクノロジレベル：各ライフサイクルプロセスの条件（例えば，法律や技術レベルにより処理可能な許容範囲）と，処理する実体の属性の制約条件（例えば，顧客要求を満たすために提供すべき機能）が，互いに充足しているか否か．例えば，マテリアルリサイクルを含む製品ライフサイクルを設計する場合を想定する．このとき，マテリアルリサイクルの対象部品を構成する 2 つの素材には相溶性がなく，一方でそれら 2 つの素材を混合したままリサイクルするようにライフサイクルフローのノミナル情報を導出している場合，リサイクル材の強度は低下する [53]．この強度低下が顧客要求を満たす機能の提供に対して許容可能かどうかが判断基準となる．

以上 3 つの整合性管理レベルのうち，國井はトポロジーレベルの整合性を管理する手法を提案している [13]．トポロジーレベルの整合性管理手法は，ライフサイクルプロセスノードに入出力する実体ノードの過不足に着目した手法である．具体的には，ライフサイクルプロセスノードに入出力するフローリンクと各フローリンクに対応付く実体ノードとの関係から，ライフサイクルプロセスノードが，(a)通過，(b)変換，(c)結合，(d)分離，(e)開始，(f)終了，という 6 つを基本タイプとした処理方法を表すものと定義する（図 3-11 参照）．

本研究では，この基本タイプを「ライフサイクルプロセスタイプ」と呼ぶこととする．ライフサイクルフローモデルがいずれのライフサイクルプロセスタイプにも分類されないライフサイクルプロセスノードを含む場合，その製品ライフサイクルのノミナル情報モデルをトポロジーレベルで不整合とみなす.

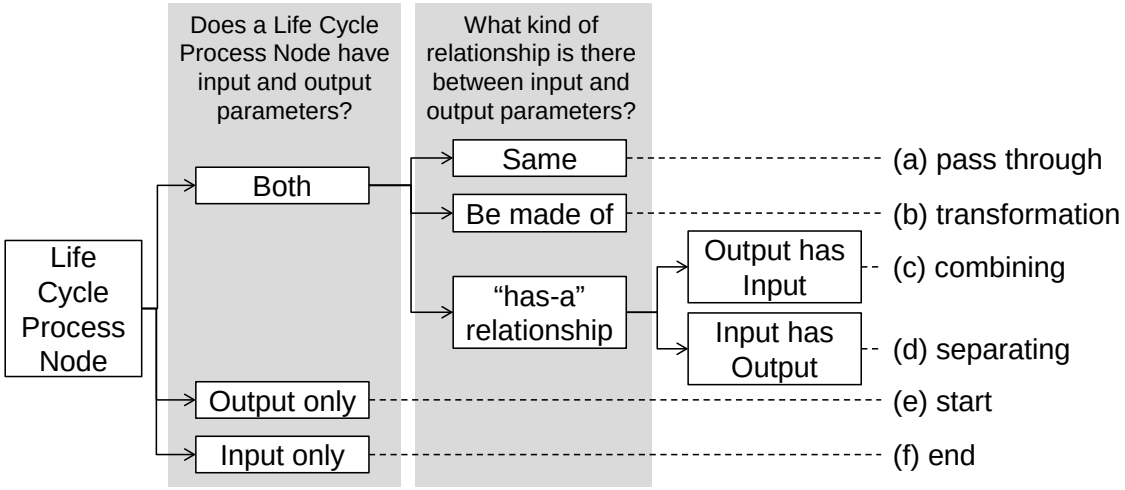


図 3-11：ライフサイクルプロセスタイプへの分類手法（ [13]を基に作成）

3.4.2. Life Cycle Simulation

Life Cycle Simulation (LCS)は，離散事象シミュレーション技術に基づいて，ライフサイクルにわたる製品，部品，素材，金銭，情報の流れをシミュレーションする手法である．LCS は，2.2.3 項で示したようにいくつかの研究で取り組まれている．本項では，[14]を例に，LCS の実行アルゴリズムを示す．

LCS を実行するためのモデル（LCS モデル）[14]は，製品の構造と属性を表す製品モデル，およびライフサイクルプロセスのネットワークを表すライフサイクルモデルという 2 つのサブモデルから成る．

製品モデルは，製品，モジュール，部品，という 3 段階から成る．製品はモジュールの接続グラフで表現され，モジュールは部品の接続グラフで表現される（図 3-12 参照）．部品は，属性（例えば，リサイクル可能率，製造コスト，寿命，重量，構成素材）の集合であり，これ以上分解できない最小単位である．

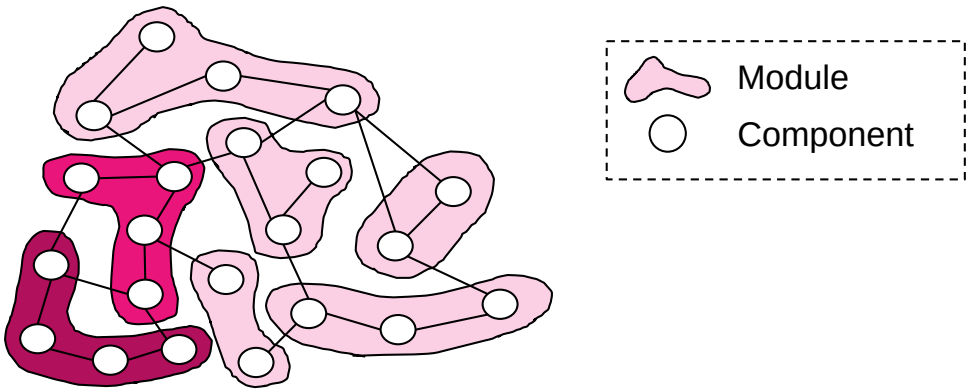


図 3-12：製品モデル

ライフサイクルモデルは，ライフサイクルプロセスを表すライフサイクルプロセスノードと，ライフサイクルプロセス間の実体や情報の流れを表すリンクから成る．ライフサイクルプロセスノードは，定数パラメータ，入力パラメータ，出力パラメータ，プロシージャーという 4 つの要素から成る．各要素について以下で詳述する．

- 定数パラメータ
  - 名前：定数パラメータ独自の名前
  - 値：定数パラメータの値
  - タイプ:定数パラメータ値の扱い方の情報であり, 2 種類のモードがある. 第一に，設計者が値を直接代入するモードである．第二に，データベースから値を参照するモードである．

- 入力パラメータ
  - 名前：入力パラメータ独自の名前
  - 値：入力パラメータの値．下記の参照パラメータの値を継承する．
  - 参照パラメータ：他のライフサイクルプロセスノードの出力パラメータもしくは定数パラメータへの参照を表す．
  - Delay：上流のライフサイクルプロセスノードの出力パラメータの値を何ターン後にパラメータ値に反映するかを決める定数である．すなわち，シミュレーションターンtにおけるパラメータpの値$u(p,t)$を，Delay を用いた式(3.2)で表現する．なお，式(3.2)のipとopは，それぞれ入力パラメータ自身と参照する出力パラメータを意味する．

$$u(ip,t) = u(op, t - Delay) \tag{3.2}$$

- 出力パラメータ
  - 名前：出力パラメータ独自の名前
  - 値：出力パラメータの値．プロシージャーに基づいた計算の結果として決まる．
  - 型:パラメータの型であり, monthly か continuous の 2 種類がある. 型が monthly のパラメータは, シミュレーションターンごとに値を初期化する. 型が continuous のパラメータは，前のシミュレーションターンでの値を保持する．このように型を分類する理由は，製品ライフサイクルには，シミュレーションターンごとの値が必要なパラメータ（例えば，月ごとの売り上げ台数）と前のシミュレーションターンでの値の繰り越しが必要なパラメータ（例えば，倉庫で管理している部品個体）の 2 種類が存在するためである．

- プロシージャー
  - 定数パラメータ，入力パラメータ，出力パラメータ間の関係を，プログラミング言語を用いて記述する．

LCS は，上記の製品モデルとライフサイクルモデルを入力に，シミュレーションを実行する．まず，シミュレーションターンごとに，各ライフサイクルプロセスノードに記述されたプロシージャーに従って，出力パラメータの値更新に関する計算を行う．次に，ライフサイクルプロセス間の依存関係に従って，パラメータ値を伝播する．つまり，入力パラメータの値に，参照パラメータの値を伝播させる．この出力パラメータの値更新の計算とパラメータ値の伝播を，すべてのライフサイクルプロセスノードについて順に行う．この計算サイクルを，シミュレーションターンが，設計者によって設定された最終ターンとなるまで繰り返す．

3.5. 本研究の位置付け

本研究の最終的な目標は，製品の実データに基づいて高い精度で予測した個体情報を用いて，資源が効率的に循環する製品ライフサイクルを設計するための方法論を構築することにある．本研究では，設計者がノミナル情報と個体情報という捉え方で製品ライフサイクルを設計するための道具立てとして，実データの有無に関わらず，仮想的な個体情報を用いて製品ライフサイクルのノミナル情報を決定する過程を支援する手法を提案する．すなわち，3.3 節で挙げた要件に対して本研究で解決する課題とそのアプローチは以下のとおりである．

- **製品ライフサイクルのノミナル情報と個体情報を表現したモデル化手法の構築**

仮想的な個体情報を用いて製品ライフサイクルを設計するための作業空間として，製品ライフサイクルをノミナル情報と個体情報の両面から表現可能としたモデル化手法を提案する．1.1.2 項で述べたように，ノミナル情報が変われば，個体の集合が示す状態の違いや量の変動の有様は変化する．そのため本研究では，現実世界の製品ライフサイクルにおける様々な変動要因によって多様となる個体情報をノミナル情報の結果として表現可能とする．そこで，本モデル化手法を，以下の 3 つの要素で構成する．

- 製品ライフサイクルのノミナル情報を表現したモデル
  要件 1-1 に対し，3.4.1 項で示した製品ライフサイクルのノミナル情報モデル [13] を用いて，製品とライフサイクルフローのノミナル情報を表現する．また，本モデル上で，個体情報の多様さの原因となる変動要因を表現可能とする．

- 個体情報を表現したモデル
  要件 1-2 に対し，ライフサイクルにわたって状態の違いや量の変動を示す個体の集合を表現するためのモデルを定義する．本モデルを「個体情報モデル」と呼ぶ．本研究では，ライフサイクルにわたって各個体が示す個体情報の集合として個体情報モデルを表現する．本表現によって，製品ライフサイクル全体，各ライフサイクルプロセス，各個体，といったライフサイクルにわたる様々な側面や粒度での個体情報の評価を可能とする．

- 製品ライフサイクルのノミナル情報モデルから個体情報モデルの作成手法
  変動要因を表現した製品ライフサイクルのノミナル情報モデルを入力に，3.4.2 項で示した LCS [14]によって製品個体のライフサイクルをシミュレーションすることで，個体情報モデルを作成可能とする．

- **製品ライフサイクルの設計サイクル実行支援手法の構築**

上記のモデル化手法を用いた製品ライフサイクルの設計サイクルにおいて生じる要件 2 に関する課題に対し，以下 2 つの手法を提案する．

- 個体情報の評価の実行支援手法

  要件 2-1 に対し，上記のモデル化手法によって個体情報の評価を実行するためには，ライフサイクルにわたる個体の流れをシミュレーションするための計算処理の記述が必要である．具体的には，①個体に施す処理方法を表す計算処理，②その処理方法の違いや処理によって生じる個体の状態の違いを表す計算処理，③個体に対する処理の過程で生じる環境負荷排出量やコストといった評価値を算出するための計算処理，という 3 種類の計算処理の記述が必要である．本研究ではこのうち，製品ライフサイクルのノミナル情報モデルのトポロジー表現を用いることで計算処理①の記述を支援し，また評価値の算出に関する計算式をパターン化することで計算処理③の記述を支援する手法を提案する．

- ノミナル情報の修正案特定支援手法

  要件 2-2 に対し，上記のモデル化手法によって個体情報の評価を実行した結果が設計要求を満たしていない場合に，製品ライフサイクルのノミナル情報モデルを構成するパラメータ間の関係の分析とパラメータの感度分析によって，修正に適した設計パラメータを特定する．

3.6. 第 3 章のまとめ

本章では，本研究における製品個体の集合に着目した製品ライフサイクルの設計方法論に関する基本的な考えを示した．まず，製品ライフサイクルを「製品とライフサイクルフローのノミナル情報」「仮想的な個体の集合」「現実世界における個体の集合」という 3 つの階層に分類した．次に，この 3 つの階層から製品ライフサイクルを捉えた場合の設計問題について述べ，本設計問題を内包する製品ライフサイクルの設計を実行するうえでの方法論の要件を挙げた．本要件に対する先行研究の取り組みを示し，本研究で取り組むべき課題を設定した．すなわち本研究では，仮想的な個体情報を用いた製品ライフサイクルのノミナル情報の決定過程を支援する手法を提案する．

第4章 製品個体の集合に着目した製品ライフサイクルのモデル化手法

第 4 章では，製品ライフサイクルをノミナル情報と個体情報の両面から表現したモデル化手法を提案する．

4.1. 製品ライフサイクルのモデル化手法のアプローチ

現実世界において状態の違いや量の変動という特徴を示す個体の集合を設計段階で模擬し，その仮想的な個体情報を用いて製品ライフサイクルのノミナル情報を決定するための作業空間として，ノミナル情報と個体情報の両面から製品ライフサイクルを表現したモデル化手法を提案する．

3.4.1 項で示した製品ライフサイクルのノミナル情報モデル [13]は，製品とライフサイクルフローのノミナル情報を対応付けて表現したモデルである．2.2.4 項②で述べたように本モデルは，製品ライフサイクルを対象製品の 1 個体のライフサイクルにより代表させて表現しており，多数存在する製品個体それぞれを表現しているわけではない．また，製品ライフサイクルの変動要因をどのように表現するかの議論はなされていない．

3.4.2 項で示した LCS [14]は，製品ライフサイクルのノミナル情報について，ライフサイクルにわたる個体の流れをシミュレーションする手法である．しかし，従来の LCS では，シミュレーション実行後に個体情報は存在しない．ライフサイクルにわたる様々な側面や粒度で個体情報の評価を可能とするために，シミュレーション後も個体情報が継続的に存在するモデルが必要である．

そこで本研究は，製品ライフサイクルのモデル化手法を，以下 3 つの要素で構成する．

- 製品ライフサイクルのノミナル情報モデル [13]を用いた変動要因の表現手法
- 個体情報の表現手法（個体情報モデル）
- LCS [14]を用いて製品ライフサイクルのノミナル情報モデルから個体情報モデルを作成する手法

4.2. 変動要因の表現手法

本節は，3.4.1 項で示した製品ライフサイクルのノミナル情報モデル [13]を用い，変動要因を表現する手法を提案する．

4.2.1. 変動要因の分類

本項では，製品ライフサイクルの変動要因を分類する．

3.1 節で示したように，現実世界における製品ライフサイクルの様々な変動要因の結果，各ライフサイクルプロセスで処理する個体の集合には状態の違いがあり，またその量が変動する．具体的には，3.1 節(a)で示したように，ライフサイクルプロセスを経た製品個体は，「(a1)個体の物理的な性質」と「(a2)個体に対する処理の違い」に起因して違いを示す．本研究ではこの分類に従い，変動要因を「個体の物理的な性質に依存する場合（製品依存)」と「個体に対する処理に依存する場合（ライフサイクルプロセス依存)」の 2 つに分類する．さらに，個体に対する処理の違いを，3.1 節(a2)(b1)(b2)(b3)から，「人や機械といった各ライフサイクルプロセスで処理を施す主体の能力や要求の違い（3.1 節(a2)(b2)が該当)」，「個体が辿るフローの分岐条件（3.1 節(b1)が該当)」，「処理自体の変化（3.1 節(b3)が該当)」によって生じるものと捉える．以上の分類から，変動要因を図 4-1 の 4 つで定義する．

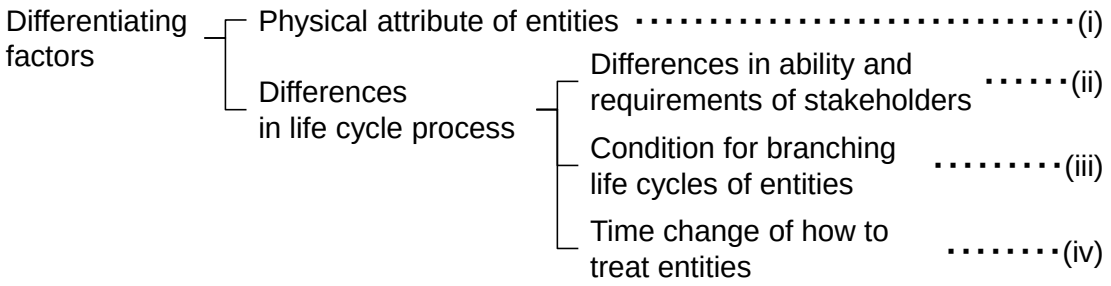


図 4-1：変動要因の分類

各変動要因について，以下で詳述する．

変動要因(i) ：個体に内在する物理的な性質

個体は，同一環境下で同一の処理が施される場合であっても，蓄積した物理的な性質が原因で異なる挙動を示し得る．例えば，同じノミナル情報に基づいて生成された個体を同じ環境下で同じ時期から使用した場合でも，一定時間経過後の個体の状態は「故障」と「正常」という別の状態になり得る．

変動要因(ii) ：主体の能力や要求の違い

人や機械などの主体が施す処理は，その能力や要求によって異なる．主体それぞれが個別の処理を個体に施すことで，各個体の属性値の変化量は個々に異なる．例えば，ユーザによって使用頻度が異なることで，一定時間経過後の個体の劣化進展度合いは異なる．また例えば，3.1 節の図 3-2(a2)で示したような，加工者の技術レベルが異なることで，加工した製品個体の幾何形状に違いが生じること [12]は，本変動要因に該当する．

変動要因(iii) ：個体が辿るフローの分岐条件

同一の主体が処理する場合であっても，個体のフローは，その状態やリユース部品の需要量といった外部からの指示に従って分岐する．例えば同一環境下の同一のユーザであっても，使用している製品個体が故障していない場合は使用し続けるが，故障している場合はその程度に応じてメンテナンスや廃棄という行動を取る．

変動要因(iv) ：処理自体の変化

同一の主体であっても，施す処理の内容は，外部環境に依存して時間変化する．例えば，技術進歩による陳腐化が原因で需要量が減少すれば，その製品の生産台数は減少し得る．また例えば，子供の成長に伴って洗濯物の量が増加すれば，洗濯機の使用頻度は増加し得る．

4.2.2. 製品ライフサイクルのノミナル情報モデルを用いた変動要因の表現手法

4.2.1 項で 4 つに分類した変動要因を，製品ライフサイクルのノミナル情報モデルを用いて表現する．具体的には，個体の物理的な性質を表す変動要因(i)を，製品階層構造モデルを用いて表現する．個体に対する処理の違いを表す変動要因(ii)(iii)(iv)を，製品階層構造モデルとライフサイクルフローモデルの両者を用いて表現する．各変動要因を表現する手法について，以下で詳述する．

- 変動要因(i)
  本研究では，同一環境下で同一の処理を施す個体が集合として示す物理的な性質を，製品属性の 1 つとして扱う．例として，同じ使用環境下で生じる故障の発生確率が挙げられる．3.4.1 項で示したように，製品階層構造モデルの実体ノードは属性A_{en}を保持する．そこで，属性a_αに，個体が集合として示す物理的な性質に関する項目と属性値を設定し，この属性a_αをA_{en}の要素とすることで変動要因(i)を表現する．

- 変動要因(ii)
  主体による能力や要求の違いを，ライフサイクルプロセスノードlcpの$gp_{lcp,i}$を用いて表現する．その結果として個体に生じる属性値の変化量の違いを，製品階層構造モデルを構成する実体ノードの属性a_αを用いて表現する．そのために実体ノードの属性値を，初期値a_α^0と変化率$\Delta_\alpha(X)$で構成する．変化率$\Delta_\alpha(X)$は，変数の集合Xに対する属性値の変化割合を表す多変数関数である．主体の能力や要求の違いを表す$gp_{lcp,i}$を変化率$\Delta_\alpha(X)$の変数とし，個体に生じる属性値の変化量の違いを式(4.1)のようにPR_{lcp}に記述する．つまり，式(4.1)では，$gp_{lcp,i} \in X$である．

$$a_\alpha = a_\alpha^0 + \Delta_\alpha(X) \tag{4.1}$$

  例えば，ノートパソコンのバッテリー容量が，経過時間とユーザの使用頻度に依存した変化率$\Delta_c(X)$で変化する場合，時間t_1経過後のバッテリー容量c[mAh]を，使用工程を表す Use プロセスノードの Given parameter GP_{use}の要素でありユーザの使用頻度を表すufと，バッテリーを表す実体ノードの属性の要素でありバッテリー容量の初期値を表すc_0[mAh]を用いて，式(4.2)のようにPR_{use}に記述する．

$$c = c_0 + \Delta_c(t_1, uf) \tag{4.2}$$

- 変動要因(iii)
  分岐の閾値を，$gp_{lcp,i}$か$ip_{lcp,j}$を用いて表現する．この閾値に基づく個体の分岐の条件式をPR_{lcp}に記述する．例えば，リユース部品の保管工程を表すReusePartsWarehouseプロセスノードにおいて，組立工程からのリユース部品需要量を表すInput parameterを閾値とし，その閾値に基づいてReusePartsWarehouseプロセスノードに存在するリユース部品個体を組立工程に出庫するか否か判断する条件式を，$PR_{\text{ReusePartsWarehouse}}$に記述する．

- 変動要因(iv)
  変動要因(ii)の属性の変化率$\Delta_\alpha(X)$の変数であり主体の能力や要求の違いを表す$gp_{lcp,i}$，および変動要因(iii)の閾値で用いる$gp_{lcp,i}$や条件式を，時間関数とすることで処理自体の時間変化を表現する．例えば，ノートパソコンのバッテリー製造工程を表すBattery Manufプロセスノードにおいて，生産台数を表すGiven parameter $gp_{BatteryManuf,i}$を時間関数として設定し，その$gp_{BatteryManuf,i}$分のバッテリー個体を生産することを$PR_{BatteryManuf}$に記述する．

4.3. 個体情報モデル

ライフサイクルにわたって状態の違いや量の変動を示す個体の集合を表し，製品ライフサイクル全体，各ライフサイクルプロセス，各個体，といった様々な側面や粒度から個体情報の評価を可能とするモデルとして，個体情報モデルを規定する．

4.3.1. 個体情報の表現手法

個体情報モデルを，個体情報eiを要素したモデル EIM として規定する．

$$EIM = \{ei\} \tag{4.3}$$

EIM の要素である個体情報eiを，式(4.4)の 6 つの要素で規定する．

$$ei = (eid, ni, t, lcp, s, assy) \tag{4.4}$$

eidは，個体の識別子を表す．

niは，個体のノミナル情報を表し，実体ノードの識別子を値として保持する．

tは，その個体が存在する時刻を表す．ここでの時刻とは，対象とする製品ライフサイクルで，すべての個体が共通とする時間を表す．この個体共通の時間を，ライフサイクルタイムと呼ぶ．

lcpは, ライフサイクルタイム tにおいて個体が位置するライフサイクルプロセスを表し，ライフサイクルプロセスノードの識別子を値として保持する．

sは, ライフサイクルタイム tに個体が示す状態を表す. この状態sを, 属性a_αの集合とし，式(4.5)のように表現する．

$$s = (a_1, a_2, \dots, a_n) \tag{4.5}$$

このとき，個体によって属性の項目は異なる．例えば，スマートフォンのディスプレイ個体は解像度や画面サイズを属性の項目に含み，一方でバッテリー個体はバッテリー容量を項目に含み得る．そのため状態sを構成する属性a_αの項目は，各個体が参照する実体ノードに定義された属性a_αに従うものとする．

$assy$は，ライフサイクルタイム tにおいて各個体を構成する個体ei_β（βは自然数）の集合を表す．

4.3.2. 個体情報モデルを用いた製品ライフサイクルの表現

評価する目的に応じて個体情報モデルから式(4.4)の個体情報*ei*を抽出することで，以下 5 つの側面から製品ライフサイクルを表現する．まず，各ライフサイクルプロセスで処理する個体の(1)量の変動と(2)状態の違い，という製品ライフサイクルがもつ 2 つの特徴を表現する手法について示す．また分解性設計やリサイクル性設計などの DfX 手法を適用するための製品ライフサイクルの対象表現として，(3)部品の劣化，(4)部品交換の履歴，(5)リユース部品と新品部品が共存する製品，という 3 つの側面を表現することが課題になる［116］．個体情報モデルを用いることで，この 3 つの側面が表現可能であることを示す．

(1) ライフサイクルプロセスで処理する個体の量の変動

各ライフサイクルプロセスで処理する個体の量の変動は，個体情報モデルから*ni*と*lcp*の値を同じくする個体情報*ei*を抽出し，ライフサイクルタイム*t*ごとに個体数を集計することで表現する．集計した結果から，ライフサイクルプロセスノード*lcp*を辿り，かつ同一実体*ni*を参照する個体数の時間変化をグラフとして表現する．例えば図 4-2 では，Manufacturing プロセスと Collection プロセスを辿った液晶テレビ（Liquid Crystal Display-Television, LCD–TV）個体の時間変化を，それぞれ異なるグラフとして表現している．

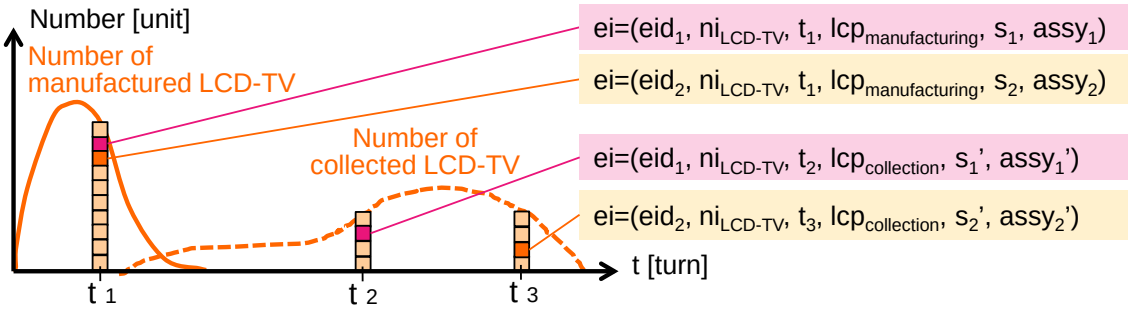


図 4-2：各ライフサイクルプロセスで処理する個体数の時間変化（対象例：液晶テレビ）

(2) ライフサイクルプロセスで処理する個体の状態の違い

各ライフサイクルプロセスで処理する個体の状態の違いを，以下 3 つの側面から表現する．

(2-1) 属性値を同じくする個体数の分布

各ライフサイクルプロセスで処理する個体の状態の違いを，ある属性に関する属性値の分布として表現する．まず個体情報モデルから，niとlcpを同じくする個体情報eiを抽出する．その個体数を「属性a_αの値」を同じくする集合ごとに集計して，ライフサイクルタイム t ごとに累積する．例えば図 4-3 は，回収された LCD-TV 個体の輝度を横軸にとることで，ライフサイクルタイム全体での輝度の値の台数分布を表現している．

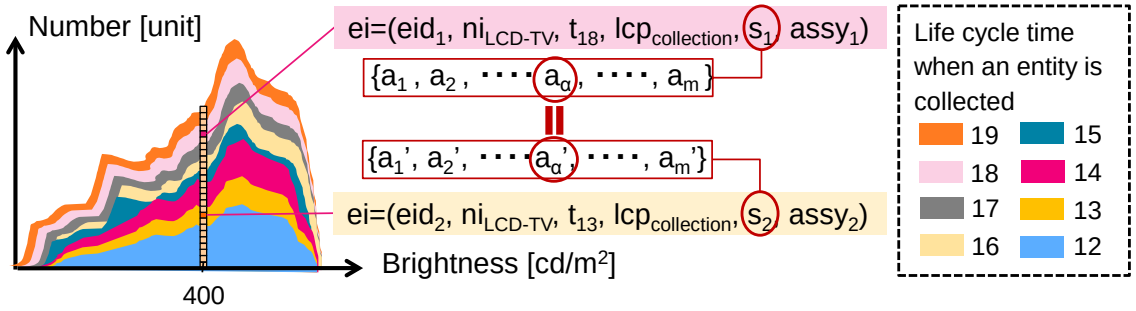


図 4-3：属性値を同じくする個体数の分布（対象例：液晶テレビ）

本表現によって，例えば図 4-4 のように，検査を経てリユース可能とリユース不可とにみなされたLCD-TV個体の集合それぞれの属性値の分布を表すことができる．図 4-4 の 1 から 4 は，各属性値の輝度とスピーカ音質をもつLCD–TV個体が各ライフサイクルタイムに何個体ずつ存在するかを，個体情報モデルを用いて表現した例である．この属性値に関する個体数の分布表現によって，リユース可能なLCD-TV個体数の増加のために着目すべき属性を明示的に示唆できる．例えば図 4-4 のように，4 よりも 3 の集合に属するLCD-TV個体数が多い場合は，3 の属性である輝度が検査基準の 400[cd/m²]以下となる個体を減らすようにLCD-TVのノミナル情報を変更することが，リユース可能な個体数を増加させる有効なアプローチと推定できる．

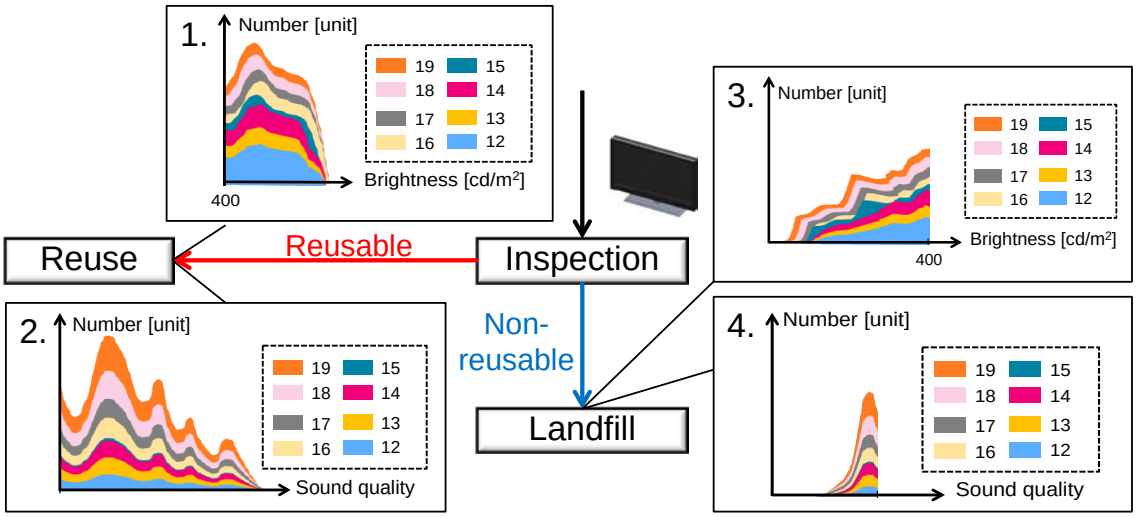


図 4-4：個体の状態の違いの表現によるノミナル情報の決定支援の一例

(2-2) 各個体の状態を階層的に表現

個体の状態の違いを，各個体情報eiの比較によって表現する．具体的には，個体情報eiの要素であるniとsと$assy$を用い，比較対象の個体が参照している実体ノードとその個体を構成する個体$assy$が参照している実体ノードによって，各個体をそれぞれ階層的に表現する．例えば図 4-5 のように，(a)LCD Unit 個体（eid = hu58ha7b-f68o-c3gq）と(b)LCD Unit 個体（eid= da469db5-a44c-49e9）が，同一のライフサイクルタイムt=10[year]に同一のライフサイクルプロセス（Repair プロセス）において，輝度の値や残余寿命が異なることを階層的に表す．

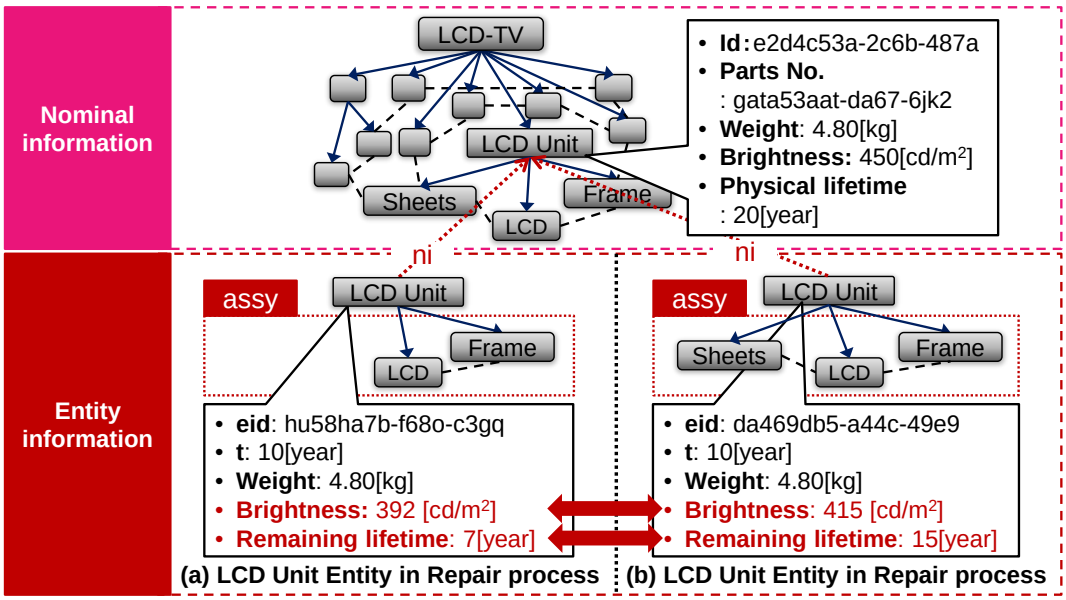


図 4-5：個々の個体の状態の比較表現（対象例：液晶テレビ）

(2-3) 各個体の状態をソリッドモデルとして表現

個体情報eiが保持する実体ノードへの参照niと実体ノードの一部であるソリッドモデル（3.4.1 項(A)参照）を用い，各個体の属性値a_αをソリッドモデルに反映することで，各個体の幾何形状を表現する．

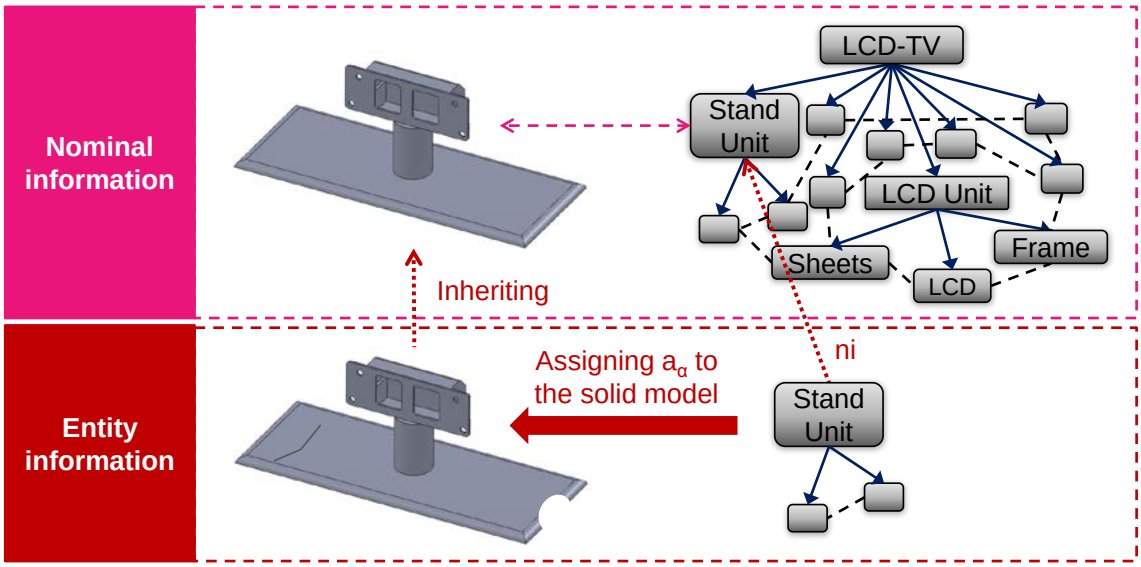


図 4-6：各個体の幾何形状のソリッドモデルとしての表現

(3) 部品の劣化

部品の劣化は，(2-2)と同様のアプローチで表現する．具体的には，識別子eidが同一であり，ライフサイクルタイムtが異なる個体情報eiについて，niとsと$assy$を用いて比較表現することにより，個体が劣化していることを示す．例えば図 4-7 では，ライフサイクルタイムが(a) t=1[year]，(b) t=10[year]，(c) t=19[year]と変化するに従って，同一のeidをもつ LCD Unit 個体の輝度と残余寿命が減少していることを表している．なお，図 4-7 の(a)(b)(c)は，それぞれ異なる LCD Unit のeiを表す．

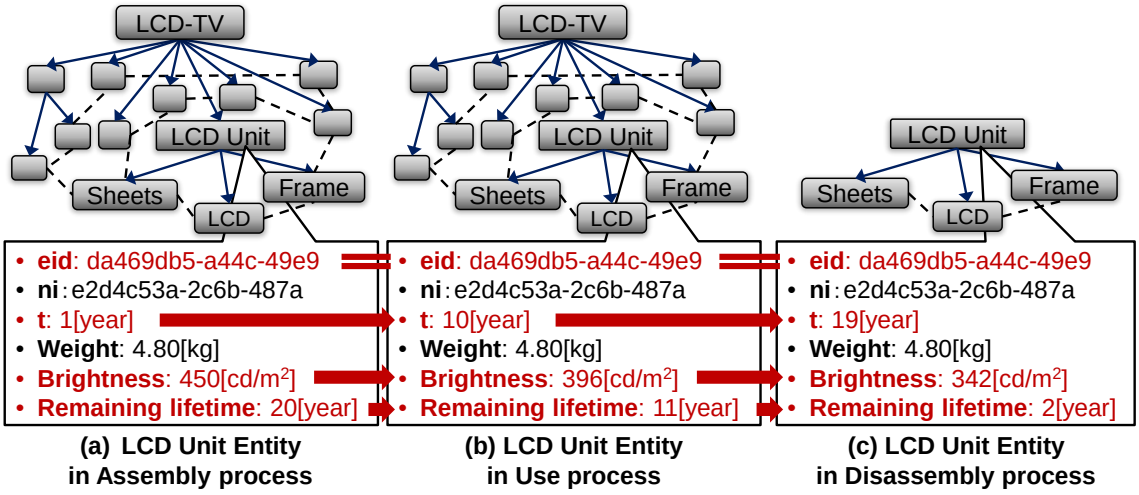


図 4-7：部品個体の劣化（対象例：液晶テレビ）

(4) 部品交換の履歴

部品交換の履歴は，交換対象の個体を構成部品とする個体を表す 2 つの個体情報eiについて，その$assy$を構成する部品個体の識別子eidが変化したことによって表す．例えば図 4-8 は，LCD-TV 個体(eid=gbc967bz-9f4c-dg3s)について，ライフサイクルタイムt =10[year]に，LCD Unit 個体(eid=da469db5-a44c-49e9)から LCD Unit 個体(eid=hu58ha7b-f68o-c3gq)への部品交換が行われたことを表している．なお，図 4-8 の(a)(b)は，それぞれ異なる LCD-TV のeiを表す．

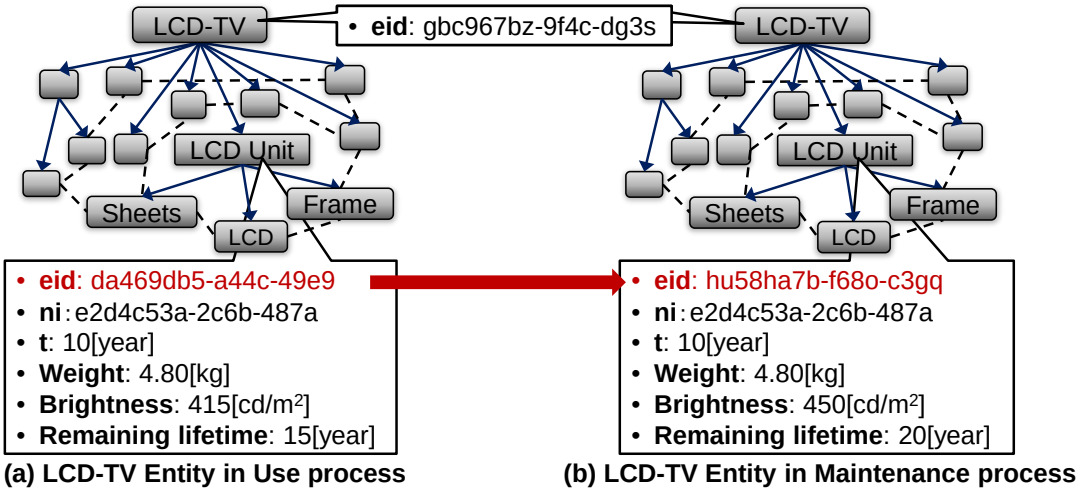


図 4-8：部品交換の履歴（対象例：液晶テレビ）

(5) リユース部品と新品部品が共存する製品

設計者は製品の構成部品について，リユース部品か新品部品のいずれかを許容するようにノミナル情報を決める場合がある．このとき，市場に流通する製品個体には，「リユース部品個体を組み込んだ製品個体」と「新品部品個体を組み込んだ製品個体」が共存し得る．市場におけるリユース部品と新品部品の共存を，個体情報モデルを用いて表現する．

製品 P を構成する部品 p1 が，リユース部品か新品部品のいずれかを許容するようにノミナル情報が決められている場合を想定する．部品 p1 を参照する個体 x が，そのライフサイクル履歴から，製品 P を参照する個体 X に含まれており，かつそのライフサイクルタイム以前に異なるeidの個体 Y の$assy$に複数のライフサイクルタイムにわたって含まれている場合，個体 x はリユース部品であることを表す．例えば図 4-9 では，LCD Unit 個体(eid=da469db5-a44c-49e9)が，ライフサイクルタイムt =20[year]においては LCD-TV 個体(eid=dua5l9ha-a9ot-qap5)に含まれている（図 4-9(c)参照）ものの，ライフサイクルタイムt =10[year]やt =15[year]においては LCD-TV 個体(eid=gbc967bz-9f4c-dg3s)に含まれている（図 4-9(a)(b)参照）ことによって，図 4-9(c)の LCD Unit 個体がリユース部品であることを表している．また図 4-9 では，ライフサイクルタイムt =20[year]に，(c)リユース部品である LCD Unit 個体 x(eid=da469db5-a44c-49e9)と(d)新品部品である LCD Unit 個体 x'(eid=hu58ha7b-f68o-c3gq)が共存していることを表している．

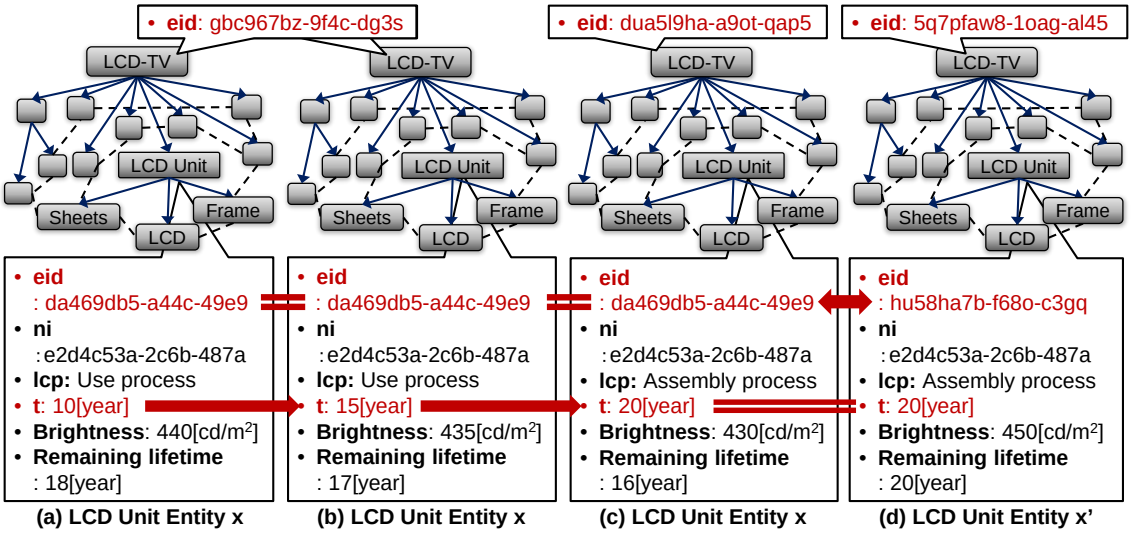


図 4-9：リユース部品と新品部品が共存する製品（対象例：液晶テレビ）

4.4. 製品ライフサイクルのノミナル情報モデルから個体情報モデルを作成する手法

本節では，4.2 節の変動要因を表現した製品ライフサイクルのノミナル情報モデルを入力に Life Cycle Simulation (LCS) [14]を実行することで，4.3 節で規定した個体情報モデル EIM を作成する手法を提案する．式(4.4)で規定した個体情報eiを要素として EIM を作成するために，本手法は以下の手順を取る（図 4-10 参照）．

STEP1：製品ライフサイクルのノミナル情報モデルの作成と変動要因の設定

設計者は製品ライフサイクルのノミナル情報を，製品階層構造モデルとライフサイクルフローモデルを用いてモデル作成する．またそのモデルに，変動要因を設定する．

STEP2：対象期間と時間ステップの設定

設計者は，作成する個体情報モデルの対象期間（開始時期t_sと終了時期t_e）と時間ステップΔt（シミュレーションの刻み）を設定する．

STEP3：個体情報eiの生成と個体情報モデルへの格納

各個体がライフサイクルタイムtに各ライフサイクルプロセスノードで示す個体情報を，eiとして生成する．生成した個体情報eiを，個体情報モデルへ格納する．

STEP4：ライフサイクルタイムの更新

ライフサイクルタイム tを，$t+\Delta t$とする．

以上の STEP3 から STEP4 までを，$t=t_e$となるまで繰り返す．

なお，各個体情報は，以下に従って生成する．ある個体 e が，ライフサイクルプロセスノード$lcpm$のPR_{lcpm}に従って，実体ノードni_eに基づき，識別子eid_eとして生成される場合を想定する．この個体 e が，ライフサイクルタイムtにおいて，ライフサイクルプロセスノード$lcpm'$に位置し，状態をs_e，構成部品個体の情報を$assy_e$とするとき，個体 e の個体情報eiを式(4.6)のように生成する．

$$ei=(eid_e, ni_e, t, lcpm', s_e, assy_e) \tag{4.6}$$

この個体 e が，時間Δt後に，ライフサイクルプロセスノード$lcpm''$において，$PR_{lcpm''}$に従って処理された結果，状態がs_e'，構成部品個体の情報が$assy_e'$に変化した場合，式(4.6)の個体情報を更新した式(4.7)の個体情報ei'を新たに生成する．

$$ei'=(eid_e, ni_e, t+\Delta t, lcpm'', s_e', assy_e') \tag{4.7}$$

式(4.6)の個体情報eiと式(4.7)の個体情報ei'それぞれを，個体情報モデルの要素とする．

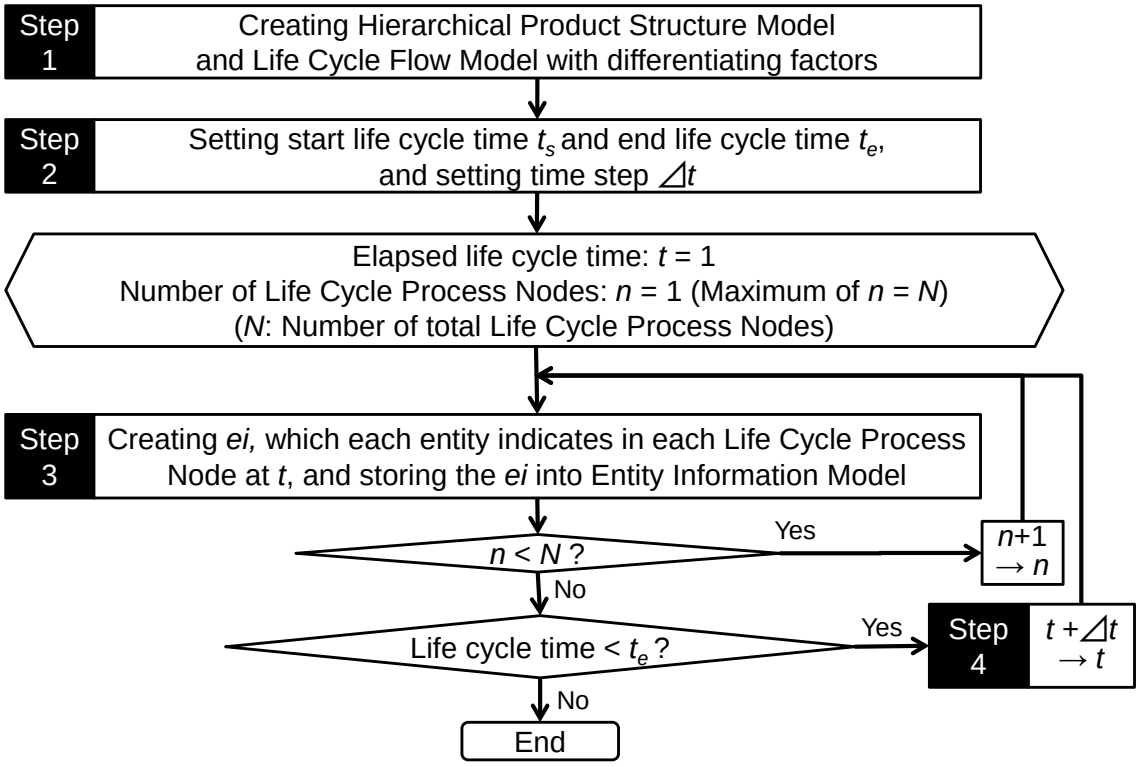


図 4-10：LCS を用いた個体情報モデルの作成手順

4.5. 第 4 章のまとめ

本章では，製品ライフサイクルをノミナル情報と個体情報の両面から表現するモデル化手法を提案した．本モデル化手法を以下 3 つの要素で構成した．第一に，製品とライフサイクルフローのノミナル情報を表す製品ライフサイクルのノミナル情報モデル [13]を用い，個体情報に違いを与える要因（変動要因）を表現するものとした．本研究では変動要因を，(i)個体に内在する物理的な性質，(ii)主体の能力や要求による処理の違い，(iii)個体が辿るフローの分岐条件，(iv)処理自体の変化，の 4 つに分類した．このうち変動要因(i)を製品階層構造モデルの実体ノードの属性を用いて，また変動要因(ii)(iii)(iv)を製品階層構造モデルとライフサイクルフローモデルの対応関係を用いて表現する手法を示した．第二に，ライフサイクルにわたって各個体が示す個体情報の集合を表現したモデルとして，個体情報モデルを規定した．個体情報モデルを用いて，各ライフサイクルプロセスで処理する個体の集合が示す状態の違いと量の変動，および各個体の状態やライフサイクル履歴を表現する手法を示した．第三に，変動要因を表現した製品ライフサイクルのノミナル情報モデルを入力に，離散事象シミュレーションである LCS を実行することで，個体情報モデルを作成する手法を示した．

第5章 製品個体の集合に着目した製品ライフサイクルの設計サイクル実行支援手法

第 5 章では，第 4 章で提案したモデル化手法を用いての製品ライフサイクルの設計サイクルにおける「個体情報の評価」と「ノミナル情報の修正」という 2 つの設計過程を実行するための支援手法を提案する．

5.1. 個体情報の評価の実行支援手法

5.1.1. 個体情報の評価の実行における問題点

第 4 章で提案したモデル化手法を用いて個体情報の評価を実行するためには，LCS の実行過程において各ライフサイクルプロセスノードlcpの振る舞いを決める Procedure PR_{lcp} の記述によって，個体のライフサイクルやその違いの原因となる変動要因を表現しなければならない．しかし， 3.3 節の要件 2-1 で述べたように，製品ライフサイクルのノミナル情報を構成するパラメータは膨大であり，かつパラメータ間の関係は複雑である．特に，PR_{lcp}を記述するうえでは，「設計者の手間と労力」と「表現内容の不一致」という 2 つの問題が生じる．

- **設計者の手間と労力**

第 4 章で提案したモデル化手法では，LCS の実行過程において，各ライフサイクルプロセスノードlcpの振る舞いを決定するPR_{lcp}に従い，lcpで各個体が示す個体情報eiを生成する．このPR_{lcp}は，計算機が理解できる言語，つまりプログラミング言語を用いて記述する必要がある [14]．プログラミング言語を用いた Procedure の記述は，設計者にとって手間と労力がかかるものである．図 5-1 は，液晶テレビの使用工程を表す Use プロセスの Procedure の一部を，一例として示している．図 5-1 の記述例から想定できるように，プログラミング言語の知識がない設計者にとって，Procedure の記述は困難である．

```
//使用し続けるEntityの追加
[[keepUsingLCDTV]] .AddRange([[LCDTV]]);

//故障率に基づいて廃棄するEntityを決定
[[disposedLCDTV]] =LCSFunction.DisposeBasedOnFailureRateOfProductsAndComponents([[keepUsingLCDTV]]);

//廃棄に回るEntityを，使用し続けるEntityのListから削除
foreach(EntityData entity in [[disposedLCDTV]])
{
   [[keepUsingLCDTV]].Remove(entity);
}
[[disposedAmount]]= [[disposedLCDTV]].Count;

//使用し続けるEntityの残余寿命を1[turn]減らす
[[keepUsingLCDTV]]= LCSFunction.UseOneTurn([[keepUsingLCDTV]]);

//使用によるCO2排出量とユーザコストの計算
foreach(EntityData entity in [[keepUsingLCDTV]])
{
    [[UseCO2AsDefaultParameter1]]
    += entity.IndividualIndicationofUsage * entity.IndividualPerformanceList["PowerConsumption"].Invariable * [[electricityUseCO2]];
    [[UseCostAsDefaultParameter1]]
    += entity.IndividualIndicationofUsage * entity.IndividualPerformanceList["PowerConsumption"].Invariable * [[electricityPrice]];
}

//使用段階に存在する個体数から，新規製造台数を算出する
if([[keepUsingLCDTV]].Count  < 2400 && LCSimulator.CurrentTime < 24)
{
    [[newProductionAmount]] = 150;
}
else
```

