

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-92855

(P2005-92855A)

(43) 公開日 平成17年4月7日(2005.4.7)

(51) Int.Cl.<sup>7</sup>

G06F 9/44

F I

G06F 9/06 620A

テーマコード (参考)

5B076

審査請求 未請求 請求項の数 18 O L (全 35 頁)

(21) 出願番号 特願2004-141535 (P2004-141535)  
 (22) 出願日 平成16年5月11日 (2004.5.11)  
 (31) 優先権主張番号 60/502, 229  
 (32) 優先日 平成15年9月12日 (2003.9.12)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 593165487  
 学校法人金沢工業大学  
 石川県石川郡野々市町扇が丘7番1号  
 (74) 代理人 100105924  
 弁理士 森下 賢樹  
 (72) 発明者 國井 利▲泰▼  
 東京都文京区本郷1丁目25番21号  
 Fターム(参考) 5B076 DD01

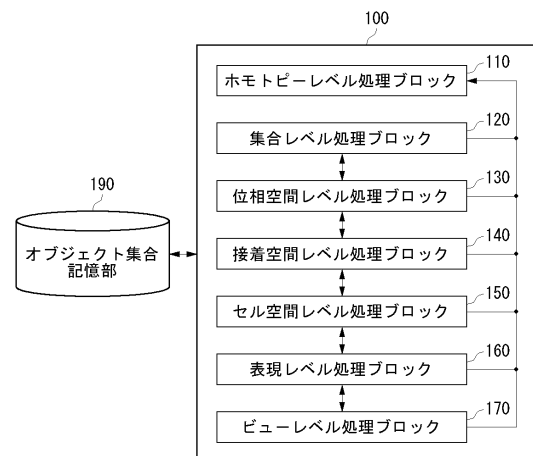
(54) 【発明の名称】 ソフトウェア開発支援装置およびソフトウェア開発支援方法

## (57) 【要約】

【課題】 要求仕様を満たすソフトウェアの開発には膨大な工数がかかる。

【解決手段】 集合レベル処理ブロック120は、オブジェクト集合記憶部190から複数のプログラムを抽出し、位相空間レベル処理ブロック130は、プログラムを位相空間として設計する。接着空間レベル処理ブロック140は、複数のプログラムの上に同値関係を規定し、同値関係で対応づけられたプログラムモジュールを接着した接着空間を設計する。セル空間レベル処理ブロック150は、モジュールの属性をセルで表現し、セルの次元を設計し、表現レベル処理ブロック160は、セルの表現を設計する。ビューレベル処理ブロック170は、ユーザに対するビューを設計する。ホモトピーレベル処理ブロック110は、各処理ブロックによる操作によって生じるプログラムの変化をホモトピーとして設計する。

【選択図】 図4



## 【特許請求の範囲】

## 【請求項 1】

各階層においてソフトウェア部品集合に対する操作が可能に構成されたモジュラ化された抽象階層ブロックを有し、前記抽象階層ブロック間で前記ソフトウェア部品集合の同値関係を不変量として継承しながら、前記ソフトウェア部品集合が満たすべき仕様を前記抽象階層ブロック毎にインクリメンタルに詳細化することにより、与えられた要求仕様に合致するソフトウェアを前記ソフトウェア部品集合から自動生成することを特徴とするソフトウェア開発支援装置。

## 【請求項 2】

前記モジュラ化された抽象階層ブロックは、

10

前記ソフトウェア部品集合に集合演算を施して、所望の複数のソフトウェア部品集合を設計する集合設計部と、

前記ソフトウェア部品集合の部分集合を要素とする集合に位相を規定することにより、前記ソフトウェア部品集合を位相空間として設計する位相空間設計部と、

それぞれが位相空間として設計された前記複数のソフトウェア部品集合の上に同値関係を規定し、前記同値関係で対応づけられた部分空間を接着した接着空間を設計する接着空間設計部と、

前記接着空間において前記複数のソフトウェア部品集合に属するソフトウェア部品の仕様に係る属性をセルで表現し、前記セルの次元を設計するセル空間設計部と、

前記セルの表現形式を設計する表現設計部とを含むことを特徴とする請求項 1 に記載のソフトウェア開発支援装置。

20

## 【請求項 3】

前記モジュラ化された抽象階層ブロックは、前記複数のソフトウェア部品集合に関して、ソフトウェア開発者に対するビューを設計し、ソフトウェア開発者毎に前記複数のソフトウェア部品集合の操作可能範囲を異ならせるビュー設計部をさらに含むことを特徴とする請求項 2 に記載のソフトウェア開発支援装置。