図 5-1：Procedure の記述例（対象：液晶テレビの Use プロセスの Procedure の一部）

- **表現内容の不一致**

第 4 章で提案したモデル化手法では，製品階層構造モデルとライフサイクルフローモデルの対応関係によって，各ライフサイクルプロセスノードlcpの振る舞いを表す．つまり，製品ライフサイクルのノミナル情報モデルを作成する過程で，設計者が各ライフサイクルプロセスノードに入出力するフローリンクと実体ノードとの対応関係を定義することによって，ライフサイクルプロセスの振る舞いをトポロジー表現している [13]．また，上述したように，PR_{lcp}はライフサイクルプロセスの振る舞いを表す．つまり，製品ライフサイクルのノミナル情報モデルは，トポロジー表現とPR_{lcp}という 2 つの形式によって，ライフサイクルプロセスの振る舞いを表現する．この 2 つの形式による表現の間に違いが生じ得る．その例を，図 5-2 に示す．図 5-2(a)は，製品階層構造モデルとライフサイクルフローモデルの対応関係により，Disassembly プロセスノードでは入力する製品を「Module A」および「Module B の構成部品である Part Ba と Part Bb」まで分解することを表現している．一方，図 5-2(b)の$PR_{disassembly}$では，入力する製品から Module A だけ取り出すことを表現している．図 5-2 の例のように，製品ライフサイクルのノミナル情報モデルでは， 2 つの形式による表現の間に不一致が生じ得る．しかし，2 つの形式による表現の間の一致関係を把握しながら設計過程を進めることは困難である．

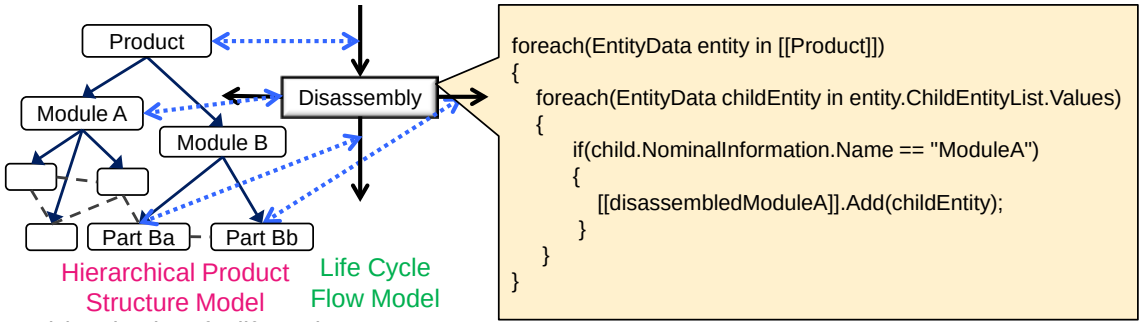


図 5-2：2 つの形式によるライフサイクルプロセスの振る舞いの表現の不一致の例

本節では，以上に示した個体情報の評価の実行過程における「設計者の手間と労力」と「表現内容の不一致」という 2 つの問題解決を目的とした支援手法を提案する．

5.1.2. 個体情報の評価の実行支援のアプローチ

PR_{lcp}は，3.4.1 項(B)で述べたように，GP_{lcp}，IP_{lcp}，OP_{lcp}間の関係を定義し，OP_{lcp}を構成する$op_{lcp,k}$の値を変化させる役割を担う．PR_{lcp}によって値が変化する$op_{lcp,k}$は，個体の集合か評価値を表現している．そこで本研究は，PR_{lcp}の計算処理を，以下の 2 つに分類する．

(1) 個体の集合に対する処理方法

個体の集合に対して，生成（例えば，部品製造），状態の変化（例えば，組立，使用，分解，破砕），流れ（例えば，輸送，選別）などの処理を施して出力する方法を表す．

(2) 処理に伴う評価値の算出方法

(1)の処理に伴って生じる環境負荷排出量，資源消費量，コストなどに関しての評価値を算出する方法を表す．

上記の計算処理の分類に基づいて，以下 2 つのアプローチにより，PR_{lcp}の記述における「設計者の手間と労力」と「表現内容の不一致」の問題を解消する．

第一に，個体の集合に対する処理方法を表す計算処理を，製品階層構造モデルとライフサイクルフローモデルの対応関係によるトポロジー表現に従って生成する．具体的には，各ライフサイクルプロセスノードが表す個体の集合に対しての処理方法を，2 つのモデル間の対応関係によって表現可能なライフサイクルプロセスタイプ（3.4.1 項(D)参照）に分類する．分類したライフサイクルプロセスノードlcpの計算処理を，ライフサイクルプロセスタイプそれぞれの Procedure テンプレートを基にして生成する．

第二に，評価値の算出方法を表す計算処理を，個体の集合に対する処理の過程で生じる環境負荷排出量やコストといった評価値に関するパラメータ，およびその算出に必要なパラメータの間の関係を設計者が設定することで生成する．

5.1.3. ライフサイクルプロセスタイプ

各ライフサイクルプロセスノードが表す個体の集合に対する処理方法は，製品階層構造モデルの実体ノードとライフサイクルフローモデルのフローリンクとの対応関係によって表現可能な基本タイプから成るものと定義する．3.4.1 項(D)で示したように，この基本タイプをライフサイクルプロセスと呼び，これを「通過」「変換」「結合」「分離」「開始」「終了」の 6 つで構成する［13］．しかし本分類には，選別などによって各個体が異なる経路を辿る「分岐」や，新品部品個体とリユース部品個体のいずれかが組立部品となる「合流」といった，個体情報が異なる個体に対しての処理方法であり，かつトポロジー表現可能な処理方法が含まれていない．そこで本研究では，ライフサイクルプロセスタイプを「(a)通過」「(b)変換」「(c)組立」「(d)分離」「(e)合流」「(f)分岐」「(g)開始」「(h)終了」の 8 つに分類する（図 5-3 参照）．

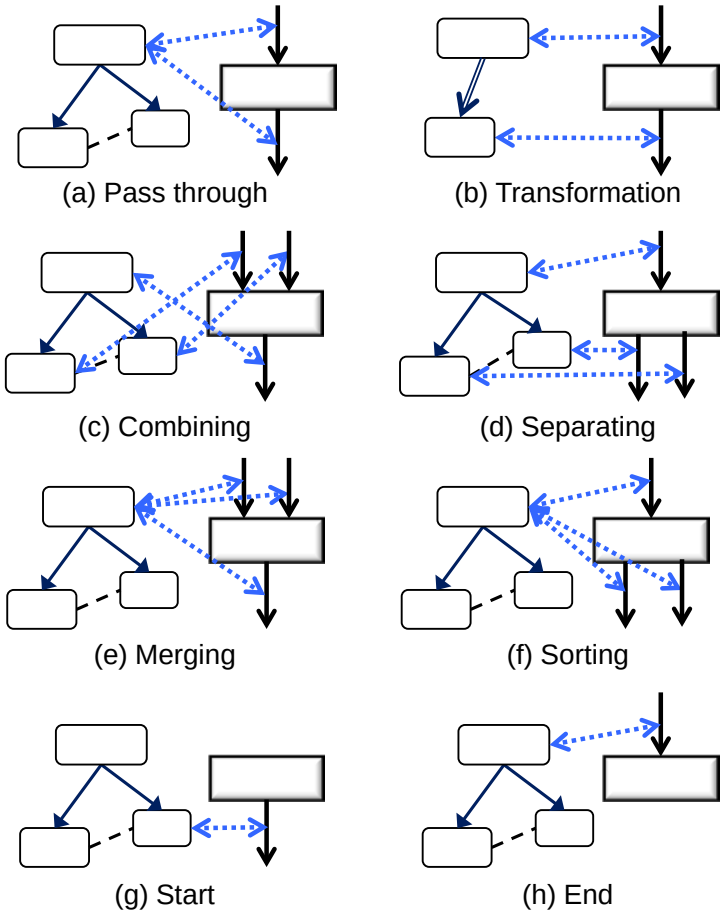


図 5-3：製品階層構造モデルとライフサイクルフローモデルの対応関係に基づくライフサイクルプロセスタイプへの分類

各ライフサイクルプロセスタイプを, 以下の(a)から(h)で定義する. なお, FL_{lcp}^{in}とFL_{lcp}^{out}は, それぞれライフサイクルプロセスノードlcpに入出力するフローリンクの集合を表し，個体の集合の流れに関するフローリンクのみを要素とする．また，同一のライフサイクルプロセスノードの間を流れる個体の集合が複数種類存在する場合はフローリンクを 1 本のリンクで表現し，異なるライフサイクルプロセスノードへ入出力する場合は複数のリンクで表現する．なお，$n(FL_{lcp}^{in})$と$n(FL_{lcp}^{out})$は，それぞれライフサイクルプロセスノードlcpに入出力するフローリンクの数を表す.

(a) 通過：FL_{lcp}^{in}とFL_{lcp}^{out}を構成するフローリンクに対応関係をもつ実体ノードが同一，かつ $n(FL_{lcp}^{in}) = n(FL_{lcp}^{out}) = 1$

(b) 変換：FL_{lcp}^{in}とFL_{lcp}^{out}を構成するフローリンクに対応関係をもつ実体ノードが異なり，かつその実体ノードが変換リンクで関係づいている．かつ，$n(FL_{lcp}^{in}) = n(FL_{lcp}^{out}) = 1$

(c) 結合：FL_{lcp}^{in}を構成するフローリンクに対応関係をもつ実体ノードが，FL_{lcp}^{out}を構成するフローリンクに対応関係をもつ実体ノードの下位階層にあり，かつ両者の最下層にある実体ノードの集合が互いに等しい．かつ，$n(FL_{lcp}^{in}) > n(FL_{lcp}^{out})$

(d) 分離：FL_{lcp}^{out}を構成するフローリンクに対応関係をもつ実体ノードが，FL_{lcp}^{in}を構成するフローリンクに対応関係をもつ実体ノードの下位階層にあり，かつ両者の最下層にある実体ノードの集合が互いに等しい．かつ，$n(FL_{lcp}^{in}) < n(FL_{lcp}^{out})$

(e) 合流：FL_{lcp}^{in}とFL_{lcp}^{out}を構成するフローリンクに対応関係をもつ実体ノードがすべて同一, かつ$(n(FL_{lcp}^{in}) \geq 2) \cap (n(FL_{lcp}^{out}) = 1)$

(f) 分岐：FL_{lcp}^{in}とFL_{lcp}^{out}を構成するフローリンクに対応関係をもつ実体ノードがすべて同一, かつ$(n(FL_{lcp}^{in}) = 1) \cap (n(FL_{lcp}^{out}) \geq 2)$

(g) 開始：$(n(FL_{lcp}^{in}) = 0) \cap (n(FL_{lcp}^{out}) \geq 1)$

(h) 終了：$(n(FL_{lcp}^{in}) \geq 1) \cap (n(FL_{lcp}^{out}) = 0)$

5.1.4. Procedure の生成手順

ライフサイクルプロセスノードlcpのPR_{lcp}を生成する手順を，下記に示す（図 5-4 参照）．

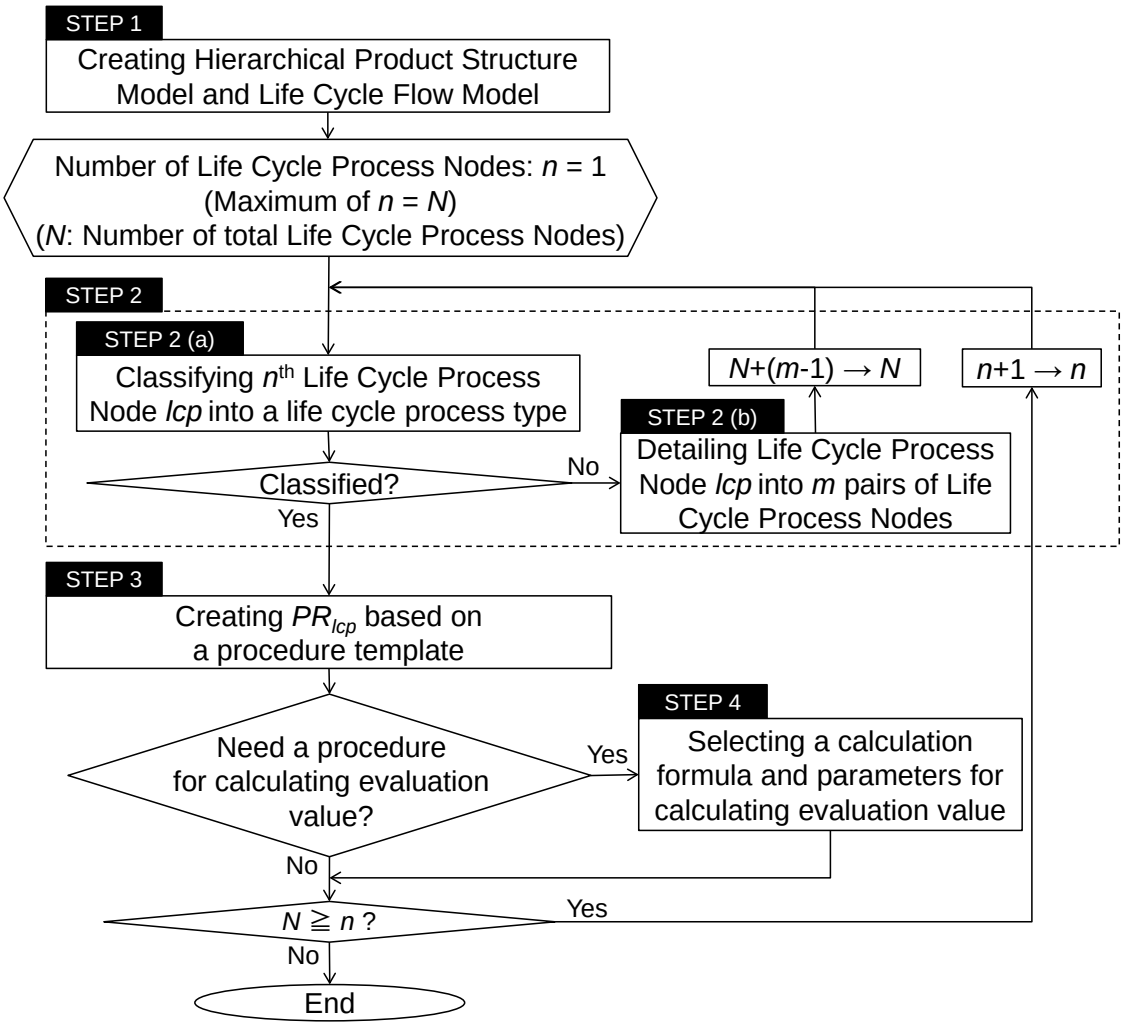


図 5-4：Procedure の生成手順

STEP1：製品ライフサイクルのノミナル情報モデルの作成

製品階層構造モデルとライフサイクルフローモデルを用いて，製品ライフサイクルのノミナル情報をモデル作成する．

STEP2(a)：ライフサイクルプロセスノードのライフサイクルプロセスタイプへの分類

5.1.3 項に示した分類方法に基づいて，ライフサイクルフローモデルを構成するライフサイクルプロセスノードを，ライフサイクルプロセスタイプに分類する．例えば図 5-5 は，フィーチャーフォンのライフサイクルシナリオ [117]に基づいて作成した製品ライフサイクルのノミナル情報モデルを表す．図 5-5 のライフサイクルフローモデルでは，14 個のライフサイクルプロセスノードのうち 9 個を，ライフサイクルプロセスタイプに分類できる．ただし図 5-5 では，製品階層構造モデルとライフサイクルフローモデルの対応関係を，一部のみ示している．

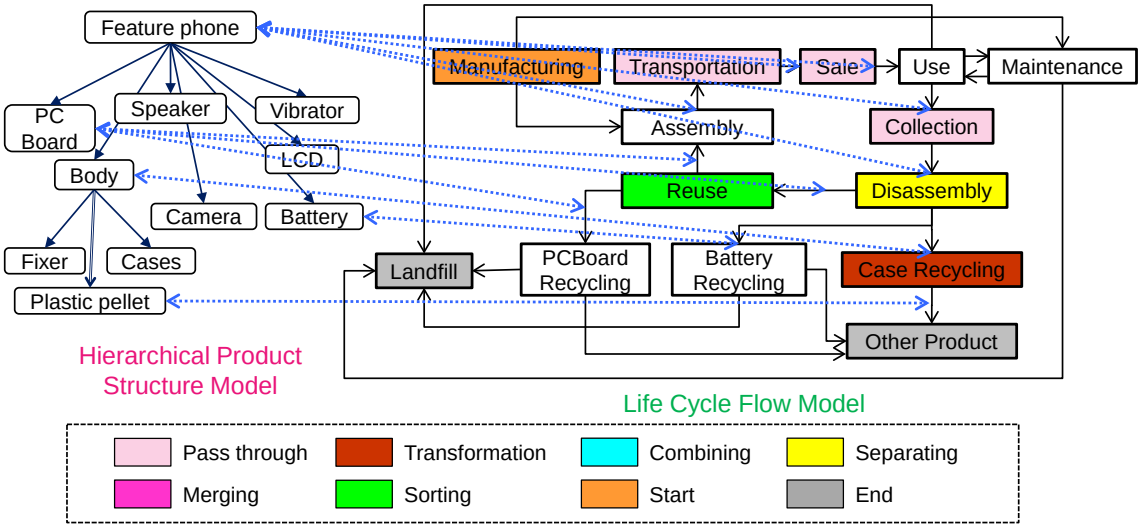


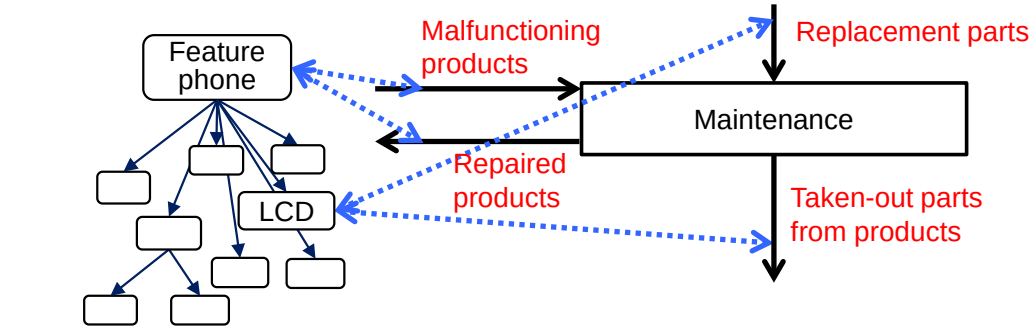
図 5-5：ライフサイクルプロセスタイプへの分類例（対象例：フィーチャーフォン）

STEP2(b)：ライフサイクルプロセスタイプに分類されないライフサイクルプロセスノードの詳細化

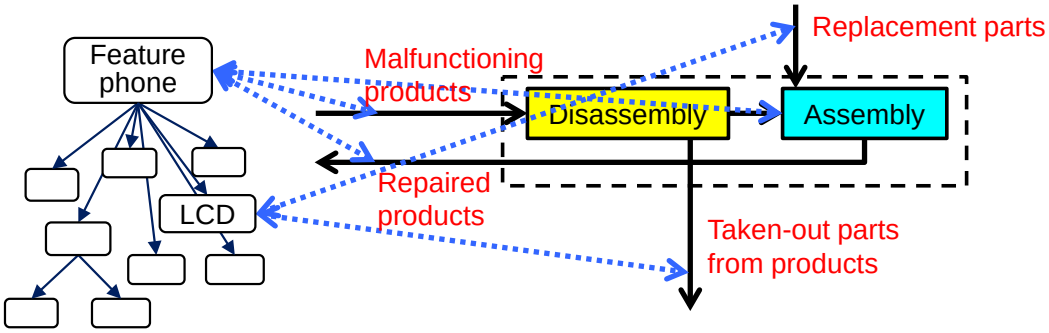
STEP2(a)において，ライフサイクルプロセスタイプに分類できないライフサイクルプロセスノードが存在する．なぜならば，製品ライフサイクルのノミナル情報モデルを用いたモデル作成は，自由度が高いためである．

ライフサイクルプロセスタイプに分類できないライフサイクルプロセスノードは，ライフサイクルプロセスタイプに分類される複数のライフサイクルプロセスノードの集合と捉え，分類可能なライフサイクルプロセスノードのみで構成される粒度まで展開するというアプローチをとる．例えば図 5-5 の Maintenance プロセスノードは，$n(FL_{lcp}^{in}) = n(FL_{lcp}^{out}) = 2$であり，いずれのライフサイクルプロセスタイプにも属さない（図 5-6 (a)参照）．この

Maintenance プロセスノードは，「故障した製品を分解し，交換が必要な部品を取り出す処理」と「交換用の新品部品を，故障していた製品に組み込んで組み立てる処理」という 2 つの処理方法を表現している．そのため，本 Maintenance プロセスノードを，図 5-6 (b)に示すように，Disassembly と Assembly という 2 つのライフサイクルプロセスノードに展開して表現する．この展開によって，ライフサイクルプロセスタイプに分類されていなかった Maintenance プロセスノードは，「(d)分離」と「(c)結合」というライフサイクルプロセスタイプに分類された 2 つのライフサイクルプロセスノードで構成することができる．図 5-6 (a)と図 5-6 (b)は，モデルの粒度が異なるものの，個体の集合に対する同一の処理方法を表現している．



(a) Not enabled to be classified into any life cycle process type because Maintenance process includes more than two types of treatments



(b) Enabled to be classified into a life cycle process type by expanding Maintenance process to Disassembly and Assembly process

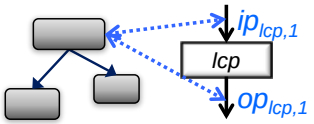
図 5-6：分類不可のライフサイクルプロセスノードの詳細化による
ライフサイクルプロセスタイプへの分類例

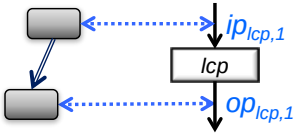
STEP3：Procedure テンプレートを用いた個体の集合に対する処理方法に関しての計算処理の生成

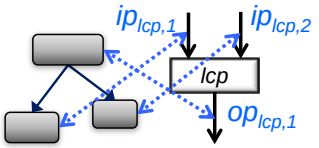
各ライフサイクルプロセスノードlcpのPR_{lcp}のうち，個体の集合に対する処理方法に関しての計算処理を，ライフサイクルプロセスタイプごとに用意する Procedure テンプレートに基づいて生成する．各ライフサイクルプロセスタイプの Procedure テンプレートを，表 5-1 に示す. ただし,「(f)分岐」と「(g)開始」に分類したライフサイクルプロセスノードlcpは，それぞれ「分岐条件の閾値を表すパラメータと条件判定するパラメータ」と「出力する個体の数を表すパラメータ（例えば，需要量)」を，GP_{lcp}かIP_{lcp}から選択する．

表 5-1 の”EntityData”は，個体情報eiを定義したクラスを表す．表 5-1 の”x.NonimalInformation”は, 個体 x の個体情報eiの要素である実体のノミナル情報niを表す．また，”LCSFunction.関数名”は，記述頻度が高い計算処理をモジュール化した関数を表す．各関数の機能，引数，出力パラメータの型を，表 5-2 に示す．さらに，X.AddRange()は，個体の集合 X の要素に，引数の個体の集合を追加する関数を表す．X.Add()は，個体の集合 X の要素に，引数の個体を 1 個体のみ追加する関数を表す．

表 5-1：ライフサイクルプロセスタイプごとの Procedure テンプレート

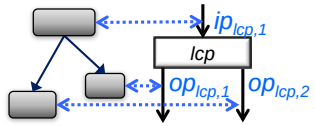
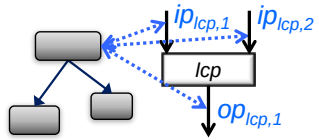
Life cycle process type	Procedure template	Representation
(a) Pass through	$op_{lcp,1}$.AddRange($ip_{lcp,1}$)	

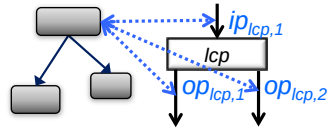
(b) Transformation	foreach(EntityData entity in $ip_{lcp,1}$)
	{
	$op_{lcp,1}$.Add(LCSFunction.Transform(entity, outputParameterName));
	}
	
	where
	outputParameterName: Name of $op_{lcp,1}$
		

(c) Combining	for(int i = 0; i < $ip_{lcp,1}$.Count; i++)
	{
	List<EntityData> componentList = new List<EntityData>();
	componentList.Add($ip_{lcp,1}$[i]);
	componentList.Add($ip_{lcp,2}$[i]);
	$op_{lcp,1}$.Add(LCSFunction.Assembly(componentList, outputParameterName);
	}
	
	where
	outputParameterName: Name of $op_{lcp,1}$
	$ip_{lcp,j}$[i]: i^{th} entity composing of $ip_{lcp,j}$
		

<table>
<tr>
<td>(d) Separating</td>
<td>
foreach(EntityData entity in $ip_{lcp,1}$)
{
　foreach(EntityData childEntity in entity.ChildEntityList)
　{
　　if(/* ni of childEntity is same as id of an Entity Node relating to $op_{lcp,1}$*/)
　　{
　　　$op_{lcp,1}$.Add(childEntity);
　　}

　　if(/*ni of childEntity is same as id of an Entity Node relating to $op_{lcp,2}$ */)
　　{
　　　$op_{lcp,2}$.Add(childEntity);
　　}
　}
}

where
entity.ChildEntityList: Aggregation of entities composing of an entity
</td>
<td> 
</td>
</tr>
<tr>
<td>(e) Merging</td>
<td>
$op_{lcp,1}$.AddRange($ip_{lcp,1}$);
$op_{lcp,1}$.AddRange($ip_{lcp,2}$);
</td>
<td> 
</td>
</tr>
</table>

(f) Sorting	foreach(EntityData entity in $ip_{lcp,1}$)
	{
	if(/* conditional expression */)
	{
	$op_{lcp,1}$.Add(entity);
	}
	else
	{
	$op_{lcp,2}$.Add(entity);
	}
	}
		

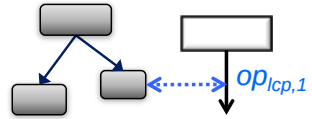
(g) Start	$op_{lcp,1}$.AddRange(LCSFunction.MakeComponentBasedOnProductionPlan(output ParameterName, productionPlanName, LCSimulator.CurrentTime)); *where* outputParameterName: Name of $op_{lcp,1}$ productionPlanName: Name of production plan LCSimulator.CurrentTime: Life cycle time	
(h) End	//Since this type of Life Cycle Process Node has no output parameter of entities, no procedure about how to treat entities is defined.	

表 5-2：関数リスト

Name	Role	Argument			Return value
.ProductionAmount	Acquiring number of manufactured entities	string production plan name	int life cycle time		double number of manufactured entities
.MakeComponent	Manufacturing an entity	string name of a manufactured entity	int life cycle time		EntityData Manufactured entity
.MakeComponent BasedOn ProductionPlan	Manufacturing entities based on a production plan	string name of manufactured entities	string production plan name	int life cycle time	List<EntityData> Aggregation of manufactured entities
.Assemble	Assembling an entity from input entities	List<EntityData> Aggregation of entities for assembly	string name of an assembled entity		EntityData Assembled entity
.SetTargetUser	Setting a user type to an entity and its child entities	EntityData Target entity	string user type name		EntityData Target entity
.UseOneTurn	Reducing 1 turn of remaining life of entities	List<EntityData> Aggregation of entities under use			List<EntityData> Aggregation of used entities
.DisposeBased OnFailureRate	Extracting disposed entities based on failure rate	List<EntityData> Aggregation of candidate for disposed entities			List<EntityData> Aggregation of disposed entities
.DisposeBased OnOtherStandard	Extracting disposed entities based on a threshold	List<EntityData> Aggregation of candidate for disposed entities	double threshold		List<EntityData> Aggregation of disposed entities

.Collecting	Extracting collected entities based on collection rate	List<EntityData> Aggregation of candidate for collected entities	double collection rate	List<EntityData> Aggregation of collected entities
.Disassemble	Disassembling an entity to its child entities	EntityData An entity for disassembly		List<EntityData> Aggregation of entities extracted through disassembly
.Disassemble ToBottomLayer	Disassembling an entity to its bottom layer	EntityData An entity for disassembly		List<EntityData> Aggregation of entities extracted through disassembly
.Transform	Transforming an entity to another entity	EntityData A transforming entity	string name of an transformed entity	EntityData Transformed entity

STEP4：計算式とパラメータ選択による評価値の算出方法に関しての計算処理の生成

評価値の算出方法に関する計算処理は，計算式とパラメータを選択することで生成する．設計者はまず，評価値を表すパラメータをOP_{lcp}から選択する．次に，評価値の算出に関する計算式を，計算式リスト（表 5-3 参照）から選択する．最後に，選択した計算式の変数に代入するパラメータを，GP_{lcp}，IP_{lcp}，OP_{lcp}，およびすべての実体ノードの属性の集合Aから選択する．

表 5-3：計算式リスト

Calculation formula	Substituted parameter
[[A]].Count*[[B]][0].WeightOfConstituentMaterial(“C”)*[[D]]	A,B: Aggregation of entities C: Material name D: Basic unit
[[A]][0].WeightOfConstituentMaterial(“B”)*[[C]]	A: Aggregation of entities B: Material name C: Basic unit
[[A]].Count*$\prod_{n=0}[[B]]_n$	A: Aggregation of entities B: Basic unit, etc.

for(int i = 0 ; i < [[A]].Count; i++)
{
[[B]] += [[A]][i].WeightOfConstituentMaterial(“C”)*[[D]];
}
	A: Aggregation of entities
	B: Target parameter
	C: Material name
	D: Basic unit

（注：[[ ]]に，選択したパラメータ名を代入する）

5.1.5. Procedure の生成例

本項では, 液晶テレビの製品ライフサイクルを対象に, Procedure の生成例を示す. まず, 液晶テレビの製品ライフサイクルのノミナル情報モデルを, 図 5-7 に示すように作成した. このとき, 作成したライフサイクルフローモデルを構成するライフサイクルプロセスノードは, それぞれ表 5-4 のようにライフサイクルプロセスタイプに分類できる.

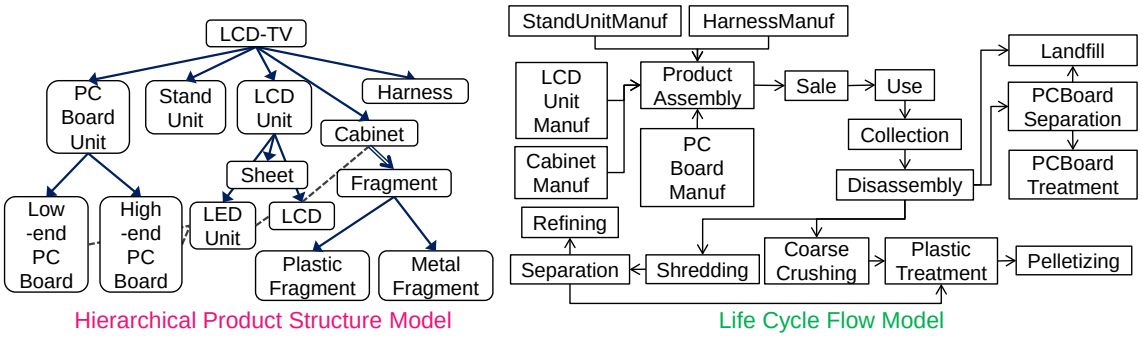


図 5-7：製品ライフサイクルのノミナル情報モデル（対象例：液晶テレビ）

表 5-4：ライフサイクルプロセスノードのライフサイクルプロセスタイプへの分類

Life cycle process type		Name of life cycle process node
(a)	Pass through	Sale, Use, Collection
(b)	Transformation	Shredding, CoarseCrushing
(c)	Combining	ProductAssembly
(d)	Separating	Disassembly
(e)	Merging	PlasticTreatment
(f)	Sorting	PCBoardSeparation, Separation
(g)	Start	LCDUnitManuf, HarnessManuf, PCBoardManuf, StandUnitManuf, CabinetManuf, SpeakerManuf
(h)	End	Refining, Landfill, Pelletizing

表 5-4 のように分類したライフサイクルプロセスノード*lcp*について, 個体の集合に対する処理方法を表す計算処理は, Procedure テンプレートに基づいて以下のように生成することができる.

- 「(a)通過」に分類されたライフサイクルプロセスノードの Procedure 生成

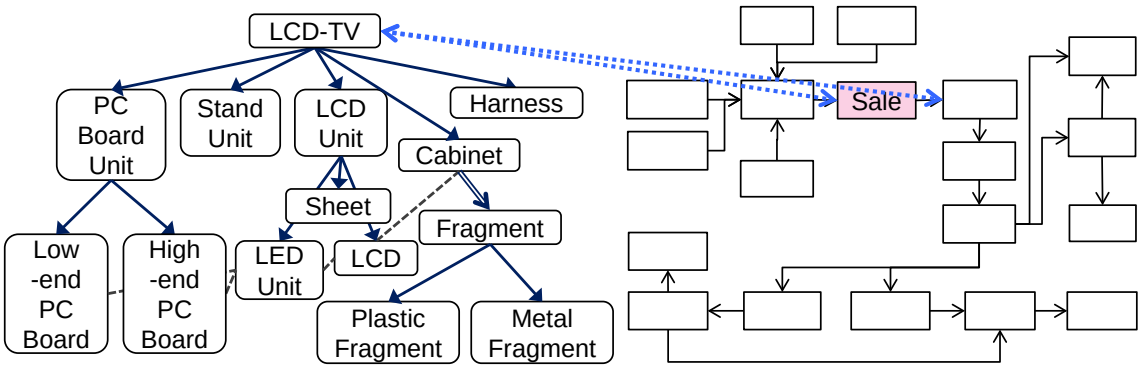


図 5-8：実体ノードとフローリンクとの対応関係に基づく
ライフサイクルプロセスタイプへの分類（図 5-7 の Sale プロセス）

```
[[soldLCDTV]].AddRange([[assembledLCDTV]]);
```

図 5-9：「(a)通過」に分類されたライフサイクルプロセスノードの Procedure 生成例
（図 5-7 の Sale プロセスの Procedure）

- 「(b)変換」に分類されたライフサイクルプロセスノードの Procedure 生成

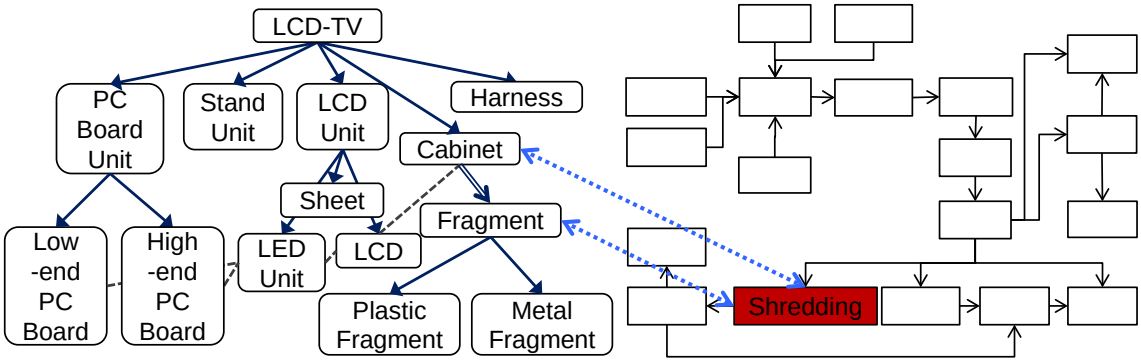


図 5-10：実体ノードとフローリンクとの対応関係に基づく
ライフサイクルプロセスタイプへの分類（図 5-7 の Shredding プロセス）

```
foreach(EntityData entity in [[Cabinet]])
{
   [[Fragment]].Add(LCSFunction.Transform(entity, "Fragment"));
}
```

図 5-11：「(b)変換」に分類されたライフサイクルプロセスノードの Procedure 生成例
（図 5-7 の Shredding プロセスの Procedure）

- 「(c)組立」に分類されたライフサイクルプロセスノードの Procedure 生成

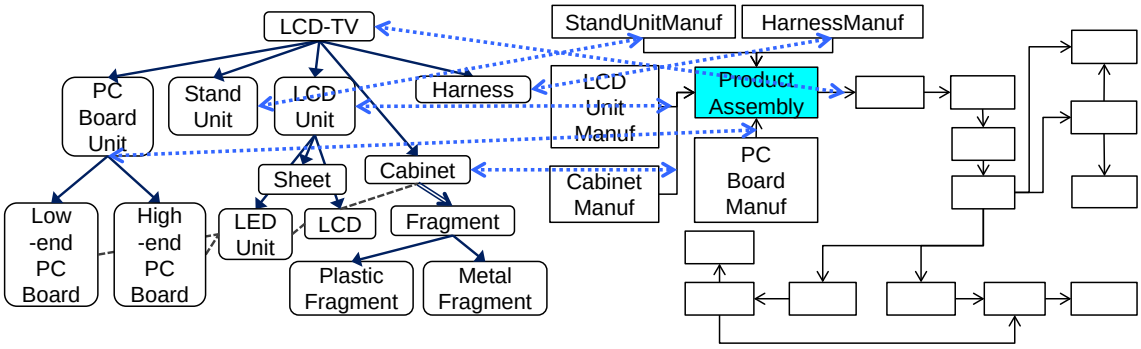


図 5-12：実体ノードとフローリンクとの対応関係に基づく
ライフサイクルプロセスタイプへの分類（図 5-7 の ProductAssembly プロセス）

```
for(int i = 0 ; i < [[StandUnit]].Count; i++ )
{
  List<EntityData> componentList = new List<EntityData>();

   componentList.Add([[StandUnit]][i]);
   componentList.Add([[LCDUnit]][i]);
   componentList.Add([[Harness]][i]);
   componentList.Add([[PCBoardUnit]][i]);
   componentList.Add([[Cabinet]][i]);
   [[assembledLCDTV]].Add(LCSFunction.Assembly(componentList, "assembledLCDTV "));
}
```

図 5-13：「(c)組立」に分類されたライフサイクルプロセスノードの Procedure 生成例
（図 5-7 の Product Assembly プロセスの Procedure）

- 「(d)分解」に分類されたライフサイクルプロセスノードの Procedure 生成

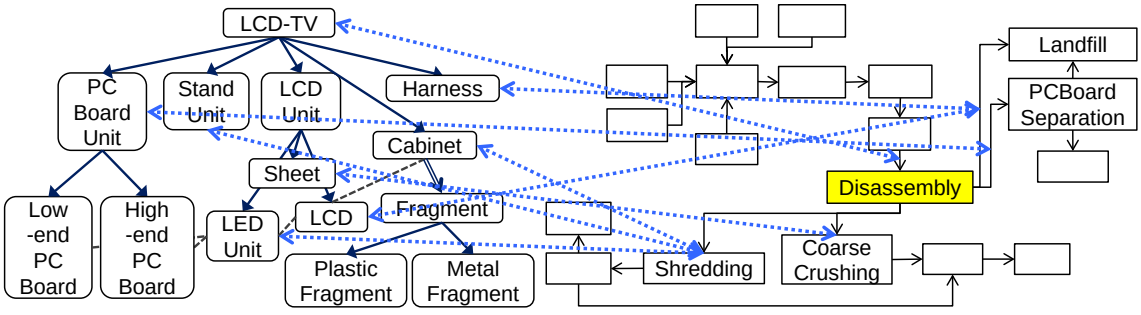


図 5-14：実体ノードとフローリンクとの対応関係に基づく
ライフサイクルプロセスタイプへの分類（図 5-7 の Disassembly プロセス）

```
foreach(EntityData entity in [[collectedLCDTV]])
{
   foreach(EntityData childEntity in entity.ChildEntityList.Values)
   {
      if(childEntity.NominalInformation.Name=="PCBoardUnit")
      {
         [[PCBoardUnit]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="StandUnit")
      {
         [[StandUnit]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="LCD")
      {
         [[LCD]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="LEDUnit")
      {
         [[LEDUnit]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Sheet")
      {
         [[Sheet]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Cabinet")
      {
         [[Cabinet]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Harness")
      {
         [[Harness]].Add(childEntity);
      }
   }
}
```

図 5-15：「(d)分解」に分類されたライフサイクルプロセスノードの Procedure 生成例
（図 5-7 の Disassembly プロセスの Procedure）

- 「(e)合流」に分類されたライフサイクルプロセスノードの Procedure 生成

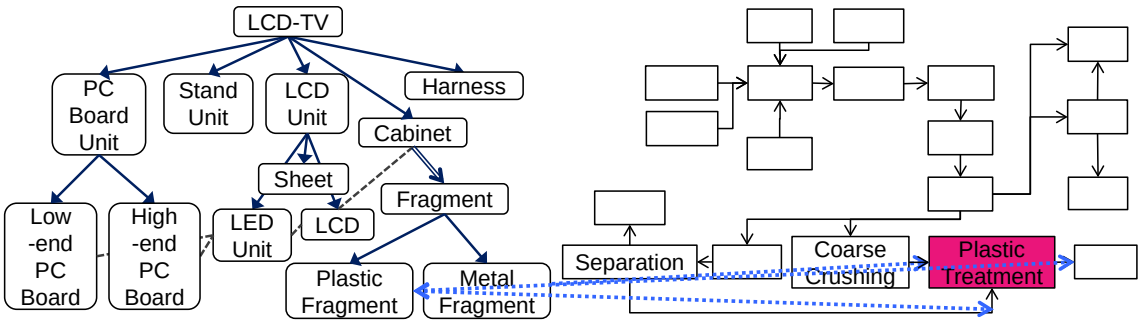


図 5-16：実体ノードとフローリンクとの対応関係に基づく
ライフサイクルプロセスタイプへの分類（図 5-7 の PlasticTreatment プロセス）

```
[[PlasticFragment]].AddRange([[CoarseCrushedFragment]]);
[[PlasticFragment]].AddRange([[ShreddedFragment]]);
```

図 5-17：「(e)合流」に分類されたライフサイクルプロセスノードの Procedure 生成例
（図 5-7 の PlasticTreatment プロセスの Procedure）

- 「(f)分岐」に分類されたライフサイクルプロセスノードの Procedure 生成

本ライフサイクルプロセスタイプに分類されるライフサイクルプロセスノードでは，分岐条件の閾値を表すパラメータや条件判定するパラメータを選択する必要がある．図 5-19 の例では，条件判定するパラメータとして構成素材を選定し，その構成素材が PCBoard (High end)）であるかどうかによって分岐させるように，計算処理を生成している．

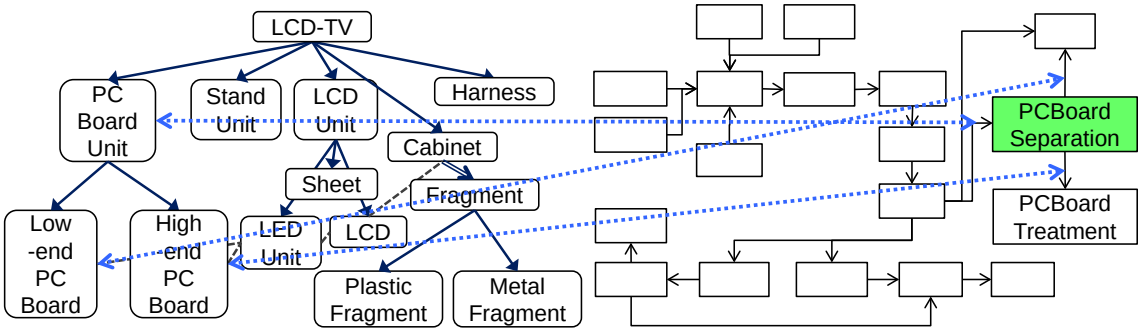


図 5-18：実体ノードとフローリンクとの対応関係に基づく
ライフサイクルプロセスタイプへの分類（図 5-7 の PCBoardSeparation プロセス）

```
foreach(EntityData entity in [[PCBoardUnit]])
{
    if(entity.NominalInformation.MaterialList.ContainsKey("PCBoard (High end)"))
    {
        [[HighEndPCBoard]].Add(entity);
    }
     else
     {
        [[LowEndPCBoard]].Add(entity);
     }
}
```

図 5-19：「(f)分岐」に分類されたライフサイクルプロセスノードの Procedure 生成例
（図 5-7 の PCBoardSeparation プロセスの Procedure）

- **「(g)開始」に分類されたライフサイクルプロセスノードの Procedure 生成**

本ライフサイクルプロセスタイプに分類されるライフサイクルプロセスノードでは，出力する個体の数を表すパラメータを選定する必要がある．図 5-21 の例では，出力する個体の数を表すパラメータとして，LCD-TV の生産計画を表す「ProductionPlanForLCDTV」を選択している．

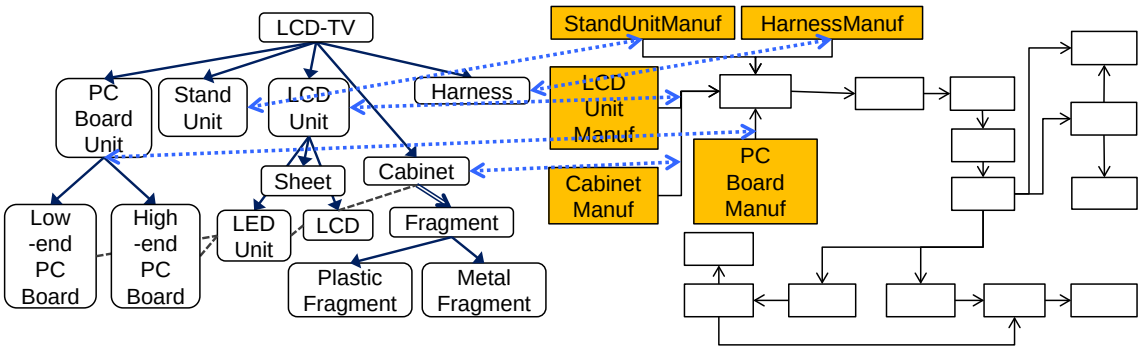


図 5-20：実体ノードとフローリンクとの対応関係に基づく
ライフサイクルプロセスタイプへの分類（図 5-7 の CabinetManuf プロセス）

```
[[Cabinet]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan
("Cabinet","ProductionPlanForLCDTV",LCSimulator.CurrentTime));
```

図 5-21：「(g)開始」に分類されたライフサイクルプロセスノードの Procedure 生成例
（図 5-7 の CabinetManuf プロセスの Procedure）

5.2. 製品ライフサイクルの修正案特定支援手法

本節では，製品ライフサイクルのノミナル情報について個体情報の評価を実行した結果が設計要求を満たしていない場合に，ノミナル情報の修正を支援する手法を提案する．

5.2.1. 修正案特定支援手法のアプローチ

第 4 章で提案したモデル化手法を構成する製品ライフサイクルのノミナル情報モデルは，製品とライフサイクルフローのノミナル情報を構成するパラメータ間の関係を，ライフサイクルプロセスノードの Procedure とノード間のリンクによって表現している．なお本研究では，すべてのライフサイクルプロセスノードの Given parameter の集合GP，Input parameter の集合IP，Output parameter の集合OP，およびすべての実体ノードの属性の集合Aの要素を，パラメータと捉える．また設計パラメータを，設計者がパラメータ値を直接代入できるパラメータと捉える．つまり，$pr_{lcp,l}$によってパラメータ値が決まる$op_{lcp,k}$や，上流のライフサイクルプロセスノードlcp'の$op_{lcp',k}$の値によってパラメータ値が決まる$ip_{lcp,j}$は設計パラメータに含めない．

本研究では，製品ライフサイクルのノミナル情報のうち，設計パラメータである Given parameter の集合GPか属性の集合Aから修正案を特定するための支援手法を提案する．本手法を，以下 2 つの手順で構成する（図 5-22 参照）．

(a) 製品階層構造モデルとライフサイクルフローモデルを構成するパラメータの因果関係グラフを作成する．本因果関係グラフを用いることで，要求値を満たしていない評価値に影響を与える設計パラメータ候補を抽出する．

(b) 対象の評価値への影響が大きい設計パラメータを，感度分析によって特定する．

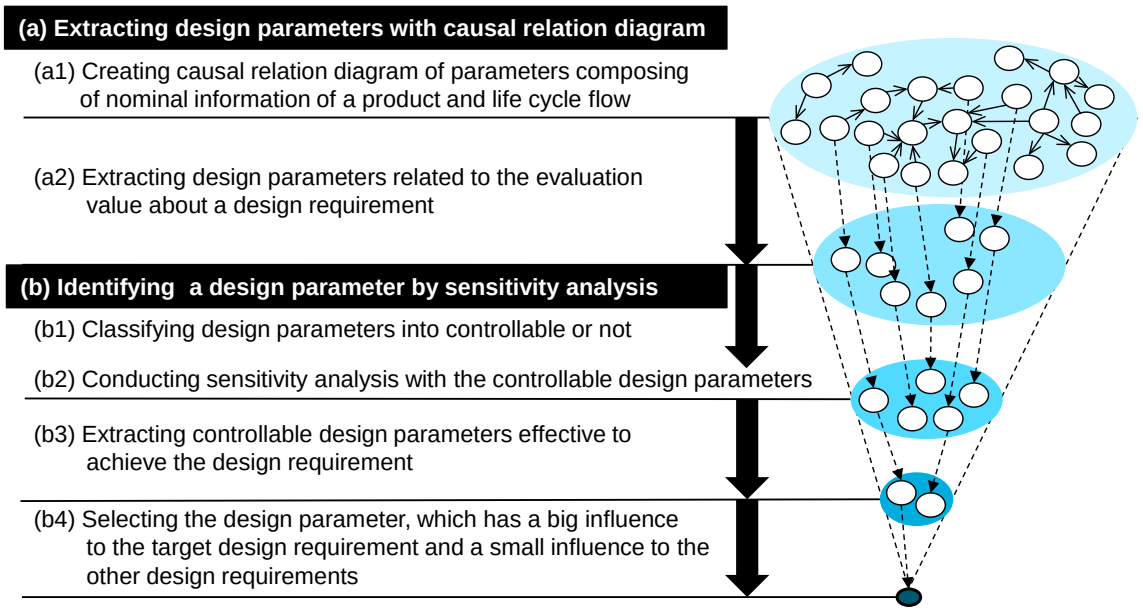


図 5-22：設計パラメータの特定手順

5.2.2. 因果関係グラフを用いた設計パラメータ候補の抽出

要求値を満たしていない評価値に影響を与える設計パラメータを抽出する手法（図5-22(a)参照）を示す．具体的には，まず，製品ライフサイクルのノミナル情報モデルを構成するパラメータ間の関係を，因果関係グラフとして表現する手法について示す．次に，因果関係グラフを用い，対象の評価値に影響を与えるパラメータを探索することで，修正すべき設計パラメータ候補を抽出する手法を示す．

(a1) 因果関係グラフの作成

製品ライフサイクルのノミナル情報モデルを用いて表現した製品とライフサイクルフローを構成するパラメータについて，以下の 3 段階で因果関係を特定する．

(ア) ライフサイクルプロセスノードlcpの$pr_{lcp,l}$において，$op_{lcp,k}$が式(3.1)を用いて表現されている場合，$\{gp_{lcp,i'}\}$，$\{ip_{lcp,j'}\}$，$\{op_{lcp,k'}\}$を$op_{lcp,k}$に影響を与えるパラメータとして特定する．このとき，$ip_{lcp,j'}$や$op_{lcp,k'}$が個体の集合を表し，かつ式(3.1)の関数が個体の属性a_αを変数としている場合，$ip_{lcp,j'}$や$op_{lcp,k'}$と実体ノードとの対応関係［13］から，個体のノミナル情報を表す実体ノードniの属性a_αも$op_{lcp,k}$に影響を与えるパラメータとして特定する．

(イ) $op_{lcp,k}$が条件文の後件部や反復処理文のループ内に含まれる場合，条件文の前件部および反復処理文の条件式に含まれるパラメータを，$op_{lcp,k}$に影響を与えるパラメータとして特定する．

(ウ) ライフサイクルフローモデルはライフサイクルプロセス間の個体の集合や無形物の流れを表す．そのため，ライフサイクルプロセスノードlcpの$op_{lcp,k}$の値は，下流のライフサイクルプロセスノードlcp'の$ip_{lcp',j}$の値に影響を与える．この$op_{lcp,k}$から$ip_{lcp',j}$への影響を因果関係として特定する．

上記の 3 段階で特定されるパラメータ間の関係を，因果関係グラフとして表現する（図5-23 参照）．本手法では因果関係グラフを，パラメータ（$gp_{lcp,i}$，$ip_{lcp,j}$，$op_{lcp,k}$，a_α）を表すパラメータノードと，パラメータ間の関係を表す因果リンクによって表現する．なお，図 5-23 の長方形ノードは，実体ノードとライフサイクルプロセスノードを表している．

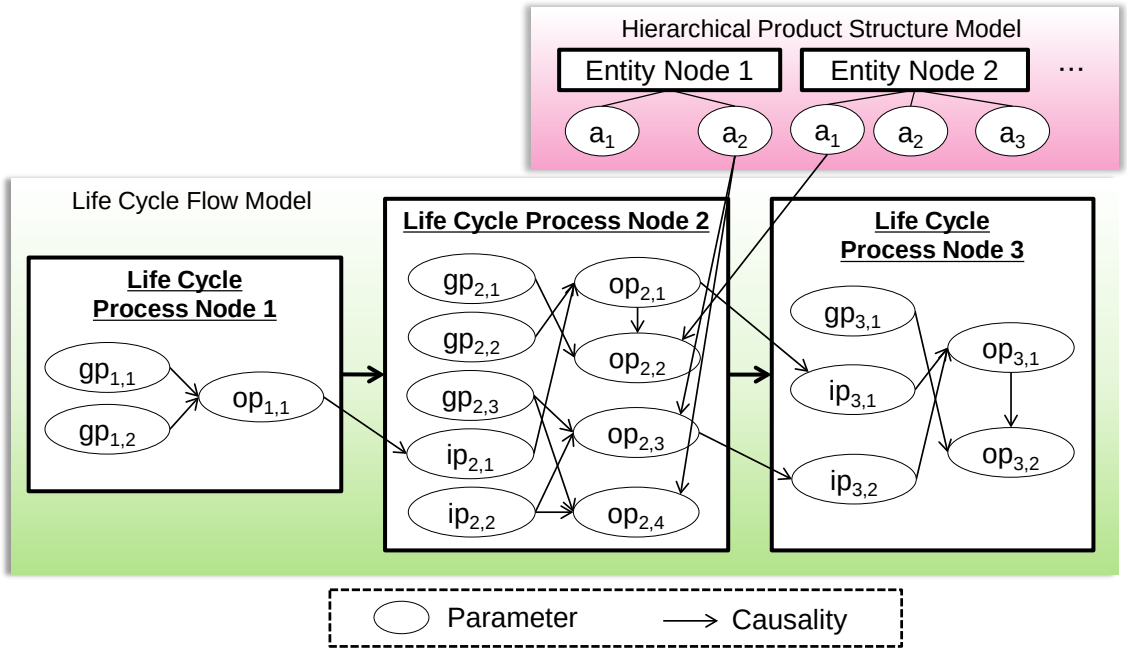


図 5-23：製品ライフサイクルのノミナル情報を構成するパラメータの因果関係グラフ

(a2) 因果関係グラフを用いた設計パラメータの探索

(a1)の過程で作成する因果関係グラフ上で，要求値を満たしていない評価値を表すパラメータノードを始点にして，因果リンクの逆向きにパラメータノードを探索する．探索したパラメータノードが Given parameter $gp_{lcp,i}$か製品属性a_αの場合，そのパラメータを修正すべき設計パラメータ候補として抽出する．

5.2.3. 感度分析による設計パラメータの特定

5.2.2 項の手法で抽出した設計パラメータ候補群の中で，設計要求に関する評価値への影響が大きい設計パラメータを特定する．本手法では，以下の手順で特定する（図 5-22(b) 参照）．

(b1) 5.2.2 項の手法で抽出される設計パラメータは膨大となり得るため，抽出した設計パラメータすべてを感度分析するには膨大な時間がかかる．設計サイクルを効率よく進めるために，抽出した設計パラメータ候補を設計者にとって制御可能か制御不可かに分類することで，感度分析する設計パラメータの数を削減する．分類は，設計者の知識と経験に基づくものとする．

(b2) 制御可能な設計パラメータ候補群について，感度分析を実施する．感度分析では，制御可能な設計パラメータ候補のパラメータ値を増減両方に一定割合変動させる．製品ライフサイクルのノミナル情報モデル上で各パラメータ値をそれぞれ独立に変更し，LCS を実行して個体情報モデルを作成する．つまり，制御可能な設計パラメータ候補が n 個ある場合，計 2n 回の LCS の実行によって 2n 個の個体情報モデルを作成する．

(b3) 感度分析の過程で作成した個体情報モデルを用いて，個体情報の評価を行う．評価結果から，要求値を満たしていない評価値への影響が大きい順に，設計パラメータ候補を並べる．

(b4) 対象とする評価値への影響がより大きく，かつ他の設計要求に関する評価値への影響が小さい設計パラメータを，修正すべき設計パラメータとする．

5.3. 第 5 章のまとめ

本章では，製品ライフサイクルの設計における設計サイクルを円滑に実行可能とするために，「個体情報の評価」と「ノミナル情報の修正」という 2 つの設計過程を実行するための支援手法を提案した．

第一に，ライフサイクルフローモデルの要素であり，ライフサイクルプロセスの振る舞いを表す Procedure の記述を支援する手法を提案した．本手法では，ライフサイクルプロセスノードlcpのPR_{lcp}のうち，Output parameter の値変化に関する計算処理を，「個体の集合に対する処理方法」と「評価値の算出方法」の 2 つに分類した．このうち，個体の集合に対する処理方法に関しての計算処理は，ライフサイクルプロセスノードを製品階層構造モデルとライフサイクルフローモデルの対応関係 [13]から 8 つのライフサイクルプロセスタイプのいずれかに分類することで，ライフサイクルプロセスタイプごとに用意した Procedure テンプレートに基づいて生成するものとした．また，評価値の算出方法に関する計算処理は，個体の集合に対する処理に伴って生じる環境負荷排出量やコストといった評価値を表すパラメータを選択し，その評価値を算出する計算式と関連するパラメータを設計者が選択することによって生成するものとした．

第二に，第 4 章で提案したモデル化手法を用いて表現した製品ライフサイクルのノミナル情報について，個体情報の評価を実行した結果が設計要求を満たしていない場合に，修正すべき設計パラメータを特定する手法を提案した．本手法では，まず製品階層構造モデルとライフサイクルフローモデルを用いて表現した製品ライフサイクルのノミナル情報を構成するパラメータ間の関係を，因果関係グラフとして作成するものとした．次に，因果関係グラフ上で，対象の設計要求に関する評価値に影響を与える設計パラメータを抽出するものとした．さらに，抽出した設計パラメータの感度分析を実施することで，対象の評価値に対する影響が大きい設計パラメータを特定するものとした．

第6章 計算機実装

第 4 章と第 5 章で提案した 2 つの手法から成る製品ライフサイクルの設計支援手法を計算機に実装したシステムとして，Entity-oriented Product Life Cycle-CAD システム（e-PLC-CAD システム）を構築した．第 6 章では，e-PLC-CAD システムの仕様と構成，およびサブシステムの機能について述べる．

6.1. システムの仕様

第 4 章と第 5 章で提案した 2 つの手法から成る製品ライフサイクルの設計支援手法を計算機上で実行可能とするために，以下に示す 5 つの基本機能を実装する．

(1) 製品ライフサイクルのノミナル情報モデルの作成と変動要因の設定
- 製品階層構造モデルを用いた製品のノミナル情報のモデル作成
- ライフサイクルフローモデルを用いたライフサイクルフローのノミナル情報のモデル作成
- 製品階層構造モデルとライフサイクルフローモデルの対応関係の管理
- 製品ライフサイクルのノミナル情報モデルを用いた変動要因の設定
- 環境性や経済性評価のためのデータ（以下，ライフサイクルインベントリデータ）の管理と利用

(2) 個体情報モデルの格納と個体情報の表示
- 各ライフサイクルプロセスにおける個体の状態の違いの表示
- 各ライフサイクルプロセスにおける個体の量の時間変化の表示
- 各個体の状態やライフサイクル履歴の表示

(3) LCS を用いた製品ライフサイクルのノミナル情報モデルから個体情報モデルの作成
- 製品階層構造モデルとライフサイクルフローモデルを入力とした LCS の実行
- LCS の実行過程における個体情報モデルの作成

(4) 個体情報の評価の実行支援
- ライフサイクルプロセスノードのライフサイクルプロセスタイプへの分類
- Procedure テンプレートを用いた計算処理の生成
- 計算式リストを用いた計算処理の生成

(5) 製品ライフサイクルのノミナル情報の修正案特定支援
- 製品とライフサイクルフローのノミナル情報を構成するパラメータの因果関係グラフの生成
- 対象の設計要求に関する評価値に影響を与える設計パラメータの抽出
- 設計パラメータ候補群の感度分析の実施

6.2. システムの基本構成

6.1 節で挙げた 5 つの基本機能を実現するため，本研究では図 6-1 に示す構成で，e-PLC-CAD システムを構築した. 本システムは, Microsoft 社の Windows OS 上で動作し，実装には Visual C#を用いている.

e-PLC-CAD システムを，以下 6 つのサブシステムで構成する.

1. **Entity-oriented Product Life Cycle Design Manager**：製品ライフサイクルの設計過程における設計サイクルの実行を統括するシステム. 他のサブシステムは本サブシステムから起動する.
2. **Multiple Products Modeling System**：製品のノミナル情報を表す製品階層構造モデルを作成するシステム.
3. **Life Cycle Flow Modeling System**：ライフサイクルフローモデルを作成するシステム. 本サブシステム上で, 製品階層構造モデルとライフサイクルフローモデルとの対応関係を定義する. 個体情報の評価実行支援は，本サブシステム内の Procedure 記述支援システムによって実行する.
4. **Entity Information Model Manager**：個体情報を格納した個体情報モデルの管理と，管理している個体情報モデルから任意の個体情報を抽出して各ライフサイクルプロセスでの個体の状態の違いや量の変動を表示するシステム.
5. **Life Cycle Simulator**：LCS を実行するシステム.
6. **Design Modification Support System**：対象の設計要求に関する評価値に影響を与える設計パラメータを抽出し，感度分析を実施するシステム.

本システムは，製品ライフサイクルのノミナル情報モデル化システム [13]を拡張したサブシテム（図 6-1 の黒枠黒文字のノード）と，本研究で追加した新たな機能をもつサブシステム (図 6-1 の赤枠赤文字のノード) から成る. 具体的には,「1. Entity-oriented Product Life Cycle Design Manager」と「2. Multiple Products Modeling System」と「3. Life Cycle Flow Modeling System」は， [13]のシステムを拡張したサブシステムである. また「5. Life Cycle Simulator」は，ライフサイクルシミュレーションシステム [14]のアルゴリズムを用いたサブシステムである. 次節以降で，各サブシステムの詳細機能について述べる.

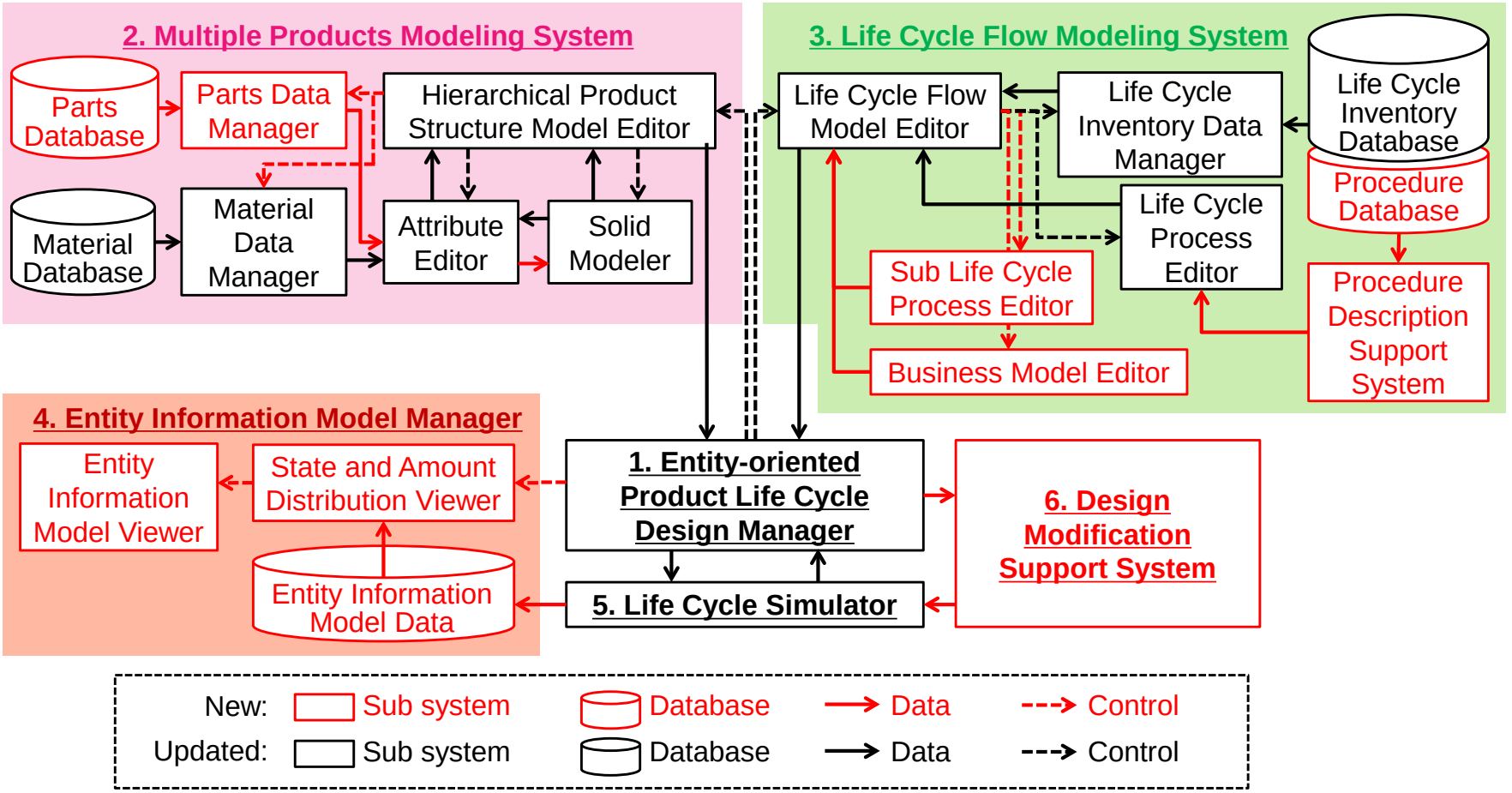


図 6-1：e-PLC-CAD システムの基本構成

6.3. Entity-oriented Product Life Cycle Design Manager

Entity-oriented Product Life Cycle Design Manager は，製品個体の集合に着目した製品ライフサイクルの設計過程を統括するサブシステムである．他 5 つのサブシステムは，本サブシステムから起動する（図 6-2 参照）．また本サブシステムは，他 5 つのサブシステムで編集した情報を管理する．

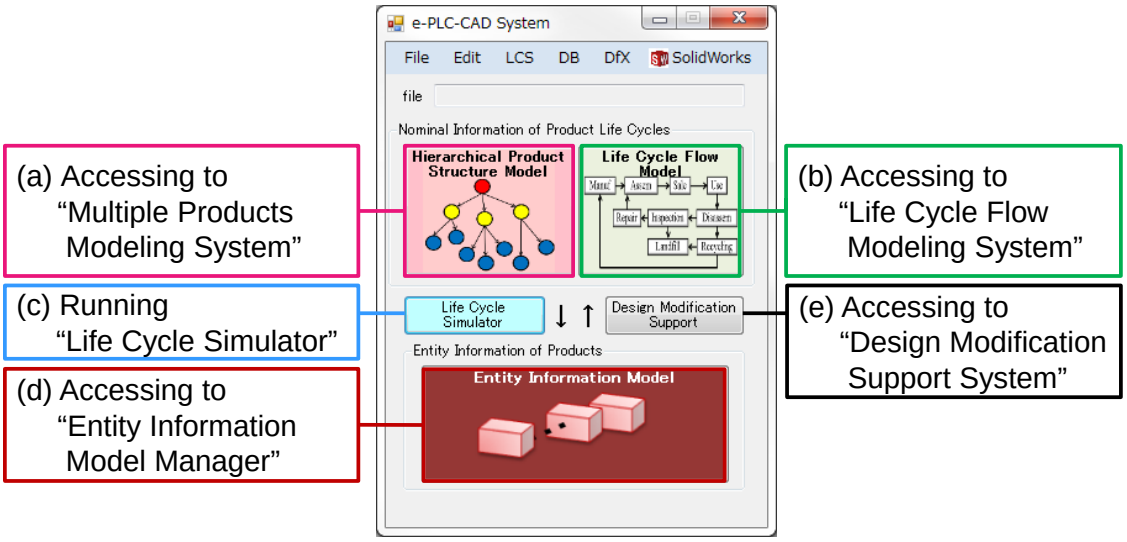


図 6-2：Entity-oriented Product Life Cycle Design Manager のスクリーンショット

6.4. Multiple Products Modeling System

Multiple Products Modeling System は，製品階層構造モデルのデータを編集や表示する Hierarchical Product Structure Model Editor，各実体ノードの属性を編集する Attribute Editor，属性のひとつである幾何形状を編集する Solid Modeler，過去に設計した製品や部品のノミナル情報を管理や参照可能とする Parts Data Manager，属性のひとつである構成素材のデータを管理や参照可能とする Material Data Manager から成る．

6.4.1. Hierarchical Product Structure Model Editor

Hierarchical Product Structure Model Editor は，製品階層構造モデルを作成するためのシステム [13]を拡張し，複数製品の製品階層構造モデルを編集や表示可能としたツールである（図 6-3 参照）．

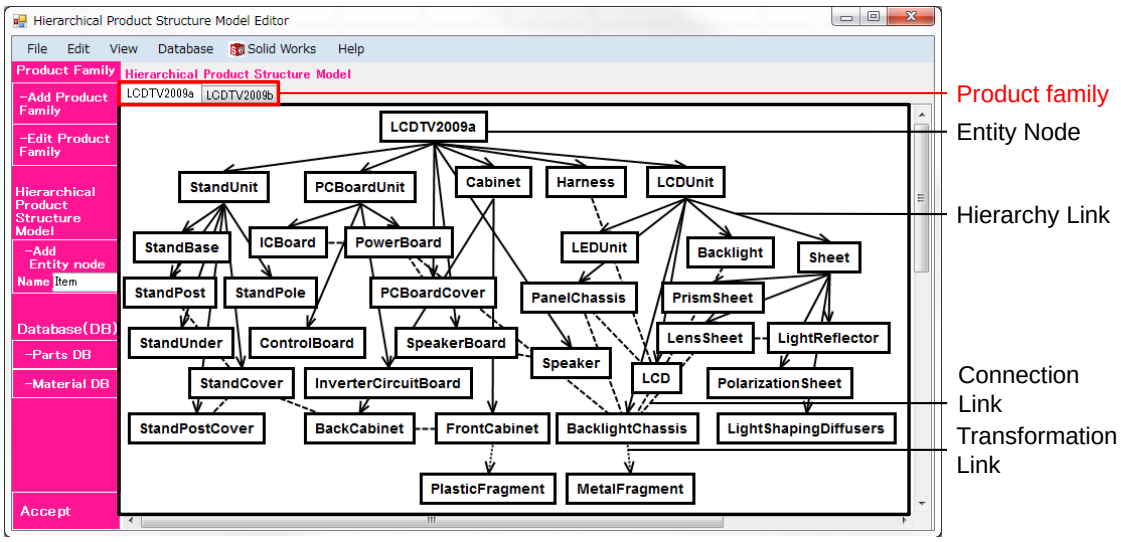


図 6-3：Hierarchical Product Structure Model Editor のスクリーンショット（対象例：液晶テレビ）

本ツールで作成する実体ノードの属性は，各実体ノードから Attribute Editor（6.4.2 項参照）にアクセスして編集できる．また本ツールは，Parts Data Manager（0 項参照）や Material Data Manager（0 項参照）と接続している．そのため設計者は，過去に設計した部品のノミナル情報や素材情報を用いて，属性値の初期値や変化率を設定することができる．

6.4.2. Attribute Editor

Attribute Editor は，製品階層構造モデルの実体ノードの属性を編集するためのツールである（図 6-4 参照）．ノミナル情報モデル化システム [13]では，実体の属性として，「物理寿命」「構成素材とその重量」が編集可能である．なお実体の重量は，設定した素材重量の総和として決定される．本ツールではこれらの機能に加えて，値が変化する属性の代表例としての「性能」と変動要因(i)の代表例としての「故障率」を設定可能とするため，編集機能を図 6-4 の右側に示すように拡張した．

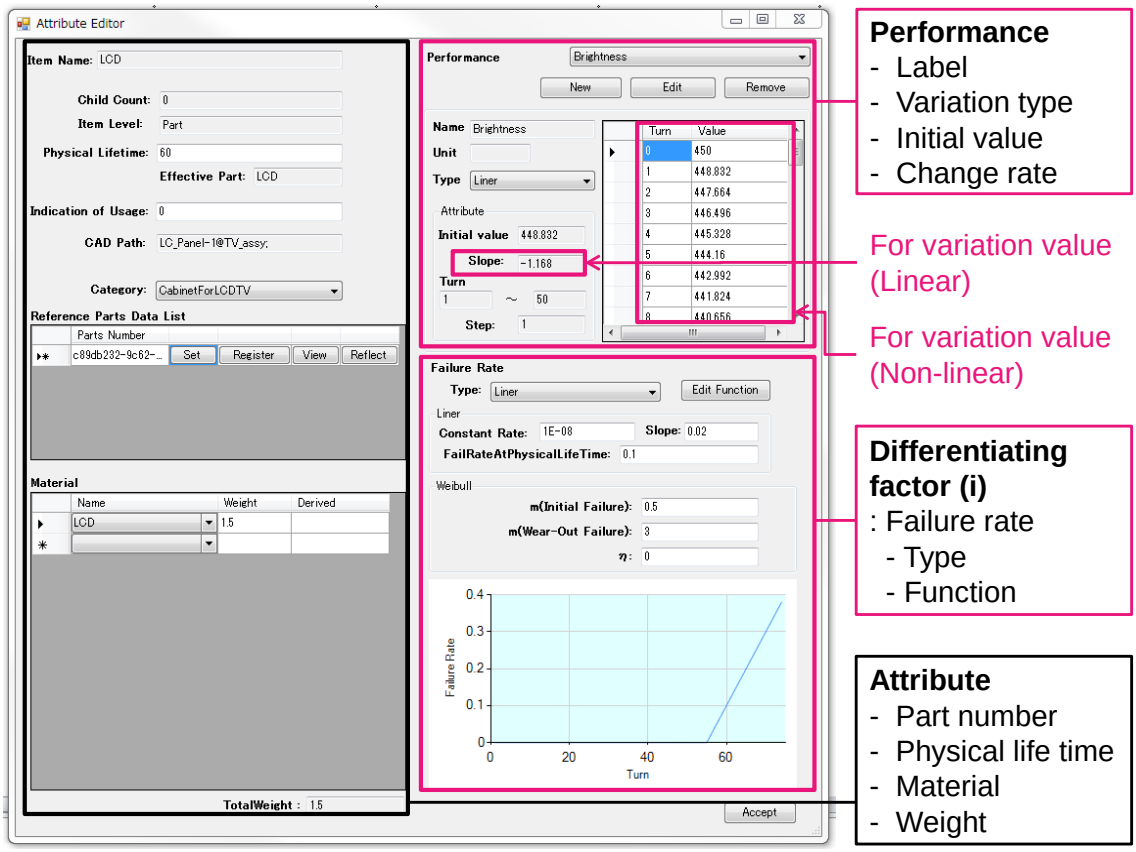


図 6-4 : Attribute Editor のスクリーンショット

性能には，項目，型，初期値，変化率$\Delta_\alpha(X)$の 4 つの値を設定する．本ツールでは，まず性能の型を，固定値，変動値（線形変化），変動値（非線形変化），という 3 つから選択する．固定値の場合，変化率$\Delta_\alpha(X) = 0$となる．変動値（線形変化）の場合，変化率$\Delta_\alpha(X)$を定数で設定する．変動値（非線形変化）の場合，変数x（例えば，ライフサイクルタイム）に対する性能値y（例えば，輝度）を入力することによって，変化率$\Delta_\alpha(X)$を設定する．

各実体の故障率には，型と関数を設定する．故障率の型は，バスタブ曲線型と非線形型の 2 つの型から選択する．バスタブ曲線型では，初期故障はないものとし，偶発期間故障

率$fr_{constant}$，摩耗期間故障率傾き$\Delta_{wear-out}$，物理寿命pl，物理寿命時の故障率fr_{pl}，という4つのパラメータを用い，式(6.1)に従って故障率関数$fr(t)$を設定する．

$$fr(t) = fr_{constant} \qquad (if\ 0 < t < pl - \frac{fr_{pl} - fr_{constant}}{\Delta_{wear-out}})$$
$$fr(t) = \Delta_{wear-out} \times (t - pl) + fr_{pl} \quad (if\ pl - \frac{fr_{pl} - fr_{constant}}{\Delta_{wear-out}} \leq t) \tag{6.1}$$

非線形型では，時間tに対する故障率yの値を入力することによって，故障率関数を設定する．

6.4.3. Solid Modeler

ノミナル情報モデル化システム [13]と同様に，属性のひとつである幾何形状を編集や表示可能とするため，本システムはソリッドモデラのひとつである SOLIDWORKS [90]と接続している．SOLIDWORKS の Application Programming Interface (API)には，Visual C# を用いて呼び出し可能な関数が多く定義されている．6.4.1 項の Hierarchical Product Structure Model Editor は，この API の関数を用いることで，ソリッドモデラ上のソリッドモデルの操作や情報の取得が可能である．具体的には，作成したソリッドモデルから素材や重量などのデータを取得し，実体ノードの属性値に設定することができる．

6.4.4. Parts Data Manager

Parts Data Manager は，Parts Database からの部品データの取得と，Parts Database への部品データの追加や更新を行うツールである（図 6-5 参照）.

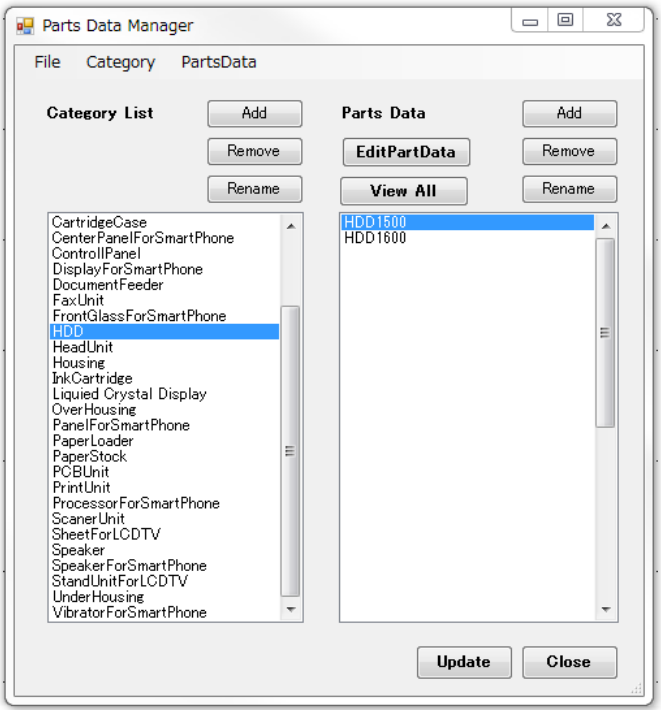


図 6-5：Parts Data Manager のスクリーンショット

Parts Database は，格納する部品データに，カテゴリと品番を割り当てる．Parts Data Manager を通じ，複数の実体ノードに共通の品番を割り当てることで，それら実体ノードに同じ属性を保持させ，共通部品として扱うことができる．

6.4.5. Material Data Manager

Material Data Manager は，Material Database からの素材データの取得と，Material Database への素材データの追加や更新を行うツールである（図 6-6 参照）．Material Database が格納する素材データは，名前と材料特性（例えば，密度，ヤング率，熱伝導率，など）を保持する．材料特性は，項目，値，単位から成る．設計者は，任意の材料特性を追加や削除することができる．

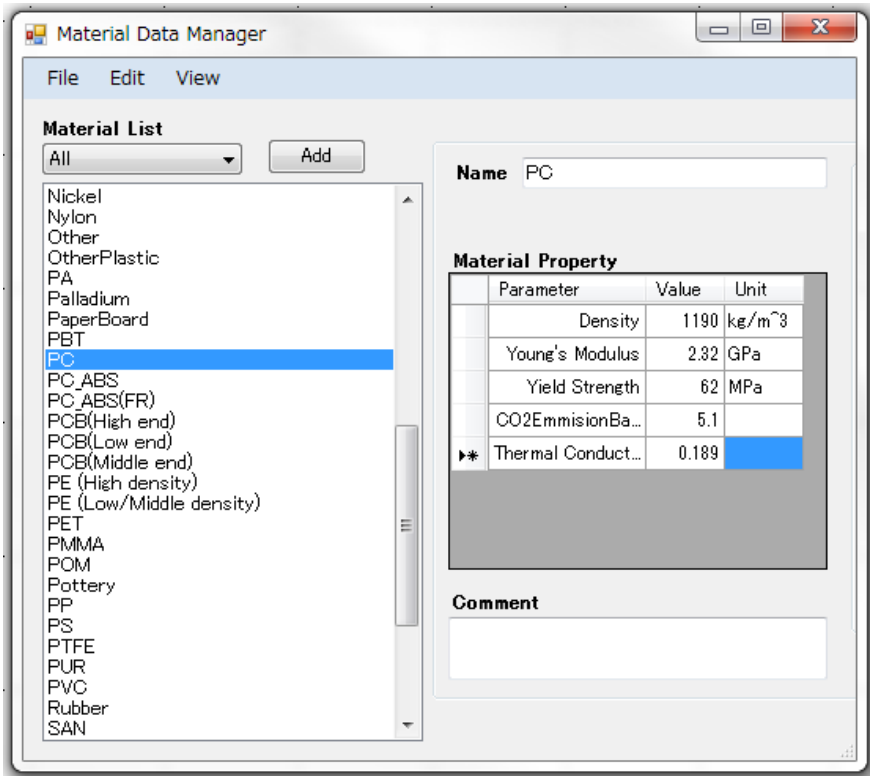


図 6-6：Material Data Manager のスクリーンショット

6.5. Life Cycle Flow Modeling System

Life Cycle Flow Modeling System は，ライフサイクルフローモデルのデータを編集や表示する Life Cycle Flow Model Editor，ライフサイクルフローモデルの要素であるライフサイクルプロセスノードlcpの$GP_{lcp}, IP_{lcp}, OP_{lcp}, PR_{lcp}$を編集する Life Cycle Process Editor，ライフサイクルプロセスノードを展開して編集する Sub Life Cycle Process Editor，生産計画とユーザモデルを編集する Business Model Editor，ライフサイクルインベントリデータを編集と管理する Life Cycle Inventory Data Manager，Procedure の記述を支援する Procedure Description Support System から成る．

6.5.1. Life Cycle Flow Model Editor

Life Cycle Flow Model Editor は，ライフサイクルフローモデルを作成するためのツールであり，ノミナル情報モデル化システム [13]に基づく（図 6-7 参照）．

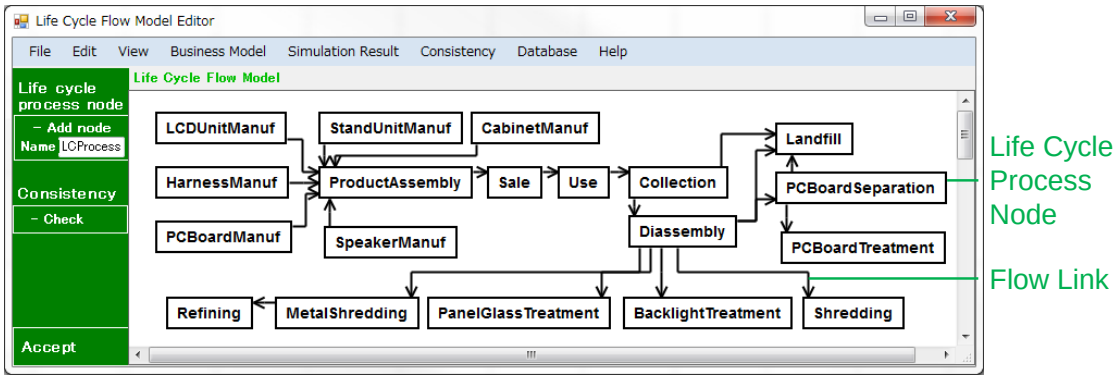


図 6-7：Life Cycle Flow Model Editor のスクリーンショット（対象例：液晶テレビ）

5.1.4 項で示したように，ライフサイクルフローモデルの作成過程では，ライフサイクルプロセスノードを展開して詳細化する場合がある．そのため本ツールは，ライフサイクルプロセスノードを階層的に作成可能とする．この目的のため，Life Cycle Flow Modeling Systemで扱うライフサイクルフローのデータを，図 6-8 に示すような階層構造で管理する．つまり，ライフサイクルフローのデータは，図 6-7 のLife Cycle Flow Model Editor上で作成されているライフサイクルフローモデルを 2nd Layerとし，ライフサイクルフロー全体で共有する情報やライフサイクルフロー全体の計算結果を表す情報を保持する階層を 1st Layerとする．また，ライフサイクルフローのデータは，2nd Layerのライフサイクルプロセスノードを展開したライフサイクルプロセスノード群やその間のフローリンクから成る情報を保持する階層をnth Layer（nは 3 以上の整数）とする．

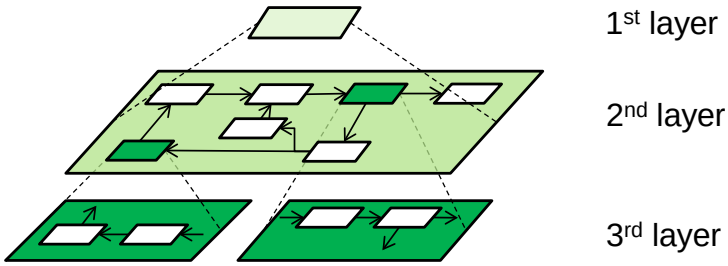


図 6-8：ライフサイクルフローモデルのデータ構造のイメージ図

6.5.2. Life Cycle Process Editor

Life Cycle Process Editor は, 各ライフサイクルプロセスノードlcpのGP_{lcp}, IP_{lcp}, OP_{lcp}, PR_{lcp}を編集するツールである.

GP_{lcp}, IP_{lcp}, OP_{lcp}を構成する$gp_{lcp,i}$, $ip_{lcp,j}$, $op_{lcp,k}$には，それぞれパラメータ名，パラメータ値，単位を設定する．これらの要素に加え，$gp_{lcp,i}$には，Life Cycle Inventory Data Manager のライフサイクルインベントリデータ（6.5.5 項参照）への参照パスを値として保持させることで，ライフサイクルインベントリデータを定数属性値として扱うことができる．$ip_{lcp,j}$には，他のライフサイクルプロセスノードlcp'から出力する$op_{lcp',k}$との関係を設定する．この関係によって，ライフサイクルプロセスノードlcp'からlcpへの個体の集合や無形物の流れを表現し，Life Cycle Flow Model Editor 上のライフサイクルプロセスノード間にフローリンクを生成する．$op_{lcp,k}$には，パラメータが表す対象に応じた型を設定する．$op_{lcp,k}$の型は，double と List<EntityData>という 2 種類から成る．型 double の$op_{lcp,k}$は「評価値」であることを示し，型 List<EntityData>の$op_{lcp,k}$は「個体の集合」であることを示す．個体の集合を表す$op_{lcp,k}$には，実体ノードへの参照パスを設定する．この参照パスが，フローリンクと実体ノードとの対応関係となる．

PR_{lcp}には，そのライフサイクルプロセスノードlcpでの個体の集合に対する処理方法と評価値の算出方法に関する計算処理を記述する．4.2.2 項で示したように，PR_{lcp}の記述にはプログラミング言語（本ツールでは Visual C#）を用いている．

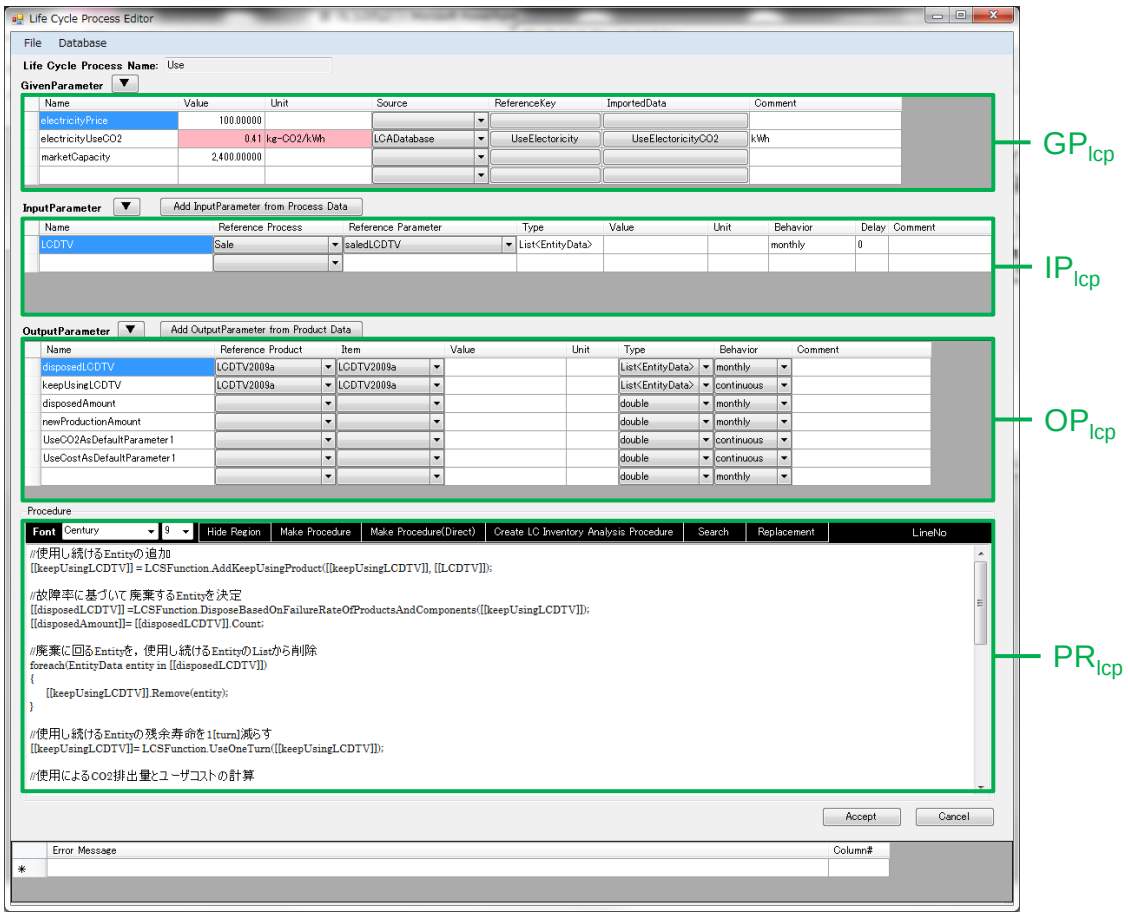


図 6-9：Life Cycle Process Editor のスクリーンショット

6.5.3. Sub Life Cycle Process Editor

Sub Life Cycle Process Editorは，図 6-8 のnth Layerを編集するツールである．Sub Life Cycle Process Editorは，親ライフサイクルプロセスノードlcpのIP_{lcp}とOP_{lcp}を継承する．継承されたすべてのパラメータを子ライフサイクルプロセスノードのいずれかに割り当てることで，親と子の間のパラメータの整合性を保ちながらのモデリングが可能である．

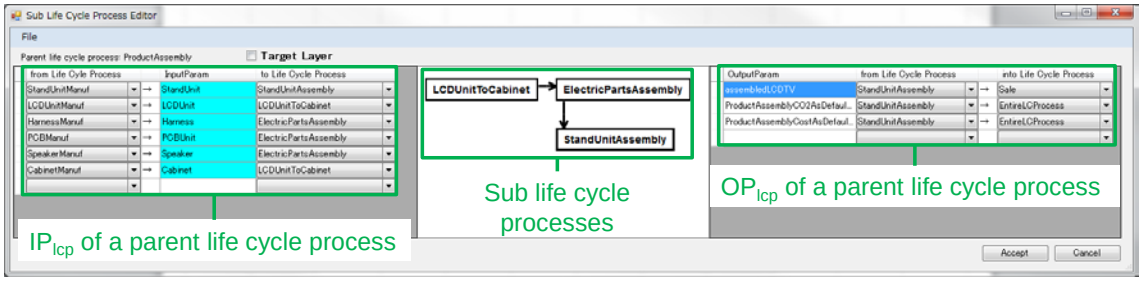


図 6-10：Sub Life Cycle Process Editor のスクリーンショット

6.5.4. Business Model Editor

Business Model Editor は，設計対象である製品の生産計画，およびユーザモデルを編集するツールである（図 6-11 参照）．本ツールで編集した生産計画やユーザモデルは，Procedure の変数として引用可能である．

生産計画は，製品種ごとに設定可能である．具体的には，6.4.1 項の Hierarchical Product Structure Model Editor でモデル作成した製品について，各製品種を各ライフサイクルタイムに何台生産するかを設定する．なお生産計画には，LCS 実行時の生産行動を以下の 2 通りから選択する．

(i) 生産計画で設定した台数に常に従って生産する．

(ii) 市場台数が最大容量以下の時は生産計画で設定した台数に従って生産し，最大容量以上のときは生産台数をゼロとする．ここでの最大容量とは，対象製品を購入する意欲のあるユーザ数を表す．つまり(i)では，購入意欲のあるユーザ数を無限大とみなしている．なお，本ツールでは，購入意欲のあるユーザ数は時間変化せず，そのため生産台数の最大容量は一定値としている．

ユーザモデルは，市場を構成するユーザタイプの情報を表す．各ユーザタイプは，使用頻度と構成人数を属性として保持する．例えば図 6-11 の右側では，Heavy User，Standard User，Light User という 3 つのユーザタイプを想定している．各ユーザタイプの使用頻度と構成人数を，該当の行に記述する．

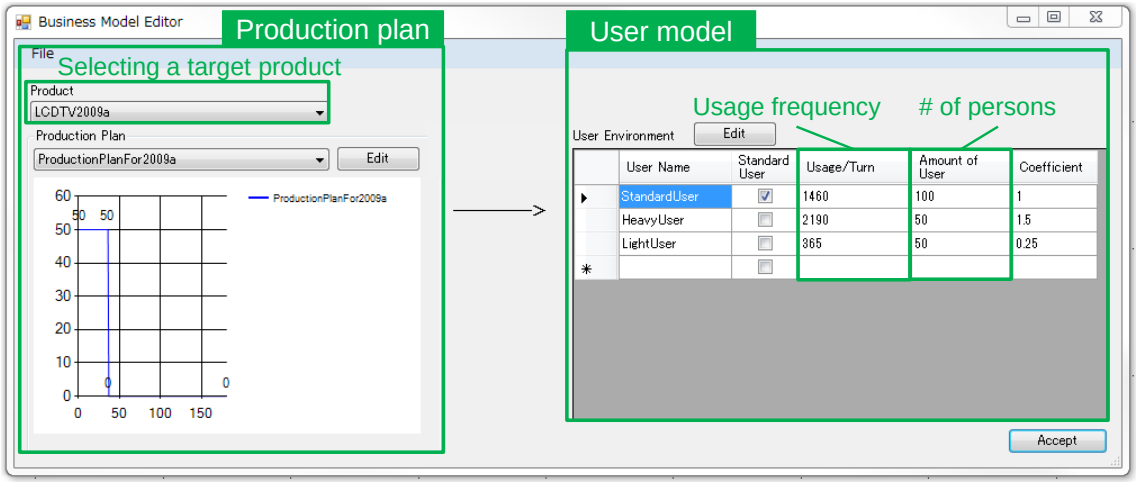


図 6-11：Business Model Editor のスクリーンショット

6.5.5. Life Cycle Inventory Data Manager

Life Cycle Inventory Data Manager は, 既存の LCA ツールや文献などから収集した様々なライフサイクルインベントリデータを Life Cycle Inventory Database に格納し，また Life Cycle Inventory Database からライフサイクルインベントリデータを取得して編集するサブシステムである（図 6-12 参照）．前述したように，本 Life Cycle Inventory Data Manager で編集したライフサイクルインベントリデータは，6.5.2 項の Life Cycle Process Editor の Given parameter に引用可能である．

各ライフサイクルインベントリデータは，(a)対象とするライフサイクルプロセス，(b)対象地域，(c)関連する構成素材，(d)投入資源や排出物名，(e)原単位，(f)単位，(g)排出エリア，(h)データの参照情報，から成る．各ライフサイクルインベントリデータの「(c)関連する構成素材」は，0 項に示した Material Database の素材データから選択する．

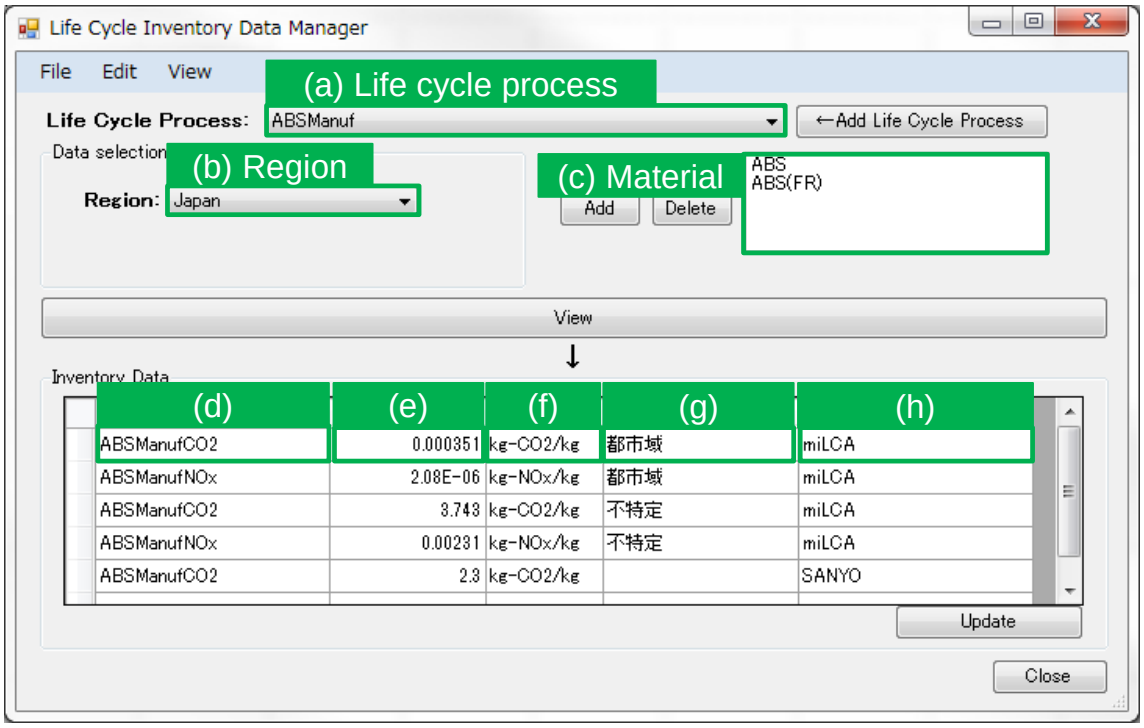


図 6-12：Life Cycle Inventory Data Manager のスクリーンショット

6.5.6. Procedure Description Support System

Procedure Description Support System は，5.1 節の手法に基づいてライフサイクルプロセスノードlcpのPR_{lcp}を生成するサブシステムである．

本サブシステムでは，まず Life Cycle Flow Model Editor に記述されたライフサイクルプロセスノードを，5.1.3 項のアルゴリズムに従ってライフサイクルプロセスタイプに分類する．例えば図 6-7 の Life Cycle Flow Model Editor に記述されたライフサイクルプロセスノードは，図 6-13 のようにライフサイクルプロセスタイプに分類できる．

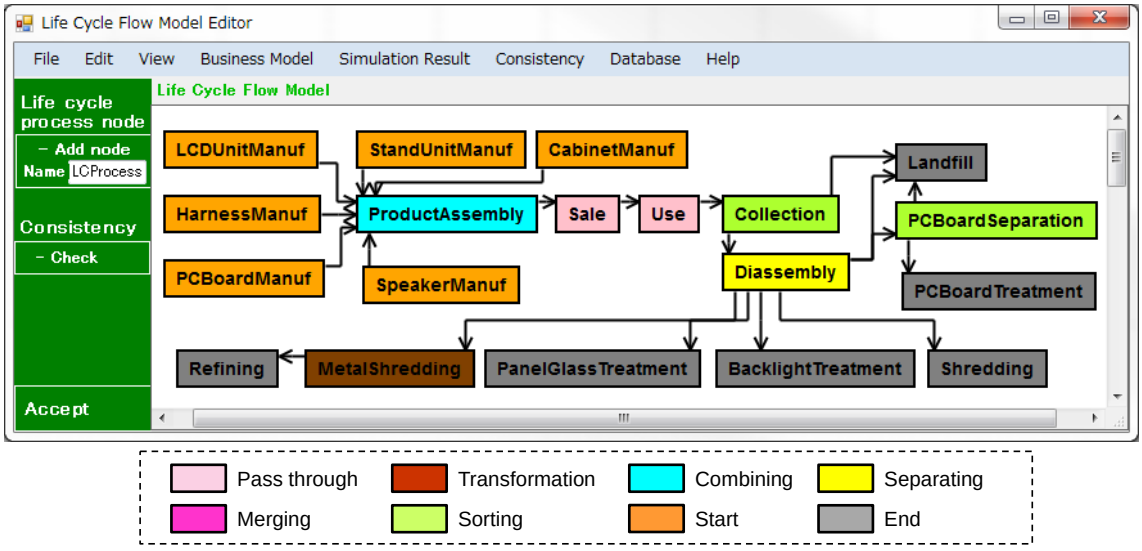


図 6-13：ライフサイクルプロセスノードのライフサイクルプロセスタイプへの分類（対象例：液晶テレビ）

次に，ライフサイクルプロセスノードlcpに関して，以下 2 通りの計算処理を生成できる．

(1) 個体の集合に対する処理方法に関しての計算処理の生成

Procedure Database は，ライフサイクルプロセスタイプごとの Procedure テンプレート（表 5-1 参照）を格納している．ライフサイクルプロセスタイプに分類したライフサイクルプロセスノードに対し，該当する Procedure テンプレートに基づいて，個体の集合に対する処理方法に関しての計算処理を生成できる．

(2) 評価値の算出方法に関しての計算処理の生成

設計者は，図 6-14 のツールを用い，以下の手順で，評価値の算出方法に関しての計算処理を生成できる．

① 評価値を表すパラメータを，OP_{lcp}を格納したリストから選択する．
② 評価値を算出するための計算式を，計算式リスト（表 5-3 参照）から選択する．
③ 選択した計算式に代入するパラメータを，GP_{lcp}，IP_{lcp}，OP_{lcp}を格納したリストから選択する．

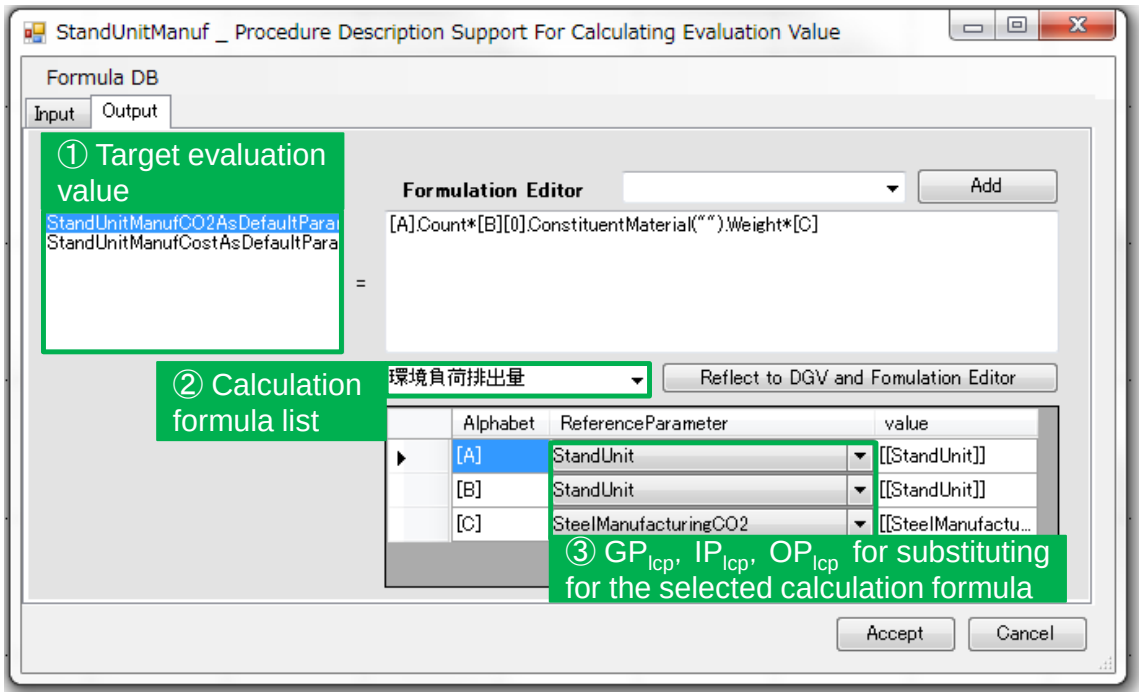


図 6-14：評価値の算出方法に関しての計算処理の生成支援ツール

さらに，Procedure Database は，上記の「ライフサイクルプロセスタイプごとの Procedure テンプレート」と「計算式リスト」に加え，記述頻度が高い計算処理を任意に格納や取得可能なデータリストを保持している．データリストには，例えば表 5-2 の関数（LCSFunction.）を格納している．

6.6. Entity Information Model Manager

Entity Information Model Manager は，LCS の実行過程で生成した個体情報を格納する Entity Information Model Data を用いて，各ライフサイクルプロセスで処理する個体の状態の違いや量の変動が表示可能な State and Amount Distribution Viewer と各個体情報*ei* が表示可能な Entity Information Viewer から成る．

6.6.1. State and Amount Distribution Viewer

State and Amount Distribution Viewer は，Entity Information Model Data から個体情報*ei*を抽出し，各ライフサイクルプロセスにおける個体の状態の違いや量の変動を表示するツールである．本ツールでは個体情報を，以下 4 つの側面から表示できる（図 6-15 参照）．

(1) 量の時間変化：各ライフサイクルプロセスにおける個体数の時間変化
(2) 属性の平均値の時間変化：あるライフサイクルプロセスにおける個体の属性の平均値の時間変化
(3) 属性値の分布：あるライフサイクルプロセスにおける個体のある属性値についての，個体数の分布とライフサイクルタイムごとの内訳
(4) 属性値間の相関関係：あるライフサイクルタイムでの，あるライフサイクルプロセスにおける個体の 2 つの属性値間の関係．各プロットが個体を表す．

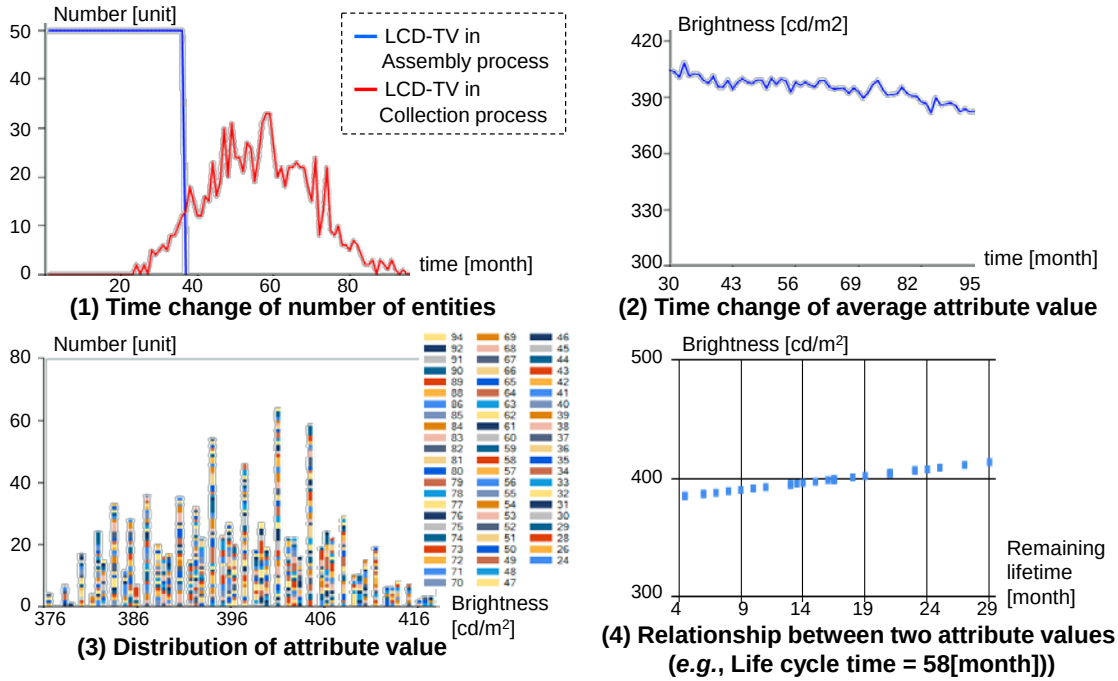


図 6-15 : State and Amount Distribution Viewer が表示する個体の状態の違いと量の変動
（対象例：液晶テレビ）

6.6.2. Entity Information Viewer

Entity Information Viewer は, Entity Information Model Data から特定の個体情報*ei*を抽出して表示するツールである.