## 【請求項 4】

前記モジュラ化された抽象階層ブロックは、各設計部による前記ソフトウェア部品集合に対する操作によって生じる前記ソフトウェア部品集合の変化がホモトピー同値になるように設計し、前記ソフトウェア部品集合を変化前の状態に戻せるように、その操作手順を記録するホモトピー設計部をさらに含むことを特徴とする請求項 2 または 3 に記載のソフトウェア開発支援装置。

30

## 【請求項 5】

前記接着空間設計部は、

互いに素である第 1 の位相空間  $X$  と第 2 の位相空間  $Y$  に対して、前記第 2 の位相空間の  $Y$  の部分空間  $Y_0$  から前記第 1 の位相空間  $X$  への連続写像を、前記同値関係を表す接着写像  $f$  として取得する接着写像取得部と、

前記第 2 の位相空間  $Y$  の前記部分空間  $Y_0$  における各点  $y$  を、前記接着写像  $f$  による前記第 1 の位相空間  $X$  における像  $f(y)$  と同一視することによって、前記第 2 の位相空間  $Y$  を前記第 1 の位相空間に接着し、前記接着空間を同値類の排他的論理和として取得する等化写像取得部とを含むことを特徴とする請求項 2 に記載のソフトウェア開発支援装置。

40

## 【請求項 6】

前記セル空間設計部は、

前記接着写像  $f$  によって同一視された前記第 1 の位相空間  $X$  のソフトウェア部品と前記第 2 の位相空間  $Y$  のソフトウェア部品をセル空間において対応づけて接合するセル接合部と、

前記セル空間において対応づけられた前記ソフトウェア部品の仕様に係る属性の次元に関する不整合を調整するセル次元調整部を含むことを特徴とする請求項 5 に記載のソフトウェア開発支援装置。

## 【請求項 7】

50

前記ソフトウェア部品は、データ構造体であり、

前記セル次元調整部は、前記セル空間において対応づけられた前記データ構造体の仕様に係る属性の名前または個数に不整合があるとき、その不整合を調整することを特徴とする請求項 6 に記載のソフトウェア開発支援装置。

【請求項 8】

前記ソフトウェア部品は、プログラムモジュールであり、

前記セル次元調整部は、前記セル空間において対応づけられた前記プログラムモジュールの仕様に係る属性の次元として、前記プログラムモジュールの入出力インタフェースの名前または個数に不整合があるとき、その不整合を調整することを特徴とする請求項 6 に記載のソフトウェア開発支援装置。

10

【請求項 9】

前記ソフトウェア部品は、プログラムモジュールであり、

前記セル次元調整部は、前記セル空間において対応づけられた前記プログラムモジュールの仕様に係る属性の次元として、前記プログラムモジュールで参照される変数の名前または個数に不整合があるとき、その不整合を調整することを特徴とする請求項 6 に記載のソフトウェア開発支援装置。

【請求項 10】

前記ソフトウェア部品は、プログラムモジュールであり、

前記セル次元調整部は、前記セル空間において対応づけられた前記プログラムモジュールの仕様に係る属性の次元として、前記プログラムモジュールで利用される関数の名前または個数に不整合があるとき、その不整合を調整することを特徴とする請求項 6 に記載のソフトウェア開発支援装置。

20

【請求項 11】

前記表現設計部は、前記セル空間において対応づけられた前記ソフトウェア部品の仕様に係る属性の表現形式に関する不整合を受容表現にしたがって調整する表現形式調整部を含むことを特徴とする請求項 6 に記載のソフトウェア開発支援装置。

【請求項 12】

前記ソフトウェア部品は、データ構造体であり、

前記表現形式調整部は、前記セル空間において対応づけられた前記データ構造体の仕様に係る属性のデータ型または桁数に不整合があるとき、その不整合を調整することを特徴とする請求項 11 に記載のソフトウェア開発支援装置。

30

【請求項 13】

前記ソフトウェア部品は、プログラムモジュールであり、

前記表現形式調整部は、前記セル空間において対応づけられた前記プログラムモジュールの仕様に係る属性の表現形式として、前記プログラムモジュールの入出力インタフェースのデータ型または桁数に不整合があるとき、その不整合を調整することを特徴とする請求項 11 に記載のソフトウェア開発支援装置。