表示する個体情報*ei*は，6.6.1 項の State and Amount Distribution Viewer と Entity Information Selector を用いて選択する．まず，State and Amount Distribution Viewer が表示しているグラフから，抽出の対象とする個体情報*ei*の条件を選択する．例えば図 6-16(a)では，回収されたライフサイクルタイムが 58 [month]である製品個体を抽出の条件として選択している．次に，抽出の条件に該当する個体情報の候補を，図 6-16(b)に示すように Entity Information Selector に列挙する．設計者は，この個体情報候補のリストの中から，表示対象とする個体情報*ei*を任意に選択する．

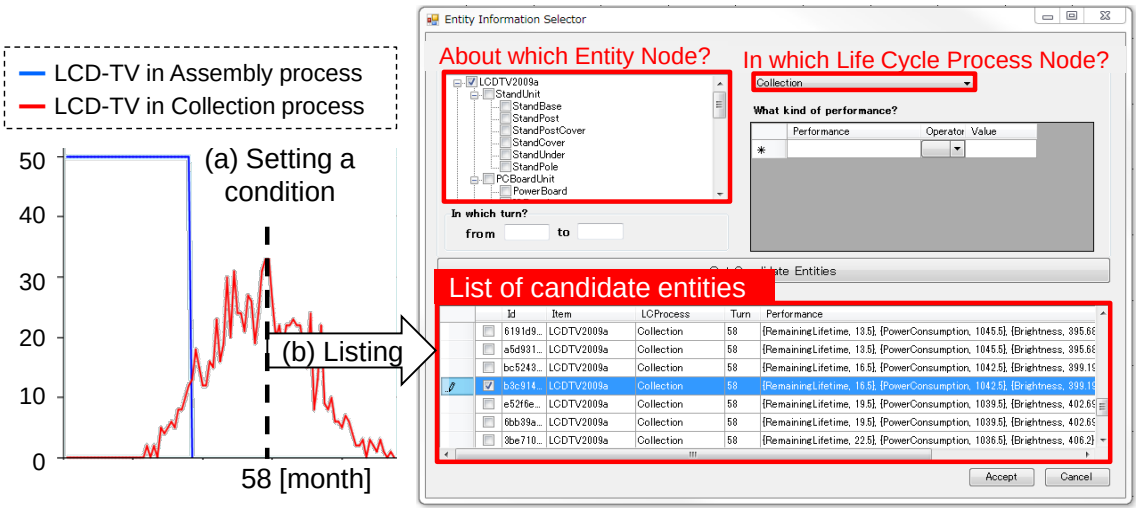


図 6-16：State and Amount Distribution Viewer からの個体情報候補の抽出と Entity Information Selector での表示（対象例：液晶テレビ）

Entity Information Selector で選択した個体情報eiは，4.3.2 項で示した通り，実体ノードとリンクを用いて階層的に表示できる．また，Entity Information Viewer では，各実体ノードが表す個体情報eiの状態sを構成する属性値と「故障しているか否か」が表示できる．さらに，Entity Information Viewer では，Entity Information Model Data 内に存在する同一eidの個体情報eiから「属性値の変化の履歴」「辿ったライフサイクルプロセスノードの履歴」を表示することができる（図 6-17 参照）．

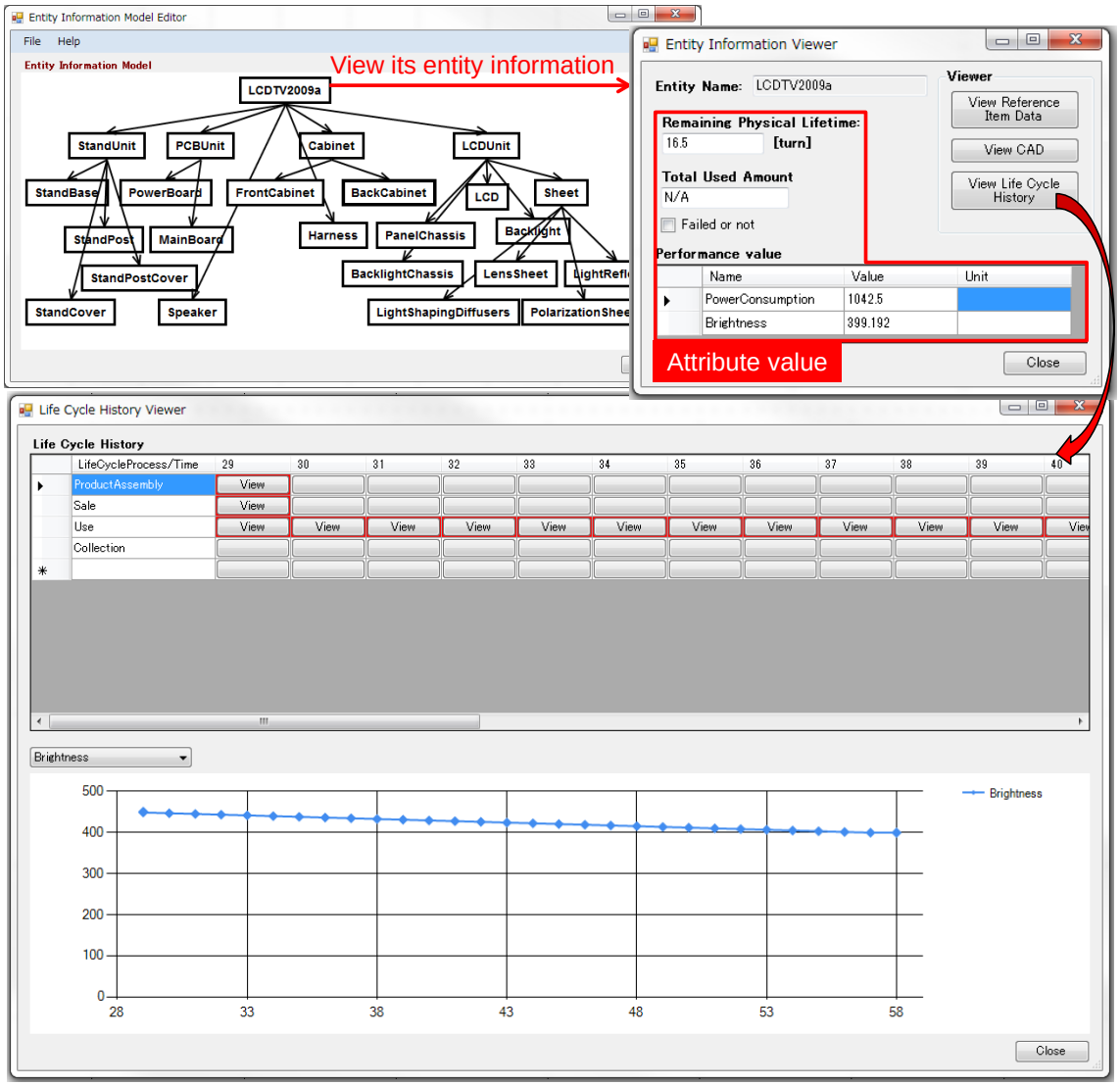


図 6-17：個体の階層構造表現，および状態とライフサイクル履歴の表示
（対象例：液晶テレビ）

6.7. Life Cycle Simulator

Life Cycle Simulator は，6.4 節の Multiple Products Modeling System を用いて作成した製品階層構造モデルと 6.5 節の Life Cycle Flow Modeling System を用いて作成したライフサイクルフローモデルを入力に LCS を実行し，個体情報モデルを作成するサブシステムである．

LCS の実行アルゴリズムは，3.4.2 項で示した手法 [14]に基づく．LCS を実行するために，各ライフサイクルプロセスノードlcpの$ip_{lcp,j}$と$op_{lcp,k}$には，6.5.2 項の Life Cycle Process Editor を用いて，それぞれ「Delay」と「monthly と continuous」について値を設定する．

6.8. Design Modification Support System

Design Modification Support System は，製品ライフサイクルのノミナル情報モデルを構成するパラメータのネットワークを因果関係グラフとして生成する Causal Relation Diagram Creator, 因果関係グラフ上で対象の設計要求に関する評価値に影響を与える設計パラメータを抽出する Related Parameter Searching Tool，抽出した設計パラメータ候補群の感度分析を実施する Sensitivity Analysis Support Tool から成る.

6.8.1. Causal Relation Diagram Creator

Causal Relation Diagram Creator は，6.4 節の Multiple Products Modeling System を用いて作成した製品階層構造モデルと 6.5 節の Life Cycle Flow Modeling System を用いて作成したライフサイクルフローモデルを構成するパラメータ間の関係を, Procedure の記述内容とノード間のリンクに従って特定し，因果関係グラフとして生成するツールである（図 6-18 参照）.

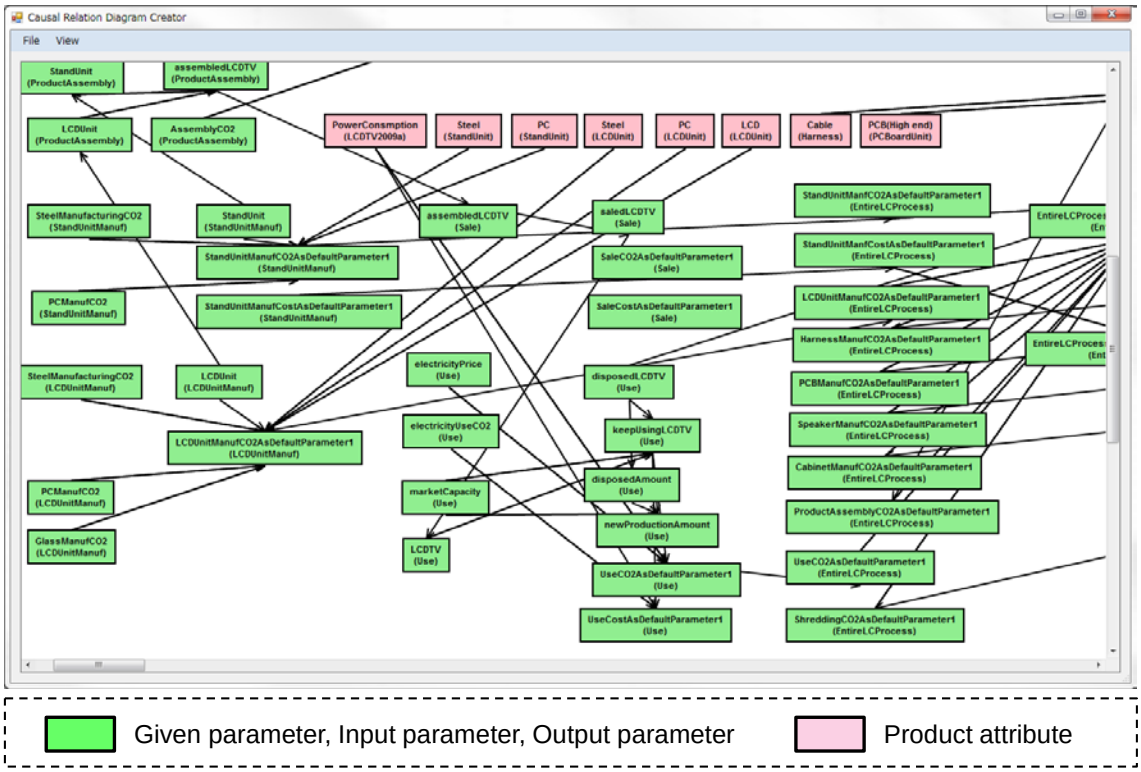


図 6-18：Causal Relation Diagram Creator のスクリーンショット

6.8.2. Related Parameter Searching Tool

Related Parameter Searching Tool は，6.8.1 項の Causal Relation Diagram Creator によって生成した因果関係グラフを用いて，対象の設計要求に関する評価値に影響を与える設計パラメータを探索および表示するツールである（図 6-19 参照）．本ツールでは因果関係グラフを用いて探索したパラメータノードが，ライフサイクルプロセスノードの Given Parameter か製品属性である場合，そのパラメータを修正すべき設計パラメータ候補として抽出する．すべてのパラメータ探索が終了したとき，抽出した設計パラメータ候補をリスト形式で表示する．

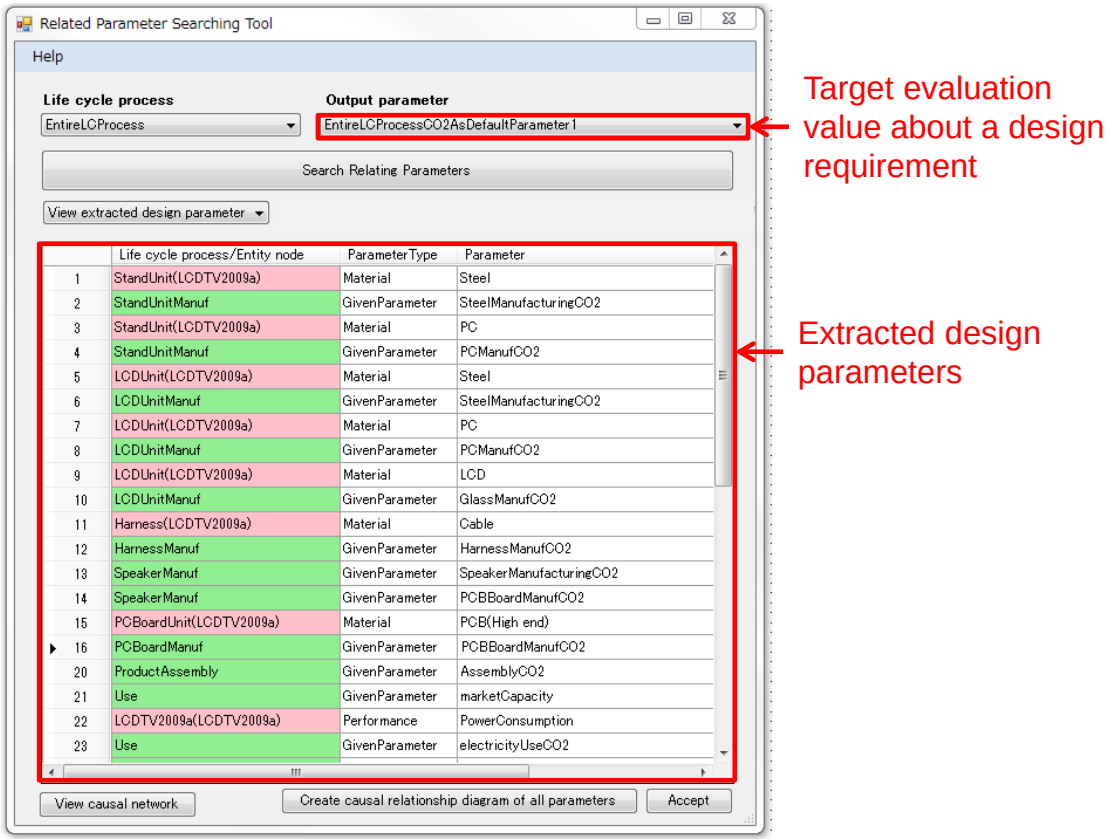


図 6-19：Related Parameter Searching Tool のスクリーンショット

6.8.3. Sensitivity Analysis Support Tool

Sensitivity Analysis Support Tool は，6.8.2 項の Related Parameter Searching Tool を用いて抽出した設計パラメータ候補群の感度分析を実施するツールである．感度分析を実施するために，抽出した設計パラメータ候補それぞれに以下の値を設定する．なお，変動幅は，0 から 1 の間の値を選択するものとし，初期設定は 0.1 である．

- 制御可能か否か
- 最大値：現状のパラメータ値×(1 + 変動幅)
- 最小値：現状のパラメータ値×(1 - 変動幅)
- 水準：2

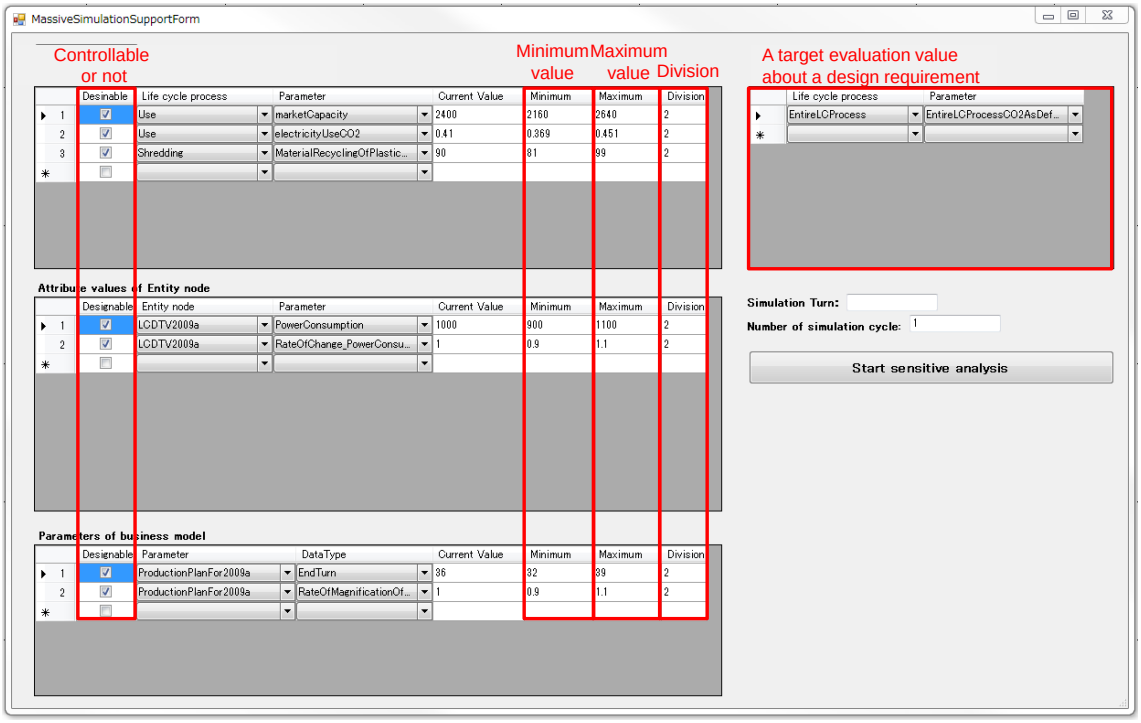


図 6-20：Sensitivity Analysis Support Tool のスクリーンショット

各設計パラメータの最小値と最大値を，それぞれ個別に製品階層構造モデルかライフサイクルフローモデルに入力し，LCS を実行して個体情報モデルを作成する．作成した個体情報モデルそれぞれを用いて，設計要求に関する評価値を算出する．Sensitivity Analysis Support Tool は，感度分析の実行結果を図 6-21 のように表示する．具体的には，各設計パラメータについて最小値と最大値をとった場合それぞれの評価値を，変動幅として棒グラフで表す．つまり，棒グラフの長い設計パラメータが，対象の評価値に対する影響の大きい設計パラメータである．

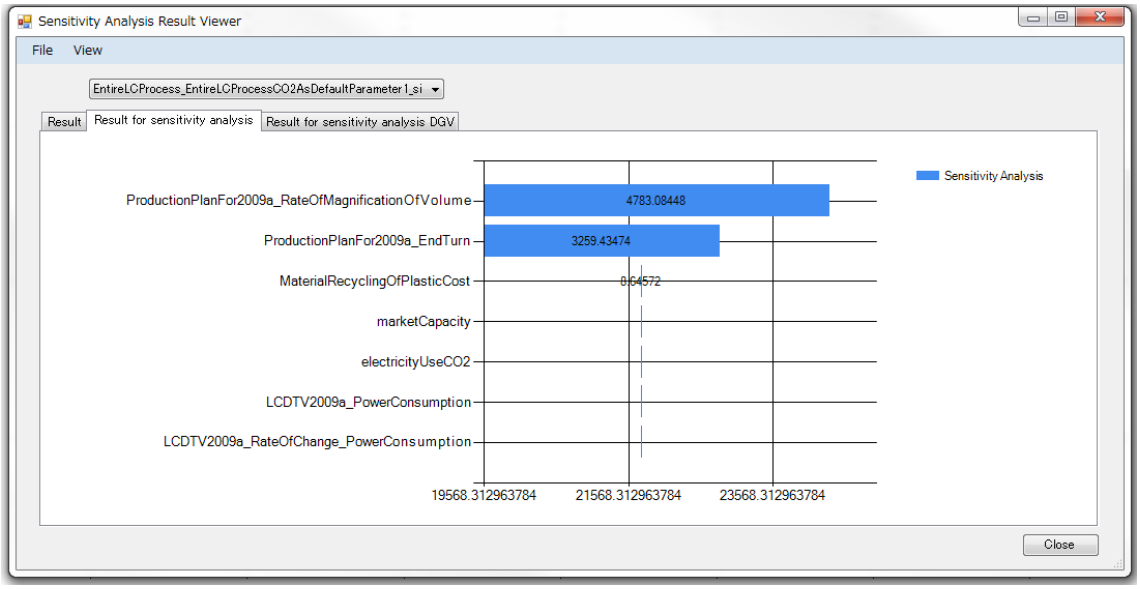


図 6-21：感度分析の実行結果の表示例

第7章 ケーススタディ

第 7 章では，第 4 章と第 5 章で提案した 2 つの手法から成る製品ライフサイクルの設計支援手法の有効性を，第 6 章で示したシステムを用い，スマートフォンを対象としたケーススタディを通じて検証する．

本ケーススタディは，以下の手順で行う．

- 製品ライフサイクルの現状分析
- ライフサイクル戦略と設計要求の設定
- 製品ライフサイクルのノミナル情報モデルを用いた，製品とライフサイクルフローのノミナル情報のモデル作成
- 変動要因の設定
- Procedure の生成
- 個体情報モデルの作成
- 製品ライフサイクルのノミナル情報の修正

7.1. 対象製品

7.1.1. 選定理由

スマートフォンを対象製品とした理由は，下記の 2 点にある．

- 大量生産・大量消費されている電気電子機器であり，スマートフォンの製造業者は経済性を維持しつつも，環境負荷排出量や資源消費量を削減する必要がある．
- 買い替えサイクルが比較的短いため，同一世代もしくは二世代にわたる製品ライフサイクル内で資源を循環できる可能性が高い．しかし，現状では循環型の製品ライフサイクルは構築されておらず，提案した設計支援手法を用いることでその原因やノミナル情報の修正点が特定可能と考えられる．

7.1.2. 製品情報の調査

スマートフォンの部品構成，および各部品の構成素材や幾何形状を明確にするために，市販のスマートフォンを分解調査した．分解対象のスマートフォンとして， Galaxy SⅡを選定した．Galaxy SⅡは，2011 年に発売された Samsung 製のスマートフォンである．

Galaxy SⅡを分解したところ，図 7-1 に示すような部品構成であることが分かった．また，分解調査から得た各部品の構成素材と重量の情報，およびいくつかの文献（例えば，[118] [119]）から得た部品情報を，表 7-1 のように整理した．

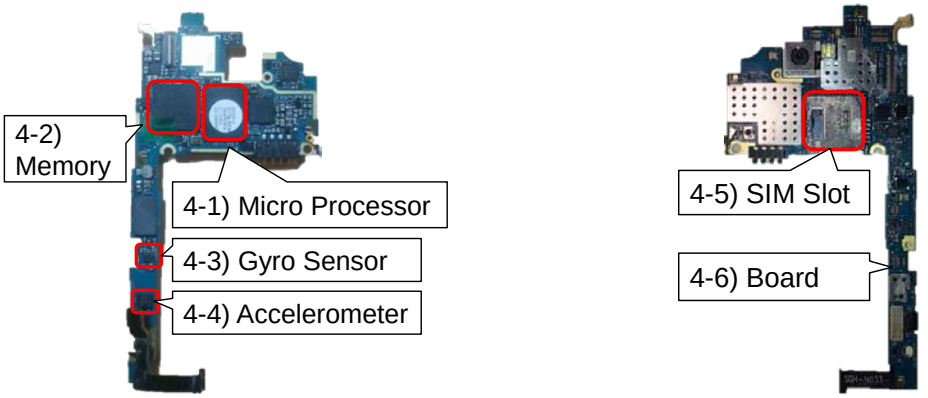
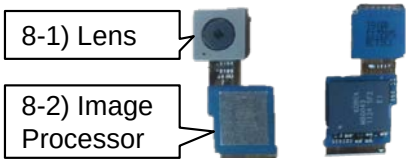
<table>
<tr><td>1) Front Glass
  </td><td colspan="2">2) Organic Light Emitting diode Display (OLED)
  </td><td>3) Center Panel
  </td></tr>
<tr><td colspan="4">4) Main Board
 
</td></tr>
<tr><td>5) Micro USB Board
  </td><td>6) Rear Panel
  </td><td colspan="2">7) Battery Cover
  </td></tr>
<tr><td>8) Rear Facing Camera Unit
 
</td><td>9) Front Facing Camera
  </td><td colspan="2">10) Battery
  </td></tr>
<tr><td>11) Antenna
  </td><td>12) Vibrator
  </td><td>13) Speaker
  </td><td>14) Home Button
  </td></tr>
</table>

図 7-1：Galaxy SⅡの分解調査を通じて特定したスマートフォンの部品構成

表 7-1：分解調査と文献調査を通じて特定した製品情報

	Parts name		Parts information		Reference
1.	Front Glass		Material	{Polycarbonate, 3.32[g]}, {Glass, 27.10[g]}	*
2.	OLED		Material	{OLED, 1.37[g]}	[118], *
			Attribute	{Screen size, 4.3[inch]}	[118]
3.	Center Panel		Material	{Polycarbonate, 5.82[g]}, {Magnesium, 0.05[g]}	*
4.	Main Board		Material	{Gold, 0.03[g]} {Silver, 0.35[g]} {Copper, 16.00[g]} {Palladium, 0.02[g]} {Lead, 0.26[g]}	[119]
	4-1	Micro Processor	Attribute	{Clock speed, 1.2[GHz]}	[118]
	4-2	Memory	Attribute	{Internal memory, 16[GB]}	[118]
	4-3	Gyro Sensor	Function	Sensing directional motions	[118]
	4-4	Accelerometer	Function	Picking up the tilting motion of the phone and enabling to turn the screen when the user tilts the phone.	[118]
	4-5	SIM Slot	Function	To insert SIM/Micro SD	[118]
	4-6	Board	—	—	—
5.	Micro USB Board		Material	{PCBoard, 2.40[g]}	*
6.	Rear Panel		Material	{Polycarbonate, 9.41[g]}	*
7.	Battery Cover		Material	{Polycarbonate, 7.37[g]}	*
8.	Rear Facing Camera Unit (Main Camera Module)		Material	{Camera, 1.20[g]}	*
			Attribute	{Effective pixels, 8.0[million pixels]}	[118]
	8-1	Lens	—	—	—
	8-2	Image Processor	Function	Sensing as CMOS	[120]

9.	Front Facing Camera (Sub-Camera)	Material	{Camera, 0.30[g]}	*
		Attribute	{Effective pixels, 2.0[million pixels]}	[118]
10.	Battery	Material	{Li-ion Battery, 32.60[g]}	*
		Attribute	{Battery capacity, 1650[mAh]}	[118]
11.	Antenna	Material	{Polycarbonate, 0.28[g]}, {Steel, 2.00[g]}	*
12.	Vibrator	Material	{Vibrator, 2.60[g]}	*
13.	Speaker	Material	{Speaker, 3.30[g]}	*
14.	Home Button (Button)	Material	{Polycarbonate, 0.40[g]}	*
Smart phone		Weight	118.18[g]	

*は，分解調査を通じて特定した情報であることを示す.

7.2. ライフサイクル戦略と設計要求の設定

スマートフォンは，2011 年 4 月に日本で施行された小型家電リサイクル法の対象製品である．小型家電リサイクル法の対象製品に指定されている理由は，スマートフォンが希少金属を含むためである．現状の日本では，スマートフォンに含まれる素材について，下記のような EoL 処理を施している [121].

- 金属類（鉄，金，銀，銅，パラジウム）：鉄以外はマテリアルリサイクル．鉄は埋立処理
- プラスチック類：ハンガーなどの日用品，プラスチック収納容器，玩具の筐体などへマテリアルリサイクル
- 金属類の精錬過程で生じるスラグ：路盤材や湾岸施設（テトラポット中込材）などへマテリアルリサイクル

また携帯電話のユーザは，新機種の買い替え後も前機種を退蔵しておくことが多く，そのため携帯電話の回収率は低い．特にスマートフォンは，音楽再生機能やカメラ機能がフィーチャーフォン以上に優れており，この機能性の高さが原因で携帯電話の回収率は低下の傾向をたどっている．具体的には，2010 年の携帯電話の回収率は 37.8[%]であるのに対し，小型家電リサイクル法が 2011 年に施行されたにもかかわらず，2012 年の回収率は 21.7%に留まっている [122].

大量生産品であるスマートフォンの製造業者は，企業利益を維持しつつも，環境負荷排出量や資源消費量を削減する必要がある．しかし，分解調査の結果，スマートフォンの構成部品の大半は，分離が困難な状態で複数の素材が混在しており，貴金属類を含有しない部品のリサイクル効果は小さいと考えられる．そこで本ケーススタディでは，製品組み込みリユース（以下，部品リユース）の実現可能性について検討することとした．

スマートフォンの買い替えサイクルは 2.0 年程度 [123]である．スマートフォンは半年ないし 1 年に 1 度の頻度で新機種が販売されるため，限界リユース率 [84]に基づくと短期・中期パターンに分類できる．そのため，「販売期間の長期化」と「多世代間での製品組み込みリユース」が，リユース性の向上に効果的である [84]．製品開発サイクルが買い替えサイクルの半分程度であるスマートフォンでは，二世代にわたって部品リユースすることが望ましいと考えられる．しかし，世代ごとに形状変更している現状の製品設計では，部品リユースの実現は困難である．以上より本ケーススタディでは，リユース効果の高い部品を次世代機種に部品リユースし，貴金属を含む部品を従来通りリサイクルすることで，経済性が高く，かつ環境負荷排出量と資源消費量を削減するようなスマートフォンの製品ライフサイクルのノミナル情報を探索する．

設計要求の要求項目として，生産するリユース対象部品個体のうちリユースする個体の割合（リユース率と呼ぶ），スマートフォン製造業者がライフサイクル全体で得る利益（以

下，メーカ利益)，およびライフサイクル全体で排出されるCO_2（以下，$LCCO_2$と呼ぶ）を選定した．選定した要求項目の要求値として，リユース率を 40%以上，メーカ利益を現状値以上，$LCCO_2$を現状値以下に設定した．

リユース対象部品としては，7.1.2 項に示した製品情報と上記で設定した設計要求より，Center Panel と Rear Facing Camera Unit（以下，Main Camera Module）という 2 つの部品に着目した．Center Panel は，OLED，Main Board，Battery などを保持する筐体である．分解調査を通じて，主な構成素材がポリカーボネート樹脂（以下，PC 樹脂）であり，またマグネシウム（以下，Mg）を構成素材とした耐熱材や塩化ビニルテープが付着した部品であることがわかった．主構成素材である PC 樹脂をリサイクルするためには，塩化ビニルテープを手作業で取り外し，また破砕して Mg などの素材と選別する必要がある．一方で，図 7-2 に示すように，近年では画面サイズが世代間で統一されつつある．そのため OLED を保持する Center Panel のサイズも，世代間で統一できる可能性が高い．また Center Panel はユーザの目に触れない部品であるため，部品自体の価値寿命が長い．

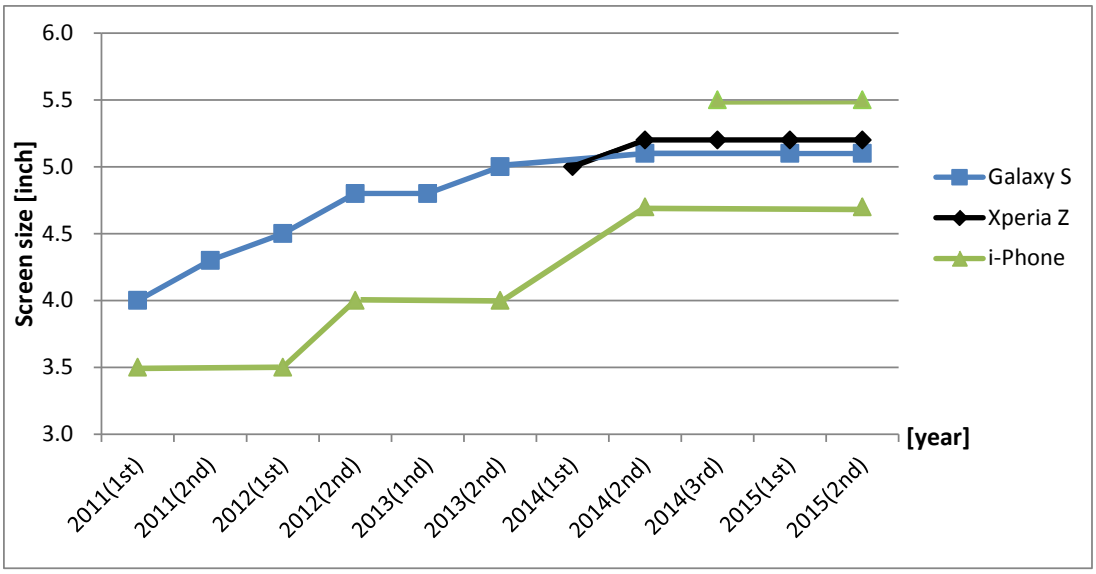


図 7-2：スマートフォンの画面サイズの変遷（ [124]の情報を基に作成）

Main Camera Module は，背面カメラ機能を実現する部品群（Lens, Image Processor, Gyro Sensor, Accelerometer）のうち，Lens と Image Processor を構成部品とする部品群を指す．また Main Camera Module は，OLED，Main Board，Touch Panel に次いで製造コストが高い [125]．Main Camera Module は，年々性能が向上してきたが，近年では世代間での性能はほぼ横ばいである（図 7-3 参照）．その背景に，スマートフォン購入時にカメラ機能を重視するユーザが全体の 36.9%を占める程度であり，購入重視点の順位が 15 位と低い点が挙げられる [126]．そのため，ユーザの約 6 割は，他機種との性能差が大きくない限り，Main Camera Module のリユースを性能面からは受け入れ可能と考えられる．

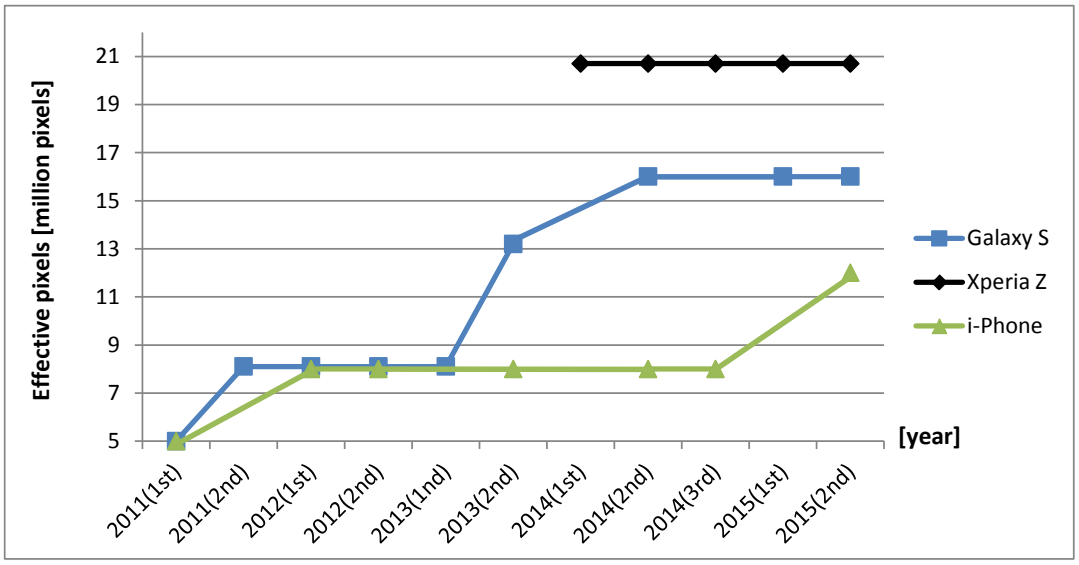


図 7-3：スマートフォンのカメラ性能の変遷（ [124]の情報を基に作成）

Center Panel と Main Camera Module は，ともに多数の素材が混在しており，素材選別に手間がかかるものの，含有する素材価値は高くない．つまり，これら 2 つの部品のリサイクルによる利益は小さいものと予測される．

以上より，Center Panel と Main Camera Module に対して，次世代機種に部品リユースするというライフサイクル･オプションを選択した. Center Panel と Main Camera Module 以外の部品に対するライフサイクル・オプションを，表 7-2 に示すように選択した．リサイクルを選択した部品は，貴金属を含んでいる，もしくは素材選別に手間がかからないことを判断基準とした．

表 7-2：各構成部品に対するライフサイクル・オプションの選択

Life cycle option	Parts
Material Recycling	2) OLED, 4) Main Board, 6) Rear Panel, 7) Battery Cover, 10) Battery, 11) Antenna
Appropriate Disposal	1) Front Glass, 5) Micro USB Board, 9) Front Facing Camera, 12) Vibrator, 13) Speaker
Landfill	14) Button

7.3. 製品ライフサイクルのノミナル情報のモデル作成

7.2 節で策定したライフサイクル戦略を実現する製品とライフサイクルフローのノミナル情報を，製品ライフサイクルのノミナル情報モデルを用いてモデル作成した．作成したスマートフォンの製品階層構造モデルを図 7-4 に，ライフサイクルフローモデルを図 7-5 に示す．また，分解調査を通じて特定した部品形状や配置から，スマートフォンのソリッドモデルを作成し，製品階層構造モデルの実体ノードと接続した．

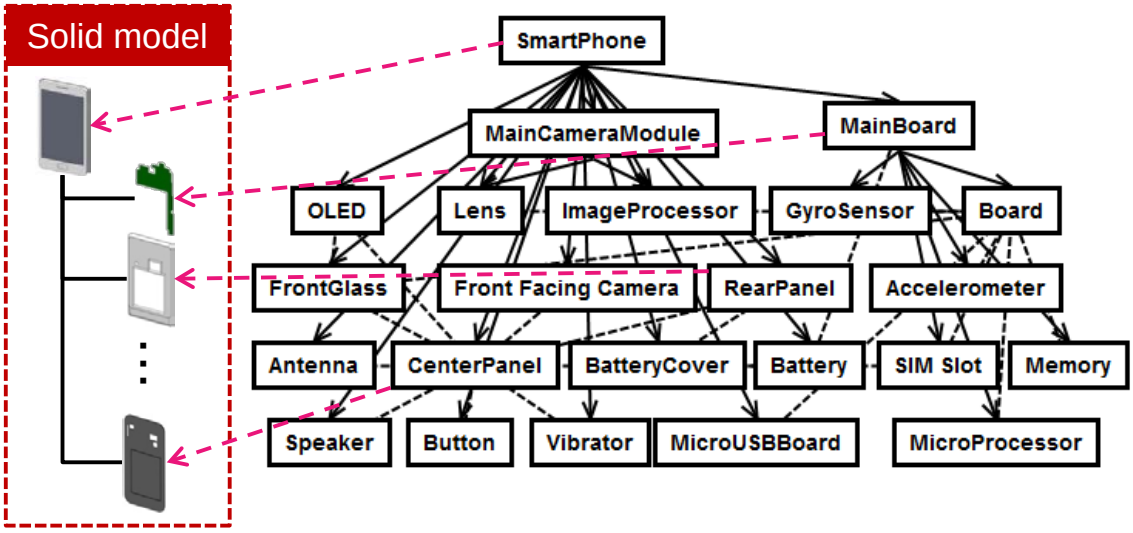


図 7-4：対象とするスマートフォンの製品階層構造モデル

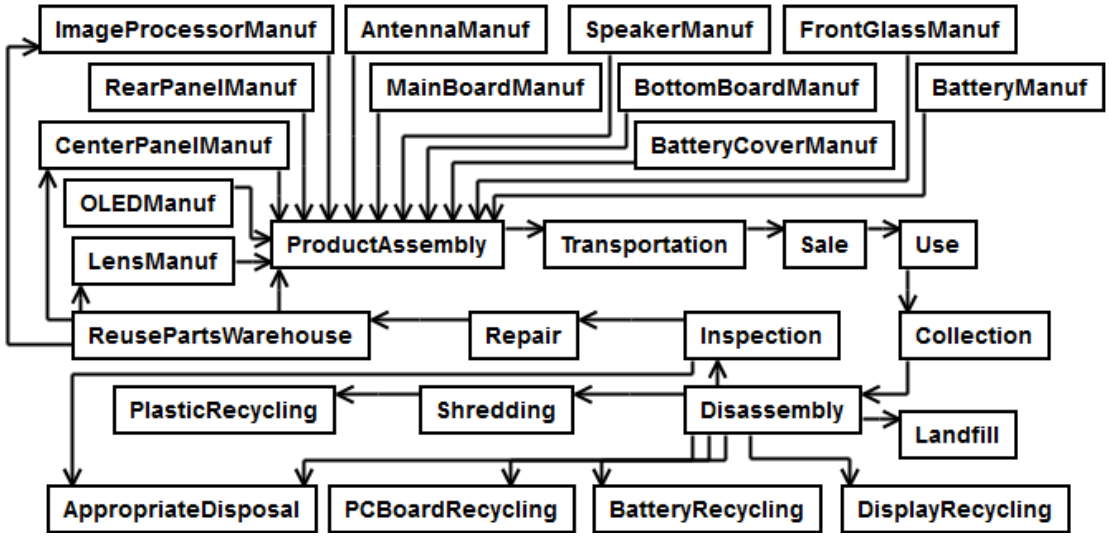


図 7-5：対象とするスマートフォンのライフサイクルフローモデル

7.3.1. 製品階層構造モデルの作成

対象のスマートフォンの製品階層構造モデルの実体ノードには，表 7-1 の部品情報を基に，構成素材とその重量および初期性能値を設定した．なお本ケーススタディでは，一世代目と二世代目の製品構造と属性を同一と仮定してモデル作成した．

7.3.2. ライフサイクルフローモデルのモデル作成

対象のスマートフォンのライフサイクルフローモデルを構成する各ライフサイクルプロセスノードの役割や振る舞いを，以下に示す．

- **Parts Manufacturing（OLEDManuf，MainBoardManuf，BottomBoardManuf，BatteryManuf，SpeakerManuf，FrontGlassManuf，CenterPanelManuf，RearPanelManuf，LensManuf，ImageProcessorManuf，BatteryCoverManuf，AntennaManuf)**

本ライフサイクルプロセスノードは，スマートフォンの構成部品を製造するライフサイクルプロセスを表す．

各部品個体の製造コストを，表 7-3 のように設定した．なお，Galaxy SⅡの製造コストに関する情報は公表されていなかったため，いくつかの部品に関しては，次世代機種である Galaxy SⅢの部品データ [125]や，Galaxy SⅡと同時期に発売されていた Nexus One の部品データ [127]を用いることとした．また本ケーススタディでは，1$を 100 円と換算した．また，簡単のため，部品共通化による部品製造コストの削減効果は考えないものと仮定した．

表 7-3：各部品の部品製造コスト

	Parts name	Manufacturing cost	Reference
1.	Front Glass	2,350[yen/unit]	[127]
2.	OLED	6,500[yen/unit]	[125]
3.	Center Panel	*	**
4.	Main Board	7,620[yen/unit] (Breakdown: 1,750 [yen/unit] (Micro Processor)，2,900[yen/unit] (ROM), 1,450[yen/unit] (Wireless Section), 820[yen/unit] (WLAN), 700[yen/unit] (Power Management))	[125]
5.	Micro USB Board	150[yen/unit]	**
6.	Rear Panel	*	**
7.	Battery Cover	*	**

8.	Main Camera Module	Lens	1,270 [yen/unit]	[125]
		Image Processor	1,520 [yen/unit]	
9.	Front Facing Camera		380 [yen/unit]	[125]
10.	Battery		490 [yen/unit]	[125]
11.	Antenna		2,140 [yen/unit] (Mechanical/Electro-Mechanical)	[125]
12.	Vibrator			
13.	Speaker			
14.	Button			

*は，製造コスト[yen/unit]=素材製造コスト[yen/kg]×素材重量[kg]であることを示す.
**は，仮定した情報であることを示す.

部品製造に伴うその他のコストデータやCO_2排出量に関するライフサイクルインベントリデータを，表 7-4 に示す．なお本ケーススタディでは，電子部品以外の部品加工時のCO_2排出量は，材料採掘から材料加工までのCO_2排出量と比較すると十分に小さいと考え，電子部品以外の部品加工時のCO_2排出量をゼロと仮定した.

表 7-4：ライフサイクルインベントリデータ

Life cycle inventory data	Value	Unit	Reference
Steel manufacturing CO_2	1.98	kg-CO_2/kg	[128]
Magnesium manufacturing CO_2	2.90*10	kg-CO_2/kg	[128]
Silver manufacturing CO_2	5.71*10	kg-CO_2/kg	[128]
Copper manufacturing CO_2	2.69	kg-CO_2/kg	[128]
PC manufacturing CO_2	7.08	kg-CO_2/kg	[128]
Glass manufacturing CO_2	$1.52*10^{-2}$	kg-CO_2/yen	[128]
OLED manufacturing CO_2	3.34*10	kg-CO_2/kg	[128]
Lens manufacturing CO_2	$3.71*10^{-3}$	kg-CO_2/yen	[128]
Speaker manufacturing CO_2	$8.78*10^{-1}$	kg-CO_2/unit	[128]
Li-ion battery manufacturing CO_2	1.32	kg-CO_2/unit	[128]
PCBoard manufacturing CO_2	$3.53*10^{-3}$	kg-CO_2/yen	[128]
IC manufacturing CO_2	$1.71*10^{-3}$	kg-CO_2/yen	[128]
PC manufacturing cost	$2.90*10^{2}$	yen/kg	[129]
Magnesium manufacturing cost	$2.75*10^{2}$	yen/kg	[130]

- **Product Assembly**

Product Assemblyプロセスは，Parts Manufacturingプロセスで製造された部品個体やReuse Parts Warehouseプロセスで保管されていたリユース部品個体を，スマートフォン個体に組み立てるライフサイクルプロセスを表す．スマートフォン 1 個体を組て立る際にかかるコストを，[125]に基づいて 850[yen/unit]とした．なお本ケーススタディでは，部品製造と比較すると製品組立時のCO_2排出量は十分に小さいと考え，製品組立によるCO_2排出量をゼロと仮定した．

- **Transportation**

Transportation プロセスは，Product Assembly プロセスで組み立てられた製品個体を専売店まで輸送するライフサイクルプロセスを表す．なお，輸送による組立から販売への時間遅れはないものと仮定した．

本ケーススタディではSANYOのLCA評価モデル [131]を用いて，総輸送距離d=1,000[km]を 5tトラックで輸送する際に生じるコストやCO_2排出量を算出した．なお，海外から国内への製品輸送による負荷は計上しないものと仮定した．輸送によるCO_2排出量$CO2_{transport}$[kg-CO_2]とコスト$Cost_{transport}$[yen]は，以下の式で表現した [131].

$$CO2_{transport} = lu_{transport} \times CO2_{lu} \tag{7.1}$$

$$Cost_{transport} = lu_{transport} \times Cost_{lu} + Cost_{package} \times N_{transport} \tag{7.2}$$

なお，$CO2_{lu}$は軽油使用によるCO_2排出量原単位（=2.6[kg-CO_2/L] [131]）を，$Cost_{lu}$は軽油使用コスト（=60[yen/L] [131]）を，$Cost_{package}$は製品を梱包するコスト（=600[yen/unit] [125]）を，$N_{transport}$は輸送する製品個体数[unit]を表す．また，輸送における軽油使用量$lu_{transport}$ [L]は，以下の式で表現した [131].

$$lu_{transport} = \frac{w \times N_{transport} \times d}{fc \times lc} \tag{7.3}$$

wは製品 1 個体の重量[kg]を，fcは燃料消費（=6.13 [km/L] [131]）を，lcはトラック容量（=5,000 [kg] [131]）を表す．

- **Sale**

Sale プロセスは，Transportation プロセスから輸送された製品個体をユーザに販売するライフサイクルプロセスを表す.

Galaxy SⅡの発売当初の販売価格は 56,448[yen/unit]であった [132]. 時間経過に伴って販売価格は低下していくものと考えられるが，本ケーススタディでは一律で 50,000[yen/unit]と仮定した．また，リユース部品を組み込んだ製品（以下，リユース部品組み込み製品）の販売価格は，新品製品の 99%と仮定した.

- **Use**

Use プロセスは，Sale プロセスで販売された製品個体をユーザが使用するライフサイクルプロセスを表す.

ユーザはスマートフォンを日々使用する中で，バッテリーを充電する．1 回の充電にかかる電気料金ecp[yen]を，式(7.4)で表現した.

$$ecp = epw \times ecw \tag{7.4}$$

なお，epwはキロワットあたりの電気料金[yen/kWh]を，ecwは 1 回の充電で消費する電力[kWh]を表す．epwは，平成 25 年 7 月現在の東京電力の料金に基づき，$1.89*10^{-2}$[yen/kWh]と仮定した．またecwは，式(7.5)で表現した.

$$ecw = nrp \times cr \times crpd \tag{7.5}$$

nrpはバッテリーの定格電圧 (=3.7[V]) を表す. crはバッテリー容量[mAh]を表す. $crpd$は，1 回に充電する割合を表し，本ケーススタディではバッテリー容量が 20%まで低下した場合に 100%まで充電するものと仮定して，$crpd$=0.8 とした．以上より，バッテリー初期容量時（=1,650[mAh]）の 1 回の充電にかかる電気料金ecpは，約 0.12[yen]となる．この電気料金ecpに，1 日あたりの Battery の充電回数cnを掛け合わせることで，1 ヶ月（ひと月を 30 日と仮定）あたりに充電によって発生する電気料金$epcm$[yen]を算出するものとした.

$$epcm = (30 \times cn) \times ecp \tag{7.6}$$

crとcnは 7.4 節で後述するように，ユーザの使用頻度（表 7-8 参照）に依存して変化する.

- **Collection**

Collection プロセスは，Use プロセスで使用された製品個体が，ユーザの手元を離れて回収されるライフサイクルプロセスを表す.

- **Disassembly**

Disassembly プロセスは, Collection プロセスで回収された製品個体を分解するライフサイクルプロセスを表す．本ライフサイクルプロセスで生じる分解コスト$Cost_{disassembly}$を，式(7.7)のように表現した．

$$Cost_{disassembly} = (T_{disassembly} \times Cost_{labor}) \times N_{disassembly} \quad (7.7)$$

$T_{disassembly}$はスマートフォン 1 個体あたりの分解時間を表し，10.0[min/unit]と仮定した．また，$Cost_{labor}$は人件費を表し，2,000[yen/hour]と仮定した．$N_{disassembly}$は，分解する製品個体数[unit]を表す．

- **Inspection**

Inspection プロセスは，Disassembly プロセスで分解されて取り出されたリユース対象部品がリユース条件を満たしているかを検査するライフサイクルプロセスを表す．リユース条件は，変動要因として設定し，7.4 節で後述する．リユース対象部品 1 個体あたりの検査コストを，300[yen/unit]と仮定した．リユースが不可な部品は，適正処理された後に埋立処理されるものと仮定した．

- **Repair**

Repair プロセスは，Inspection プロセスで検査されてリユース可能とみなされた部品個体を修理するライフサイクルプロセスを表す．ただし本ケーススタディでは，修理による性能改善を考えないものとした．つまり，リユース部品個体を，回収時と同じ性能値でリユースするものとした．

- **Reuse Parts Warehouse**

Reuse Parts Warehouse プロセスは，リユース部品個体を保管するライフサイクルプロセスを表す．本ケーススタディでは，1 か月あたりのスマートフォン 1 個体の管理費を 300 [yen/unit/month]と仮定した．また，在庫中に部品個体は劣化しないものと仮定した．

- **Recycling（PCBoardRecycling，BatteryRecycling，DisplayRecycling，PlasticRecycling)**

本ライフサイクルプロセスノードは，リサイクル対象部品をリサイクル材に還元するライフサイクルプロセスを表す．スマートフォンに含まれる素材の純度や重量は小さいため，リサイクルによるコストやCO_2削減メリットは大きくない. そのため本ケーススタディでは，リサイクルによるメーカ利益や$LCCO_2$への影響をゼロと仮定した．

- **Appropriate Disposal，Landfill**

Appropriate DisposalプロセスやLandfillプロセスは，リユースもリサイクルもされない部品個体を適正処理や埋立処理するライフサイクルプロセスを表す．適正処理や埋立処理によるコストおよびCO_2排出量は発生しないものとした．

- **Entire Life Cycle**

Entire Life Cycleプロセスは，各ライフサイクルプロセスで発生したメーカ利益やCO_2排出量の総和を算出するライフサイクルプロセスを表す．なお本ケーススタディではメーカ利益を，スマートフォン販売による売り上げから，Parts Manufacturing，Product Assembly，Transportation，Sale，Collection，Disassembly，Inspection，Repair，ReusePartsWarehouseプロセスで生じるコストを引いた値として算出した．簡単のため，スマートフォン販売による売り上げがすべてメーカ利益になるものと仮定した．また，本ライフサイクルプロセスノードで，リユース率の計算を行うものとした．

7.4. 変動要因の設定

7.3 節のように作成した製品ライフサイクルのノミナル情報モデルに，変動要因を設定した．設定した変動要因を，変動要因(i)から(iv)に分類した．それぞれを，7.4.1 項から 7.4.4 項に示す．

7.4.1. 変動要因(i)：個体に内在する物理的な性質

(i-a) 故障率

スマートフォンの故障原因は，落下による破損や水没によって生じる突発故障（Accident）と，機能不全による故障（Malfunction）の 2 つに大別される [133]．Square Trade [133] は，スマートフォン購入後に突発故障と機能不全を経験したユーザ割合についての時間推移を，図 7-6 のように報告している．本データは，i-phone 3G を対象に調査されたデータであり，分解調査した Galaxy SⅡとは異なる機種のデータである．しかし，i-Phone と Galaxy シリーズは共通した部品を用いている点から，本データを Galaxy SⅡの故障率データとして採用することとした．

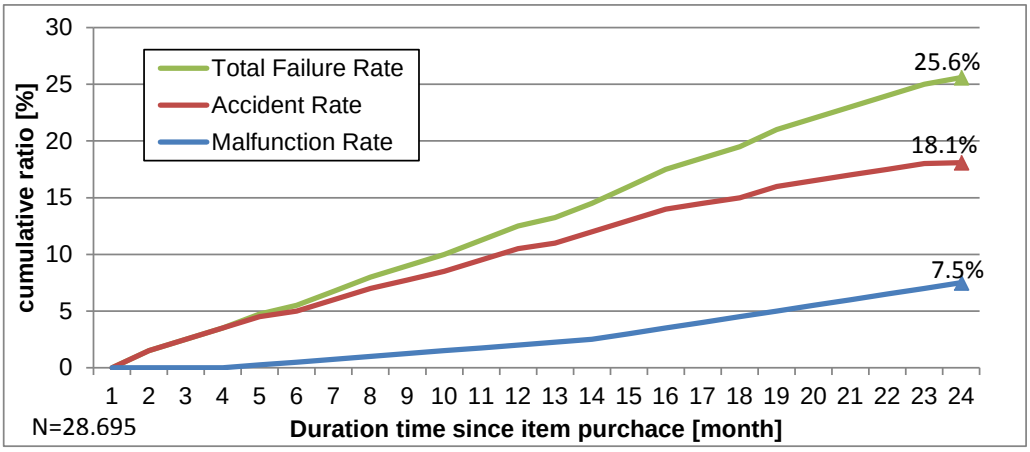


図 7-6：i-Phone 3G の故障を経験したユーザ割合の時間推移 [133]

図 7-6 のユーザ割合の時間推移を基にして，スマートフォン購入後の各故障率の時間推移を算出した．算出結果を，図 7-7 に示す．なお，SquareTrade [133]は，24[month]までの故障率の時間推移のみを報告していた．そのため，図 7-7 の 24[month]以降の故障率は，それ以前の故障率から概算した．また，本ケーススタディでは，突発故障を変動要因(iv)として設定した．さらに，機能不全が発生する部品の割合は，表 7-5 のように報告されている．この割合と図 7-7 を基に，各部品の故障率の時間推移を，図 7-8 に示すように算出した．機能不全は，ユーザの使用頻度に依存して発生するものと考えられるため，横軸をユーザの累積使用時間と置き換えた．図 7-8 の機能不全による故障率を，各部品を表す実体ノードの属性に追加した．なお，設定した故障率は，すべて独立とした．

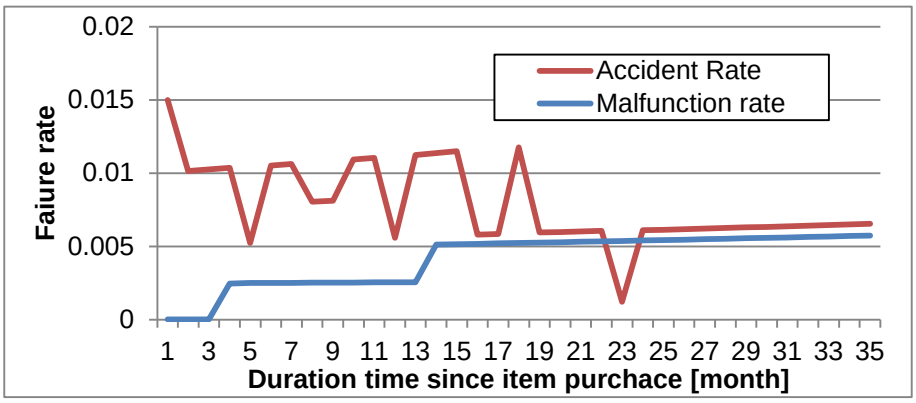


図 7-7：突発故障率の時間推移（ [133]を基に作成）

表 7-5：機能不全を引き起こす部品の割合（ [133]を基に作成）

Parts		Ratio [%]
4-1)	Micro Processor	35.04
2)	OLED	26.00
14)	Button	12.00
10)	Battery	9.50
13)	Speaker	9.00
12)	Vibrator	1.47
6)	Rear Panel	1.47
5)	Micro USB Board	1.26
4-5)	SIM Slot	1.26

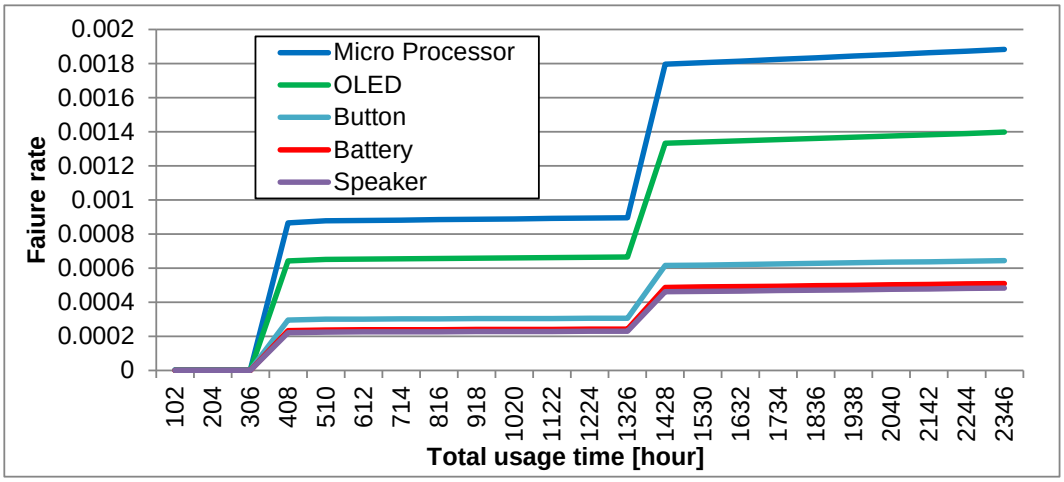


図 7-8：構成部品の機能不全による故障率の時間推移（ [133]を基に作成）

[133]で報告されていない部品の故障率関数$fr(t)$は，6.4.2 項に示した式(6.1)のように，偶発期間故障率$fr_{constant}$，摩耗期間故障率傾き$\Delta_{wear-out}$，物理寿命pl，物理寿命時の故障率fr_{pl}，という 4 つのパラメータを用いて，以下の式で表現した．σは，各スマートフォン個体の累積使用時間[min]を表す．

$$fr(\sigma) = fr_{constant} \qquad \left(if\ 0 < \sigma < pl - \frac{fr_{pl} - fr_{constant}}{\Delta_{wear-out}}\right)$$
$$fr(\sigma) = \Delta_{wear-out} \times (\sigma - pl) + fr_{pl} \quad \left(if\ pl - \frac{fr_{pl} - fr_{constant}}{\Delta_{wear-out}} \leq \sigma\right) \tag{7.8}$$

各部品の$fr_{constant}$，$\Delta_{wear-out}$，pl，fr_{pl}の値を表 7-6 に示す．なお，表 7-6 の値はすべて仮定である．

表 7-6：故障率関数の変数

Parts		$fr_{constant}$	$\Delta_{wear-out}$	pl [min]	fr_{pl}
1)	Front Glass	$1.0*10^{-8}$	0.05	$3.67*10^{5}$	0.1
3)	Center Panel	$1.0*10^{-8}$	0.05	$7.34*10^{5}$	0.1
4-2)	Memory	$1.0*10^{-8}$	0.05	$3.67*10^{5}$	0.1
4-3)	Gyro Sensor	$1.0*10^{-8}$	0.05	$3.67*10^{5}$	0.1
4-4)	Accelerometer	$1.0*10^{-8}$	0.05	$3.67*10^{5}$	0.1
4-6)	Board	$1.0*10^{-8}$	0.05	$3.67*10^{5}$	0.1
7)	Battery Cover	$1.0*10^{-8}$	0.05	$3.67*10^{5}$	0.1
8-1)	Lens	$1.0*10^{-8}$	0.05	$6.12*10^{5}$	0.1
8-2)	Image Processor	$1.0*10^{-8}$	0.05	$6.12*10^{5}$	0.1
9)	Front Facing Camera	$1.0*10^{-8}$	0.05	$6.12*10^{5}$	0.1
11)	Antenna	$1.0*10^{-8}$	0.05	$3.67*10^{5}$	0.1

7.4.2. 変動要因(ii)：主体の能力や要求の違い

(ii-a) 使用頻度の違い

ユーザによってスマートフォンの使用頻度は異なる．本ケーススタディではユーザを，Flipsen らの調査データ [134]に基づいて，Heavy User，Standard User，Light User という 3 つのタイプに分類した（表 7-7 参照）．つまり，使用時間が異なる 3 つの市場が存在するものとみなした．

表 7-7：ユーザタイプの分類 [134]

User type	Usage time [min/day]	Ratio [%]
Heavy User	300	20
Standard User	204	60
Light User	108	20

本ケーススタディでは，ユーザごとの使用頻度の違いによって，リユース対象部品である Center Panel と Main Camera Module, および廃棄要因となり得る Battery の性能劣化を，以下のように表現した．なお，Center Panel と Main Camera Module の代表性能として，「シャルピー衝撃強さ」と「有効画素数」を選定した．

- **Center Panel のシャルピー衝撃強さの劣化関数**

Center Panel は，接している部品（OLED，Main Board，Battery）が使用時に発する熱によって，最高で 90 度近い熱に曝される．そのため使用時間に比例して，機械特性であるシャルピー衝撃強さが劣化するものと仮定した．使用時間に比例してシャルピー衝撃強さが変化することを，ライフサイクルフローモデルのGPを変数とした Center Panel の実体ノードの属性の変化率$\Delta_\alpha(X)$として，以下の式で表現した．

$$\Delta_\alpha(X) = \gamma \times \frac{uf}{uf_s} \tag{7.9}$$

ufは各スマートフォン個体を使用するユーザの使用時間[min/day]，uf_sは Standard User の使用時間[min/day]を表す．γは，Standard User が 100[month]使用した場合に，シャルピー衝撃強さがゼロになるものと仮定し，γ=-0.84[(J/m)/month]とした．なお，シャルピー衝撃強さの初期値cis_0は，PC 樹脂の衝撃強さcis_{pc}と Center Panel の厚みthを用いて，以下の式で表現した．

$$cis_0 = cis_{pc} \times th \tag{7.10}$$

cis_{pc}はシャルピー衝撃試験 [135]によって求められ，cis_{pc}= 84.00 [KJ/m^2] [136]とした．thは，分解調査の結果，th=1.00*10^{-3} [m]であった．

➢ **Main Camera Module の有効画素数の劣化関数**

Main Camera Module の属性のひとつである有効画素数は，撮影した写真の画質に影響する要因のひとつである．スマートフォンの Main Camera Module の撮像素子（CMOS イメージセンサ）は半導体素子であるため，スマートフォン本体の使用によって発生する熱などの影響を受けて劣化が起こり得る．そのため有効画素数がスマートフォン本体の使用時間に比例して減少し，画質を低下させるものと仮定した．その変化を，ライフサイクルフローモデルのGPを用いた Main Camera Module の実体ノードの属性の変化率$\Delta_\alpha{}'(X)$として，以下の式で表現した．

$$\Delta_\alpha{}'(X) = \gamma' \times \frac{uf}{uf_s} \tag{7.11}$$

なおγ'を，γ'=-1.000*10^{-5} [million pixels/month]と仮定した．ufとuf_sは，式(7.9)と同様である．

➢ **バッテリー容量の劣化関数**

各ユーザタイプの 1 日あたりの充電回数N_{bc}を，バッテリー容量が初期値の x[%]のときに必要な充電間隔N_xを用いて，以下の式で表現した．

$$N_{bc} = 1/N_x \tag{7.12}$$

充電間隔N_xは，[134]によるバッテリー容量 100%（初期値）と 70%のときの各ユーザタイプの充電間隔の値（表 7-8 参照）を用い，式(7.13)に従って線形的に減少するものと仮定した．

$$N_x = -\alpha \times (100 - x) + N_{100} \tag{7.13}$$

表 7-8：ユーザタイプごとの充電間隔 [134]

User type	N_{100}：Number of days between recharge	N_{70}：Number of days between recharge	Slope (α)
Heavy User	1.4	0.95	0.011
Standard User	$\frac{1.4 + 2.7}{2}$	$\frac{0.95 + 1.9}{2}$	0.007
Light User	2.7	1.9	0.005

Li-ion バッテリーは，理論的には永久的に機能するが，「充電サイクル」「高温度」「aging」によって，容量が減少する [137]．本ケーススタディでは簡単のため，充電サイクルに

よってのみバッテリーが劣化するものと仮定した．充電サイクルによるバッテリー容量の低下は，Battery University の報告書 [137]に基づいて，図 7-9 のように記述できる．図 7-9 の線形近似より，累積充電回数N_{total}のときのバッテリー容量cr[mAh]を，以下の式で表現した．なお，cr_0はバッテリー容量の初期値（=1,650[mAh]）を表す

$$cr = cr_0 \times (\frac{-1.68}{50} N_{total} + 94.57)/100 \tag{7.14}$$

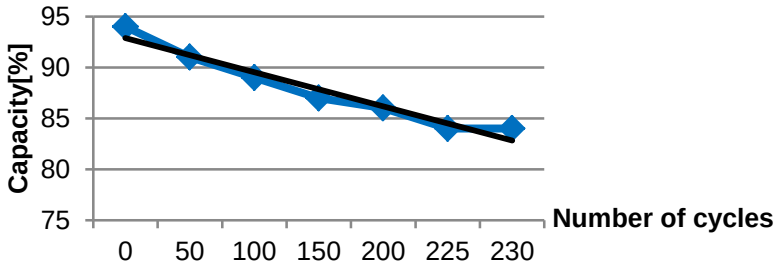


図 7-9：充電サイクルによるバッテリー容量の低下（ [137]を基に作成）

(ii-b) スマートフォンの回収率

7.2 節の分析に基づいて，回収率を 22%とした．そのため，Use プロセスで廃棄判断がなされた製品個体のうち 22%だけが Collection プロセスから Disassembly プロセスに流入するものと仮定した．つまり，回収されない製品個体はユーザによって退蔵され，Collection プロセスに留まるものとみなした．

7.4.3. 変動要因(iii)：個体が辿るフローの分岐条件

(iii-a) 廃棄行動の条件

ユーザが使用中のスマートフォンを廃棄する条件として，以下 3 つの基準に従うものと仮定した．ここでの「廃棄」とは，ユーザが正規の回収ルートに持参する場合（(ii-b)の回収率 22%に該当）と退蔵する場合（(ii-b)の残りの 78%に該当）の両方を包含する行為を指す．

- 故障率に基づいてスマートフォン個体が故障した場合
- 機能の陳腐化や飽きといった価値寿命が原因となる場合
- バッテリー容量の低下が原因となる場合

廃棄条件の一つ目の発生要因は，7.4.1 項の変動要因(i)に示した通りである．なお本ケーススタディでは，構成部品のうち一つでも故障した場合，すべてのユーザが廃棄行動をとるものと仮定した．二つ目は，陳腐化の発生確率の時間推移として，変動要因(iv)に設定する（7.4.4 項の変動要因(iv-b)参照）．三つ目として，1 日に 1 回以上の充電が必要な場合（つまり，式(7.12)のN_{bc}が 1 以上となる場合），ユーザは廃棄行動をとるものと仮定した．

(iii-b) リユース部品の制約条件

リユース部品は，新品部品と同程度の性能を満たす必要がある．この前提に基づいて，リユース対象部品である Center Panel と Main Camera Module のリユース条件を，それぞれ以下のように設定した．

- **Center Panel の部品リユースに対する制約条件**

  新品の Center Panel が満たすべき強度の条件として，高さ 1.5m から落下した場合にスマートフォンに加わる力以上のシャルピー衝撃強さ[J/m]をもつことと仮定した．重量m=0.118[kg]（表 7-1 参照）のスマートフォンが落下した際に加わる力Fは，衝突直前の速度をv，衝突直後の速度v'を$v' = 0$，衝突にかかる時間Δtを$\Delta t = 0.01$[s]としたとき，

$$F = \frac{(mv - mv')}{\Delta t} = \frac{0.118 \times v}{0.01} \tag{7.15}$$

  となる．高さh=1.5m から落下した場合の衝突直前の速度vは，

$$v = \sqrt{2gh} \tag{7.16}$$

  で算出できるので，力Fは，

$$F = \frac{(mv - mv')}{\Delta t} = \frac{0.118 \times \sqrt{2 \times 9.8 \times 1.5}}{0.01} = 63.98[\mathrm{J/m}] \tag{7.17}$$

となる．式(7.17)より，Center Panel のリユース可能条件を，以下の式で表現した．

$$\text{Charpy impact strength} \geq 64.00[\mathrm{J/m}] \tag{7.18}$$

- **Main Camera Module の部品リユースに対する制約条件**

Main Camera Module は，新品部品の 99.98%以上の画質であれば，リユース部品として十分な性能であると仮定した．

$$Effective\ pixels \geq 8.0000[million\ pixels] \times 0.9998 = 7.9984 \tag{7.19}$$

(iii-c) リユース部品組み込み製品のシェア

市場のすべてのユーザがリユース部品を受け入れるわけではない．Main Camera Module に関しては，リユース部品自体を許容しないユーザや高性能カメラをスマートフォンに求めるユーザは少なからず存在する．本ケーススタディでは，二世代目製品のうちリユース部品を組み込んだ製品のシェアを，7.2 節に示したカメラ機能を重視するユーザの割合から，全体の 63%と仮定した．

7.4.4. 変動要因(iv)：処理自体の変化

(iv-a) 生産台数の時間変化

スマートフォンの生産台数は，式(7.20)に従うものと仮定した．ただし，シミュレーションでは，1,000 個体を 1 個体で代表させ，1,000 分の 1 スケールで実行した．

$$
\begin{aligned}
&P_1(t) = 150{,}000[\text{unit}] \quad (if\ M(t) \leq 3{,}000{,}000[\text{unit}], 0 < t \leq 18\ [\text{month}])\\
&P_1(t) = D_1(t-1) \quad (if\ M(t) > 3{,}000{,}000[\text{unit}], 0 < t \leq 18\ [\text{month}])\\
&P_2(t) = 150{,}000[\text{unit}] \quad (if\ M(t) \leq 3{,}000{,}000[\text{unit}], 15 \leq t < 33\ [\text{month}])\\
&P_2(t) = D_2(t-1) \quad (if\ M(t) > 3{,}000{,}000[\text{unit}], 15 \leq t < 33\ [\text{month}])
\end{aligned}
\tag{7.20}
$$

なお，$P_n(t)$はライフサイクルタイム t における n 世代目製品の生産台数，$M(t)$はライフサイクルタイム t における製品の市場台数，$D_n(t)$はライフサイクルタイム t における n 世代目製品の廃棄台数を表す．

(iv-b) 価値寿命による廃棄行動

バッテリー性能の低下を除いた陳腐化が原因でスマートフォンを買い替えるユーザ割合は，各ユーザが製品を購入してからの時間経過に伴って増加すると考えられる．本ケーススタディでは，陳腐化によって廃棄するユーザ割合の時間推移$or(\tau)$を，式(7.21)のように表現した．

$$
or(\tau) = \begin{cases} or_1 & (0 < \tau \leq \tau_1) \\ or_2 \times (\tau - \tau_1) + or_1 & (\tau_1 < \tau\ (\tau \neq \tau_2)) \\ or_3 & (\tau = \tau_2) \end{cases}
\tag{7.21}
$$

τは，各ユーザがスマートフォン個体を購入してから経過したライフサイクルタイム[month]を表す．なお，購入後 1 年は陳腐化による買い替えはほぼ発生せず（or_1=0.0002），1 年後から徐々に増加する（or_2=0.014）ものと仮定し，τ_1=12[month]とした．また，日本では，スマートフォン購入からちょうど 2 年目以外に機種変更すると違約金が発生するという 2 年縛りの契約をしているユーザが大半を占める．そのため，購入から 2 年経過したタイミングでスマートフォンを買い替えるユーザが 30%存在するものと仮定し，τ_2=24[month] or_3=0.3 とした．

7.5. Procedure の生成

6.5.6 項の Procedure Description Support System を用いることで, 作成したライフサイクルフローモデル（図 7-5 参照）を構成するライフサイクルプロセスノードを，ライフサイクルプロセスタイプに分類した（図 7-10 参照）.

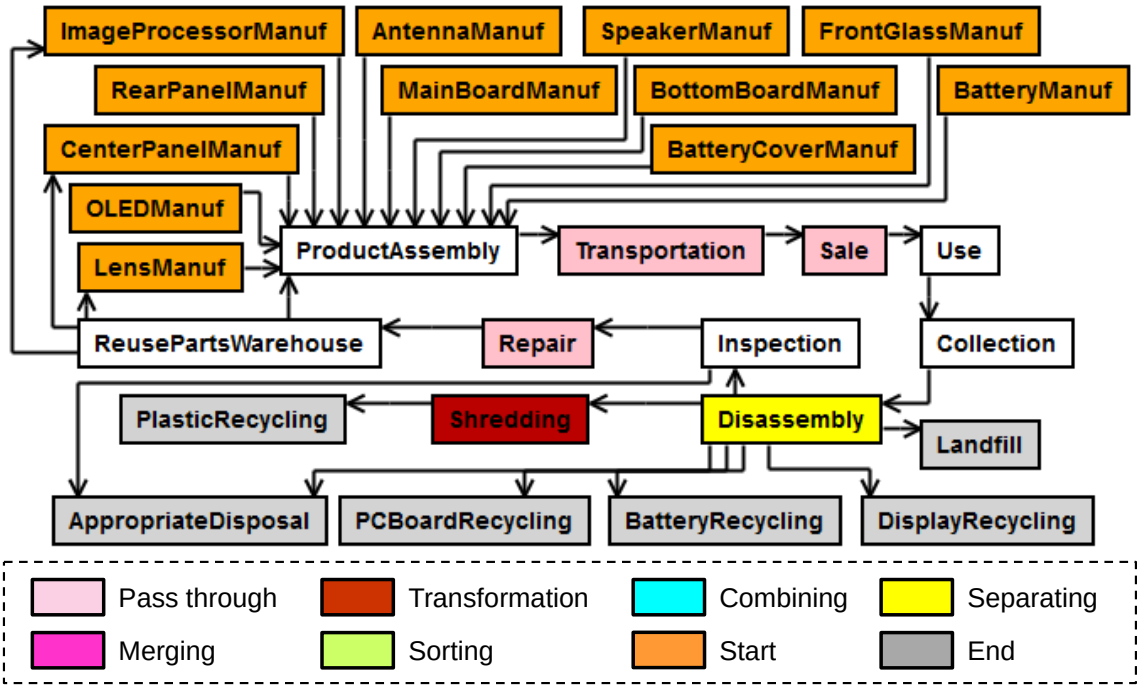
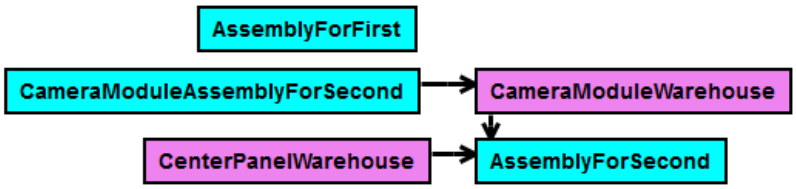
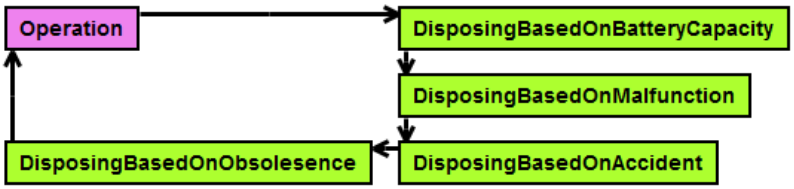
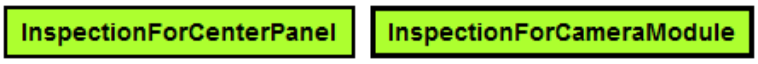
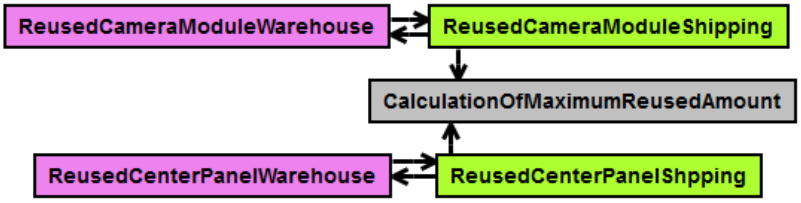


図 7-10：ライフサイクルプロセスノードのライフサイクルプロセスタイプへの分類

このとき，いくつかのライフサイクルプロセスノードは，いずれのライフサイクルプロセスタイプにも分類されなかった. その理由として, 例えば ProductAssembly プロセスを, 新品部品とリユース部品を収集するという「合流」と収集した部品から製品を組み立てる「組立」という 2 種類の処理方法が混在したライフサイクルプロセスノードとしてモデル作成していたことが挙げられる．そこで，ライフサイクルプロセスタイプに分類されなかったライフサイクルプロセスノードを，分類可能な粒度まで詳細化した（表 7-9 参照）.

表 7-9：詳細化したライフサイクルプロセスノード

Original Life Cycle Process Node	Detailed Life Cycle Process Nodes
ProductAssembly	
Use	
Collection	
Inspection	
ReusePartsWarehouse	

ライフサイクルプロセスタイプに分類した結果，各ライフサイクルプロセスノードlcpの個体の集合に対する処理方法に関しての計算処理を，以下のように生成することができた．

(a) 通過

該当したライフサイクルプロセスノード：Transportation，Sale，Repair

```
//通過プロセスと判別されました
[[transportedSmartPhone]].AddRange([[assembledSmartPhone]]);
[[transportedSmartPhoneSecond]].AddRange([[assembledSmartPhoneSecond]]);
```

図 7-11：「(a)通過」に分類されたライフサイクルプロセスノードの Procedure 生成例（Transportation プロセス）

(b) 変換

該当したライフサイクルプロセスノード：Shredding

```
//変換プロセスと判別されました
foreach(EntityData entity in [[BatteryCover]])
{
    [[PlasticFragment]].Add(LCSFunction.Transform(entity, “PlasticFragment"));
}
```

図 7-12：「(b)変換」に分類されたライフサイクルプロセスノードの Procedure 生成例（Shredding プロセス）

(c) 結合

該当したライフサイクルプロセスノード：AssemblyForFirst，AssemblyForSecond，CameraModuleAssemblyForSecond

```
//結合プロセスと判別されました
for(int i = 0 ; i < [[CenterPanel]].Count; i++ )
{
   List<EntityData> componentList = new List<EntityData>();

    componentList.Add([[CenterPanel]][i]);
    componentList.Add([[Antenna]][i]);
    componentList.Add([[Battery]][i]);
    componentList.Add([[MainBoard]][i]);
    componentList.Add([[RearPanel]][i]);
    componentList.Add([[FrontGlass]][i]);
    componentList.Add([[MicroUSBBoard]][i]);
    componentList.Add([[BatteryCover]][i]);
    componentList.Add([[OLED]][i]);
    componentList.Add([[Speaker]][i]);
    componentList.Add([[Lens]][i]);
    componentList.Add([[ImageProcessor]][i]);
    componentList.Add([[Button]][i]);
    componentList.Add([[FrontFacingCamera]][i]);
    componentList.Add([[Vibrator]][i]);
    [[SmartPhone]].Add(LCSFunction.Assembly(componentList,"SmartPhone"));
}
```

図 7-13：「(c)結合」に分類されたライフサイクルプロセスノードの Procedure 生成例（AssemblyForFirst プロセス）

(d) 分離

該当したライフサイクルプロセスノード：Disassembly

```
//分離プロセスと判別されました
foreach(EntityData entity in [[collectedSmartPhone]])
{
    foreach(EntityData childEntity in entity.ChildEntityList.Values)
    {
      if(childEntity.NominalInformation.Name=="CenterPanel")
      {
        [[CenterPanel]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="RearPanel")
      {
        [[RearPanel]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="MicroUSBBoard")
      {
        [[MicroUSBBoard]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Battery")
      {
        [[Battery]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="MainCameraModule")
      {
        [[CameraModule]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="MainBoard")
      {
        [[MainBoard]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="BatteryCover")
      {
        [[BatteryCover]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="OLED")
      {
        [[OLED]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="FrontGlass")
      {
        [[FrontGlass]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Antenna")
      {
        [[Antenna]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="FrontFacingCamera")
      {
        [[FrontFacingCamera]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Speaker")
      {
        [[Speaker]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Vibrator")
      {
        [[Vibrator]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Button")
      {
        [[Button]].Add(childEntity);
      }
    }
}
```

図 7-14：「(d)分離」に分類されたライフサイクルプロセスノードの Procedure 生成例（Disassembly プロセス）

(e) 合流

該当したライフサイクルプロセスノード: Operation，CollectionOfDisposingProducts，CenterPanelWarehouse，CameraModuleWarehouse，ReusedCenterPanelWarehouse，ReusedCameraModuleWarehouse

```
//合流プロセスと判別されました
[[CenterPanelSecond]].AddRange([[NewCenterPanelSecond]]);
[[CenterPanelSecond]].AddRange([[UsedCenterPanel]]);
```

図 7-15：「(e)合流」に分類されたライフサイクルプロセスノードの Procedure 生成例（CenterPanelWarehouse プロセス）

(f) 分岐

該当したライフサイクルプロセスノードは，DisposingBasedOnBatteryCapacity，DisposingBasedOnMalfunction，DisposingBasedOnAccident，DisposingBasedOnObsolescence，CollectedOrUncollected，InspectionForCenterPanel，InspectionForCameraModule，ReusedCenterPanelShpping，ReusedCameraModuleShipping の 9 つである．「(f)分岐」に分類されたライフサイクルプロセスノードの Procedure 生成においては，設計者自身が分岐条件の閾値と条件判定するパラメータを選択する必要があった．例として，InspectionForCenterPanel プロセスでの Center Panel 個体の集合に対する処理方法に関しての計算処理の生成を図 7-16 に示す．$PR_{InspectionForCenterPanel}$の生成においては，7.4.3 項の変動要因(iii-b)に基づき，Center Panel の性能値のひとつであるシャルピー衝撃強さを条件判定するパラメータとし，また$GP_{InspectionForCenterPanel}$の要素である ReusableConditionForCenterPanel（=64.00[J/m]）を分岐の閾値として，設計者自身で設定した．

```
//分岐プロセスと判別されました
foreach(EntityData entity in [[CenterPanel]])
{
  if(entity.IndividualPerformanceList["CharpyImpactStrength"].Invariable > [[ReusableConditionForCenterPanel]])
  {
    [[ReusableCenterPanel]].Add(entity);
  }
  else
  {
    [[UnreusableCenterPanel]].Add(entity);
  }
}
```

図 7-16：「(f)分岐」に分類されたライフサイクルプロセスノードの Procedure 生成例（InspectionForCenterPanel プロセス）

(g) 開始

該当したライフサイクルプロセスノードは，OLEDManuf，MainBoardManuf，BottomBoardManuf，BatteryManuf，SpeakerManuf，FrontGlassManuf，CenterPanel Manuf，RearPanelManuf，LensManuf，ImageProcessorManuf，BatteryCoverManuf，AntennaManufである．ただし，一世代目製品の構成部品の生産計画として，式(7.20)の$P_1(t)$を表す”Plan1”を，設計者が対応付ける必要があった．同様に，二世代目製品の構成部品の生産計画として，式(7.20)の$P_2(t)$を表す”Plan2”を対応付ける必要があった．

```
//開始プロセスと判別されました
[[FrontGlass]].AddRange
(LCSFunction.MakeComponentBasedOnProductionPlan("FrontGlass","Plan1",LCSimulator.CurrentTime));
[[FrontGlassSecond]].AddRange
(LCSFunction.MakeComponentBasedOnProductionPlan("FrontGlassSecond","Plan2",LCSimulator.CurrentTime));
```

図 7-17：「(g)開始」に分類されたライフサイクルプロセスノードの Procedure 生成例（FrontGlassManuf プロセス）

7.6. 個体情報モデルの作成

各ライフサイクルプロセスノードlcpでのメーカ利益とLCCO$_2$を算出するために必要な計算処理を，7.5 節で生成したPR_{lcp}に追記した．各lcpのGP_{lcp}，IP_{lcp}，OP_{lcp}，PR_{lcp}は，付録を参照されたい．

以上に示した製品ライフサイクルのノミナル情報を入力に，個体情報モデルを作成した．個体情報モデルの対象期間は，一世代目機種の製品個体が使用済みとなって回収され得る十分な期間を想定し，60[month]と設定した．この対象期間で，単位時間ステップ 1[month]として LCS を実行し，個体情報モデルを作成した．

作成した個体情報モデルを用いることで，組み立てられた一世代目スマートフォン個体と二世代目スマートフォン個体，および回収された Center Panel 個体と Main Camera Module 個体数の時間変化を表現できた（図 7-18 参照）．回収されたスマートフォン個体と Main Camera Module 個体は同数であるため，図 7-18 における 2 つのグラフは重なりを示している．

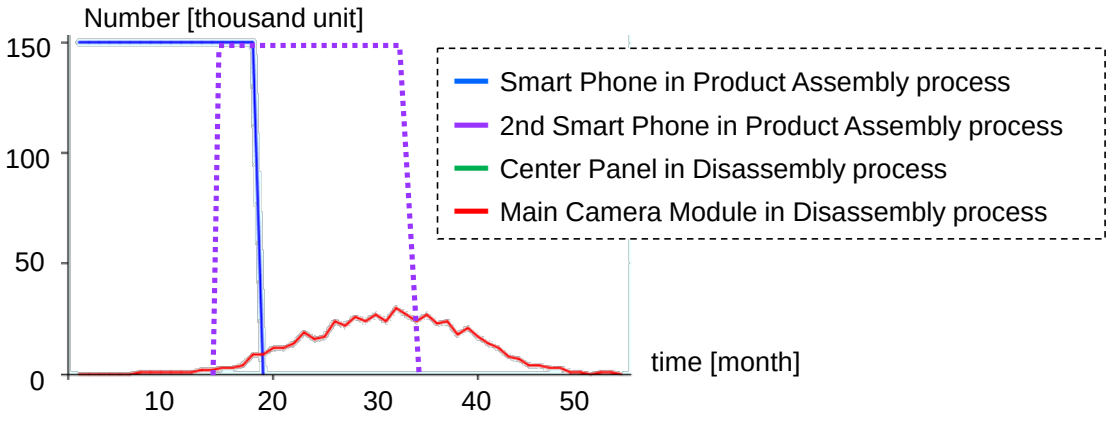


図 7-18：生産台数と回収台数の時間変化（回収率：22%）

図 7-18 のグラフ作例に利用した個体情報のうち，二世代目機種の製造期間終了（t =32[month]）前までに回収された Center Panel 個体と Main Camera Module 個体の個体情報を抽出することで，Center Panel のシャルピー衝撃強さと Main Camera Module の有効画素数は，それぞれ図 7-19 と図 7-20 のように違いを示すことがわかった．なお，図 7-19 と図 7-20 での積み上げ棒グラフの色は，各個体が回収されたライフサイクルタイムを示している．

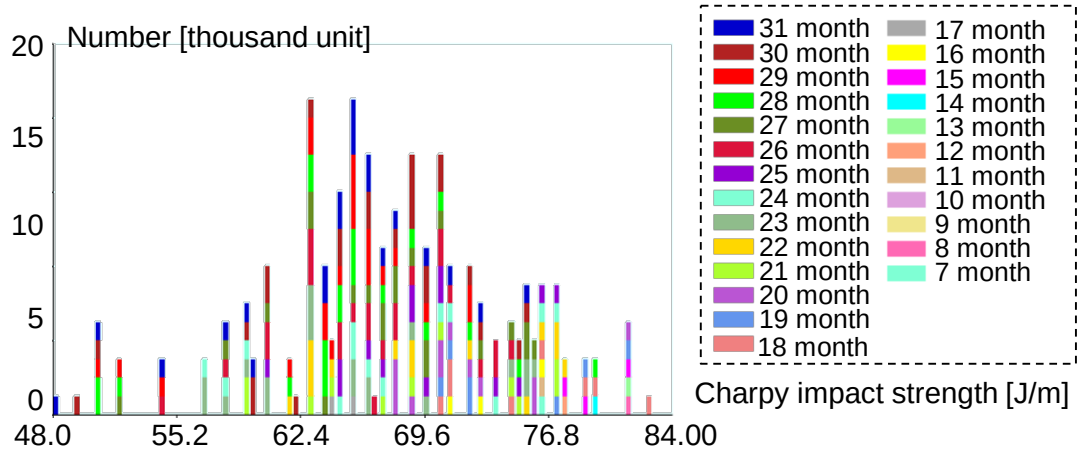


図 7-19：回収された Center Panel 個体の状態の違い（回収率：22%）

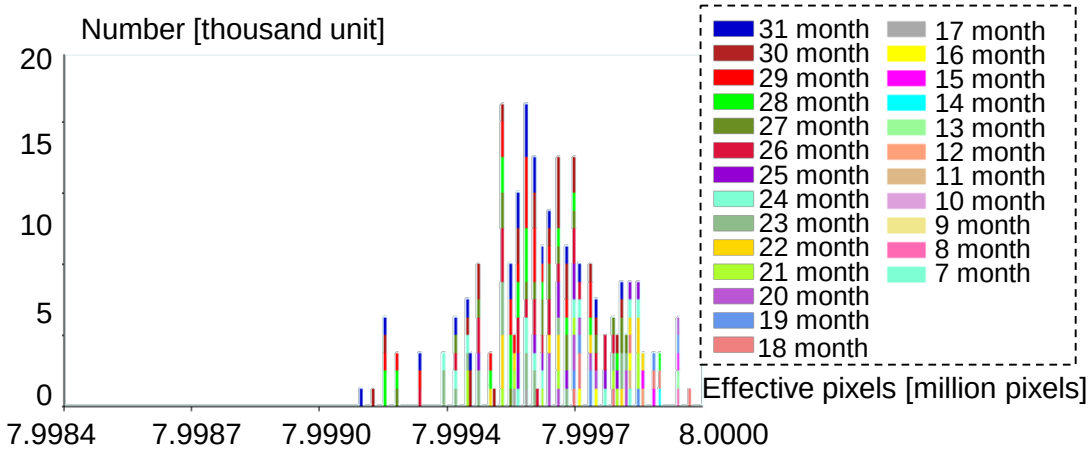


図 7-20：回収された Main Camera Module 個体の状態の違い（回収率：22%）

作成した個体情報モデルから，設計要求項目である「Center Panelのリユース率」「Main Camera Moduleのリユース率」「メーカ総利益（現状に対する比率）」「$LCCO_2$（現状に対する比率）」を算出した．その結果，それぞれ 7.8%，10.3%，100.3%，99.5%であることがわかった．なお，現状のリサイクルシナリオに関して，個体情報モデルを作成し，メーカ総利益と$LCCO_2$を求めたところ，$1.64*10^8$[yen]と $2.46*10^5$[kg-CO_2]であった．

上記の結果に基づくと，導出したノミナル情報の初期解は，リユース率が要求値を大幅に下回っていることがわかる．その原因のひとつとして，図 7-18 より，二世代目スマートフォン個体の製造台数に比べて Center Panel 個体や Main Camera Module 個体の回収台数の少なさがあると推定できる．現状の回収率は 22%であるため，この改善が必要と考えられる．回収率を改善するためには，例えばユーザの新機種購入時に製造業者が一定額を支払って，前機種を回収するという有償での製品引取制度の導入が考えられる．そこで，現状の回収率 22%以上の製品個体を回収する際には，メーカがユーザに対して引取価格

2,000[yen/unit]を支払うものとした．なお本ケーススタディでは，引取価格導入後の回収率を 85%と仮定した．回収率を現状より大幅に高く設定した理由は，スマートフォンの部品リユースがビジネスとして成立するために最低限必要な回収率を求めるためである．

回収率 85%として製品ライフサイクルのノミナル情報モデルを変更し，再度個体情報モデルを作成した．作成した個体情報モデルから，組み立てられた一世代目スマートフォン個体と二世代目スマートフォン個体，および回収されたCenter Panel個体とMain Camera Module個体数の時間変化を，図 7-21 に示すように表現することができた．また，そのうち二世代目機種の製造期間終了前（t=32[month]）までに回収された個体の状態の違いを，図 7-22 と図 7-23 に示すように表現することができた．なお，図 7-22 と図 7-23 での積み上げ棒グラフの色は，図 7-21 の棒グラフに対応しており，各個体が回収されたライフサイクルタイムを示している．このとき，設計要求項目である「Center Panelのリユース率」「Main Camera Moduleのリユース率」「メーカ利益（現状に対する比率）」「$LCCO_2$（現状に対する比率）」が，それぞれ 33.3%, 41.6%, 100.2%, 97.8%であることがわかった．図 7-23 に示すように，Main Camera Moduleはリユース可能な個体が 100%であるにもかかわらず，リユース率が 41.6%であった．その理由として，二世代目スマートフォン製造時期終了後に回収されている個体（図 7-21 の棒グラフの右側部分）を多く含むことが要因として考えられる．つまり，限界リユース率 [84]として議論されているように，製造時期に対する回収時期の遅れによって，約 60%の部品個体がリユースされない可能性がある．ここで，Main Camera Moduleのリユース率は要求値を満たしていたものの，Center Panelのリユース率の評価結果が 40%以下であり，依然として設定した要求値を満たしていなかった．2 つのリユース対象部品の生産台数と回収台数は同じであるため，リユース率の違いは図 7-22 と図 7-23 より，リユース条件以下の部品個体を含むことに原因があると考えられる．具体的な原因と，その問題を解消する設計パラメータを，5.2 節の手法を用いて抽出した．抽出した設計パラメータについて，次節で示す．

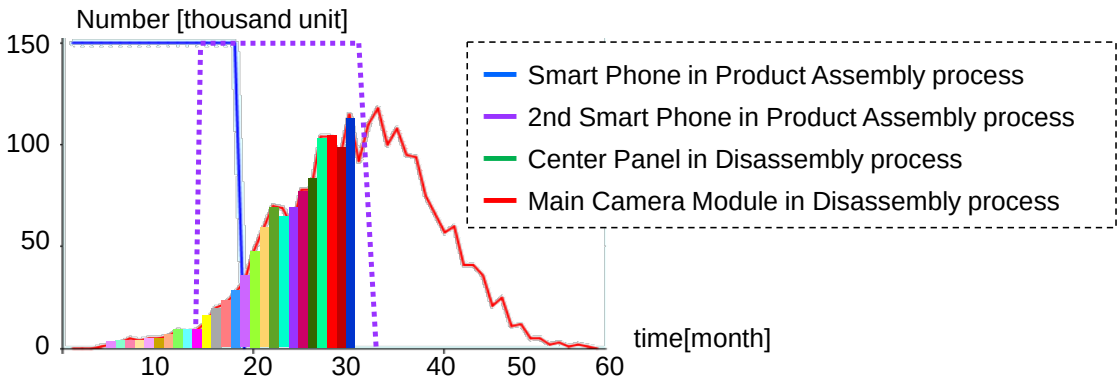


図 7-21：生産台数と回収台数の時間変化（回収率：85%）

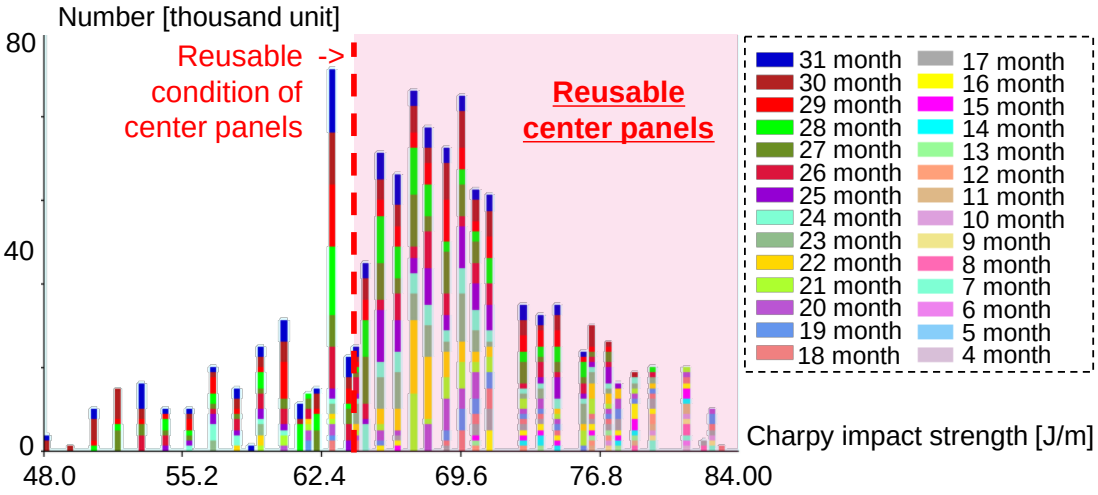


図 7-22：回収された Center Panel 個体の状態の違い（回収率：85%）

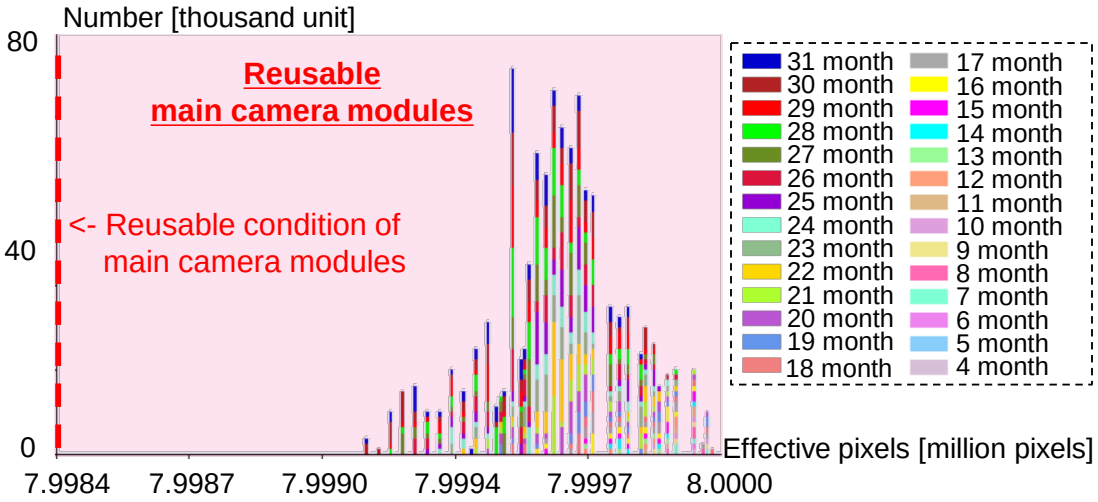


図 7-23：回収された Main Camera Module 個体の状態の違い（回収率 85%）

7.7. 製品ライフサイクルのノミナル情報の修正

7.7.1. 設計パラメータ候補の抽出

作成したライフサイクルフローモデルを構成するすべてのライフサイクルプロセスノードlcpのPR_{lcp}とノード間のリンクを用い，スマートフォンの製品ライフサイクルのノミナル情報を構成するパラメータについての因果関係グラフを生成した．因果関係グラフは，549個のパラメータノードで構成されていた．生成した因果関係グラフの一部を図 7-24 に示す．

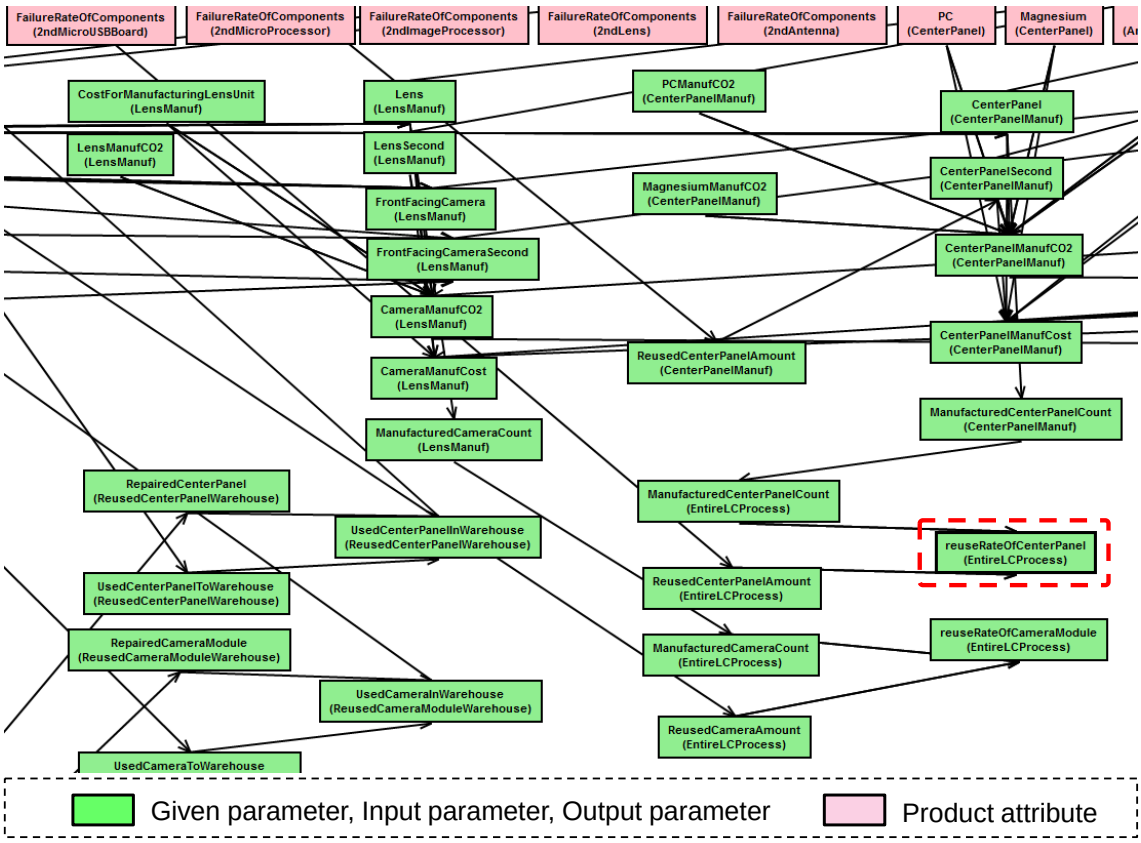


図 7-24：スマートフォンのノミナル情報を構成するパラメータの因果関係グラフ（一部）

生成した因果関係グラフ上で，Center Panel のリユース率を表すパラメータノード（図 7-24 の赤色破線参照）を始点に，因果リンクの逆向きにパラメータノードを探索した．その結果，157 個のパラメータが設計パラメータ候補として抽出された．抽出された 157 個の設計パラメータ候補のうち，Given parameter が 24 個，製品属性が 133 個であった．表 7-10 に，抽出された設計パラメータ候補の一部を示す．

表 7-10：抽出された設計パラメータ候補（一部）

Type	Design parameter
Given parameter	Collection rate
	Reusable condition for Main Camera Module
	Reusable condition for Center Panel
	End production turn of the first smart phone
	Start production turn of the second smart phone
	End production turn of the second smart phone
	Number of heavy user
	Usage frequency of heavy user
	Number of standard user
	Usage frequency of standard user
	Number of light user
	Usage frequency of light user
Product attribute	Initial value of Charpy impact strength of Center Panel
	Rate of change of Charpy impact strength of Center Panel
	Initial value of effective pixels of Main Camera Module
	Rate of change of effective pixels of Main Camera Module

7.7.2. 感度分析による設計パラメータの特定

抽出した 157 個の設計パラメータ候補を，設計者の判断によって制御可能か否かに分類した．制御可能な設計パラメータは，157 個中 109 個であった．制御不可と判断した 48 個の設計パラメータを，表 7-11 に示す．具体的には，ユーザの使用行動に関連する設計パラメータや, Square Trade [133]が報告している故障率などの統計値を, 制御不可と判断した.