【請求項 14】

前記ソフトウェア部品は、プログラムモジュールであり、

前記表現形式調整部は、前記セル空間において対応づけられた前記プログラムモジュールの仕様に係る属性の表現形式として、前記プログラムモジュールで参照される変数のデータ型または桁数に不整合があるとき、その不整合を調整することを特徴とする請求項 11 に記載のソフトウェア開発支援装置。

40

【請求項 15】

各階層においてソフトウェア部品集合に対する操作が可能に構成されたモジュラ化された抽象階層ステップを有し、前記抽象階層ステップ間で前記ソフトウェア部品集合の同値関係を不変量として継承しながら、前記ソフトウェア部品集合が満たすべき仕様を前記抽象階層ブロック毎にインクリメンタルに詳細化することにより、与えられた要求仕様に合致するソフトウェアを前記ソフトウェア部品集合から自動生成することを特徴とするソフトウェア開発支援方法。

50

## 【請求項 16】

ソフトウェア部品集合に集合演算を施して、所望の複数のソフトウェア部品集合を設計する集合設計ステップと、

前記ソフトウェア部品集合の部分集合を要素とする集合に位相を規定することにより、前記ソフトウェア部品集合を位相空間として設計する位相空間設計ステップと、

それぞれが位相空間として設計された前記複数のソフトウェア部品集合の上に同値関係を規定し、前記同値関係で対応づけられた部分空間を接着した接着空間を設計する接着空間設計ステップと、

前記接着空間において前記複数のソフトウェア部品集合に属するソフトウェア部品の仕様に係る属性をセルで表現し、前記セルの次元を設計するセル空間設計ステップと、

前記セルの表現形式を設計する表現設計ステップとをコンピュータに実行させることを特徴とするプログラム。

10

## 【請求項 17】

前記複数のソフトウェア部品集合に関して、ソフトウェア開発者に対するビューを設計し、ソフトウェア開発者毎に前記複数のソフトウェア部品集合の操作可能範囲を異ならせるビュー設計ステップをさらに含むことを特徴とする請求項 16 に記載のプログラム。

## 【請求項 18】

各設計ステップによる前記ソフトウェア部品集合に対する操作によって生じる前記ソフトウェア部品集合の変化がホモトピー同値になるように設計し、前記ソフトウェア部品集合を変化前の状態に戻せるように、その操作手順を記録するホモトピー設計ステップをさらに含むことを特徴とする請求項 16 または 17 に記載のプログラム。

20

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

この発明はソフトウェア開発支援技術、とくに、複数の異なるオブジェクト集合間でデータの一貫性維持、インタフェイスの確定その他の処理をなすソフトウェア開発支援装置およびソフトウェア開発支援方法に関する。

## 【背景技術】

## 【0002】

「ソフトウェア工学」(Software Engineering)という用語は、1968年のNATO会議で最初に使われたといわれる。この会議において討議されたソフトウェア設計法、プロジェクト管理、技術者教育、ソフトウェアの価格および流通の4つの項目は、今なお、ソフトウェア工学の中心課題である(非特許文献1参照)。

30

## 【0003】

ソフトウェアの開発プロセスは、要求分析、システム設計、実装、テスト、運用、および保守の各段階からなり、これをソフトウェアのライフサイクルという。最初の要求分析が甘く、顧客の要求を取りこぼしたり、取り違えたりすると、それ以降のシステム設計、実装に顧客の要求が正しく反映されない。そのため、テストや運用段階で発見された不具合を修正するために、プログラムを改良したり、大規模な変更の場合はシステム設計をやり直すなど、ソフトウェアの保守に大変な時間と労力がかかる。最悪の場合、保守に耐えられないソフトウェアは廃棄されることもあり、最初から開発をやり直すことになる。このように、ソフトウェアの開発プロセスの初期段階において、要求分析に時間と労力を惜しむと、保守段階においてコストがかかり、開発プロセスをトータルでみた場合に、全体のコストが肥大する。

40

## 【0004】

ソフトウェア開発の一例として、データベースシステムの開発を例にあげる。データベースシステム開発におけるデータ設計とは、顧客の業務内容をヒアリングして、管理すべきデータ項目を洗い出し、それらの項目を関連付け、利用者が検索しやすいようなビューを定義して、顧客ニーズに合ったデータベースシステムを構築することである。業務内容を分析し、どのようなデータを管理すべきかを決める概念設計から始め、データ構造とそ