表 7-11：制御不可と判断した設計パラメータ

Type	Design parameter	
Given parameter	Last turn for reusing	Usage frequency of standard user
	Constant value of obsolescence	Usage frequency of heavy user
	Start turn of obsolescence	Usage frequency of light user
	Rate of obsolescence at 24 month	Ratio of standard user
	Slope of obsolescence	Ratio of heavy user
	Start production turn of the first smart phone	Ratio of light user
Product attribute	Physical life time of Micro Processor	Physical life time of 2nd Micro Processor
	Failure rate of Micro Processor	Failure rate of 2nd Micro Processor
	Physical life time of OLED	Physical life time of 2nd OLED
	Failure rate of OLED	Failure rate of 2nd OLED
	Physical life time of Button	Physical life time of 2nd Button
	Failure rate of Button	Failure rate of 2nd Button
	Physical life time of Battery	Physical life time of 2nd Battery
	Failure rate of Battery	Failure rate of 2nd Battery
	Physical life time of Speaker	Physical life time of 2nd Speaker
	Failure rate of Speaker	Failure rate of 2nd Speaker
	Physical life time of Vibrator	Physical life time of 2nd Vibrator
	Failure rate of Vibrator	Failure rate of 2nd Vibrator
	Physical life time of Rear Panel	Physical life time of 2nd Rear Panel
	Failure rate of Rear Panel	Failure rate of 2nd Rear Panel
	Physical life time of Micro USB Board	Physical life time of 2nd Micro USB Board
	Failure rate of Micro USB Board	Failure rate of 2nd Micro USB Board
	Physical life time of SIM Slot	Physical life time of 2nd SIM Slot
	Failure rate of SIM Slot	Failure rate of 2nd SIM Slot

制御可能な 109 個の設計パラメータについて，Center Panel のリユース率に対する感度分析を実施した．本ケーススタディでは，抽出した設計パラメータの値を±10%ずつ変動させた．図 7-25 に，感度分析の実行結果の一部を示す．

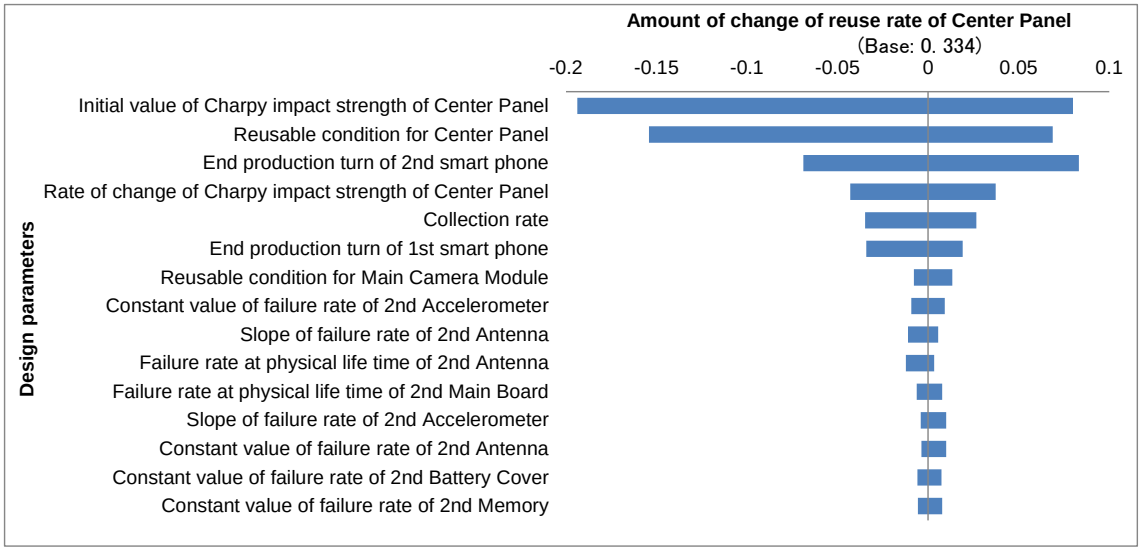


図 7-25：Center Panel のリユース率に関する感度分析の実行結果（上位 15 個）

感度分析の結果より，影響の大きい 10 個の設計パラメータについて，Center Panelのリユース率以外の設計要求への感度を調べた．その結果を表 7-12 に示す．図 7-25 と表 7-12 の結果より，Center Panelのシャルピー衝撃強さの初期値が，Center Panelのリユース率への影響が最も大きく，かつMain Camera Moduleのリユース率，メーカ利益，$LCCO_2$への影響が小さい設計パラメータであることがわかった．

表 7-12：Center Panel のリユース率以外の設計要求項目への影響度（上位 10 個）

	Changed range		
	Reuse rate of Main Camera Module [%]	Total benefit of manufactures [yen]	$LCCO_2$ [kg-CO2]
Initial value of Charpy impact strength of Center Panel	1.15	$0.13*10^6$	$0.18*10^3$
Reusable condition for Center Panel	1.33	$0.12*10^6$	$0.20*10^3$
End production turn of 2nd smart phone	21.52	$28.17*10^6$	$28.27*10^3$
Rate of change of Charpy impact strength of Center Panel	0.37	$0.08*10^6$	$0.06*10^3$
Collection rate	5.00	$0.60*10^6$	$0.64*10^3$
End production turn of 1st smart phone	5.54	$7.40*10^6$	$10.91*10^3$
Reusable condition for Main Camera Module	42.04	$3.12*10^6$	$5.35*10^3$
Constant value of failure rate of 2nd Accelerometer	0.59	$0.05*10^6$	$0.08*10^3$
Slope of failure rate of 2nd Antenna	0.30	$0.10*10^6$	$0.04*10^3$
Failure rate at physical life time of 2nd Antenna	0.56	$0.03*10^6$	$0.07*10^3$

7.7.3. 設計パラメータの修正と評価

式(7.10)で示したように本ケーススタディでは, Center Panel のシャルピー衝撃強さの初期値を，Center Panel の厚みと構成素材のシャルピー衝撃強さに依存するものと仮定した．Center Panel の主な構成素材である PC 樹脂は，他の汎用プラスチックと比較してシャルピー衝撃強さが高い．また PC 樹脂は，金属類と比較して熱伝導率が低い．Center Panel は，Main Board や OLED や Battery を保持する部品であり，OLED や Battery が発する熱を極力 Main Board に伝えないようにする必要がある．以上の理由から，構成素材の変更は修正案に適していないと判断し，本ケーススタディでは Center Panel の厚みを変更することでシャルピー衝撃強さの初期値を向上させることとした．

7.7.2 項で実施した感度分析の結果から，Center Panelのシャルピー衝撃強さの初期値を10%増加させた場合，Center Panelのリユース率の評価結果は 40.9%であり，要求値を満たすことが分かった．そこで，Center Panelの厚みを現状の 1.1 倍である 1.10[mm]に修正した．Center Panelの厚み変更を製品ライフサイクルのノミナル情報モデルに反映して，個体情報モデルを作成した．作成した個体情報モデルを用いて製品ライフサイクルを評価した結果，設計要求項目である「Center Panelのリユース率」「Main Camera Moduleのリ

ユース率」「メーカ利益（現状に対する比率）」「$LCCO_2$（現状に対する比率）」は，それぞれ 40.9%，41.7%，100.1%，97.5%となり，設定したすべての設計要求を満たしていた．

厚み 1.10[mm]とした場合，Center Panel のリユース率の評価結果は要求値を満たしていたが，依然としてリユース条件を満たさない Center Panel 個体が存在した（図 7-26 参照）．例えば厚みを 1.20[mm]とした場合，すべての Center Panel 個体がリユース条件を満たし，Center Panel のリユース率は 41.7%となることを確認した（図 7-27 参照）．どちらの解を選択するかは設計者次第であるが，本ケーススタディでは必要最小限の厚み増加によって設計要求を満たすよう，厚み 1.10[mm]に修正することとした．

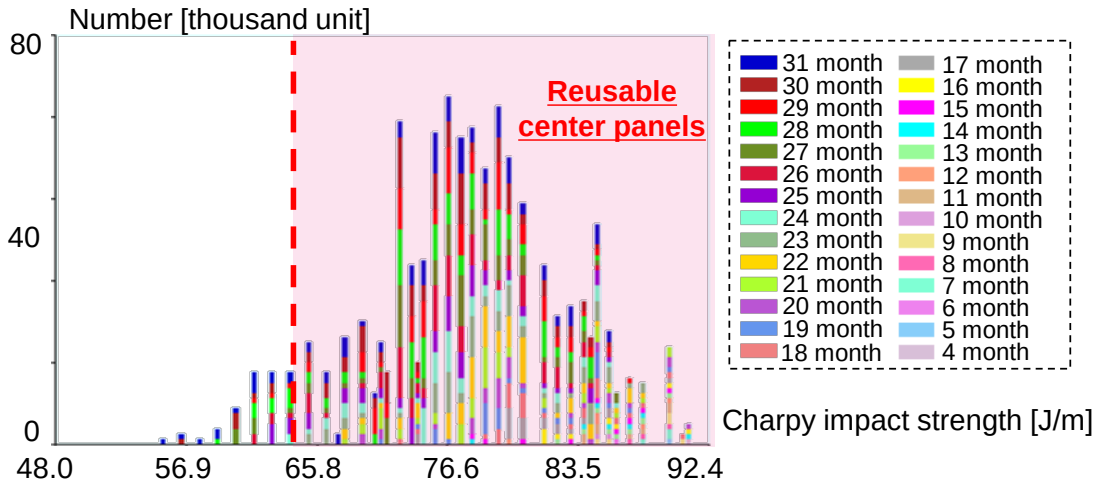


図 7-26：回収された Center Panel 個体の状態の違い（厚み：1.10mm）

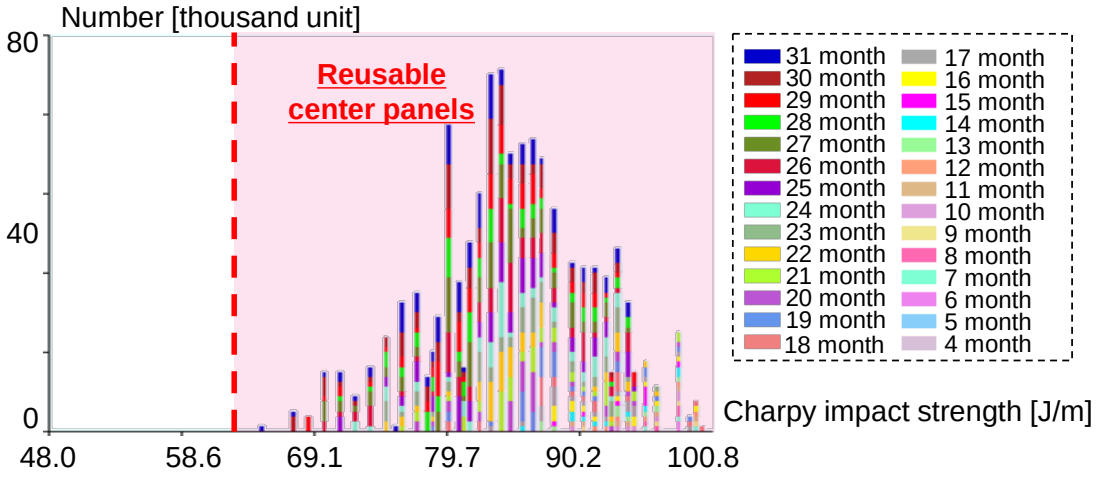


図 7-27：回収された Center Panel 個体の状態の違い（厚み：1.20mm）

また本研究では，[13]のモデル化手法を用いて製品ライフサイクルのノミナル情報を表現しているため，Center Panel の厚み変更が幾何的に実現可能かを検証することができた．具体的には，変更した Center Panel の厚みについて，製品階層構造モデルの一部であるソリッドモデルに反映し，干渉がないことを確認した．その結果，スマートフォン全体の厚みを変更することなく，Center Panel の厚みを 1.10[mm]に変更可能であることが確認できた（図 7-28 参照）．

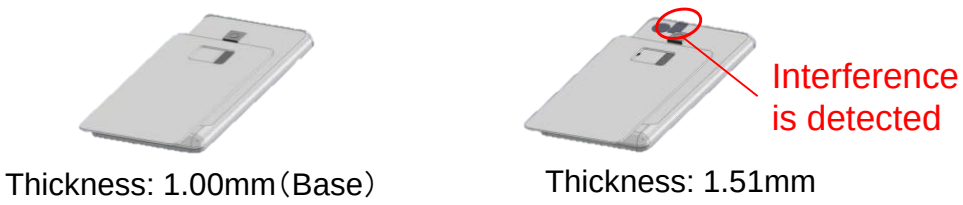


図 7-28：Center Panel の厚みを増加した場合の干渉判定

以上より，提案した設計支援手法を用いることで，次世代機種への部品リユースを含む製品ライフサイクルのノミナル情報を，個体情報が設計要求を満たす範囲で多様さを示すように導出することができた．

7.8. ケーススタディの考察

第 4 章で提案したモデル化手法は，「製品のノミナル情報」「ライフサイクルフローのノミナル情報」「個体情報」という製品ライフサイクルに関する 3 つの情報を，製品階層構造モデル，ライフサイクルフローモデル，個体情報モデルを用いて表現する手法である．製品ライフサイクルに関する 3 つの情報を対応付けて表現する本モデル化手法を用いることで，導出した製品ライフサイクルのノミナル情報について，各ライフサイクルプロセスで処理する個体の状態の違いや量の変動を，個体情報の集合として表現可能であった．この表現によって本ケーススタディでは，導出したノミナル情報のリユース率が要求値以下であるという資源循環の問題とその原因を，設計段階で明示的に示唆することができた．具体的には，作成した個体情報モデルから，製造したスマートフォン個体，および回収した Center Panel 個体と Main Camera Module 個体の個体情報を抽出し，それぞれ個体数の時間変化を表現した．その結果，回収個体数と二世代目スマートフォンの製造個体数の重なりが少ないことを明示的に示唆できた．特に，図 7-18 のようなグラフ表現によって回収個体数を明示化することで，回収率 22%であることが重なりの少ない主要な原因であると推定できた．ただし，こういった回収個体数の時間変化と製造個体数の時間変化の重なりは，2.3 節で挙げたマテリアルフロー分析や 2.2.3 項で挙げた限界リユース率など，既存の統計的手法を用いても表現できる．一方で，製品や部品 1 個体ごとの個体情報から成る個体情報モデルでは，個体数の時間変化に加えて，回収された Center Panel 個体のシャルピー衝撃強さの違いと Main Camera Module 個体の有効画素数の違いを表現できた．この表現によって，設計変更前の Center Panel では，回収された個体にリユース条件を満たす個体と満たさない個体の両方を含んでいることを明示的に示唆できた．また，回収された Main Camera Module 個体がすべてリユース条件を満たしており，かつリユース率に関する要求値を満たしていたことから，リユース条件を満たさない Center Panel 個体を含むことがリユース率に関する要求値を満たさない原因のひとつであることを推定できた．さらに，各個体の回収時期を個体情報から読み取ることで，図 7-22 の棒グラフのように回収時期ごとに色分けして状態の違いを表すことができた．このように個体の回収時期と属性値を関連付けて表現することで，リユース条件以下の Center Panel 個体の大半は，25[month]以降に回収された個体であることが分かった．また，図 7-22 の棒グラフを形成する個体情報と同一のeidを$assy$に含み，かつ使用工程に位置するスマートフォン個体の個体情報eiを個体情報モデルから読み取ることで，Center Panel 個体の属性値の分布をユーザタイプごとに表現できる（図 7-29 参照）．その結果，Heavy User によって使用されていたスマートフォン個体の構成部品である Center Panel 個体の大半がリユース条件以下であることを明示化できる．

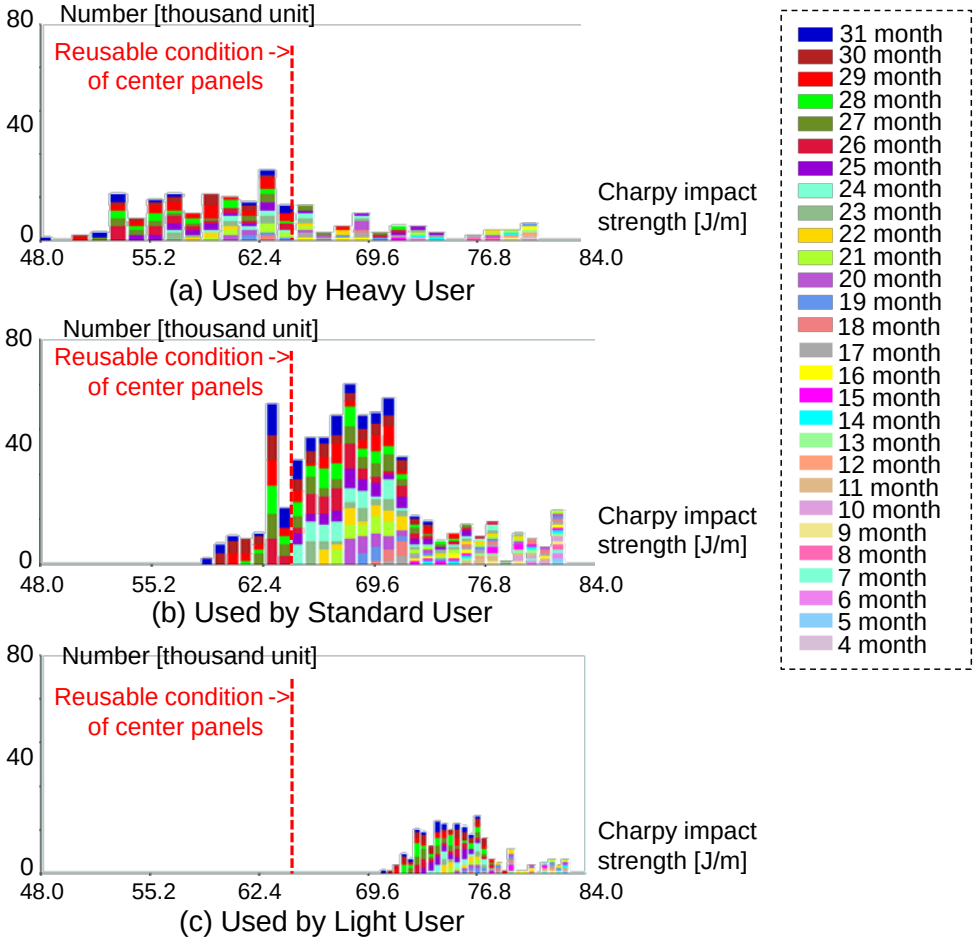


図 7-29：ユーザタイプごとの Center Panel 個体のシャルピー衝撃強さの違い

個体情報モデルを用いたこれらの表現により，Center Panel のリユース率を向上させるには，Heavy User の使用頻度下においても Center Panel 個体がリユース条件以上のシャルピー衝撃強さを確保できるよう，その初期値を増加させるか変化率を鈍化させる必要があると推定できる．実際，シャルピー衝撃強さの初期値と変化率は，5.2 節で提案した修正案特定支援手法によって，Center Panel のリユース率に対して影響の大きい設計パラメータとして探索されていた（7.7.2 項の図 7-25 参照）．つまり，提案した設計支援手法は，個体情報の評価結果に対する修正案を特定支援可能としただけでなく，解くべき設計問題に対して様々な側面や粒度から個体情報を明示的に表現可能とすることで，個体情報の多様さに起因して生じる製品ライフサイクルの設計問題やその原因を明示的かつ定量的に示唆できる．この明示的かつ定量的な方法で設計支援が可能な点に，提案手法の有効性があると考える．

また，5.1 節で提案した個体情報の評価の実行支援手法によって，製品ライフサイクルのノミナル情報モデルの作成から個体情報モデルを作成する過程の手間と労力を削減できた．具体的には，ライフサイクルフローモデル全体で 281 行から成る Procedure のうち，222

行を Procedure Description Support System（6.5.6 項参照）によって生成できた．この 222 行のうち 128 行の計算処理を，製品階層構造モデルとライフサイクルフローモデルの対応関係に基づいたライフサイクルプロセスタイプへの分類によって生成できた．ただし，ライフサイクルプロセスタイプ「(f)分岐」と「(g)開始」に分類されたライフサイクルプロセスノードの計算処理の生成には，それぞれ分岐条件と生産台数に関するパラメータを，設計者自身で選択する必要があった（7.5 節(f)(g)参照）．なお，生成したPR_{lcp}によるライフサイクルプロセスノードlcpにおける個体の集合に対する処理方法の表現が，製品階層構造モデルとライフサイクルフローモデルの対応関係による表現と一致していることを，設計者自身で確認した．つまり本ケーススタディでは，個体情報の評価を実行するうえで生じる「設計者の手間と労力」と「表現内容の不一致」という 2 つの問題（5.1.1 項参照）を，提案手法によって解消することができた．ただし，222 行の計算処理を生成可能とするために，もともと 28 個のライフサイクルプロセスノードから成るライフサイクルフローモデルを詳細化して，42 個のライフサイクルプロセスノードで構成する必要があった（表 7-9 参照）．ライフサイクルプロセスノードを詳細化することでライフサイクルプロセスのネットワークが複雑になるが，Procedure を記述する手間と労力が削減できた点は，本手法の利点と考えられる．

さらに，5.2 節で提案した製品ライフサイクルの修正案特定支援手法によって，設計対象の製品ライフサイクルを構成するパラメータ間の関係が明示的に表現できたとともに，設計要求を満たすために修正すべきノミナル情報を段階的に特定することができた．本ケーススタディでは，スマートフォンの製品ライフサイクルのノミナル情報を構成する 549 個のパラメータのネットワークを，因果関係グラフとして生成することができた（5.2 節(a1)の過程に該当）．549 個のパラメータのうち，設計パラメータは 268 個であった．因果関係グラフを用いることで，Center Panelのリユース率に影響を与える設計パラメータ候補を，設計パラメータ全体の 59%にあたる 157 個に絞り込むことができた（5.2 節(a2)の過程に該当）．さらに，109 個の制御可能な設計パラメータに絞り込んでから感度分析を実施することができた（5.2 節(b1)(b2)の過程に該当）．なお，5.2 節(a2)の過程で，e-PLC-CADシステムによって抽出されなかった 111 個の設計パラメータを，表 7-13 と表 7-14 に示す．この 111 個の設計パラメータのうち，CO_2排出量原単位を表す設計パラメータが 22 個，コスト原単位を表す設計パラメータが 24 個，CO_2排出量やコスト計算のための設計パラメータが 9 個，変動要因ではない性能値を表す設計パラメータが 12 個，構成素材を表す設計パラメータが 44 個であった．このうちいずれの設計パラメータも，Center Panelのリユース率に影響を与えないことを設計者自身で確認した．すなわち本ケーススタディでは，5.2 節(a1)に示した 3 つの段階でパラメータ間の因果関係を特定することによって，対象の評価値に影響を与えるすべての設計パラメータを漏れなく抽出することができた．このことから，提案手法を用いることで，必要最小限のパラメータについて感度分析を実施し，修正案を特定することができたと考える．

表 7-13：抽出されなかった設計パラメータ（Given parameter）

Life Cycle Process Node	Given parameter
FrontGlassManuf	CostForManufacturingTouchScreen
	PCManufCO2
	GlassManufCO2
OLEDManuf	CostForManufacturingOLED
	OELDManufCO2
BottomBoardManuf	CostForMicroUSBBoardManuf
	IntegratedCircuitManufCO2
	PCBoardManufCO2
AntennaManuf	PCManufCO2
	SteelManufCO2
BatteryManuf	CostForManufacturingBattery
	LiIonBatteryManufCO2
BatteryCoverManuf	PCManufCO2
	RateForPCGlass
CenterPanelManuf	PCManufCO2
	MagnesiumManufCO2
RearPanelManuf	PCManufCO2
MainBoardManuf	CostForManufacturingProcessor
	CostForManufacturingWirelessSection
	CostForManufacturingWLAN
	CostForManufacturingMemory
	CostForManufacturingIntegratedCircuit
	IntegratedCircuitManufCO2
	SilverManufacturingCO2
	CopperManufacturingCO2
	GoldManufacturingCO2
LensManuf	CostForManufacturingLens
	CostForManufacturingFrontFacingCamera
	LensManufCO2
ImageProcessorManuf	CostForManufacturingImageProcessor
	IntegratedCircuitManufCO2
SpeakerManuf	SpeakerManufacturingCO2
AssemblyForFirst	CostForAssembly

AssemblyForSecond	CostForAssembly
Transportation	TransportationDistance
	FuelConsumption
	LoadCapacity
	CostForBoxContents
	LightOilUseCost
	LightOilUseCO2
Sale	salesPrice
	discountedRate
Operation	costForCharging
	UseElectoricityCO2
CollectedOrUncollected	collectionDeposit
	currentCollectionRate
Disassembly	disassemblyTime
InspectionForCenterPanel	InspectionFee
InspectionForCameraModule	InspectionFee
ReusedCenterPanelShipping	ChargesForCustody
ReusedCameraModuleShipping	ChargesForCustody
EntireLifeCycle	yenToTheDollar
	laborCost
	PCManufCost
	MgManufCost

表 7-14：抽出されなかった設計パラメータ（製品属性）

Entity Node	Product attribute	
OLED	Performance	Screen size
	Material	OLED
Front Glass	Material	PC
	Material	Glass
Center Panel	Material	PC
	Material	Magnesium
Rear Panel	Material	PC
Antenna	Material	PC
	Material	Steel
Image Processor	Material	PCBoard
Lens	Material	Camera
Front Facing Camera	Performance	Effective pixels
	Material	Camera
Battery	Material	Li-ion Battery
Battery Cover	Material	PC
Micro Processor	Performance	Clock Speed
	Material	Gold
	Material	Silver
	Material	Copper
	Material	Palladium
	Material	Lead
Memory	Performance	Internal memory size
Micro USB Board	Material	PCBoard
Vibrator	Material	Vibrator
Speaker	Material	Speaker
Button	Material	PC
2nd OLED	Performance	Screen size
	Material	OLED
2nd Front Glass	Material	PC
	Material	Glass
2nd Center Panel	Performance	Initial value of Charpy impact strength
	Performance	Rate change of Charpy impact strength
	Material	PC

	Material	Magnesium
2nd Rear Panel	Material	PC
2nd Antenna	Material	PC
	Material	Steel
2nd Image Processor	Performance	Initial value of effective pixels
	Performance	Rate change of effective pixels
	Material	PCBoard
2nd Lens	Material	Camera
2nd Front Facing Camera	Performance	Effective pixels
	Material	Camera
2nd Battery	Material	Li-ion Battery
2nd Battery Cover	Material	PC
2nd Micro Processor	Performance	Clock speed
	Material	Gold
	Material	Silver
	Material	Copper
	Material	Palladium
	Material	Lead
2nd Memory	Performance	Internal memory size
2nd Micro USB Board	Material	PCBoard
2nd Vibrator	Material	Vibrator
2nd Speaker	Material	Speaker
2nd Button	Material	PC

一方，本ケーススタディを e-PLC-CAD システム上で行うことによって，個体情報の評価の実行支援手法に課題があることが分かった．上述したように本ケーススタディでは，ライフサイクルフローモデルの詳細化によってすべてのライフサイクルプロセスノードをライフサイクルプロセスタイプに分類できた．しかし，e-PLC-CAD システムによって生成されず，設計者自身で記述しなければならない計算処理が複数行存在した．本ケーススタディでは，以下の場合において，Procedure テンプレートや計算式リストを用いた計算処理の生成ができなかった．

ケース1：あるライフサイクルプロセスにおいて製品個体の属性が変化する場合（例えば，劣化，使用中の製品個体の故障発生，など）

ケース2：条件文の前件部に計算処理を含む場合（例えば，リユース部品個体数を考慮した新品部品個体の生産台数の算出，倉庫から出庫する最大個体数の算出，など）

ケース3：条件文の前件部が複数の条件式から成る場合

ケース4：Given parameter，Input parameter，Output parameter を用いないテンポラリ変数の定義に関する計算処理

例えば，陳腐化によって使用中のスマートフォンを廃棄するライフサイクルプロセスを表す DisposingBasedOnObsolescence プロセスは，ライフサイクルプロセスタイプ「(f)分岐」に分類できた．しかし，全 27 行から成る Procedure うち 9 行の計算処理を設計者自身が記述しなければならなかった（図 7-30 参照）．設計者自身で記述した計算処理は，「乱数の取得といった Given parameter，Input parameter，Output parameter を用いないテンポラリ変数の定義に関する計算処理（図 7-30①参照）」および「条件文の前件部が複数の条件式から成る場合（図 7-30②参照）」であった．

```
Random cRandom = new Random();                                                    ①

//1世代目
[[keepUsingSmartPhoneWithoutObsolescence]].AddRange([[keepUsingSmartPhoneWithoutAccident]]);

foreach(EntityData entity in [[keepUsingSmartPhoneWithoutObsolescence]])
{
    double dRandom = cRandom.NextDouble();                                        ①
    if((entity.ElapsedTime < [[ObsolescenceStartTurn]]  && dRandom < [[obsolescenceConstantValue]])
      || (entity.ElapsedTime == 24 &&  dRandom < [[ObsolescenceValueAt24]])       ②
      || (entity.ElapsedTime >= [[ObsolescenceStartTurn]] &&  entity.ElapsedTime != 24 &&
          dRandom < [[ObsolescenceSlope]] * (entity.ElapsedTime - [[ObsolescenceStartTurn]]) + [[obsolescenceConstantValue]]))
    {
        [[disposedSmartPhoneDueToObsolescence]].Add(entity);
    }
}
foreach(EntityData entity in  [[disposedSmartPhoneDueToObsolescence]])
{
    [[keepUsingSmartPhoneWithoutObsolescence]].Remove(entity);
}

//2世代目
[[keepUsingSmartPhoneSecondWithoutObsolescence]].AddRange([[keepUsingSmartPhoneSecondWithoutAccident]]);

foreach(EntityData entity in [[keepUsingSmartPhoneSecondWithoutObsolescence]])
{
    double dRandom = cRandom.NextDouble();                                        ①
    if((entity.ElapsedTime < [[ObsolescenceStartTurn]]  && dRandom < [[obsolescenceConstantValue]])
      || (entity.ElapsedTime == 24 &&  dRandom < [[ObsolescenceValueAt24]])       ②
      || (entity.ElapsedTime >= [[ObsolescenceStartTurn]] &&  entity.ElapsedTime != 24 &&
          dRandom < [[ObsolescenceSlope]] * (entity.ElapsedTime - [[ObsolescenceStartTurn]]) + [[obsolescenceConstantValue]]))
    {
        [[disposedSmartPhoneSecondDueToObsolescence]].Add(entity);
    }
}
foreach(EntityData entity in  [[disposedSmartPhoneSecondDueToObsolescence]])
{
    [[keepUsingSmartPhoneSecondWithoutObsolescence]].Remove(entity);
}
```

図 7-30：DisposingBasedOnObsolescence プロセスの Procedure
（赤色ハイライト部分：e-PLC-CAD システムが生成した計算処理）

対象とする製品種を増やし，Procedure テンプレートや計算式リストの汎用性を向上させることが今後の課題である．特に，上記のケース 1 は，変動要因に関する計算処理である．3.3 節の要件 2-1 を満たすために，ケース 1 に関する計算処理の記述支援は解決すべき課題のひとつと考える．

第8章　考察

第 8 章では，本研究で提案した製品ライフサイクルの設計支援手法の有効性を考察し，課題を抽出する．

8.1. 製品ライフサイクルの設計支援手法の有効性

8.1.1. 製品個体の集合に着目した製品ライフサイクルのモデル化手法の有効性

第 4 章で提案したモデル化手法は, 1)製品ライフサイクルのノミナル情報モデルを用いた変動要因の表現，2)個体情報を表現した個体情報モデル，3)製品ライフサイクルのノミナル情報モデルを入力とした LCS の実行による個体情報モデルの作成，という 3 つの要素から成る手法であった．個体情報は LCS の実行過程で生成した結果であるが，本手法では各個体情報を式(4.3)と式(4.4)の規定に従って個体情報モデルの要素としている．そのため，従来の LCS 手法での個体情報はシミュレーション実行後に消えてしまうが，本手法での個体情報はシミュレーション実行後も仮想的に存在し，随時観測や操作できる．具体的には，4.3.2 項で示したような，各ライフサイクルプロセスで処理する個体の(1)量の変動，(2)状態の違い，および(3)部品の劣化，(4)部品交換の履歴，(5)リユース部品と新品部品が共存する製品，というライフサイクルにわたる様々な側面や粒度から個体情報を表現できる．

このようにライフサイクルにわたって各個体が示す個体情報を表現する本モデル化手法が，既存の製品ライフサイクルのモデル化手法に比べて優位な点について考察する．既存のモデル化手法としては例えば，製品ライフサイクルのノミナル情報のモデル化手法 [13] や LCA で用いられるモデルのように，個体情報の異なる多数の個体が存在する製品ライフサイクルを, 対象製品の 1 個体のライフサイクルにより代表させて表現するモデルがある．また，回収量の時間変化を，統計的な分布として表現するモデルがある（例えば， [84]）．これら 2 つの表現手法（以下，ノミナルモデルと分布モデル）と本モデル化手法による製品ライフサイクルの表現には，以下の違いがある．

まず，ノミナルモデルでの回収量の変化$C_{nominal}(t)$と分布モデルでの回収量の変化$C_{distribution}(t)$を，それぞれ式(8.1)と式(8.2)で表現する．$p(i)$はライフサイクルタイムiで製造する製品個体数を表す．$f(t-i)$は，$t-i$が製品の使用期間と一致する場合は 1 を，一致しない場合は 0 を取る関数を表す．また分布モデルは，使用期間が正規分布に従うことを例として，式(8.2)のように平均使用期間μとその標準偏差σを用いて表す．

$$C_{nominal}(t) = \sum_{i=1}^{t} p(i) \times f(t-i) \tag{8.1}$$

$$C_{distribution}(t) = \sum\nolimits_{i=1}^{t} \left(\frac{p(i)}{\sqrt{2\pi}\sigma} \times \exp\left(-\frac{((t-i)-\mu)}{2\sigma^2} \right) \right) \tag{8.2}$$

このとき，ノミナルモデルでは，設計要求に対する資源循環の実行効果を現実世界よりも過大評価もしくは過小評価する可能性がある．例えば第 7 章のケーススタディでは，提案手法を用い，「偶発故障率」「機能不全による故障率」「陳腐化による廃棄発生率」「使用頻度の違い」といった現実の使用工程に存在し得る変動要因を複数設定して，スマートフォ

ンの回収個体数の時間変化やその状態の違いを表現した（図 8-1(c)参照）．一方で，すべての個体のライフサイクル履歴を一様に表現するノミナルモデルを用いた場合，すべての個体の使用期間や使用頻度は同じであるため，リユース条件以下の Center Panel 個体が表現されない（図 8-1(a)の右側参照）．その結果，ノミナルモデルを用いた Center Panel のリユース率の評価結果は，提案手法を用いた評価結果よりも高い値を示している．ただし，Center Panel のシャルピー衝撃強さの変化率やリユース条件の設定値次第では，すべての個体がリユース条件以下となり，その場合はリユース率 0%となる．

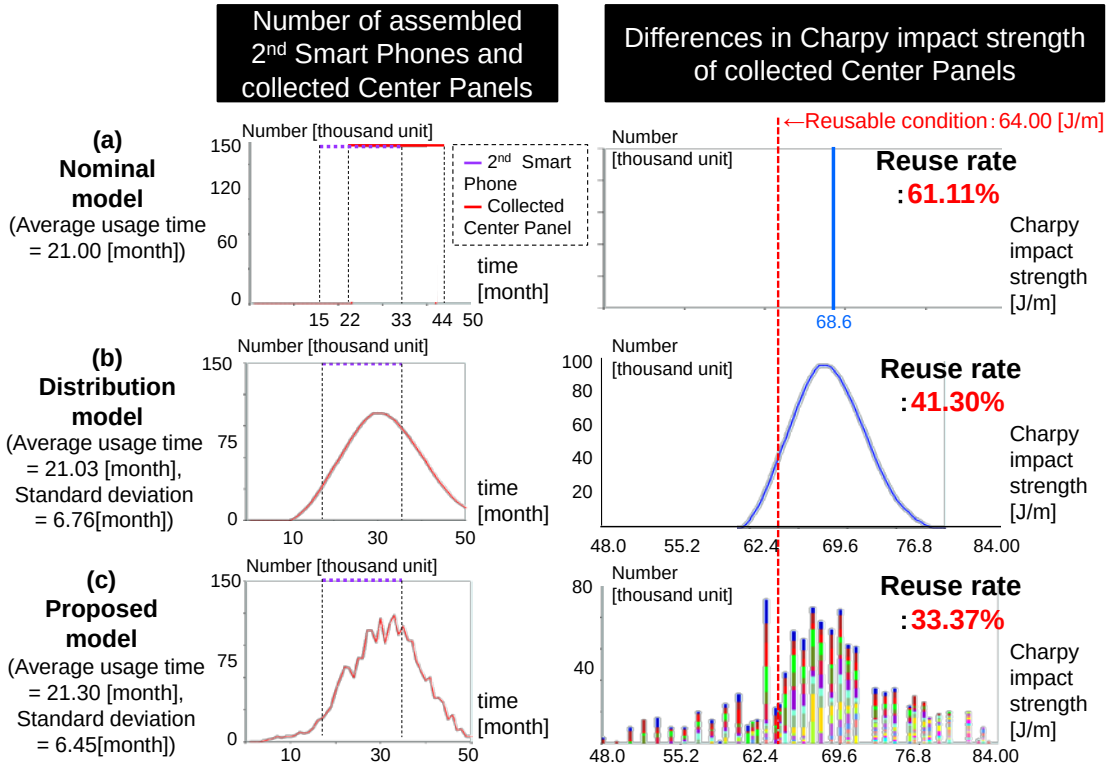


図 8-1：既存手法と提案手法による製品ライフサイクルの表現の違い

また図 8-1(a)(b)では，高い性能のまま回収される Center Panel 個体が表現されていない．図 8-1 (b)と(c)では，平均使用期間と標準偏差がほぼ等しく，そのため回収個体数の時間変化を表すグラフは同等の形状を示している．しかし，Center Panel 個体の状態のバラつきは，図 8-1 (b)の分布モデルよりも図 8-1 (c)の本モデル化手法を用いた表現の方が大きい．その理由のひとつとして，分布モデルでは属性値の変化率を経過時間のみに依存するものとして表現しており，そのため使用頻度などの他の要因を状態の違いに反映していない点が挙げられる．以上より，ノミナルモデルを用いた設計では，導出したノミナル情報が設計要求を満たすかのように製品ライフサイクルを過大評価する可能性がある．また反対に，ノミナルモデルや分布モデルを用いたライフサイクル設計では，高い性能のリユース部品

を求めるユーザに対して部品リユースするという潜在的な需要を満たすビジネス戦略が，設計段階で検討対象から外れてしまうため，部品リユースの効果を過小評価する可能性がある．

ただし，分布モデルでは，平均使用期間とその標準偏差に加えて，使用頻度などの統計値を組み合わせてモデル作成することで，図 8-1 (c)と類似した分布形状の状態の違いを表現できる．このように統計値の組み合わせによって状態の違いや量の変動を表現する分布モデルに対しての本モデル化手法の利点は，状態の違いや量の変動に起因した設計問題の原因となる個体の集合の特徴を明示的に表現できる点にある．例えば，資源循環の供給資源として十分な量の個体が確保できないと評価された場合，多くの個体が所望のライフサイクルを辿らない原因を，その個体の集合を任意のライフサイクル履歴ごとにクラスタリングして表現することで分析可能となる．例えば第 7 章のケーススタディでは，7.8 節で示したように，Center Panel 個体のシャルピー衝撃強さの違いを，回収時期や使用ユーザごとに明示化することで，Heavy User の使用による性能低下が，Center Panel のリユース率を要求値以下とする要因のひとつであることを把握できた．さらに本モデル化手法は，4.3.2 項で示したように，各個体の幾何形状をソリッドモデルによって表現できる．そのため各個体の性能を，CAE などの解析シミュレーションを用いて評価できる．個体ごとの解析シミュレーションは，様々な属性が複雑に関係し合って決まる性能値をもつ個体を評価する場合に有効である．例えば，第 7 章のケーススタディで Center Panel のリユース条件として設定したシャルピー衝撃強さは，使用時間だけでなく，腐食や破損や変形などに依存して，個体ごとに異なる値を示す．個体情報モデルを用いることで，各 Center Panel 個体の腐食進展，破損，変形度合を設定したソリッドモデルを入力として落下衝撃シミュレーションを実行することで，Center Panel 個体のシャルピー衝撃強さをより具体的に評価できる．ただし，個体ごとの解析シミュレーションを実行可能とするようソリッドモデルを作成するためには，幾何形状の変化率をあらかじめ製品のノミナル情報として設定する必要がある．例えば，摩耗による形状変化に関しては，どの面がどの基準面に対してどのような角度にどういった割合で変化するかを，評価対象個体のノミナル情報を表す実体ノードの属性値に設定する必要がある．こういった変化率の設定に関する課題は，8.3 節(d)で考察する．

8.1.2. 製品個体の集合に着目した製品ライフサイクルの設計サイクル実行支援手法の有効性

5.1 節で提案した個体情報の評価の実行支援手法を用いることで，部分的には設計者による記述が必要であるものの，Procedure の大半をシステムによって生成可能とした．その結果，個体情報の評価を実行するためのノミナル情報モデルの作成過程における，設計者の手間と労力を削減可能としたと考えられる．

5.2 節で提案した修正案特定支援手法は，個体情報モデルを用いた評価の実行結果を受けて，修正すべき設計パラメータ候補を導出可能であった．本手法で設計パラメータの探索に用いた製品ライフサイクルのノミナル情報モデルは，製品とライフサイクルフローのノミナル情報を対応付けて表現した手法である [13]．そのため本手法では，設計要求を満たすために修正すべき設計パラメータ候補を，製品のノミナル情報だけでなく，そのライフサイクルフローのノミナル情報からも抽出することができた．また，本手法を用いて設計パラメータを探索するためには，ライフサイクルプロセスノードの Procedure やノード間のリンクを用いてパラメータ間の関係を記述しておく必要がある．上述したように Procedure の大半は，5.1 節の手法に基づいて生成できるため，本手法はノミナル情報の修正過程における設計者の手間と労力を削減可能としたと考えられる．

8.1.3. 製品ライフサイクルの設計支援手法全体の有効性

本設計支援手法で用いた製品ライフサイクルのノミナル情報モデルは，製品とライフサイクルフローのノミナル情報をトポロジーレベルで整合性管理（3.4.1 項(D)参照）することができ，またソリッドモデルを一部としている [13]．そのため，5.2 節の手法によって特定した設計パラメータの修正が実行可能かどうかを，トポロジーレベルとジオメトリレベルで評価することができる．例えば第 7 章のケーススタディでは，Center Panel の厚みを現状の 1.1 倍に変更することが幾何的に実現可能かを，ソリッドモデルを用いて確認できた（図 7-28 参照）．また，本手法では，厚みを 1.1 倍にした場合の Center Panel の重量やシャルピー衝撃強さの初期値を製品階層構造モデル上で変更し，変更したモデルを入力に LCS を実行して個体情報モデルを作成するといった，ノミナル情報の修正から個体情報の再評価が同一の設計対象モデル上で実行可能であった．さらに，8.1.2 項で示したように，第 5 章で提案した設計サイクルの実行支援手法を用いることで，「製品とライフサイクルフローのノミナル情報の導出」から「個体情報の評価」の過程と，「個体情報の評価」から新たな「製品ライフサイクルのノミナル情報の導出」の過程が円滑に実行可能であった．以上より，本研究で提案した設計支援手法は，3.3 節で示した製品ライフサイクルの設計サイクルに沿って，導出した製品ライフサイクルのノミナル情報についての個体情報の評価と，評価結果から提起される問題を解決する新たなノミナル情報の導出，および新たに導出したノミナル情報の評価，という一般的な設計と同様の設計サイクルを，1 つの設計対象モデル上で実行可能とした．このようにライフサイクル設計における特定の設計段階での円滑

な設計サイクルの実行を支援している点は，製品ライフサイクルのノミナル情報を入力に個体情報をシミュレーションする技術である LCS に対しての，本手法の利点のひとつと考えられる．

一方，提案手法を用いて製品ライフサイクルを表現する場合，既存手法と比較して，ユーザの使用頻度や廃棄行動といった設計者にとって入手が難しく，不確実性をもつパラメータの入力が必要である．例えば，第 7 章のケーススタディでは，部品の属性値の変化率やリユース条件の閾値などを予測値によって，また故障率や使用頻度などを類似製品の実データによってモデル作成した．対象製品の実データではなく，これら予測値や類似製品の実データを用いたことで，仮想的に生成した個体情報と現実世界の個体情報との間に乖離が生じる可能性は高まる．その結果，循環を意図した設計による環境性や経済性の改善効果が十分に得られない可能性もある．しかし，現実世界に先立って個体情報を仮想的に生成しておくことで，実データを取得した場合に，設計時の予測とどのくらい乖離しているかを分析できる．さらに，5.2 節に示した修正案特定支援手法を用いて，設計要求に関する評価値への影響が大きい設計パラメータを特定しておくことで，製品ライフサイクルの変動要因すべてをモニタリングや情報収集することを回避し，モニタリングや情報収集の負担を軽減できる．以上の「現実世界との乖離の分析」と「モニタリングや収集が必要な情報の特定」の観点からも，設計段階で個体情報を模擬してノミナル情報を決定することの意義があり，提案手法はその支援に有効であると考える．

8.2. 計算機実装の有効性

本研究で提案した製品ライフサイクルの設計支援手法を計算機上で実装した e-PLC-CAD システムは，第 7 章のケーススタディで示したように，4 種類の変動要因を含めた製品とライフサイクルフローのノミナル情報を統合的にモデル作成可能であった．また，製品ライフサイクルのノミナル情報モデルを入力に LCS を実行して作成した個体情報モデルによって，個体情報の多様さを考慮した製品ライフサイクル全体の環境性や経済性の評価を可能としただけでなく，個体が集合として示す状態の違いや量の変動を表現することで，ライフサイクルプロセスの実行効果（例えば，リユース率）を評価可能とするといった，製品ライフサイクルの任意の側面や粒度からの個体情報の評価が支援可能であった．また，ソリッドモデルとの連携により，各個体の劣化シミュレーションなど，個体レベルでの評価が可能であることを示唆した．さらに本システムは，個体情報の評価を実行した結果が設計要求を満たしていない場合，製品ライフサイクルのノミナル情報を構成するパラメータの中から，対象の設計要求に関する評価値に対する影響が大きい設計パラメータとその他の設計要求に関する評価値に対する影響の小さい設計パラメータを列挙することで，修正に適した設計パラメータの特定を支援することができた．

e-PLC-CAD システム上で製品ライフサイクルの設計過程を進める場合，実体ノードやライフサイクルプロセスノードに多くのパラメータ値を入力し，またこれらパラメータ間の関係を Procedure やリンクによって表現する必要がある．そのため，製品ライフサイクルの設計には時間と労力を要する．この問題に対して本システムは，製品設計過程で製品設計者が作成するであろうソリッドモデルを一部とするため，構成素材や重量などの属性値をソリッドモデルから直接取得できる．さらに，5.1 節の手法に基づいて，Procedure の記述を支援している．Procedure の記述を支援するサブシステムは，例えば，7.8 節で示したように，スマートフォンの製品ライフサイクルのノミナル情報モデルの大半の計算処理を生成可能としているため，設計過程における設計者の負担を削減することができたと考えられる．また本サブシステムは，Procedure テンプレートと計算式リストに加えて，記述頻度が高い計算処理を任意に格納や取得可能としたデータベースと接続している．そのため本システムを用いた設計を重ね, Procedure 記述のためのデータベースを充実させていくことで，手入力による記述をさらに減らしていくことができる．

製品開発現場で e-PLC-CAD システムを用いることで, 製品設計者や各ライフサイクルプロセス設計者は，個体情報モデルを用いた製品個体の状態の違いや量の変動の評価結果を受けて，製品や各ライフサイクルプロセスの設計変更ができる．ただし，上記の設計を実現するためには，製品設計者や各ライフサイクルプロセス従事者といったライフサイクルにわたるステークホルダ間の連携，および連携のためのファシリテータとなるライフサイクル設計者の存在が必要不可欠である（図 8-2 参照）．

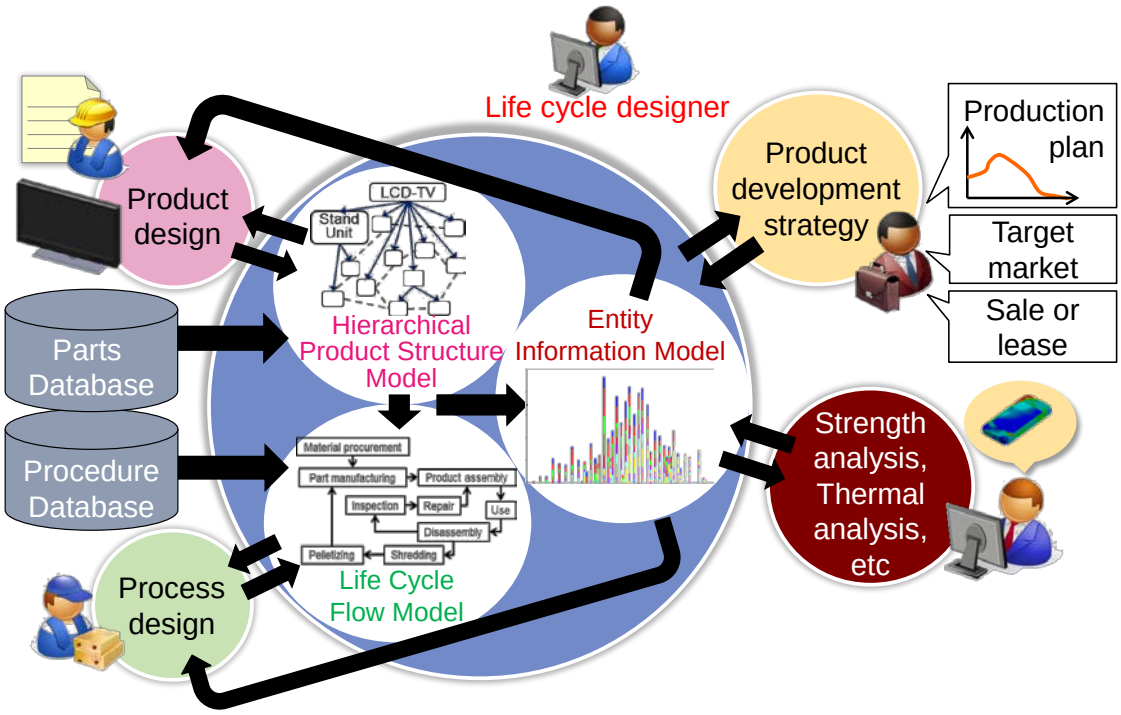


図 8-2：製品開発現場での e-PLC-CAD システムの活用イメージ図

8.3. 製品ライフサイクルの設計への提案手法の適用限界

提案した設計支援手法には 8.1 節で示した有効性があり，本設計支援手法を計算機実装した e-PLC-CAD システムには 8.2 節で示した有効性がある．一方で，設計支援手法や e-PLC-CAD システムには以下 4 つの課題がある．

(a) 個体の多様性を模擬するためのモデル作成の困難性

個体情報の評価の実行支援においては，製品ライフサイクルのノミナル情報モデルのトポロジー表現を用いて Procedure の一部を生成するものとした．つまり，製品階層構造モデルとライフサイクルフローモデルの対応関係に基づいて，ライフサイクルプロセスにおける個体の集合に対する処理方法を基本タイプに分類し，基本タイプごとの Procedure テンプレートに基づいて計算処理を生成するものとした．本トポロジー表現は，ライフサイクルにわたる個体の一様な流れに加え，「(f)分岐」という変動要因(iii)やその結果として生じる個体のフローの違いを表すことができる．しかし，個体の状態変化の違いを表現することができない．そのため，変動要因(ii)「主体の能力や要求の違い」の結果として生じる個体の状態の違いに関する計算処理は，設計者自身で記述する必要がある．また，個体の状態に違いを与える計算処理（例えば，使用段階の劣化進展モデル）は，様々な変動要因が関係し得る．そのため，設計者にとって記述が困難，もしくはそもそも因果関係の特定が困難なケースが多い．例えば第 7 章のケーススタディでリユース対象部品とした Main Camera Module の劣化は，実際には撮影回数，使用環境の温度や湿度，光曝露など，様々な要因が影響すると考えられる．しかし，これら変動要因の関係を特定することは困難であり，本ケーススタディでは使用時間のみに依存するものと表現した．こういった個体情報に違いを与える計算処理のモデル化支援は課題のひとつであるが，製品個別の問題であり，一般化することは困難である．

(b) 設計パラメータ探索の困難性

本手法で特定可能な設計パラメータは，製品ライフサイクルのノミナル情報モデルを用いて設計者が表現するパラメータやその間の関係といったモデル構造の粒度に依存する．例えば，第 7 章の実行例で修正した Center Panel の厚みのように，製品ライフサイクルのノミナル情報モデル上で表現していない設計パラメータは，修正案として直接特定することはできない．作成するモデル構造の粒度が設計者に依存している点は，本手法の適用限界である．特に，製品階層構造モデルの現在の実装では，製品属性間の関係を表現することができないため，設計対象モデルの対象表現にも限界がある．製品属性間の関係を表現可能な実装とすることは，今後の課題である．

(c) 最適解の導出における困難性

本設計支援手法は，複数のノミナル情報の解候補から最適解を選択するための支援が十分ではない．例えば 5.2 節で提案した手法は，感度分析の実行結果から，対象の設計要求に関する評価値への影響が大きく，かつ他の設計要求に関する評価値への影響が小さい設計パラメータを，修正すべきパラメータとして特定するものとした．しかし，導出したノミナル情報が複数の設計要求を満たしていない場合，修正すべき設計パラメータの間にトレードオフが生じる可能性がある．また反対に，すべての設計要求に関する評価値に正の影響を与える設計パラメータも存在し得る．しかし本手法では，対象とする設計要求以外の設計要求に関する評価値への影響が小さい設計パラメータから，設計者による判断で 1 つの設計パラメータを選択することとした．さらに，設計要求を満たすために修正すべきノミナル情報には，本手法が対象とした設計パラメータだけでなく，ライフサイクルプロセス間のネットワークやProcedureというパラメータ間の関係も修正対象として挙げられる．すべての修正対象から，すべての設計要求を満たすために最も修正に適したノミナル情報を探索する手法の確立が今後の課題である．

(d) 現実世界との乖離が小さい製品ライフサイクルを表現するうえでの困難性

提案したモデル化手法を用い，現実世界との乖離が小さい製品ライフサイクルを表現するには課題がある．具体的には，3.3 節の要件 3-1 に挙げたような，現実世界における製品の実データを収集し，製品ライフサイクルのノミナル情報モデルに入力する設計パラメータ値やパラメータ間の関係の正当性を高める点に課題がある．設計段階で生成した個体情報と現実世界における個体情報との乖離を埋めるためには，「変動要因」と「ライフサイクル履歴」という 2 つの側面から実データを収集および分析し，設計にフィードバック可能とするべきである．第一に，変動要因に関する実データ（例えば，ユーザの使用頻度，回収率，生産台数）を，製品ライフサイクルのノミナル情報モデルの設計パラメータとして直接入力可能とする必要がある．第二に，ライフサイクル履歴に関する実データを，製品ライフサイクルのノミナル情報モデルに反映するためには，まず「実データのライフサイクル履歴の集合」と「個体情報モデルを用いて表現した個体のライフサイクル履歴の集合」の間の乖離を引き起こしている変動要因を特定する必要がある．次に，特定した変動要因の値を修正し，新たに作成した個体情報モデルと実データとの乖離を分析する．この分析と修正を繰り返しながら，徐々に乖離を埋めていくというアプローチで，実データを活用可能とする手法の構築が必要である．さらに提案手法は，3.3 節の要件 3-2 には対応しておらず，つまり収集した前世代以前の製品の実データを分析して，次世代の製品ライフサイクルの設計にどう活用するかは，設計者に委ねられている．この実データのモデリングへの利用方法を決定するための手法の確立も今後の課題である．

第9章 結論

第 9 章では，本研究の結論，および今後の課題と展望を述べる．

9.1. 本研究の結論

本研究の最終目標は，資源が効率的に循環するような製品ライフサイクルを構築するために，製品個体個別の状態やライフサイクル履歴を設計段階で予測して，製品ライフサイクルを設計するための方法論を構築することにあった．その実現に向けて本研究では，製品ライフサイクルにおける製品に関する情報を，設計者が設計時に決定する設計解の情報を示す「ノミナル情報」と製品個体個別の情報を示す「個体情報」という 2 つに分類し，個体情報を予測しながら製品ライフサイクルのノミナル情報を決定する過程を支援する手法を提案した．これまでにも，製品ライフサイクルの設計過程における特定の設計行為を支援する手法や，製品ライフサイクルを対象製品の 1 個体のライフサイクルにより代表させて表現した設計対象モデルやそのモデリングを支援した計算機環境は提案されている．しかし，現実世界における多様な個体情報を設計段階で模擬し，その仮想的な個体情報を用いて製品ライフサイクルのノミナル情報を決定する過程を円滑に実行可能とする手法は十分に確立しておらず，この点で本研究は新規性があるといえる．

本研究では具体的に，以下の 2 点を示した．

- 製品ライフサイクルをノミナル情報と個体情報の両面から表現するモデル化手法を提案した．本手法を 3 つの要素で構成し，それぞれ以下の点を可能とした．
  (1) 変動要因の表現手法
    - ✧ 製品ライフサイクルの変動要因を 4 種類に分類した．各変動要因を，製品とライフサイクルフローのノミナル情報を表す製品ライフサイクルのノミナル情報モデル [13]を用いて表現可能とした．
  (2) 個体情報を表現した個体情報モデル
    - ✧ ライフサイクルにわたって各個体が示す個体情報を，設計段階で仮想的に，かつシミュレーション実行後も継続的に表現可能とした．本表現によって，製品ライフサイクル全体，各ライフサイクルプロセス，各個体，といった様々な側面や粒度から，資源循環の問題やその原因を予測するという個体情報の評価を可能とした．
  (3) 製品ライフサイクルのノミナル情報モデルから個体情報モデルを作成する手法
    - ✧ 変動要因を表現した製品ライフサイクルのノミナル情報モデルを入力に LCS を実行して個体情報モデルを作成することによって，導出した製品ライフサイクルのノミナル情報についての個体情報の評価を設計段階で可能とした．

- 製品ライフサイクルのノミナル情報モデルと個体情報モデルを用いて，製品ライフサイクルのノミナル情報を決定する過程における設計サイクルの実行を支援する手法を提案した．本支援手法を 2 つの要素で構成し，ライフサイクル設計の設計サイクルにおける以下の点を可能とした．
  (1) 個体情報の評価の実行支援手法
    ✧ 製品ライフサイクルのノミナル情報モデルの要素であり，各ライフサイクルプロセスの振る舞いを表す Procedure の記述を支援する手法を提案した．本手法によって，大半の Procedure が生成可能となり，個体情報の評価を実行するためのノミナル情報モデルの作成過程における，設計者の手間と労力を削減可能とした．
  (2) 製品ライフサイクルの修正案特定支援手法
    ✧ 導出した製品ライフサイクルのノミナル情報について，個体情報の評価を実行した結果が設計要求を満たしていない場合に，修正すべき設計パラメータを特定するための支援手法を提案した．本手法によって，個体情報の評価から新たなノミナル情報の導出過程における，設計者の手間と労力を削減可能とした．

本研究では，上記の設計支援手法を実装した設計支援システムを用い，ケーススタディとしてスマートフォンの製品ライフサイクルを設計した．ケーススタディの結果，提案手法によって，設計対象の製品ライフサイクルのノミナル情報について，状態の違いや量の変動に起因して生じる資源循環の問題やその問題の原因を設計段階で明示的に表現することができた．また，製品ライフサイクルのノミナル情報が設計要求を満たすよう修正すべき設計パラメータを，設計支援システムによる Procedure の生成とその Procedure を用いたパラメータ間の分析を通じ，段階的に絞り込むことができた．本設計支援手法を用いることで，製品ライフサイクルの設計過程における特定の設計段階での設計サイクルを，1 つの対象モデル上で円滑に実行可能であることを確認した．

9.2. 今後の課題と展望

e-PLC-CAD システムを活用し，持続可能な生産システムへの転換を支援するためには，以下の 2 点が解決すべき課題として挙げられる．

(a) ライフサイクル戦略から製品とライフサイクルフローの設計への展開支援

第 2 章で示したように，顧客要求を満たす製品ライフサイクルを，製品とサービスの両面から展開して設計することが，大量生産・大量消費・大量廃棄型の生産システムから持続可能な生産システムへと転換するための有効な手段である．そのためには，ライフサイクル戦略策定段階での「製品コンセプトの策定」「ライフサイクル・オプションの選定」「ビジネス・オプションの選定」から，製品とライフサイクルフローの設計への展開を支援可能な計算機環境を実現し，製品ライフサイクルに対する様々な要求を満たすノミナル情報を段階的に詳細化可能とする必要がある．

(b) ライフサイクル設計とライフサイクルマネジメントの統合支援環境の構築

資源を有効利用するためには，設計段階とマネジメント段階の両方において，個体が効率よく循環するよう製品ライフサイクルをコントロールすることが重要である．そのためには，以下の 3 点から，ライフサイクル設計とライフサイクルマネジメントの実行サイクルを支援する計算機環境が必要である．第一に，導出した製品ライフサイクルのノミナル情報についての個体情報を予測し，予測結果に基づいた適切なノミナル情報を決定可能とする．第二に，設計後のマネジメント段階で，決定したノミナル情報を基にして適切な時期に適切な量の製品個体をメンテナンスや回収できるように，使用中の製品個体をモニタリングし，またモニタリングの結果から製品個体の状態や回収個体数を評価可能とする．第三に，設計段階で予測した個体情報と現実世界における個体情報の乖離を分析し，その分析結果を設計段階にフィードバックして，次世代・次々世代の製品ライフサイクルの設計に活用可能とする．1 つ目は本研究で提案した設計支援手法によって支援可能であるため，2 つ目と 3 つ目の課題を解決する必要がある．特に 3 つ目の課題は，3.3 節の要件 3 で述べたように，資源が効率的に循環するよう製品ライフサイクルを設計するための設計方法論の構築に向けた重要な要素である．

上記の課題を解決した方法論を計算機上で実装し，製品種や世代を超えた製品ライフサイクルシステム全体の設計やマネジメントに活用することで，環境負荷排出量や資源・エネルギー消費量を最小化する持続可能な社会の実現に貢献できると考えている．

謝辞

本論文を執筆するにあたり，多くの方々から御協力と御助言をいただきました．

指導教員であり，本論文の主査を務めてくださった福重真一准教授には，長きにわたり温かくご指導して頂きました．福重准教授は，幅広い知識や柔軟の発想から，研究に行き詰っている私に多くの打開策を提示してくださいました．

本論文の査読を担当していただいた東京大学の梅田靖教授には，学部 4 回生の時から，修士課程での 2 年間，博士課程での 2 年間という計 5 年間にわたり，指導教員を務めて頂きました．研究の進め方，論文の書き方，発表資料の作成方法といった基礎技術から，問題の本質に着目する方法といった研究活動の核となる技術に至るまでを，決して優秀ではなかった私に対し，厳しいながらも的確なアドバイスをして頂きました．

本論文の査読を担当して頂いた，小林英樹教授（機械工学専攻），藤田喜久雄教授（機械工学専攻），荒井栄司教授（マテリアル生産科学専攻）には，深く感謝申し上げます．特に小林教授には，1 年あまりではありましたが，研究室の指導教員として，また企業御出身というお立場から，幅広い分野の多様な知識や考えを提供して頂きました．

本研究を進めるにあたり，有益なコメントを頂きました，木下裕介氏（(独）産業技術総合研究所），水野有智氏（(財）エネルギー総合工学研究所），高本仁志氏（(独）産業技術総合研究所）に深く感謝いたします．

本研究をともに進めてきた國井英輔氏，水野貴広氏，鹿田憲吾氏，松野智彦氏，西岡昌輝氏，松本拓也氏には，研究を進めるうえで必要不可欠なディスカッションに付き合って頂きました．また，梅田研究室，梅田・福重研究室，福重研究室，小林・福重研究室のメンバーには，公私にわたって様々な会話に付き合ってもらい，おかけで研究が苦しい中でも楽しい研究生活を送ることができました．さらに，研究生活を陰で支えてくださった事務補佐員の山下佳子さん，北林淑子さん，阪本知子さんには大変感謝しております．

最後に，研究生活を応援してくれた母と，私生活で私を支えてくれ，研究への活力を与えてくれた妻の麗伊に，感謝の意を表したいと思います．

なお，本研究の成果の一部は，JSPS 科研費（No. No. A267420）の助成を受けたものです．筆者は，2014 年度から 2015 年度にかけて，日本学術振興会特別研究員（DC2）として（独）日本学術振興会より助成を受けました．

引用文献

[1] 吉川弘之, テクノグローブ, 工業調査会, 1993.

[2] United Nations, Department of Economic and Social Affairs, Population Division, “World Population Prospects: The 2012 Revision, Highlights and Advance Tables,” Working Paper No.ESA/P/WP.228, 2013.

[3] 小渋弘明，藤本淳，梅田靖，須賀唯知，小暮啓，「エコデザイン革命-環境とビジネスの両立-」，丸善, 2003.

[4] European Commision , “Analysis associated with the Roadmap to a Resource Efficient Europe, Part I,” European Commission Staff Working Paper, 2011.

[5] 梅田靖，比地原邦彦，大野雅史，小川康暢，小林英樹，服部光郎，増井慶次郎，深野彰，“廃棄要因分析表を用いたライフサイクル戦略決定支援手法の提案,” 精密工学会誌，vol.69，No.9，pp.1270-1276, 2003.

[6] 木村文彦，高橋慎治，田中信寿，梅田靖，永田勝也，「インバース・マニュファクチャリングハンドブック - ポストリサイクルの循環型ものづくり」，丸善, 2004.

[7] 高田祥三，梅田靖，加藤悟，“ライフサイクルデザインへのロードマップ,” 精密工学会誌，vol.66，No.2，pp.1853-1857, 2000.

[8] Alting L, and Jogensen JD, “The Life Cycle Concept as a Basis for Sustainable Industrial Production,” CIRP Annals - Manufacturing Technology, vol.42-1, pp.163-167, 1993.

[9] Umeda Y, Takata S, Kimura F, Tomiyama T, Sutherland JW, Kara S, Herrmann C, and Duflou JR, “Toward integrated product and process life cycle planning - An environmental perspective,” CIRP Annals - Manufacturing Technology, vol.61-2, pp.681-702, 2012.

[10] Umeda Y, Fukushige S, Kunii K, and Matsuyama Y, “LC-CAD: A CAD system for life cycle design,” CIRP Annals - Manufacturing Technology, vol.61-1, pp. 175-178, 2012.

[11] 梅田靖，高田祥三，近藤伸亮，蔵川圭，加藤悟，“環境調和型ビジネスの設計支援技術に向けて,” 精密工学会誌，vol.71, No.10, pp.1214-1218, 2005.

[12] Nguyen DS, Vignat F, and Brissaud D, “Taking into account geometrical variation effect on product performance,” International Journal of Product Lifecycle Management, vol.5, pp.102-121, 2011.

[13] 國井英輔，“ライフサイクル CAD の構築に向けた製品ライフサイクルのモデル化手

法に関する研究,” 大阪大学, 2013.

[14] Umeda Y, Nonomura A, and Tomiyama T, “Study on life-cycle design for the post mass production paradigm,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol.14, pp.149-161, 2000.

[15] 吉川弘之，富山哲男，「設計学 -ものづくりの理論-」，放送大学教育振興会, 2000.

[16] Kimura F, “Life Cycle Design for Inverse Manufacturing,” Proceedings of EcoDesign 1999: 1st International Symposium on Environmentally Conscious Design and Inverse Manufacturing, pp.995-999, 1999.

[17] Ishii K, Eubanks CF, and Marco PD,“Design for product retirement and material life-cycle,” Materials & Design, vol.15, No.4, pp.225-233, 1994.

[18] 鎌田光郎，内田祥一，“レンズ付きフィルム「写ルンです」の循環生産システム,” 富士フィルム研究報告，vol.45，pp.28-34, 2000.

[19] 小川俊一，田渕俊，“複写機外装用樹脂のマテリアルリサイクルに於ける LCA,” リコー技報, 1998.

[20] NEC パーソナル商品総合情報サイト，“NEC Refreshed PC(NEC リフレッシュ PC), ” http://121ware.com/psp/PA121/LEARN/ENTP/h/?tab=LRN_Z_REFRESHEDPC.

[21] Warwick Manufacturing Group, “Design for X,” University of Warwick, 2007.

[22] Shu LH, Wallace DR, and Flowers WC, “Probabilistic Methods in Life-Cycle Design,” Proceedings of the 1996 IEEE International Symposium on Electronics and the Environment, pp.7-12, 1996.

[23] 山際康之，“組立性・分解性,” 精密工学会誌，vol.74，No.8，pp.805-808, 2008.

[24] Takeda H, Veerkamp P, Tomiyama T, and Yoshikawa H, “Modeling Design Processes,” AI Magazine, vol.11, No.4, pp.37-48, 1990.

[25] Wimmer W, and Zust R, “ECODESIGN Pilot,” Springer, 2001.

[26] Pahl G, Wallace K, Beitz W, 設計工学研究グループ（翻訳），「工学設計 - 体系的アプローチ」，培風館, 1995.

[27] 大富浩一，“製品開発における上流設計の重要性とその方法,” 東芝レビュー, vol.60, No.1, 2005.

[28] Bhamra TA, Evans S, McAloone TC, Simon M, Poole S, and Sweatman A, “Integrating environmental decisions into the product development process. I. The early stages,” Proceedings of EcoDesign 1999: 1st International Symposium on Environmentally Conscious Design and Inverse Manufacturing, pp.329-333,

1999.

[29] Rose CM, Ishii K, and Masui K, “How Product Characteristics Determine End-of-Life Strategies,” Proceedings of the 1998 IEEE International Symposium on Electronics and the Environment, pp.322-327, 1998.

[30] Masui K, Sakao T, Aizawa S, and Inaba A, “Quality Function Deployment for Environment (QFDE) to Support Design for Environment (DFE),” ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol.3, pp.852-857, 2002.

[31] Low Bihong，木下裕介，福重真一，梅田靖，鈴木敦，川邊隆男，“重み付け環境配慮設計チェックリストの作成手法に関する研究,” 精密工学会誌，vol.76，No.11, pp.1288-1292, 2010.

[32] Darrell M，中川徹（監修），知識創造研究グループ(翻訳），「体系的技術革新 - 新版矛盾マトリックス Matrix2010 採用（TRIZ 実践と効用）」，クレプス研究所, 2014.

[33] Pugh S, Clausing D, and Andrade R, “Creating Innovative Products Using Total Design: The Living Legacy of Stuart Pugh,” Prentice Hall, 1996.

[34] Brissaud D, and Zwolinski P, “End-of-Life Based Negotiation Throughout the Design Process,” CIRP Annals - Manufacturing Technology, vol.53, pp.155-158, 2004.

[35] Rao RV, and Padmanabhan KK, “Selection of best product end-of-life scenario using diagraph and matrix methods,” Journal of Engineering Design, vol.21, No.4, pp.455-472, 2010.

[36] 小林英樹，「製品ライフサイクルプランニング： ISO/TR14062 の実践」，オーム社, 2003.

[37] Kobayashi H, “Strategic evolution of eco-products: a product life cycle planning methodology,” Research in Engineering Design, vol.16, pp.1-16, 2005.

[38] Arai T, and Shimomura Y, “Proposal of Service CAD System - A Tool for Service Engineering -,” CIRP Annals - Manufacturing Technology, vol.53-1, pp.397-400, 2004.

[39] Mont OK, “Clarifying the concept of product-service system,” Journal of Cleaner Production, vol.10-3, pp.237-245, 2002.

[40] Tukker A, and Tischner U, “Product-services as a research field: past, present and future. Reflections from a decade of research,” Journal of Cleaner Production, vol.14-17, pp.1552-1556, 2006.

[41] Meier H, Roy R, and Seliger G, “Industrial Product-Service Systems–IPS2,”

CIRP Annals - Manufacturing Technology, vol.59, pp.607-627, 2010.

[42] Smith DJ, “Power-by-the-Hour: The Role of Technology in Reshaping Business Strategy at Rolls-Royce,” Technology Analysis & Strategic Management, vol.25, No.8, pp.987-1007, 2013.

[43] IGES Kansai Resarch Centre Discussion Paper, “製品サービスシステム（PSS）とは何か～PSS 研究及び関連政策に関する考察～,” 2006.

[44] 中村信夫，萬代浩平，福重真一，梅田靖，“エコビジネス・アイデア創出支援方法論の開発,” 精密工学会誌，vol.78，No.2，pp.136-142, 2012.

[45] Kondoh S, and Mishima N, “Strategic decision-making method for eco-business planning,” CIRP Annals - Manufacturing Technology, vol.59, pp.41-44, 2010.

[46] Boothroyd G, “Product design for manufacture and assembly,” Computer-Aided Design, vol.26, No.7, pp.505-520, 1993.

[47] Takeuchi S, and Saitou K, “Design for Product-Embedded Disassembly with Maximum Profit,” Proceedings of EcoDesign 2005: 4th International Symposium on Environmentally Conscious Design and Inverse Manufacturing, pp.199-206, 2005.

[48] 宮地直也，白石優実，福重真一，梅田靖，“分割線の付加による製品の易解体性設計手法の提案,” 日本機械学会論文集，vol.80, No.818, 2014.

[49] Shalaby M, and Saito K, “Design for Disassembly With High Stiffness Heat-Reversible Locator-Snap Systems,” Journal of Mechanical Design, vol.130-12, 121701-1-121701-7, 2008.

[50] 永田勝也，松原仁志，小野田弘士，“易解体性を考慮した設計手法の検討 -各種工業製品の解体解析とそれに基づく易解体性の提案-,” 日本機械学会第 11 回環境工学総合シンポジウム 2001 講演論文集，pp.273-276, 2001.

[51] Umeda Y, Kondoh S, Shimomura Y, and Tomiyama T, “Development of design methodology for upgradable products based on functional-behavior-state modeling,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol.19, pp.161-182, 2005.

[52] Yamada S, Yamada T, Nakano K, Bracke S, and Inoue M, “An Environmental Conscious Product Design Method for Sustainability of Product's Value,” Proceedings of Going Green - Care Innovation 2014, 2014.

[53] 親里直彦，小林英樹，“材料の組み合わせと劣化を考慮したリサイクル性評価手法,” 日本機械学会論文集（C 編），vol.70，pp.343-350, 2005.

[54] Umeda Y, Fukushige S, Mizuno T, and Matsuyama Y, “Generating design

alternatives for increasing recyclability of products, ” CIRP Annals - Manufacturing Technology, vol.62, pp.135-138., 2013.

[55] 土井健志，吉村允孝，西脇眞二，泉井一浩，“機械製品のライフサイクル設計のための 3R の視点にもとづく最適化手法,” 日本機械学会論文集（C 編），vol.75，pp.2096-2104, 2009.

[56] Gu P, Hashemian M, and Sosale S, “An Integrated Modular Design Methodology for Life-Cycle Engineering,” Annals of the CIRP, vol.46, pp.71-74, 1997.

[57] Ulrich K, “The role of product architecture in the manufacturing firm,” Research Policy, vol.24, pp.419-440, 1995.

[58] Kimura F, Kato S, Hata T, and Masuda T, “Product Modularization for Parts Reuse in Inverse Manufacturing,” CIRP Annals - Manufacturing Technology, vol.50-1, pp.89-92, 2001.

[59] Yu S, Yang Q, Tao J, Tian X, and Yin F, “Product modular design incorporating life cycle issues - Group Genetic Algorithm (GGA) based method,” Journal of Cleaner Production, vol.19, pp.1016-1032, 2011.

[60] Umeda Y, Fukushige S, Tonoike K, and Kondoh S, “Product Modularity for Life Cycle Design,” CIRP Annals - Manufacturing Technology, vol.57, pp.13-16, 2008.

[61] Seliger G, and Zettl M, “Modularization as an enabler for cycle economy,” CIRP Annals - Manufacturing Technology, vol.57, pp.133-136, 2008.

[62] Koga T, and Aoyama K, “Modular Design Method for Sustainable Life-Cycle of Product Family Considering Future Market Changes,” ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol.5, pp.3-10, 2008.

[63] Kwak M, and Kim HM, “Assessing product family design from an end-of-life perspective,” Engineering Optimization, vol.43-3, pp.233-255, 2010.

[64] Gupta SM, and Gungor A, “Product Recovery Using a Disassembly Line: Challenges and Solution, ” Proceedings of the 2001 IEEE International Symposium on Electronics and the Environment, pp.36-40, 2001.

[65] Wakamatsu H, Matsuishi M, Morinaga E, and Arai E, “Disassembly Support System for Used Products Considering Destruction of Their Parts,” Service Robotics and Mechatronics, pp.37-41, 2010.

[66] Lee HB, Cho NW, and Hong YS, “A hierarchical end-of-life decision model for determining the economic levels of remanufacturing and disassembly under environmental regulations, ” Journal of Cleaner Production, vol.18,

pp.1276-1283, 2010.

[67] Takata S, Kimura F, van Houten FJAM, and Westkamper E, “Maintenance: Changing Role in Life Cycle Management,” CIRP Annals - Manufacturing Technology, vol.53-2, pp.643-655, 2004.

[68] Colledani M, and Tolio T, “Integrated process and system modelling for the design of material recycling systems,” CIRP Annals - Manufacturing Technology, vol.62, pp.447-452, 2013.

[69] International Organisation for Standardisation, “Environmental management – Life cycle assessment – Principles and framework,” ISO 14040, 1997.

[70] Hauschild M, Jeswiet J, and Alting L, “From Life Cycle Assessment to Sustainable Production: Status and Perspectives,” CIRP Annals - Manufacturing Technology, vol.54, pp.1-21, 2005.

[71] Seo KK, Park JH, Jang DS, and Wallace D, “Approximate Estimation of the Product Life Cycle Cost Using Artificial Neural Networks in Conceptual Design,” International Advanced Manufacturing Technology, vol.19, pp.461-471, 2002.

[72] Rebitzer G, and Hunkeler D, “Life Cycle Costing in LCM: Ambitions, Opportunities, and Limitations,” International Journal of Life Cycle Assessment, vol.8, No.5, pp.253-256, 2003.

[73] ステファン・シュミットハイニー，BCSD，BCSD 日本ワーキンググループ，「チェンジング・コース 持続可能な開発への挑戦」，ダイヤモンド社, 1992.

[74] Weizsacker EUW, Lovins LH, Lovins AB, 佐々木健（翻訳），「ファクター4 - 豊かさを 2 倍に、資源消費を半分に」，省エネルギーセンター, 1998.

[75] フリードリヒ・シュミット・ブレーク，「ファクター10 - エコ効率革命を実現する」，シュプリンガー・フェアラーク東京, 1997.

[76] O'Brien M, Doig A, and Cligt R, “Social and Environmental Life Cycle Assessment,” International Journal of LCA, vol.1, No.4, pp.231-237, 1996.

[77] Jorgensen A, Bocq AL, Nazarkina L, and Hauschild M, “Methodologies for Social Life Cycle Assessment,” International Journal of LCA, vol.13, pp.96-103, 2008.

[78] Kumazawa T, and Kobayashi H, “A simulation system to support the establishment of circulated business,” Advanced Engineering Informatics, vol.20, pp.127-136, 2006.

[79] Takata S, and Kimura F, “Life Cycle Simulation System for Life Cycle Process Planning,” CIRP Annals - Manufacturing Technology, vol.52, pp.37-40, 2003.

[80] Komoto H, Tomiyama T, Silvester S, and Brezet H, “Analyzing supply chain

robustness for OEMs from a life cycle perspective using life cycle simulation," International Journal of Production Economics, vol.134, pp.447-457, 2011.

[81] IEC/TR 62635, "Guidelines for end of life information provision from manufacturers and recycles, and for recyclability rate calculation of electrical and electronic equipment," The International Electrotechnical Commission (IEC), ICS 13.020.30; 31.020, 2012.

[82] 日本国経済産業省, "資源循環指標　策定ガイドライン「資源循環指標調査検討委員会」報告書 － 参考資料 A：各種製品のリサイクルに係る指標の設定状況," http://www.meti.go.jp/policy/recycle/main/data/research/12/12_a.html, 2002.06.

[83] Papakostas N, Pintzos G, and Triantafyllou C, "Computer-aided design assessment of products for end of life separation and material handling," CIRP Annals - Manufacturing Technology, vol.64, pp.185-188, 2015.

[84] 梅田靖, 近藤伸亮, 杉野隆, "製品のリユース可能性評価のための「限界リユース率」の提案," 日本機械学会論文集（C 編）, vol.73, pp.339-346, 2007.

[85] 桐山孝司, 富山哲男, 吉川弘之, "設計対象モデル統合化のためのメタモデルの研究," 人工知能学会誌, vol.6, No.3, pp.426-434, 1991.

[86] 武田英明, 富山哲男, 吉川弘之, "知的 CAD の開発のための設計過程の分析と論理による形式化," 精密工学会誌, vol.57, No.6, pp.1047-1052, 1991.

[87] 木村文彦, 高田祥三, 平岡弘之, 小林繁, 岡本政弘, 川島幸司, 大林陽一郎, "統合的設計支援システムのためのモデリング環境," 人工知能学会誌, vol.18, pp.131-137, 2003.

[88] 蔵川圭, 桐山孝司, 馬場康憲, 梅田靖, 小林英樹, "製品ライフサイクル設計支援のためのグリーンブラウザの研究," 精密工学会誌, vol.63, No.12, pp.1685-1689, 1997.

[89] Herrmann C, Frad A, and Luger T, "Integrating the end-of-life evaluation and planning in the product management process," Progress in Industrial Ecology - An International Journal, vol.5, pp.44-64, 2008.

[90] Dassault Systems, "Solid Works," http://www.solidworks.co.jp/.

[91] 井上全人, 山田哲男, 石川晴雄, "設計初期段階から性能と環境負荷削減を両立する 3D-CAD に基づくライフサイクル設計支援システム," 設計工学, vol.49, No.10, pp.45-51, 2014.

[92] Russo D, and Rizzi C, "Structural optimization strategies to design green products," Computers in Industry, vol.65, pp.470-479, 2014.

[93] Eubanks CF, and Ishii K, "AI methods for life-cycle serviceability design of mechanical systems," Artificial Intelligence in Engineering, vol.8, pp.127-140,

1993.

[94] Kalyan-Seshu U, and Bras B, “Integrating DfX Tools with Computer-Aided Design Systems,” Proceedings of DETC 98, 1998 ASME Design Engineering Technical Conferences, DETC98/DAC-562, 1998.

[95] Roche T, Man E, and Browne J, “Development of a CAD integrated DFE workbench tool,” Proceedings of the 2001 IEEE International Symposium on Electronics and the Environment, pp.16-24, 2001.

[96] Murayama T, Hatakenaka S, Narutaki N, Oba F, and Shu LH, “Simulation Based on Petri Nets for Planning Life Cycles and Supply Chains,” Proceedings of JAPAN-U.S.A. Symposium on Flexible Automation, 2000JUSFA-13001, pp.1-6, 2000.

[97] 斉藤岳，青山和浩，古賀毅，“製品ライフサイクルを考慮したモジュール設計に関する研究,” Proceedings of EcoDesign2006, MJ-2, 2006.

[98] Komoto H, “Computer Aided Product Service System Design -Service CAD and its Integration with Life Cycle Simulation-,” Delfut University, 2009.

[99] Fleischmann MF, Bloemhof-Ruwaard JM, Dekker R, Laan E, van Nunen JAEE, and Wassenhove LNV, “Quantitative models for reverse logistics: A review,” European Journal of Operational Research, vol.103, pp.1-17, 1997.

[100] Ma J, Kwak M, and Kim HM, “Demand Trend Mining for Predictive Life Cycle Design,” Journal of Cleaner Production, vol.68, pp.189-199, 2014.

[101] 中島謙一，中村愼一郎，“廃棄物産業連関に基づくマテリアルフロー分析（WIO-MFA）：鉄資源循環分析への適用,” 日本金属学会誌，vol.70, No.8, pp.618-621, 2006.

[102] Takata S, and Sakai T, “Modelling Product Returns Taking Sales Modes into Account,” International Journal of Automation Technology, vol.3, No.1, pp.71-76, 2009.

[103] 小口正弘，亀屋隆志，田崎智宏，谷川昇，浦野紘平，“製品特性に関する数量化分析を用いた電気・電子製品の平均使用年数の推定,” 廃棄物学会論文誌，vol.18, No.3, pp.182-193, 2007.

[104] Blackburn JD, Guide VD, Souza GC, and Wassenhove LNV, “Reverse Supply Chains for Commercial Returns,” California Management Review, vol.46, No.2, pp.6-22, 2004.

[105] Kara S, Rugrungruang F, and Kaebernick H, “Simulation modelling of reverse logistics networks,” International Journal of Production Economics, vol.106,

pp.61-69, 2007.

[106] Murayama T, Yoda M, Eguchi T, and Oba F, “Production Planning and Simulation for Reverse Supply Chain,” International of Conference on Leading Edge Manufacturing in 21st Century, pp.385-390, 2005.

[107] 村山長，畠中伸也，橋本雅文，江口透，大場史憲，“物の循環を含むサプライチェーンのシミュレーションによる多面的評価,” 日本機械学会論文集（C 編），vol.72, pp.2621-2628, 2006.

[108] Demirel NÖ, and Gökçen H, “A mixed integer programming model for remanufacturing in reverse logistics environment,” The International Journal of Advanced Manufacturing Technology, vol.39, pp.1197-1206, 2008.

[109] Bentaha ML, Battaia O, Dolgui A, Hu SJ, “Dealing with uncertainty in disassembly line design,” CIRP Annals - Manufacturing Technology, vol.63, pp.21-24, 2014.

[110] Sudarsan R, Fenves SJ, Sriram RD, and Wnag F, “A Product Information Modeling Framework for Product Lifecycle Management,” Computer-Aided Design, vol.37, pp.1399-1411, 2005.

[111] Kim GY, Lee JY, Park YH, and Noh SD, “Product life cycle information and process analysis methodology: Integrated information and process analysis for product life cycle management,” Concurrent Engineering Research and Applications, vol.20-4, pp.257-274, 2012.

[112] Främling K, Holmström J, Loukkola J, Nyman J, and Kaustell A, “Sustainable PLM through Intelligent Products,” Engineering Applications of Artificial Intelligence, vol.26, pp.789-799, 2013.

[113] Wang L, Wang XV, Gao L, and Vancza J, “A cloud-based approach for WEEE remanufacturing,” CIRP Annals - Manufacturing Technology, vol.63, pp.409-412, 2014.

[114] Kimura F, Matoba Y, and Mitsui K,“Designing Product Reliability based on Total Product Lifecycle Modelling,” CIRP Annals - Manufacturing Technology, vol.56, pp.163-166, 2007.

[115] 立花和夫, 「タグチメソッド入門」, 日経文庫, 2009.

[116] 梅田靖，富山哲男，“人工物のライフサイクル設計,” シミュレーション，vol.15, No.1，pp.32-pp.40, 1996.

[117] Fukushige S, Kunii E, Yamamoto K, and Umeda Y,“A Design Support System for Scenario-Based Lifecycle Design,” 14th Design for Manufacturing and the Life

Cycle Conference (DFMLC), 2011.

[118] SamsungTomorrow, “Galaxy S II Teardown - Splitting 8.9 mm of the latest Samsung Technologies, ” http://global.samsungtomorrow.com/galaxy-s-ii-teardown-splitting-8-9-mm-of-the-latest-samsung-technologies/, 2011.7.12.

[119] movaluate, “ Samsung Galaxy S II Titanium, ” http://www.movaluate.com/samsung/galaxy-s-ii-t-mobile-titanium, 2014.

[120] IT media Mobile, “「GALAXY S II WiMAX ISW11SC」3G＋WiMAX 端末の“中身”を分解して知る,” http://www.itmedia.co.jp/mobile/articles/1206/12/news011.html, 2012.06.12.

[121] 一般社団法人　電気通信事業者協会プレスリリース, “平成 25 年度携帯電話・PHS における　リサイクルの取り組み状況について, ” http://www.tca.or.jp/press_release/2014/0624_638.html, 2014.06.24.

[122] 一般社団法人 情報通信ネットワーク産業協会, “2013 年度 携帯電話の利用実態調査, ” http://www.ciaj.or.jp/jp/pressrelease/pressrelease2013/2013/07/24/10785/, 2013.07.24.

[123] GFK マーケティングサービスジャパン株式会社, “携帯電話の利用実態調査,” http://www.gfk.com/jp/news-and-events/press-room/press-releases/documents/141106%E3%80%80mobile%20phone.pdf, 2014.11.06.

[124] 価格.com, “「携帯電話・スマートフォン」,” http://kakaku.com/.

[125] IHI Pressroom, “Samsung Galaxy S4 Carries $236 Bill of Materials, IHS iSuppli Virtual Teardown Reveals, ” http://press.ihs.com/press-release/design-supply-chain/samsung-galaxy-s4-carries-236-bill-materials-ihs-isuppli-virtual-t, 2013.03.19.

[126] 情報通信ネットワーク産業協会, “「2015 年度 携帯電話の利用実態調査」を実施 ～端末買替え意向が過去最高　市場活性化に向けた選択肢多様化に期待～,” http://www.ciaj.or.jp/jp/pressrelease/pressrelease2015/2015/07/29/13032/, 2015.07.29.

[127] IT media, “ Google Nexus One の部品コストは 174.15 ドル, ” http://www.itmedia.co.jp/news/articles/1001/12/news023.html, 2010.01.12.

[128] 産業環境管理協会, “MiLCA”.

[129] ダイケン化成株式会社, “ダイケン化成 NEWS, ” http://www.daikenkasei.com/news/aydiary.php, 2015.02.26.

[130] 日本マグネシウム協会, “マグネシウム地金価格動向推移, ”

http://magnesium.or.jp/_wp/wp-content/uploads/2013/12/201406kakaku.pdf, 2014.06.

[131] SANYO, “2008年LCA評価結果,” http://panasonic.net/sanyo/environment/jp/product/concept03.html.

[132] IT media Mobile, “「GALAXY S II SC-02C」 一括で5万円台後半、実質3万円台半ばから,” http://www.itmedia.co.jp/mobile/articles/1106/24/news053.html, 2011.06.24.

[133] SquareTrade, “26% of iPhones fail over 2 years, a Decrease from 2009,” 2010.

[134] Flipsen B, Geraedts J, Reinders A, Bakker C, Dafnomilis I, and Gudadhe A, “Environmental sizing of smartphone batteries,” Electronics Goes Green 2012, pp.1-9, 2012.

[135] International Organisation for Standardisation, “Plastics − Determination of Charpy impact properties − Part 1: Non-instrumented impact test,” ISO179-1, 2010.

[136] KDA, “プラスチックの基礎知識 プラスチックの物性,” http://www.kda1969.com/pla_material/.

[137] Battery University, “How to Prolong Lithium-based Batteries,” http://batteryuniversity.com/learn/article/how_to_prolong_lithium_based_batteries.

発表論文

（注）本学位論文と関連のある発表論文に*印を付した.

原著論文（査読付）

1. *松山祐樹，福重真一，小林英樹，梅田靖，“製品個体の集合の循環に着目した製品ライフサイクルの設計支援手法”，設計工学（under revision).
2. *松山祐樹，福重真一，梅田靖，“製品個体の集合を対象とした製品ライフサイクルのモデル化手法”，精密工学会誌，vol.82，No.1，pp.106-114，2016.
3. Matsuyama Y, Fukushige S, and Umeda Y, “Proposal of a Support Method for Identifying Design Requirements on Life Cycle Planning,” International Journal of CAD/CAM, vol. 13, No.2, pp.73-79, 2013.
4. Umeda Y, Fukushige S, Mizuno T, and Matsuyama Y, “Generating design alternatives for increasing recyclability of products,” CIRP Annals - Manufacturing Technology, vol.62-1, pp.135-138, 2013.
5. Umeda Y, Fukushige S, Kunii E, and Matsuyama Y, “LC-CAD: A CAD system for life cycle design,” CIRP Annals - Manufacturing Technology, vol.61-1, pp.175-178, 2012.

国際学会講演論文（査読あり）（口頭発表）

1. *Matsuyama Y, Fukushige S, and Umeda Y, “Simulating Life Cycles of Individual Products for Life Cycle Design,” Procedia CIRP, vol.38, pp.159-164, 2015.
2. *Matsuyama Y, Matsuno T, Fukushige S, and Umeda Y, “Study of Life Cycle Design Focusing on Resource Balance throughout Product Life Cycles,” Procedia CIRP, vol.15, pp.455-460, 2014.
3. Fukushige S, Matsuyama Y, Kunii E, and Umeda Y, “Consistency Management System between Product Design and the Lifecycle,” Proceedings of ASME 2013 IDETC/CIE: 18th Design for Manufacturing and the Life Cycle Conference, DETC2013-13575, Portland, USA, (August 2013).
4. Fukushige S, Mizuno T, Matsuyama Y, Kunii E, and Umeda Y, “Quantitative Design Modification for the Recyclability of Products”, Proceedings of 20th CIRP International Conference on Life Cycle Engineering, pp.27-33, Singapore, (April 2013).
5. Matsuyama Y, Kunii E, Fukushige S, and Umeda Y, “Proposal of Life Cycle Planning Support Method for Life Cycle CAD,” 2012 Asian Conference on Design and Digital Engineering, USB, 100085, Hokkaido, Japan, (December 2012).

6. Fukushige S, **Matsuyama Y**, Kunii E, and Umeda Y, “A Computational Design Environment for Product Lifecycle,” 2012 Asian Conference on Design and Digital Engineering, USB, 100097, Hokkaido, Japan, (December 2012).

国際学会講演論文（採択審査のみ）（口頭発表）

7. Fukushige S, **Matsuyama Y**, and Umeda Y, “A Computational Design Environment for Product Lifecycle Design,” Proceedings of International Symposium on Ultraprecision Engineering and Nanotechnology (ISUPEN2015), pp.13-16, Tokyo, Japan, (March 2015).
8. Fukushige S, **Matsuyama Y**, and Umeda Y, “Change Impact Analysis on Design Parameters for Recyclability,” Proceedings of EcoDesign 2013: 8th International Symposium on Environmentally Conscious Design and Inverse Manufacturing, O-D-19, Jeju Island, Korea, (December 2013).
9. **Matsuyama Y**, Kunii E, Fukushige S, and Umeda Y, “A CAD System for Product Life Cycle Design,” EcoBalance 2012, D1-02, Tokyo, Japan, (November 2012).

国際学会講演論文（採択審査のみ）（ポスター発表）

10. **Matsuyama Y**, Kunii E, Yamamoto K, Fukushige S, and Umeda Y, “Proposal of Design Environment for Life Cycle Scenario,” Proceedings of EcoDesign 2011: 7th International Symposium on Environmentally Conscious Design and Inverse Manufacturing, G1-13, pp.821-826, Kyoto, Japan, (November 2011).

国内学会講演論文（採択審査のみ）（口頭発表）

1. *松山祐樹，西岡昌輝，福重真一，梅田靖，“製品個体の集合を対象としたライフサイクルCADのコンセプト”，Designシンポジウム 2014，USB，東京，11 月 11 日-13 日，2014.
2. *松山祐樹，西岡昌輝，福重真一，梅田靖，“製品個体の集合を対象とした製品ライフサイクルのモデリング手法の提案”，2014 年度精密工学会秋季大会学術講演会，B21，鳥取，9 月 16 日-18 日，2014.
3. *松本拓也，松山祐樹，福重真一，梅田靖，“ライフサイクルCADを用いたライフサイクルシミュレーションモデルの作成支援手法”, 2014 年度精密工学会秋季大会学術講演会，B20，鳥取，9 月 16 日-18 日，2014.
4. *松山祐樹，松野智彦，福重真一，梅田靖，“多世代製品ライフサイクルの統合設計支援手法の提案”，日本機械学会第 23 回設計工学・システム部門講演会，2305，沖縄，10 月 23 日-25 日，2013.
5. *松野智彦，松山祐樹，福重真一，梅田靖，“多世代製品ライフサイクル設計支援環境

の構築（第 1 報）基本コンセプト”，2013 年度精密工学会秋季学術講演会，M36，大阪，2013 年 9 月 12 日-14 日，2013.

6. 松山祐樹，水野貴広，福重真一，梅田靖，“ライフサイクル戦略の策定とその製品設計への展開を支援するCADシステムの開発（第 6 報 リサイクル性向上のための製品ライフサイクル設計支援手法）”，2013 年度精密工学会春季大会学術講演会，F33，pp.409-410，東京，3 月 13 日-15 日，2013.
7. 福重真一, 國井英輔, 松山祐樹, 梅田靖, “製品とライフサイクルの設計を統合するCADシステム”，Designシンポジウム 2012，pp.193-196，京都，10 月 16 日-17 日，2012.
8. 水野貴広，谷野敏樹，松山祐樹，國井英輔，福重真一，梅田靖，“ライフサイクルシナリオに基づく製品設計のための計算機環境の開発（第 4 報 液晶TVを対象にしたリサイクル性設計への適用）”，日本機械学会第 22 回設計工学・システム部門講演会，USB, 1103，pp.20-29，広島，9 月 26 日-28 日，2012.
9. 松山祐樹，國井英輔，鹿田憲吾，福重真一，梅田靖，“ライフサイクルシナリオに基づく製品設計のための計算機環境の開発（第 3 報 ライフサイクルを通じて変化する製品の表現）”, 日本機械学会第 22 回設計工学･システム部門講演会, USB, 1102, pp.11-19, 広島，9 月 26 日-28 日，2012.
10. 福重真一，谷野敏樹，水野貴広，松山祐樹，國井英輔，梅田靖，“製品のリサイクル可能率に基づくリサイクル性設計手法”，2012 年度精密工学会秋季学術講演会，B38，PP.131-132，福岡，9 月 14 日-16 日，2012.
11. 松山祐樹，國井英輔，松浦剛，福重真一，梅田靖，“ライフサイクル戦略の策定とその製品設計への展開を支援するCADシステムの開発（第 3 報 ライフサイクルフロー設計プロセスの提案”, 2011 年度精密工学会秋季大会学術講演会, L79, pp.735-736, 石川, 9 月 20 日-22 日，2011.

付録

- **FrontGlassManuf**

表 A - 1：FrontGlassManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForManufacturingTouchScreen	17.500	$/unit
	GlassManufCO2	0.015	kg-CO2/yen
	PCManufCO2	7.082	kg-CO2/kg
	yenToTheDollar	100	yen/$

表 A - 2：FrontGlassManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	FrontGlass	List<EntityData>	monthly
	FrontGlassSecond	List<EntityData>	monthly
	FrontGlassManufCost	double	continuous
	FrontGlassManufCO2	double	continuous

```
[[FrontGlass]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("FrontGlass","Plan1",LCSimulator.CurrentTime));
[[FrontGlassSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("FrontGlassSecond","Plan2",LCSimulator.CurrentTime));

for(int i = 0; i< [[FrontGlass]].Count; i++)
{
    [[FrontGlassManufCost]] += [[CostForManufacturingTouchScreen]] * [[yenToTheDollar]];
    [[FrontGlassManufCO2]] += [[CostForManufacturingTouchScreen]] * [[yenToTheDollar]] * [[GlassManufCO2]];
    [[FrontGlassManufCO2]] += [[FrontGlass]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]];
}

for(int i = 0; i< [[FrontGlassSecond]].Count; i++)
{
    [[FrontGlassManufCost]] += [[CostForManufacturingTouchScreen]] * [[yenToTheDollar]];
    [[FrontGlassManufCO2]] += [[CostForManufacturingTouchScreen]] * [[yenToTheDollar]] * [[GlassManufCO2]];
    [[FrontGlassManufCO2]] += [[FrontGlassSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]];
}
```

図 A - 1：FrontGlassManuf プロセスの Procedure

- **AntennaManuf**

表 A - 3：AntennaManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	PCManufCost	290	yen/kg
	PCManufCO2	7.082	kg-CO2/kg
	SteelManufCO2	1.979	kg-CO2/kg

表 A - 4：AntennaManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	Antenna	List<EntityData>	monthly
	AntennaSecond	List<EntityData>	monthly
	AntennaManufCost	double	continuous
	AntennaManufCO2	double	continuous

```
[[Antenna]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("Antenna","Plan1",LCSimulator.CurrentTime));
[[AntennaSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("AntennaSecond","Plan2",LCSimulator.CurrentTime));

for(int i = 0; i< [[Antenna]].Count; i++)
{
    [[AntennaManufCost]] += [[Antenna]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCost]];
    [[AntennaManufCO2]] += [[Antenna]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]];
    [[AntennaManufCO2]] += [[Antenna]][i].WeightOfConstituentMaterial(“Steel") / 1000 * [[SteelManufCO2]];
}

for(int i = 0; i< [[AntennaSecond]].Count; i++)
{
    [[AntennaManufCost]] += [[AntennaSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCost]];
    [[AntennaManufCO2]] += [[AntennaSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]];
    [[AntennaManufCO2]] += [[AntennaSecond]][i].WeightOfConstituentMaterial(“Steel") / 1000 * [[SteelManufCO2]];
}
```

図 A - 2：AntennaManuf プロセスの Procedure

- SpeakerManuf

表 A - 5：SpeakerManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	SpeakerManfacturingCO2	0.878	kg-CO2/unit

表 A - 6：SpeakerManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	Speaker	List<EntityData>	monthly
	SpeakerSecond	List<EntityData>	monthly
	SpeakerManufCost	double	continuous
	SpeakerManufCO2	double	continuous

```
[[Speaker]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("Speaker","Plan1",LCSimulator.CurrentTime));
[[SpeakerSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("SpeakerSecond","Plan2",LCSimulator.CurrentTime));

[[SpeakerManufCO2]] += [[Speaker]].Count * [[SpeakerManufacturingCO2]];
[[SpeakerManufCO2]] += [[SpeakerSecond]].Count * [[SpeakerManufacturingCO2]];
```

図 A - 3：SpeakerManuf プロセスの Procedure

- **OLEDManuf**

表 A - 7：OLEDManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForManufacturingOLED	21.4	$/unit
	OLEDManufCO2	33.4	kg-CO2/unit
	yenToTheDollar	100	yen/$

表 A - 8：OLEDManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	OLED	List<EntityData>	monthly
	OLEDSecond	List<EntityData>	monthly
	OLEDManufCost	double	continuous
	OLEDManufCO2	double	continuous

```
[[OLED]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("OLED","Plan1",LCSimulator.CurrentTime));
[[OLEDSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("OLEDSecond","Plan2",LCSimulator.CurrentTime));

[[OLEDManufCost]] += [[OLED]].Count * [[CostForManufacturingOLED]] * [[yenToTheDollar]];
[[OLEDManufCO2]] += [[OLED]].Count * [[OLEDManufCO2]];

[[OLEDManufCost]] += [[OLEDSecond]].Count * [[CostForManufacturingOLED]] * [[yenToTheDollar]];
[[OLEDManufCO2]] += [[OLEDSecond]].Count * [[OLEDManufCO2]];
```

図 A - 4：OLEDManuf プロセスの Procedure

- **BottomBoardManuf**

表 A - 9：BottomBoardManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForMicroUSBBoardManuf	150.00	yen/unit
	IntegratedCircuitManufCO2	0.0017	kg-CO2/yen
	PCBoardManufCO2	$3.53*10^{-3}$	kg-CO2/yen

表 A - 10：BottomBoardManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	MicroUSBBoard	List<EntityData>	monthly
	MicroUSBBoardSecond	List<EntityData>	monthly
	Button	List<EntityData>	monthly

	ButtonSecond	List<EntityData>	monthly
	Vibrator	List<EntityData>	monthly
	VibratorSecond	List<EntityData>	monthly
	BottomBoardManufCost	double	continuous
	ButtonBoardManufCO2	double	continuous

```
[[MicroUSBBoard]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("MicroUSBBoard","Plan1",LCSimulator.CurrentTime));
[[MicroUSBBoardSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("MicroUSBBoardSecond","Plan2",LCSimulator.CurrentTime));

[[Button]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("Button","Plan1",LCSimulator.CurrentTime));
[[ButtonSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("ButtonSecond","Plan2",LCSimulator.CurrentTime));

[[Vibrator]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("Vibrator","Plan1",LCSimulator.CurrentTime));
[[VibratorSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("VibratorSecond","Plan2",LCSimulator.CurrentTime));


[[BottomBoardManufCost]] += [[MicroUSBBoard]].Count * [[CostForMicroUSBBoardManuf]];
[[BottomBoardManufCO2]] += [[MicroUSBBoard]].Count * [[IntegratedCircuitManufCO2]] * [[CostForMicroUSBBoardManuf]] / 2;
[[BottomBoardManufCO2]] += [[MicroUSBBoard]].Count * [[PCBoardManufCO2]] * [[CostForMicroUSBBoardManuf]] / 2;

[[BottomBoardManufCost]] += [[MicroUSBBoardSecond]].Count * [[CostForMicroUSBBoardManuf]];
[[BottomBoardManufCO2]] += [[MicroUSBBoardSecond]].Count * [[IntegratedCircuitManufCO2]] * [[CostForMicroUSBBoardManuf]] / 2;
[[BottomBoardManufCO2]] += [[MicroUSBBoardSecond]].Count * [[PCBoardManufCO2]] * [[CostForMicroUSBBoardManuf]] / 2;
```

図 A - 5：BottomBoardManuf プロセスの Procedure

- LensManuf

表 A - 11：LensManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForManufacturingLens	12.70	$/unit
	CostForManufacturingFrontFacingCamera	3.80	$/unit
	LensManufCO2	0.0037	kg-CO2/yen
	yenToTheDollar	100	yen/$

表 A - 12：LensManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	ReusedLensAmount	double	monthly
Output parameter	Lens	List<EntityData>	monthly
	LensSecond	List<EntityData>	monthly
	FrontFacingCamera	List<EntityData>	monthly
	FrontFacingCameraSecond	List<EntityData>	monthly
	CameraManufCost	double	continuous
	CameraManufCO2	double	continuous
	ManufacturedCameraAmount	double	continuous

```
[[Lens]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("Lens","Plan1",LCSimulator.CurrentTime));
for(int i = 0 ; i< LCSFunction.ProductionAmount("Plan2",LCSimulator.CurrentTime) - [[ReusedLensAmount]]; i++)
{
   EntityData newEntity = LCSFunction.MakeComponent("LensSecond", LCSimulator.CurrentTime);
   [[LensSecond]].Add(newEntity);
}

[[FrontFacingCamera]].AddRange
(LCSFunction.MakeComponentBasedOnProductionPlan("FrontFacingCamera","Plan1",LCSimulator.CurrentTime));
[[FrontFacingCameraSecond]].AddRange
(LCSFunction.MakeComponentBasedOnProductionPlan("FrontFacingCameraSecond","Plan2",LCSimulator.CurrentTime));

[[CameraManufCost]] += [[Lens]].Count * [[CostForManufacturingLens]] * [[yenToTheDollar]];
[[CameraManufCost]] += [[FrontFacingCamera]].Count * [[CostForManufacturingFrontFacingCamera]] * [[yenToTheDollar]];
[[CameraManufCO2]] += [[Lens]].Count * [[CostForManufacturingLens]] * [[yenToTheDollar]] * [[LensManufCO2]];
[[CameraManufCO2]] += [[FrontFacingCamera]].Count * [[CostForManufacturingFrontFacingCamera]] * [[yenToTheDollar]] * [[LensManufCO2]];

[[CameraManufCost]] += [[LensSecond]].Count * [[CostForManufacturingLens]] * [[yenToTheDollar]];
[[CameraManufCost]] += [[FrontFacingCameraSecond]].Count * [[CostForManufacturingFrontFacingCamera]] * [[yenToTheDollar]];
[[CameraManufCO2]] += [[LensSecond]].Count * [[CostForManufacturingLens]] * [[yenToTheDollar]] * [[LensManufCO2]];
[[CameraManufCO2]] += [[FrontFacingCameraSecond]].Count * [[CostForManufacturingFrontFacingCamera]] * [[yenToTheDollar]] * [[LensManufCO2]];

[[ManufacturedCameraAmount]] += [[Lens]].Count;
```

図 A - 6：LensManuf プロセスの Procedure

- **ImageProcessorManuf**

表 A - 13：ImageProcessorManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForManufacturingImageProcessor	15.20	$/unit
	IntegratedCircuitManufCO2	$1.71*10^{-3}$	kg-CO2/yen
	yenToTheDollar	100	yen/$

表 A - 14：ImageProcessorManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	ReusedCameraAmount	double	monthly
Output parameter	ImageProcessor	List<EntityData>	monthly
	ImageProcessorSecond	List<EntityData>	monthly
	ImageProcessorManufCost	double	continuous
	ImageProcessorManufCO2	double	continuous

```
[[ImageProcessor]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("ImageProcessor","Plan1",LCSimulator.CurrentTime));
for(int i = 0 ; i< LCSFunction.ProductionAmount("Plan2",LCSimulator.CurrentTime) - [[ReusedCameraAmount]]; i++)
{
    EntityData newEntity = LCSFunction.MakeComponent("ImageProcessorSecond", LCSimulator.CurrentTime);
    [[ImageProcessorSecond]].Add(newEntity);
}

[[ImageProcessorManufCost]] += [[ImageProcessor]].Count * [[CostForManufacturingImageProcessor]] * [[yenToTheDollar]];
[[ImageProcessorManufCO2]] += [[ImageProcessor]].Count * [[CostForManufacturingImageProcessor]] * [[yenToTheDollar]] * [[IntegratedCircuitManufCO2]];

[[ImageProcessorManufCost]] += [[ImageProcessorSecond]].Count * [[CostForManufacturingImageProcessor]] * [[yenToTheDollar]];
[[ImageProcessorManufCO2]] += [[ImageProcessorSecond]].Count * [[CostForManufacturingImageProcessor]] * [[yenToTheDollar]] * [[IntegratedCircuitManufCO2]];
```

図 A - 7：ImageProcessorManuf プロセスの Procedure

- **BatteryManuf**

表 A - 15：BatteryManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForManufacturingBattery	4.90	$/unit
	LiIonBatteryManufCO2	1.32	kg-CO2/unit
	yenToTheDollar	100	yen/$