50

のアクセス方法、および処理要件を決め、データベースの表現形式で定義する論理設計を行う。そして、最後に、メモリやディスクのサイズを見積もり、ハードウェア性能要件を考慮してデータベースの物理構造を決める物理設計を行い、データ設計が完了する。

【 0 0 0 5 】

優れたデータベースシステムを構築するには、概念設計の段階からデータ構造を綿密に解析しておく必要がある。データとして管理すべき実体（エンティティ）や属性、実体間の関連（リレーション）を洗い出し、正確に分析することが重要になる。このような分析手法の一例として、E R（Entity Relationship）モデルがある。E Rモデルでは、実体間の関係をグラフで定義するが、関係をどのように定義するかは設計者の自由であるため、実装毎に異なるシステムが生成されることになる。したがって、E Rモデルにもとづいて実装された複数のシステムを統合するには、統合すべき構成要素の組み合わせ数に応じた作業工程が必要になり、システム統合に相当な時間と労力を費やすこととなる。

10

【 0 0 0 6 】

情報システム構築は、たとえば大手銀行の統合に伴う統合システムに典型的にみられるように、単一大規模サイト向けから、複数分散大規模サイト向けに加速度的に移行しつつある。ところが、その基盤技術となっている現在のソフトウェア工学、リエンジニアリング技術において、これに対応するための基本的枠組みとしてのエンジニアリングモデルが欠落している。その理由は、統合方法がモデルレベルで欠如しているために、各サイト構築担当のエンジニアが個々にモジュール群を工夫して設計・構築している点にある。このために、モジュール群仕様がサイト毎に自ずと異なる。モジュール群の仕様が相互に開示されれば、再構築も可能であるが、モジュールの数の組み合わせにより指数関数的に再構築システム数が増え、いわゆる「開発工数爆発」が生じる。しかも、実際には、企業機密保持上モジュール群の仕様が開示されない場合のほうが多い。この場合、モジュール間のインタフェイス作成情報が開示されるものの、インタフェイス数も組み合わせにより指数関数的に増加するので、一層「開発工数爆発」の様相が顕著となる。

20

【非特許文献1】シャリ・ローレンス・ブリーガー著，堀内泰輔訳，「ソフトウェア工学理論と実践」，初版，株式会社ピアソン・エデュケーション，2001年11月

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 7 】

ソフトウェアの効率的な開発を支援するための理論的基礎となるエンジニアリングモデルが欠如しているために、要求分析からシステム設計、実装に至るまでの開発のスムーズな流れが滞り、ソフトウェア開発を遅延させる原因となっている。また、運用段階で不測の事態が発生しやすく、ソフトウェア開発のリスク管理も難しくなっている。

30

【 0 0 0 8 】

国内外には、リレーショナル・データベース（RDB）モデル、E Rモデル、タグによるXML、グラフによるUMLなどが研究され応用されているが、大規模分散情報システムを構築する理論的基礎としてはモデル的に完全とはいえず、例えばモジュール間のインタフェイス仕様が、異なるサイト間で統一的に取り扱える理論体系になっていない。したがって、前述の「開発工数爆発」は解決できない。

40

【 0 0 0 9 】

また、各国において昨今の厳しい経済状況もあり、会社統合が増加の一途をたどっているが、異なる情報システム間で整合性をもって情報処理を自在に行えること、すなわち、情報システムのインターオペラビリティが確保されないため、情報システムの運用と保守の費用が増大している。

【 0 0 1 0 】

本発明はこうした背景からなされたものであり、その目的は、ソフトウェア部品のインターオペラビリティを線形モデルとして保証して、ソフトウェア開発の自動化を支援することのできるソフトウェア開発支援技術を提供することにある。

【課題を解決するための手段】

50

## 【 0 0 1 1 】

本発明のある態様はソフトウェア開発支援装置に関する。この装置は、各階層においてソフトウェア部品集合に対する操作が可能に構成されたモジュラ化された抽象階層ブロックを有し、前記抽象階層ブロック間で前記ソフトウェア部品集合の同値関係を不変量として継承しながら、前記ソフトウェア部品集合が満たすべき仕様を前記抽象階層ブロック毎にインクリメンタルに詳細化することにより、与えられた要求仕様に合致するソフトウェアを前記ソフトウェア部品集合から自動生成する。