表 A - 16：BatteryManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	Battery	List<EntityData>	monthly
	BatterySecond	List<EntityData>	monthly
	BatteryManufCost	double	continuous
	BatteryManufCO2	double	continuous

```
[[Battery]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("Battery","Plan1",LCSimulator.CurrentTime));
[[BatterySecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("BatterySecond","Plan2",LCSimulator.CurrentTime));

[[BatteryManufCost]] += [[Battery]].Count * [[CostForManufacturingBattery]] * [[yenToTheDollar]];
[[BatteryManufCO2]] += [[Battery]].Count * [[LiIonBatteryManufCO2]];

[[BatteryManufCost]] += [[BatterySecond]].Count * [[CostForManufacturingBattery]] * [[yenToTheDollar]];
[[BatteryManufCO2]] += [[BatterySecond]].Count * [[LiIonBatteryManufCO2]];
```

図 A - 8：BatteryManuf プロセスの Procedure

- **BatteryCoverManuf**

表 A - 17：BatteryCoverManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	PCManufCost	290	yen/kg
	PCManufCO2	7.082	kg-CO2/kg
	RateForPCGlass	1.2	—

表 A - 18：BatteryCoverManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	BatteryCover	List<EntityData>	monthly
	BatteryCoverSecond	List<EntityData>	monthly
	BatteryCoverManufCost	double	continuous
	BatteryCoverManufCO2	double	continuous

```
[[BatteryCover]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("BatteryCover","Plan1",LCSimulator.CurrentTime));
[[BatteryCoverSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("BatteryCoverSecond","Plan2",LCSimulator.CurrentTime));

for(int i = 0 ; i< [[BatteryCover]].Count; i++)
{
    [[BatteryCoverManufCO2]] += [[BatteryCover]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]] * [[RateForPCGlass]];
    [[BatteryCoverManufCost]] += [[BatteryCover]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCost]] * [[RateForPCGlass]];
}

for(int i = 0 ; i< [[BatteryCoverSecond]].Count; i++)
{
    [[BatteryCoverManufCO2]] += [[BatteryCoverSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]] * [[RateForPCGlass]];
    [[BatteryCoverManufCost]] += [[BatteryCoverSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCost]] * [[RateForPCGlass]];
}
```

図 A - 9：BatteryCoverManuf プロセスの Procedure

- **MainBoardManuf**

表 A - 19：MainBoardManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForManufacturingProcessor	17.50	$/unit
	CostForManufacturingWirelessSection	14.50	$/unit
	CostForManufacturingWLAN	8.20	$/unit
	CostForManufacturingMemory	29.00	$/unit
	CostForManufacturingIntegratedCircuit	7.00	$/unit
	IntegratedCircuitManufCO2	$1.71*10^{-3}$	kg-CO2/yen
	SilverManufacturingCO2	57.09	kg-CO2/kg
	CopperManufacturingCO2	2.69	kg-CO2/kg
	GoldManufacturingCO2	$1.40*10^{4}$	kg-CO2/kg
	yenToTheDollar	100	yen/$

表 A - 20：MainBoardManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	MainBoard	List<EntityData>	monthly
	MainBoardSecond	List<EntityData>	monthly
	MainBoardManufCost	double	continuous
	MainBoardManufCO2	double	continuous

```
[[MainBoard]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("MainBoard","Plan1",LCSimulator.CurrentTime));
[[MainBoardSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("MainBoardSecond","Plan2",LCSimulator.CurrentTime));

for(int i = 0 ; i< [[MainBoard]].Count; i++)
{
    [[MainBoardManufCost]] +=  [[CostForManufacturingProcessor]] * [[yenToTheDollar]];
    [[MainBoardManufCost]] +=  [[CostForManufacturingWirelessSection]] * [[yenToTheDollar]];
    [[MainBoardManufCost]] +=  [[CostForManufacturingWLAN]] * [[yenToTheDollar]];
    [[MainBoardManufCost]] +=  [[CostForManufacturingMemory]] * [[yenToTheDollar]];
    [[MainBoardManufCost]] +=  [[CostForManufacturingIntegratedCircuit]] * [[yenToTheDollar]];

    [[MainBoardManufCO2]] += [[CostForManufacturingIntegratedCircuit]] * [[yenToTheDollar]] * [[IntegratedCircuitManufCO2]];
    [[MainBoardManufCO2]] += [[MainBoard]][i].WeightOfConstituentMaterial("Silver") * [[SilverManufacturingCO2]] /1000;
    [[MainBoardManufCO2]] += [[MainBoard]][i].WeightOfConstituentMaterial("Copper") * [[CopperManufacturingCO2]] /1000;
    [[MainBoardManufCO2]] += [[MainBoard]][i].WeightOfConstituentMaterial("Gold") * [[GoldManufacturingCO2]] /1000;
}

for(int i = 0 ; i< [[MainBoardSecond]].Count; i++)
{
    [[MainBoardManufCost]] +=  [[CostForManufacturingProcessor]] * [[yenToTheDollar]];
    [[MainBoardManufCost]] +=  [[CostForManufacturingWirelessSection]] * [[yenToTheDollar]];
    [[MainBoardManufCost]] +=  [[CostForManufacturingWLAN]] * [[yenToTheDollar]];
    [[MainBoardManufCost]] +=  [[CostForManufacturingMemory]] * [[yenToTheDollar]];
    [[MainBoardManufCost]] +=  [[CostForManufacturingIntegratedCircuit]] * [[yenToTheDollar]];

    [[MainBoardManufCO2]] += [[CostForManufacturingIntegratedCircuit]] * [[yenToTheDollar]] * [[IntegratedCircuitManufCO2]];
    [[MainBoardManufCO2]] += [[MainBoardSecond]][i].WeightOfConstituentMaterial("Silver") * [[SilverManufacturingCO2]] /1000;
    [[MainBoardManufCO2]] += [[MainBoardSecond]][i].WeightOfConstituentMaterial("Copper") * [[CopperManufacturingCO2]] /1000;
    [[MainBoardManufCO2]] += [[MainBoardSecond]][i].WeightOfConstituentMaterial("Gold") * [[GoldManufacturingCO2]] /1000;
}
```

図 A - 10：MainBoardManuf プロセスの Procedure

- CenterPanelManuf

表 A - 21：CenterPanelManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	PCManufCost	290.00	yen/kg
	MagnesiumManufCost	275.00	yen/kg
	PCManufCO2	7.08	kg-CO2/kg
	MagnesiumManufCO2	29.00	kg-CO2/kg

表 A - 22：CenterPanelManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	ReusedCenterPanelAmount	double	monthly
Output parameter	CenterPanel	List<EntityData>	monthly
	CenterPanelSecond	List<EntityData>	monthly
	CenterPanelManufCost	double	continuous
	CenterPanelManufCO2	double	continuous
	ManufacturedCenterPanelAmount	double	continuous

```
[[CenterPanel]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("CenterPanel","Plan1",LCSimulator.CurrentTime));
for(int i = 0; i< LCSFunction.ProductionAmount("Plan2",LCSimulator.CurrentTime) - [[ReusedCenterPanelAmount]]; i++)
{
    [[CenterPanelSecond]].Add(LCSFunction.MakeComponent("CenterPanelSecond", LCSimulator.CurrentTime));
}

for(int i = 0; i< [[CenterPanel]].Count; i++)
{
   [[CenterPanelManufCost]] += [[CenterPanel]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCost]];
   [[CenterPanelManufCost]] += [[CenterPanel]][i].WeightOfConstituentMaterial("Magnesium") / 1000 * [[MagnesiumManufCost]];
   [[CenterPanelManufCO2]] += [[CenterPanel]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]];
   [[CenterPanelManufCO2]] += [[CenterPanel]][i].WeightOfConstituentMaterial("Magnesium") / 1000 * [[MagnesiumManufCO2]];
}

for(int i = 0 ; i< [[CenterPanelSecond]].Count; i++)
{
   [[CenterPanelManufCost]] += [[CenterPanelSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCost]];
   [[CenterPanelManufCost]] += [[CenterPanelSecond]][i].WeightOfConstituentMaterial("Magnesium") / 1000 * [[MagnesiumManufCost]];
   [[CenterPanelManufCO2]] += [[CenterPanelSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]];
   [[CenterPanelManufCO2]] += [[CenterPanelSecond]][i].WeightOfConstituentMaterial("Magnesium") / 1000 * [[MagnesiumManufCO2]];
}

[[ManufacturedCenterPanelAmount]] += [[CenterPanel]].Count;
```

図 A - 11：CenterPanelManuf プロセスの Procedure

- **RearPanelManuf**

表 A - 23：RearPanelManuf プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	PCManufCost	290.00	yen/kg
	PCManufCO2	7.08	kg-CO2/kg

表 A - 24：RearPanelManuf プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Output parameter	RearPanel	List<EntityData>	monthly
	RearPanelSecond	List<EntityData>	monthly
	RearPanelManufCost	double	continuous
	RearPanelManufCO2	double	continuous

```
[[RearPanel]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("RearPanel","Plan1",LCSimulator.CurrentTime));
[[RearPanelSecond]].AddRange(LCSFunction.MakeComponentBasedOnProductionPlan("RearPanelSecond","Plan2",LCSimulator.CurrentTime));

for(int i = 0 ; i< [[RearPanel]].Count; i++)
{
   [[RearPanelManufCost]] += [[RearPanel]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCost]];
   [[RearPanelManufCO2]] += [[RearPanel]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]];
}

for(int i = 0 ; i< [[RearPanelSecond]].Count; i++)
{
   [[RearPanelManufCost]] += [[RearPanelSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCost]];
   [[RearPanelManufCO2]] += [[RearPanelSecond]][i].WeightOfConstituentMaterial("PC") / 1000 * [[PCManufCO2]];
}
```

図 A - 12：RearPanelManuf プロセスの Procedure

- **AssemblyForFirst**

表 A - 25：AssemblyForFirst プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForAssembly	8.00	$/unit
	yenToTheDollar	100	yen/$

表 A - 26：AssemblyForFirst プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	CenterPanel	List<EntityData>	monthly
	Antenna	List<EntityData>	monthly
	Battery	List<EntityData>	monthly
	MainBoard	List<EntityData>	monthly
	RearPanel	List<EntityData>	monthly
	FrontGlass	List<EntityData>	monthly
	MicroUSBBoard	List<EntityData>	monthly
	BatteryCover	List<EntityData>	monthly
	OLED	List<EntityData>	monthly
	Speaker	List<EntityData>	monthly
	Lens	List<EntityData>	monthly
	ImageProcessor	List<EntityData>	monthly
	Button	List<EntityData>	monthly
	FrontFacingCamera	List<EntityData>	monthly
	Vibrator	List<EntityData>	monthly
Output parameter	SmartPhone	List<EntityData>	monthly
	ProductAssemblyCostForFirst	double	continuous

```
for(int i = 0 ; i < [[CenterPanel]].Count; i++ )
{
   List<EntityData> componentList = new List<EntityData>();

    componentList.Add([[CenterPanel]][i]);
    componentList.Add([[Antenna]][i]);
    componentList.Add([[Battery]][i]);
    componentList.Add([[MainBoard]][i]);
    componentList.Add([[RearPanel]][i]);
    componentList.Add([[FrontGlass]][i]);
    componentList.Add([[MicroUSBBoard]][i]);
    componentList.Add([[BatteryCover]][i]);
    componentList.Add([[OLED]][i]);
    componentList.Add([[Speaker]][i]);
    componentList.Add([[Lens]][i]);
    componentList.Add([[ImageProcessor]][i]);
    componentList.Add([[Button]][i]);
    componentList.Add([[FrontFacingCamera]][i]);
    componentList.Add([[Vibrator]][i]);
    [[SmartPhone]].Add(LCSFunction.Assembly(componentList,"SmartPhone"));
}

[[ProductAssemblyCostForFirst]] += [[SmartPhone]].Count * [[CostForAssembly]] * [[yenToTheDollar]];
```

図 A - 13：AssemblyForFirst プロセスの Procedure

- CameraModuleAssemblyForSecond

表 A - 27：CameraModuleAssemblyForSecond プロセスの
Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	LensSecond	List<EntityData>	monthly
	ImageProcessorSecond	List<EntityData>	monthly
Output parameter	NewCameraModuleSecond	List<EntityData>	monthly

```
for(int i = 0 ; i < [[LensSecond]].Count; i++ )
{
   List<EntityData> componentList = new List<EntityData>();

    componentList.Add([[LensSecond]][i]);
    componentList.Add([[ImageProcessorSecond]][i]);
    [[NewCameraModuleSecond]].Add(LCSFunction.Assembly(componentList,“NewCameraModuleSecond"));
}
```

図 A - 14：CameraModuleAssemblyForSecond プロセスの Procedure

- **CameraModuleWarehouse**

表 A - 28：CameraModuleWarehouse プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	NewCameraModuleSecond	List<EntityData>	monthly
	UsedCameraModule	List<EntityData>	monthly
Output parameter	CameraModuleSecond	List<EntityData>	monthly

```
[[CameraModuleSecond]].AddRange([[NewCameraModuleSecond]]);
[[CameraModuleSecond]].AddRange([[UsedCameraModule]]);
```

図 A - 15：CameraModuleWarehouse プロセスの Procedure

- **CenterPanelWarehouse**

表 A - 29：CenterPanelWarehouse プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	NewCenterPanelSecond	List<EntityData>	monthly
	UsedCenterPanel	List<EntityData>	monthly
Output parameter	CenterPanelSecond	List<EntityData>	monthly

```
[[CenterPanelSecond]].AddRange([[NewCenterPanelSecond]]);
[[CenterPanelSecond]].AddRange([[UsedCenterPanel]]);
```

図 A - 16：CenterPanelWarehouse プロセスの Procedure

- **AssemblyForSecond**

表 A - 30：AssemblyForSecond プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CostForAssembly	8.00	$/unit
	yenToTheDollar	100	yen/$

表 A - 31：AssemblyForSecond プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	CenterPanelSecond	List<EntityData>	monthly
	AntennaSecond	List<EntityData>	monthly
	BatterySecond	List<EntityData>	monthly
	MainBoardSecond	List<EntityData>	monthly
	RearPanelSecond	List<EntityData>	monthly
	FrontGlassSecond	List<EntityData>	monthly
	MicorUSBBoardSecond	List<EntityData>	monthly
	BatteryCoverSecond	List<EntityData>	monthly
	OLEDSecond	List<EntityData>	monthly
	SpeakerSecond	List<EntityData>	monthly
	CameraModuleSecond	List<EntityData>	monthly
	ButtonSecond	List<EntityData>	monthly
	FrontFacingCameraSecond	List<EntityData>	monthly
	VibratorSecond	List<EntityData>	monthly
Output parameter	SmartPhoneSecond	List<EntityData>	monthly
	ProductAssemblyCostForSecond	double	continuous

```
for(int i = 0 ; i < [[CenterPanelSecond]].Count; i++ )
{
  List<EntityData> componentList = new List<EntityData>();

   componentList.Add([[CenterPanelSecond]][i]);
   componentList.Add([[AntennaSecond]][i]);
   componentList.Add([[BatterySecond]][i]);
   componentList.Add([[MainBoardSecond]][i]);
   componentList.Add([[RearPanelSecond]][i]);
   componentList.Add([[FrontGlassSecond]][i]);
   componentList.Add([[MicroUSBBoardSecond]][i]);
   componentList.Add([[BatteryCoverSecond]][i]);
   componentList.Add([[OLEDSecond]][i]);
   componentList.Add([[SpeakerSecond]][i]);
   componentList.Add([[CameraModuleSecond]][i]);
   componentList.Add([[ButtonSecond]][i]);
   componentList.Add([[FrontFacingCameraSecond]][i]);
   componentList.Add([[VibratorSecond]][i]);
   [[SmartPhoneSecond]].Add(LCSFunction.Assembly(componentList, "SmartPhoneSecond "));
}

[[ProductAssemblyCostForSecond]] += [[SmartPhoneSecond]].Count * [[CostForAssembly]] * [[yenToTheDollar]];
```

図 A - 17：AssemblyForSecond プロセスの Procedure

- **Transportation**

表 A - 32 : Transportation プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	TransportationDistance	1,000.00	km
	FuelConsumption	6.13	km/L
	LoadCapacity	4,000.00	kg
	CostForBoxContents	6.00	$/unit
	LightOilUseCost	60.00	yen/L
	LightOilUseCO2	2.60	kg-CO2/L
	yenToTheDollar	100	yen/$

表 A - 33 : Transportation プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	assembledSmartPhone	List<EntityData>	monthly
	assembledSmartPhoneSecond	List<EntityData>	monthly
Output parameter	transportedSmartPhone	List<EntityData>	monthly
	transportedSmartPhoneSecond	List<EntityData>	monthly
	LightOilUse	double	monthly
	weightAllProduct	double	monthly
	TransportationCost	double	continuous
	TransportationCO2	double	continuous

```
[[transportedSmartPhone]].AddRange([[assembledSmartPhone]]);
[[transportedSmartPhoneSecond]].AddRange([[assembledSmartPhoneSecond]]);

for(int i = 0; i < [[transportedSmartPhone]].Count; i++)
{
   [[weightAllProduct]] += [[transportedSmartPhone]][i].IndividualWeight;
}

for(int i = 0; i < [[transportedSmartPhoneSecond]].Count; i++)
{
   [[weightAllProduct]] += [[transportedSmartPhoneSecond]][i].IndividualWeight;
}

[[LightOilUse]] = ([[weightAllProduct]] / 1000) * [[TransportationDistance]] / ( [[FuelConsumption]] * [[LoadCapacity]]);
[[TransportationCost]] += [[LightOilUse]] * [[LightOilUseCost]];
[[TransportationCost]] += [[CostForBoxContents]] * [[yenToTheDollar]] * [[transportedSmartPhone]].Count;
[[TransportationCost]] += [[CostForBoxContents]] * [[yenToTheDollar]] * [[transportedSmartPhoneSecond]].Count;
[[TransportationCO2]] += [[LightOilUse]] * [[LightOilUseCO2]];
```

図 A - 18 : Transportation プロセスの Procedure

- **Sale**

表 A - 34：Sale プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	salesPrice	50,000	yen/unit
	discountedRate	0.99	—
	rateOfHeavyUser	0.2	—
	rateOfStandardUser	0.6	—
	rateOfLightUser	0.2	—

表 A - 35：Sale プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	smartPhone	List<EntityData>	monthly
	smartPhoneSecond	List<EntityData>	monthly
	reusedNumber	double	monthly
Output parameter	soldSmartPhone	List<EntityData>	monthly
	soldSmartPhoneSecond	List<EntityData>	monthly
	totalSales	double	continuous

```
[[soldSmartPhone]].AddRange([[smartPhone]]);
[[soldSmartPhoneSecond]].AddRange([[smartPhoneSecond]]);

for(int i = 0; i < [[soldSmartPhone]].Count; i++)
{
    if( i <= [[soldSmartPhone]].Count * [[ratioOfHeavyUser]])
    {
        LCSFunction.SetTargetUser([[soldSmartPhone]][i], "HeavyUser");
    }
    else if([[soldSmartPhone]].Count * [[ratioOfHeavyUser]] < i
            && i <= [[soldSmartPhone]].Count * ([[ratioOfHeavyUser]] + [[ratioOfStandardUser]]))
    {
        LCSFunction.SetTargetUser([[soldSmartPhone]][i], "StandardUser");
    }
    else if([[soldSmartPhone]].Count * (1 - [[ratioOfLightUser]]) < i)
    {
        LCSFunction.SetTargetUser([[soldSmartPhone]][i], "LightUser");
    }
}

for(int i = 0; i < [[soldSmartPhoneSecond]].Count; i++)
{
    if( i <= [[soldSmartPhoneSecond]].Count * [[ratioOfHeavyUser]])
    {
        LCSFunction.SetTargetUser([[soldSmartPhoneSecond]][i], "HeavyUser");
    }
    else if([[soldSmartPhoneSecond]].Count * [[ratioOfHeavyUser]] < i
            && i <= [[soldSmartPhoneSecond]].Count * ([[ratioOfHeavyUser]] + [[ratioOfStandardUser]]))
    {
        LCSFunction.SetTargetUser([[soldSmartPhoneSecond]][i], "StandardUser");
    }
    else if([[soldSmartPhoneSecond]].Count * (1 - [[ratioOfLightUser]]) < i)
    {
        LCSFunction.SetTargetUser([[soldSmartPhoneSecond]][i], "LightUser");
    }
}

[[totalSales]] += [[soldSmartPhone]].Count * [[salesPrice]];
[[totalSales]] += ([[soldSmartPhoneSecond]].Count - [[reusedNumber]] + [[reusedNumber]] * [[discountedRate]]) * [[salesPrice]];
```

図 A - 19：Sale プロセスの Procedure

- **Operation**

表 A - 36：Operation プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	costForCharging	0.12	yen
	UseElectoricityCO2	0.41	kg-CO2/kWh

表 A - 37：Operation プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	smartPhone	List<EntityData>	monthly
	smartPhoneSecond	List<EntityData>	monthly
	keepUsingSmartPhone WithoutObsolescence	List<EntityData>	monthly
	keepUsingSmartPhoneSecond WithoutObsolescence	List<EntityData>	monthly
Output parameter	keepUsingSmartPhone	List<EntityData>	monthly
	keepUsingSmartPhoneSecond	List<EntityData>	monthly
	TotalUseCost	double	continuous
	TotalUseCO2	double	continuous

```
[[keepUsingSmartPhone]].AddRange([[smartPhone]]);
[[keepUsingSmartPhone]].AddRange([[keepUsingSmartPhoneWithoutObsolescence]]);
[[keepUsingSmartPhoneSecond]].AddRange([[smartPhoneSecond]]);
[[keepUsingSmartPhoneSecond]].AddRange([[keepUsingSmartPhoneSecondWithoutObsolescence]]);

[[keepUsingSmartPhone]] = LCSFunction.UseOneTurn([[keepUsingSmartPhone]]);
for(int i = 0; i < [[keepUsingSmartPhone]].Count; i++)
{
     EntityData imageProcessor
     = [[keepUsingSmartPhone]][i].SearchChildEntityByName([[keepUsingSmartPhone]][i], "ImageProcessor");
    imageProcessor.RemainingPhysicalLifetime -= LCSFunction.GetUserDataByUserName(imageProcessor.TargetUser).Coefficient;
     EntityData battery = LCSFunction.SearchChildEntityFromName([[keepUsingSmartPhone]][i], "Battery");
     double capacityRate =  (-1.6786/50*(battery.IndividualPerformanceList["Capacity"].Invariable/1650) + 1) * 100;
     double previousChargeNumber = battery.IndividualIndicationofUsage;
     double previousCapacity = battery.IndividualPerformanceList["Capacity"].Invariable;
     if[[keepUsingSmartPhone]][i].TargetUser == "HeavyUser")
     {
         battery.IndividualIndicationofUsage += 30 * (0.011 *(100 - capacityRate) + 0.71);
      }
     else if([[keepUsingSmartPhone]][i].TargetUser == "StandardUser")
     {
         battery.IndividualIndicationofUsage += 30 * (0.007 *(100 - capacityRate) + 0.49);
      }
      else
      {
         battery.IndividualIndicationofUsage += 30 * (0.005 *(100 - capacityRate) + 0.37);
      }

      battery.IndividualPerformanceList["Capacity"].Invariable = ( -1.6786/50 * battery.IndividualIndicationofUsage + 94.571)/100;

     [[TotalUseCost]] += [[costForCharging]] * (battery.IndividualIndicationofUsage - previousChargeNumber)
                         *(3.7 * battery.IndividualPerformanceList["Capacity"].Invariable/1000000 * 0.8);
     [[TotalUseCO2]] += ([[UseElectoricityCO2]]/1000)*(battery.IndividualIndicationofUsage - previousChargeNumber)
                         *(3.7*battery.IndividualPerformanceList["Capacity"].Invariable/1000000 * 0.8);
}

[[keepUsingSmartPhoneSecond]] = LCSFunction.UseOneTurn([[keepUsingSmartPhoneSecond]]);
for(int i = 0; i < [[keepUsingSmartPhoneSecond]].Count; i++)
{
     EntityData battery = LCSFunction.SearchChildEntityFromName([[keepUsingSmartPhoneSecond]][i], "2ndBattery");
     double capacityRate =  (-1.6786/50*(battery.IndividualPerformanceList["Capacity"].Invariable/1650) + 1) * 100;
     double previousChargeNumber = battery.IndividualIndicationofUsage;
     double previousCapacity = battery.IndividualPerformanceList["Capacity"].Invariable;
     if([[keepUsingSmartPhoneSecond]][i].TargetUser == "HeavyUser")
     {
         battery.IndividualIndicationofUsage += 30 * (0.011 *(100 - capacityRate) + 0.71);
      }
     else if([[keepUsingSmartPhoneSecond]][i].TargetUser == "StandardUser")
     {
         battery.IndividualIndicationofUsage += 30 * (0.007 *(100 - capacityRate) + 0.49);
      }
      else
      {
         battery.IndividualIndicationofUsage += 30 * (0.005 *(100 - capacityRate) + 0.37);
      }

      battery.IndividualPerformanceList["Capacity"].Invariable = ( -1.6786/50 * battery.IndividualIndicationofUsage + 94.571)/100;

    [[TotalUseCost]] += [[costForCharging]] * (battery.IndividualIndicationofUsage - previousChargeNumber)
                         *(3.7 * battery.IndividualPerformanceList["Capacity"].Invariable/1000000 * 0.8);
    [[TotalUseCO2]] += ([[UseElectoricityCO2]]/1000)*(battery.IndividualIndicationofUsage - previousChargeNumber)
                         *(3.7*battery.IndividualPerformanceList["Capacity"].Invariable/1000000 * 0.8);
}
```

図 A - 20：Operation プロセスの Procedure

- **DisposingBasedOnBatteryCapacity**

表 A - 38：DisposingBasedOnBatteryCapacity プロセスの
Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	keepUsingSmartPhone	List<EntityData>	monthly
	keepUsingSmartPhoneSecond	List<EntityData>	monthly
Output parameter	keepUsingSmartPhone WithEnoughBatteryCapacity	List<EntityData>	monthly
	disposedSmartPhone DueToBatteryCapacity	List<EntityData>	monthly
	keepUsingSmartPhoneSecond WithEnoughBatteryCapacity	List<EntityData>	monthly
	disposedSmartPhoneSecond DueToBatteryCapacity	List<EntityData>	monthly

```
[[keepUsingSmartPhoneWithEnoughBatteryCapacity]].AddRange([[keepUsingSmartPhone]]);
foreach(EntityData entity in [[keepUsingSmartPhoneWithEnoughBatteryCapacity]])
{
    EntityData battery = LCSFunction.SearchChildEntityFromName(entity, "Battery");
     double capacityRate =  (-1.6786/50*(battery.IndividualPerformanceList["Capacity"].Invariable/1650) + 1) * 100;
    if(entity.TargetUser == "PlanForHeavy" && (-0.011 *(100 - capacityRate) + 1.4) < 1
    || entity.TargetUser == "PlanForStandard" && (-0.007 *(100 - capacityRate) + 2.05) < 1
    || entity.TargetUser == "PlanForLight" && (-0.005 *(100 - capacityRate) + 1.35) < 1)
    {
        [[disposedSmartPhoneDueToBatteryCapacity]].Add(entity);
    }
}

foreach(EntityData entity in [[disposedSmartPhoneDueToBatteryCapacity]])
{
   [[keepUsingSmartPhoneWithEnoughBatteryCapacity]].Remove(entity);
}

[[keepUsingSmartPhoneSecondWithEnoughBatteryCapacity]].AddRange([[keepUsingSmartPhoneSecond]]);
foreach(EntityData entity in [[keepUsingSmartPhoneSecondWithEnoughBatteryCapacity]])
{
    EntityData battery = LCSFunction.SearchChildEntityFromName(entity, “2ndBattery");
     double capacityRate =  (-1.6786/50*(battery.IndividualPerformanceList["Capacity"].Invariable/1650) + 1) * 100;
    if(entity.TargetUser == "PlanForHeavy" && (-0.011 *(100 - capacityRate) + 1.4) < 1
    || entity.TargetUser == "PlanForStandard" && (-0.007 *(100 - capacityRate) + 2.05) < 1
    || entity.TargetUser == "PlanForLight" && (-0.005 *(100 - capacityRate) + 1.35) < 1)
    {
        [[disposedSmartPhoneSecondDueToBatteryCapacity]].Add(entity);
    }
}

foreach(EntityData entity in [[disposedSmartPhoneSecondDueToBatteryCapacity]])
{
   [[keepUsingSmartPhoneSecondWithEnoughBatteryCapacity]].Remove(entity);
}
```

図 A - 21：DisposingBasedOnBatteryCapacity プロセスの Procedure

- **DisposingBasedOnMalfunction**

表 A - 39：DisposingBasedOnMalfunction プロセスの
Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	keepUsingSmartPhone WithEnoughBatteryCapacity	List<EntityData>	monthly
	keepUsingSmartPhoneSecond WithEnoughBatteryCapacity	List<EntityData>	monthly
Output parameter	keepUsingSmartPhone WithoutMalfunction	List<EntityData>	monthly
	disposedSmartPhoneDueToMalfunction	List<EntityData>	monthly
	keepUsingSmartPhoneSecond WithoutMalfunction	List<EntityData>	monthly
	disposedSmartPhoneSecond DueToMalfunction	List<EntityData>	monthly

```
[[keepUsingSmartPhoneWithoutMalfunction]].AddRange([[keepUsingSmartPhoneWithEnoughBatteryCapacity]]);
[[disposedSmartPhoneDueToMalfunction]] =
LCSFunction.DisposeBasedOnFailureRateOfComponents([[keepUsingSmartPhoneWithoutMalfunction]]);
foreach(EntityData entity in [[disposedSmartPhoneDueToMalfunction]])
{
  [[keepUsingSmartPhoneWithoutMalfunction]].Remove(entity);
}

[[keepUsingSmartPhoneSecondWithoutMalfunction]].AddRange([[keepUsingSmartPhoneSecondWithEnoughBatteryCapacity]]);
[[disposedSmartPhoneSecondDueToMalfunction]] =
LCSFunction.DisposeBasedOnFailureRateOfComponents([[keepUsingSmartPhoneSecondWithoutMalfunction]]);
foreach(EntityData entity in [[disposedSmartPhoneSecondDueToMalfunction]])
{
  [[keepUsingSmartPhoneSecondWithoutMalfunction]].Remove(entity);
}
```

図 A - 22：DisposingBasedOnMalfunction プロセスの Procedure

- **DisposingBasedOnAccident**

表 A - 40 : DisposingBasedOnAccident プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	keepUsingSmartPhone WithoutMalfunction	List<EntityData>	monthly
	keepUsingSmartPhoneSecond WithoutMalfunction	List<EntityData>	monthly

Output parameter	keepUsingSmartPhone WithoutAccident	List<EntityData>	monthly
	disposedSmartPhoneDueToAccident	List<EntityData>	monthly
	keepUsingSmartPhoneSecond WithoutAccident	List<EntityData>	monthly
	disposedSmartPhoneSecond DueToAccident	List<EntityData>	monthly

```
[[keepUsingSmartPhoneWithoutAccident]].AddRange([[keepUsingSmartPhoneWithoutMalfunction]]);
[[disposedSmartPhoneDueToAccident]]=
LCSFunction.DisposeBasedOnFailureRate([[keepUsingSmartPhoneWithoutAccident]], "absoluteElapsedTurn");
foreach(EntityData entity in [[disposedSmartPhoneDueToAccident]])
{
    [[keepUsingSmartPhoneWithoutAccident]].Remove(entity);
}

[[keepUsingSmartPhoneSecondWithoutAccident]].AddRange([[keepUsingSmartPhoneSecondWithoutMalfunction]]);
[[disposedSmartPhoneSecondDueToAccident]]=
LCSFunction.DisposeBasedOnFailureRate([[keepUsingSmartPhoneSecondWithoutAccident]], "absoluteElapsedTurn");
foreach(EntityData entity in [[disposedSmartPhoneSecondDueToAccident]])
{
    [[keepUsingSmartPhoneSecondWithoutAccident]].Remove(entity);
}
```

図 A - 23：DisposingBasedOnAccident プロセスの Procedure

- **DisposingBasedOnObsolescence**

表 A - 41：DisposingBasedOnObsolescence プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	ObsolescenceConstantValue	$2.00*10^{-4}$	—
	ObsolescenceStartTurn	12.00	month
	ObsolescenceSlope	0.014	—
	ObsolescenceValueAt24	0.30	—

表 A - 42：DisposingBasedOnObsolescence プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	keepUsingSmartPhone WithoutAccident	List<EntityData>	monthly
	keepUsingSmartPhoneSecond WithoutAccident	List<EntityData>	monthly

Output parameter	keepUsingSmartPhone WithoutObsolescence	List<EntityData>	monthly
	disposedSmartPhone DueToObsolescence	List<EntityData>	monthly
	keepUsingSmartPhoneSecond WithoutObsolescence	List<EntityData>	monthly
	disposedSmartPhoneSecond DueToObsolescence	List<EntityData>	monthly

```
Random cRandom = new Random();

[[keepUsingSmartPhoneWithoutObsolescence]].AddRange([[keepUsingSmartPhoneWithoutAccident]]);
foreach(EntityData entity in [[keepUsingSmartPhoneWithoutObsolescence]])
{
   double dRandom = cRandom.NextDouble();
   if((entity.ElapsedTime < [[ObsolescenceStartTurn]]  && dRandom < [[obsolescenceConstantValue]])
     || (entity.ElapsedTime == 24 &&  dRandom < [[ObsolescenceValueAt24]])
     || (entity.ElapsedTime >= [[ObsolescenceStartTurn]] &&  entity.ElapsedTime != 24 &&
         dRandom < [[ObsolescenceSlope]] * (entity.ElapsedTime - [[ObsolescenceStartTurn]]) + [[obsolescenceConstantValue]]))
   {
        [[disposedSmartPhoneDueToObsolescence]].Add(entity);
    }
}
foreach(EntityData entity in  [[disposedSmartPhoneDueToObsolescence]])
{
   [[keepUsingSmartPhoneWithoutObsolescence]].Remove(entity);
}

[[keepUsingSmartPhoneSecondWithoutObsolescence]].AddRange([[keepUsingSmartPhoneSecondWithoutAccident]]);
foreach(EntityData entity in [[keepUsingSmartPhoneSecondWithoutObsolescence]])
{
   double dRandom = cRandom.NextDouble();
   if((entity.ElapsedTime < [[ObsolescenceStartTurn]]  && dRandom < [[obsolescenceConstantValue]])
     || (entity.ElapsedTime == 24 &&  dRandom < [[ObsolescenceValueAt24]])
     || (entity.ElapsedTime >= [[ObsolescenceStartTurn]] &&  entity.ElapsedTime != 24 &&
         dRandom < [[ObsolescenceSlope]] * (entity.ElapsedTime - [[ObsolescenceStartTurn]]) + [[obsolescenceConstantValue]]))
   {
        [[disposedSmartPhoneSecondDueToObsolescence]].Add(entity);
    }
}
foreach(EntityData entity in  [[disposedSmartPhoneSecondDueToObsolescence]])
{
   [[keepUsingSmartPhoneSecondWithoutObsolescence]].Remove(entity);
}
```

図 A - 24：DisposingBasedOnObsolesence プロセスの Procedure

- **CollectionOfDisposingProducts**

表 A - 43：CollectionOfDisposingProducts プロセスの
Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	disposedSmartPhone DueToBatteryCapacity	List<EntityData>	monthly
	disposedSmartPhoneSecond DueToBatteryCapacity	List<EntityData>	monthly

	disposedSmartPhoneDueToMalfunction	List<EntityData>	monthly
	disposedSmartPhoneSecond DueToMalfunction	List<EntityData>	monthly
	disposedSmartPhoneDueToAccident	List<EntityData>	monthly
	disposedSmartPhoneSecond DueToAccident	List<EntityData>	monthly
	disposedSmartPhoneDueToObsolescence	List<EntityData>	monthly
	disposedSmartPhoneSecond DueToObsolescence	List<EntityData>	monthly
Output parameter	disposedSmartPhone	List<EntityData>	monthly
	disposedSmartPhoneSecond	List<EntityData>	monthly

```
[[disposedSmartPhone]].AddRange([[disposedSmartPhoneDueToBatteryCapacity]]);
[[disposedSmartPhone]].AddRange([[disposedSmartPhoneDueToMalfunction]]);
[[disposedSmartPhone]].AddRange([[disposedSmartPhoneDueToAccident]]);
[[disposedSmartPhone]].AddRange([[disposedSmartPhoneDueToObsolescence]]);

[[disposedSmartPhoneSecond]].AddRange([[disposedSmartPhoneSecondDueToBatteryCapacity]]);
[[disposedSmartPhoneSecond]].AddRange([[disposedSmartPhoneSecondDueToMalfunction]]);
[[disposedSmartPhoneSecond]].AddRange([[disposedSmartPhoneSecondDueToAccident]]);
[[disposedSmartPhoneSecond]].AddRange([[disposedSmartPhoneSecondDueToObsolescence]]);
```

図 A - 25：CollectionOfDisposingProducts プロセスの Procedure

- **CollectedOrUncollected**

表 A - 44：CollectedOrUncollected プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	CollectionRate	85.00	%
	collectionDeposit	2,000	yen/unit
	lastTurnForReusing	32	month
	currentCollectionRate	22.00	%

表 A - 45：CollectedOrUncollected プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	disposedSmartPhone	List<EntityData>	monthly
Output parameter	collectedSmartPhone	List<EntityData>	monthly
	uncollectedSmartPhone	List<EntityData>	monthly
	CollectionCost	double	continuous

```
[[uncollectedSmartPhone]].AddRange([[disposedSmartPhone]]);
for(int i = 0 ; i < [[CollectionRate]] / 100 * [[disposedSmartPhone]].Count ; i++)
{
  [[collectedSmartPhone]].Add([[uncollectedSmartPhone]][i]);
}

foreach(EntityData entity in [[collectedSmartPhone]])
{
  [[uncollectedSmartPhone]].Remove(entity);
}

if(LCSimulator.CurrentTime < [[lastTurnForReusing]] && [[CollectionRate]] >= [[currentCollectionRate]])
{
  [[CollectionCost]] += ((([[CollectionRate]] - [[currentCollectionRate]]) / 100) * [[collectedSmartPhone]].Count * [[collectionDeposit]];
}
```

図 A - 26：CollectedOrUncollected プロセスの Procedure

- **Disassembly**

表 A - 46：Disassembly プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	disassemblyTime	10.00	min/unit
	laborCost	2,000	yen/hour
	lastTurnForReusing	32	month

表 A - 47：Disassembly プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	collectedSmartPhone	List<EntityData>	monthly
Output parameter	CameraModule	List<EntityData>	monthly
	CenterPanel	List<EntityData>	monthly
	RearPanel	List<EntityData>	monthly
	MicroUSBBoard	List<EntityData>	monthly
	Battery	List<EntityData>	monthly
	MainBoard	List<EntityData>	monthly
	BatteryCover	List<EntityData>	monthly
	OLED	List<EntityData>	monthly
	FrontGlass	List<EntityData>	monthly
	Antenna	List<EntityData>	monthly
	FrontFacingCamera	List<EntityData>	monthly
	Speaker	List<EntityData>	monthly
	Vibrator	List<EntityData>	monthly
	Button	List<EntityData>	monthly
	DisassemblyCost	double	continuous

```
foreach(EntityData entity in [[collectedSmartPhone]])
{
    foreach(EntityData childEntity in entity.ChildEntityList.Values)
    {
      if(childEntity.NominalInformation.Name=="CenterPanel")
      {
        [[CenterPanel]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="RearPanel")
      {
        [[RearPanel]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="MicroUSBBoard")
      {
        [[MicroUSBBoard]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Battery")
      {
        [[Battery]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="MainCameraModule")
      {
        [[CameraModule]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="MainBoard")
      {
        [[MainBoard]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="BatteryCover")
      {
        [[BatteryCover]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="OLED")
      {
        [[OLED]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="FrontGlass")
      {
        [[FrontGlass]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Antenna")
      {
        [[Antenna]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="FrontFacingCamera")
      {
        [[FrontFacingCamera]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Speaker")
      {
        [[Speaker]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Vibrator")
      {
        [[Vibrator]].Add(childEntity);
      }

      if(childEntity.NominalInformation.Name=="Button")
      {
        [[Button]].Add(childEntity);
      }
    }
   [[DisassemblyCost]] += [[disassemblyTime]] * [[laborCost]] / 60;
}
```

図 A - 27：Disassembly プロセスの Procedure

- **InspectionForCenterPanel**

表 A - 48：InspectionForCenterPanel プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	InspectionFee	300	yen/unit
	lastTurnForReusing	32	month
	ReusableConditionForCenterPanel	64.00	J/m

表 A - 49：InspectionForCenterPanel プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	CenterPanel	List<EntityData>	monthly
Output parameter	ReusableCenterPanel	List<EntityData>	monthly
	UnreusableCenterPanel	List<EntityData>	monthly
	CostForInspectionForCenterPanel	double	continuous

```
foreach(EntityData entity in [[CenterPanel]])
{
  if(entity.IndividualPerformanceList["CharpyImpactStrength"].Value(LCSimulator.CurrentTime) > [[ReusableConditionForCenterPanel]])
  {
    if(LCSimulator.CurrentTime < [[lastTurnForReusing]])
    {
      [[ReusableCenterPanel]].Add(entity);
    }
    else
    {
      [[UnreusableCenterPanel]].Add(entity);
    }
  }
  else
  {
    [[UnreusableCenterPanel]].Add(entity);
  }

  if(LCSimulator.CurrentTime < [[lastTurnForReusing]])
  {
    [[CostForInspectionForCenterPanel]] += [[InspectionFee]];
  }
}
```

図 A - 28：InspectionForCenterPanel プロセスの Procedure

- **InspectionForCameraModule**

表 A - 50：InspectionForCameraModule プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	InspectionFee	300	yen/unit
	lastTurnForReusing	32	month
	ReusableConditionForCameraModule	7.9984	million pixels

表 A - 51：InspectionForCameraModule プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	CameraModule	List<EntityData>	monthly
Output parameter	ReusableCameraModule	List<EntityData>	monthly
	UnreusableCameraModule	List<EntityData>	monthly
	CostForInspectionForCameraModule	double	continuous

```
foreach(EntityData entity in [[CameraModule]])
{
  if(entity.IndividualPerformanceList["EffectivePixels"].Value(LCSimulator.CurrentTime) > [[ReusableConditionForCameraModule]])
  {
    if(LCSimulator.CurrentTime < [[lastTurnForReusing]])
    {
      [[ReusableCameraModule]].Add(entity);
    }
    else
    {
      [[UnreusableCameraModule]].Add(entity);
    }
  }
  else
  {
    [[UnreusableCameraModule]].Add(entity);
  }

  if(LCSimulator.CurrentTime < [[lastTurnForReusing]])
  {
     [[CostForInspectionForCameraModule]] += [[InspectionFee]];
   }
}
```

図 A - 29：InspectionForCameraModule プロセスの Procedure

- **Repair**

表 A - 52：Repair プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	ReusableCenterPanel	List<EntityData>	monthly
	ReusableCameraModule	List<EntityData>	monthly
Output parameter	RepairedCenterPanel	List<EntityData>	monthly
	RepairedCameraModule	List<EntityData>	monthly

```
[[RepairedCenterPanel]].AddRange([[ReusableCenterPanel]]);
[[RepairedCameraModule]].AddRange([[ReusableCameraModule]]);
```

図 A - 30：Repair プロセスの Procedure

- **ReusedCenterPanelWarehouse**

表 A - 53：ReusedCenterPanelWarehouse プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	lastTurnForReusing	32	month

表 A - 54：ReusedCenterPanelWarehouse プロセスの
Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	RepairedCenterPanel	List<EntityData>	monthly
	UsedCenterPanelToWarehouse	List<EntityData>	monthly
Output parameter	UsedCenterPanelInWarehouse	List<EntityData>	monthly

```
[[UsedCenterPanelInWarehouse]].AddRange([[RepairedCenterPanel]]);
[[UsedCenterPanelInWarehouse]].AddRange([[UsedCenterPanelToWarehouse]]);

if(LCSimulator.CurrentTime >= [[lastTurnForReusing]])
{
  [[UsedCenterPanelInWarehouse]] = new List<EntityData>();
}
```

図 A - 31：ReusedCenterPanelWarehouse プロセスの Procedure

- **ReusedCenterPanelShipping**

表 A - 55：ReusedCenterPanelShipping プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	ChargesForCustody	300	yen/unit
	startTurnForReusing	14	month
	ShareOfReusePartsInProduct	0.63	—

表 A - 56：ReusedCenterPanelShipping プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	UsedCenterPanelInWarehouse	List<EntityData>	monthly
Output parameter	UsedCenterPanel	List<EntityData>	monthly
	UsedCenterPanelToWarehouse	List<EntityData>	monthly
	maximumNumberOfShipping ForCenterPanel	double	monthly
	CostForWarehouse ForReusedCenterPanel	double	continuous
	ReusedCenterPanelAmount	double	monthly
	TotalReusedCenterPanelAmount	double	continuous

```
[[maximumNumberOfShippingForCameraModule]]
= LCSFunction.ProductionAmount("Plan2", (LCSimulator.CurrentTime + 1)) * [[ShareOfReusePartsInProduct]];

[[UsedCameraModuleToWarehouse]].AddRange([[UsedCameraModuleInWarehouse]]);
for(int i = 0 ; i < [[maximumNumberOfShippingForCameraModule]]; i++)
{
    if(i < [[UsedCameraModuleToWarehouse]].Count && [[startTurnForReusing]] <= LCSimulator.CurrentTime)
    {
        [[UsedCameraModule]].Add([[UsedCameraModuleToWarehouse]][i]);
     }
}

foreach(EntityData entity in [[UsedCameraModule]])
{
   [[UsedCameraModuleToWarehouse]].Remove(entity);
}

[[CostForWarehouseForReusedCameraModule]] += [[ChargesForCustody]] * [[UsedCameraModuleToWarehouse]].Count;
[[ReusedCameraModuleAmount]] = [[UsedCameraModule]].Count;
[[TotalReusedCameraModuleAmount]] += [[UsedCameraModule]].Count;
```

図 A - 32：ReusedCenterPanelShipping プロセスの Procedure

- **ReusedCameraModuleWarehouse**

表 A - 57：ReusedCameraModuleWarehouse プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	lastTurnForReusing	32	month

表 A - 58：ReusedCameraModuleWarehouse プロセスの
Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	RepairedCameraModule	List<EntityData>	monthly
	UsedCameraModuleToWarehouse	List<EntityData>	monthly
Output parameter	UsedCameraModuleInWarehouse	List<EntityData>	monthly

```
[[UsedCameraModuleInWarehouse]].AddRange([[RepairedCameraModule]]);
[[UsedCameraModuleInWarehouse]].AddRange([[UsedCameraModuleToWarehouse]]);

if(LCSimulator.CurrentTime >= [[lastTurnForReusing]])
{
  [[UsedCameraModuleInWarehouse]] = new List<EntityData>();
}
```

図 A - 33：ReusedCameraModuleWarehouse プロセスの Procedure

- **ReusedCameraModuleShipping**

表 A - 59：ReusedCameraModuleShipping プロセスの Given parameter

	Parameter	Value	Unit
Given parameter	ChargesForCustody	300	yen/unit
	startTurnForReusing	14	month
	ShareOfReusePartsInProduct	0.63	—

表 A - 60：ReusedCameraModuleShipping プロセスの
Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	UsedCamera ModuleInWarehouse	List<EntityData>	monthly
Output parameter	UsedCameraModule	List<EntityData>	monthly
	UsedCameraModuleToWarehouse	List<EntityData>	monthly

	maximumNumberOfShipping ForCameraModule	double	monthly
	CostForWarehouse ForReusedCameraModule	double	continuous
	ReusedCameraModuleAmount	double	monthly
	TotalReusedCameraModuleAmount	double	continuous

```
[[maximumNumberOfShippingForCameraModule]]
= LCSFunction.ProductionAmount("Plan2", (LCSimulator.CurrentTime + 1)) * [[ShareOfReusePartsInProduct]];

[[UsedCameraModuleToWarehouse]].AddRange([[UsedCameraModuleInWarehouse]]);
for(int i = 0 ; i < [[maximumNumberOfShippingForCameraModule]]; i++)
{
    if(i < [[UsedCameraModuleToWarehouse]].Count && [[startTurnForReusing]] <= LCSimulator.CurrentTime)
    {
        [[UsedCameraModule]].Add([[UsedCameraModuleToWarehouse]][i]);
     }
}

foreach(EntityData entity in [[UsedCameraModule]])
{
   [[UsedCameraModuleToWarehouse]].Remove(entity);
}

[[CostForWarehouseForReusedCameraModule]] += [[ChargesForCustody]] * [[UsedCameraModuleToWarehouse]].Count;
[[ReusedCameraModuleAmount]] = [[UsedCameraModule]].Count;
[[TotalReusedCameraModuleAmount]] += [[UsedCameraModule]].Count;
```

図 A - 34：ReusedCameraModuleShipping プロセスの Procedure

- CalculationOfMaximumReusedAmount

表 A - 61：CalculationOfMaximumReusedAmount プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	ReusedCenterPanelAmount	double	monthly
	ReusedCameraAmount	double	monthly
Output parameter	reusedNumberForSalesCalculation	double	monthly

```
if([[ReusedCameraAmount]] < [[ReusedCenterPanelAmount]])
{
   [[reusedNumberForSalesCalculation]] = [[ReusedCenterPanelAmount]];
}
else
{
   [[reusedNumberForSalesCalculation]] = [[ReusedCameraAmount]];
}
```

図 A - 35：CalculationOfMaximumReusedAmount プロセスの Procedure

- **Shredding**

表 A - 62：Shredding プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	BatteryCover	List<EntityData>	monthly
	RearPanel	List<EntityData>	monthly
	Antenna	List<EntityData>	monthly
Output parameter	PlasticFragment	List<EntityData>	monthly

```
foreach(EntityData entity in [[BatteryCover]])
{
   [[PlasticFragment]].Add(LCSFunction.Transform(entity, “PlasticFragment"));
}

foreach(EntityData entity in [[RearPanel]])
{
   [[PlasticFragment]].Add(LCSFunction.Transform(entity, “PlasticFragment"));
}

foreach(EntityData entity in [[Antenna]])
{
   [[PlasticFragment]].Add(LCSFunction.Transform(entity, “PlasticFragment"));
}
```

図 A - 36：Shredding プロセスの Procedure

- **EntireLifeCycle**

表 A - 63：EntireLifeCycle プロセスの Input parameter と Output parameter

	Parameter	Type	Behavior
Input parameter	FrontGlassManufTotalCost	double	continuous
	OLEDManufTotalCost	double	continuous
	BottomBoardManufTotalCost	double	continuous
	AntennaManufTotalCost	double	continuous
	BatteryManufTotalCost	double	continuous
	BatteryCoverManufTotalCost	double	continuous
	CenterPanelManufTotalCost	double	continuous
	RearPanelManufTotalCost	double	continuous
	MainBoardManufTotalCost	double	continuous
	LensManufTotalCost	double	continuous
	ImageProcessorManufTotalCost	double	continuous

	SpeakerManufTotalCost	double	continuous
	AssemblyCostForFirst	double	continuous
	AssemblyCostForSecond	double	continuous
	TransportationCost	double	continuous
	TotalSales	double	continuous
	CollectionCost	double	continuous
	DisassemblyCost	double	continuous
	InspectionCostForCenterPanel	double	continuous
	InspectionCostForCameraModule	double	continuous
	CostForWarehouseForReusedCenterPanel	double	continuous
	CostForWarehouseForReusedCameraModule	double	continuous
	FrontGlassManufCO2	double	continuous
	OLEDManufCO2	double	continuous
	BottomBoardManufCO2	double	continuous
	AntennaManufCO2	double	continuous
	BatteryManufCO2	double	continuous
	BatteryCoverManufCO2	double	continuous
	CenterPanelManufCO2	double	continuous
	RearPanelManufCO2	double	continuous
	MainBoardManfCO2	double	continuous
	LensManufCO2	double	continuous
	ImageProcessorManufCO2	double	continuous
	SpeakerManufCO2	double	continuous
	TransportationCO2	double	continuous
	UseCO2	double	continuous
	ManufacturedCenterPanelAmount	double	continuous
	ReusedCenterPanelAmount	double	continuous
	ManufacturedCameraAmount	double	continuous
	ReusedCameraAmount	double	continuous
Output parameter	reuseRateOfCenterPanel	double	continuous
	reuseRateOfCameraModule	double	continuous
	BenefitForManufacturers	double	continuous
	LCCO2	double	continuous

```
[[reuseRateOfCenterPanel]] = [[ReusedCenterPanelAmount]] / [[ManufacturedCenterPanelAmount]];
[[reuseRateOfCameraModule]] = [[ReusedCameraAmount]] / [[ManufacturedCameraAmount]];

[[BenefitForManufacturers]]
= [[TotalSales]] - ([[FrontGlassManufTotalCost]] + [[OLEDManufTotalCost]] + [[BottomBoardManufTotalCost]]
+ [[AntennaManufTotalCost]] + [[BatteryManufTotalCost]] + [[BatteryCoverManufTotalCost]] + [[CenterPanelManufTotalCost]]
+ [[RearPanelManufTotalCost]] + [[MainBoardManufTotalCost]] + [[LensManufTotalCost]] + [[ImageProcessorManufTotalCost]]
+ [[SpeakerManufTotalCost]] + [[AssemblyCostForFirst]] + [[AssemblyCostForSecond]] + [[TransportationCost]]
+ [[CollectionCost]] + [[DisassemblyCost]] + [[InspectionCostForCenterPanel]] + [[InspectionCostForCameraModule]]
+ [[CostForWarehouseForReusedCenterPanel]] + [[CostForWarehouseForReusedCameraModule]]);

[[LCCO2]]
= [[FrontGlassManufCO2]] + [[OLEDManufCO2]] +[[BottomBoardManufCO2]] + [[AntennaManufCO2]] + [[BatteryManufCO2]]
+ [[BatteryCoverManufCO2]] + [[CenterPanelManufCO2]] + [[RearPanelManufCO2]] + [[MainBoardManfCO2]]
+ [[LensManufCO2]] + [[ImageProcessorManufCO2]] + [[SpeakerManufCO2]] + [[TransportationCO2]] + [[UseCO2]];
```

図 A - 37：EntireLifeCycle プロセスの Procedure