## 【 0 0 1 2 】

「ソフトウェア部品」は、本装置により自動生成されるソフトウェアを構成する部品であり、ソフトウェアで利用されるデータ、ソフトウェアを構成するプログラムなどを含む。

## 【 0 0 1 3 】

前記モジュラ化された抽象階層ブロックは、前記ソフトウェア部品集合に集合演算を施して、所望の複数のソフトウェア部品集合を設計する集合設計部と、前記ソフトウェア部品集合の部分集合を要素とする集合に位相を規定することにより、前記ソフトウェア部品集合を位相空間として設計する位相空間設計部と、それぞれが位相空間として設計された前記複数のソフトウェア部品集合の上に同値関係を規定し、前記同値関係で対応づけられた部分空間を接着した接着空間を設計する接着空間設計部と、前記接着空間において前記複数のソフトウェア部品集合に属するソフトウェア部品の仕様に係る属性をセルで表現し、前記セルの次元を設計するセル空間設計部と、前記セルの表現形式を設計する表現設計部とを含んでもよい。前記モジュラ化された抽象階層ブロックは、前記複数のソフトウェア部品集合に関して、ソフトウェア開発者に対するビューを設計し、ソフトウェア開発者毎に前記複数のソフトウェア部品集合の操作可能範囲を異ならせるビュー設計部をさらに含んでもよい。

## 【 0 0 1 4 】

前記モジュラ化された抽象階層ブロックは、各設計部による前記ソフトウェア部品集合に対する操作によって生じる前記ソフトウェア部品集合の変化がホモトピー同値になるように設計し、前記ソフトウェア部品集合を変化前の状態に戻せるように、その操作手順を記録するホモトピー設計部をさらに含んでもよい。

## 【 0 0 1 5 】

この装置によれば、複数のソフトウェア部品集合の間で同値関係を規定し、その同値関係を不変量として抽象階層間で継承しながら、各抽象階層において段階的にソフトウェア部品集合の満たすべき仕様を設計することができる。集合レベル、位相空間レベル、接着空間レベル、セル空間レベル、表現レベル、およびビューレベルを通して、同値関係が保証され、最終的にソフトウェアが数学的に整合性のある形で生成される。また、各抽象階層はモジュール化されており、ソフトウェア部品の追加、削除、および修正などを行っても、抽象階層を上下に行き来しながら、インクリメンタルに設計変更することができる。さらに、ソフトウェア部品集合に対してなされる操作をホモトピーとして扱うことにより、ソフトウェア部品集合を変化前の状態と変化後の状態の間で相互に移行させることができ、数学的な整合性を維持しながら、ソフトウェアの部品化、再利用、デバッグなどの開発工程を支援することができる。

## 【 0 0 1 6 】

本発明の別の態様はソフトウェア開発支援方法に関する。この方法は、各階層においてソフトウェア部品集合に対する操作が可能に構成されたモジュラ化された抽象階層ステップを有し、前記抽象階層ステップ間で前記ソフトウェア部品集合の同値関係を不変量として継承しながら、前記ソフトウェア部品集合が満たすべき仕様を前記抽象階層ブロック毎にインクリメンタルに詳細化することにより、与えられた要求仕様に合致するソフトウェアを前記ソフトウェア部品集合から自動生成する。

## 【 0 0 1 7 】

本発明のさらに別の態様はプログラムに関する。このプログラムは、ソフトウェア部品

10

20

30

40

50

集合に集合演算を施して、所望の複数のソフトウェア部品集合を設計する集合設計ステップと、前記ソフトウェア部品集合の部分集合を要素とする集合に位相を規定することにより、前記ソフトウェア部品集合を位相空間として設計する位相空間設計ステップと、それぞれが位相空間として設計された前記複数のソフトウェア部品集合の上に同値関係を規定し、前記同値関係で対応づけられた部分空間を接着した接着空間を設計する接着空間設計ステップと、前記接着空間において前記複数のソフトウェア部品集合に属するソフトウェア部品の仕様に係る属性をセルで表現し、前記セルの次元を設計するセル空間設計ステップと、前記セルの表現形式を設計する表現設計ステップとをコンピュータに実行させる。

【0018】

なお、以上の構成要素の任意の組み合わせ、本発明の表現を方法、装置、システム、記録媒体、コンピュータプログラム、データ構造などの間で変換したものもまた、本発明の態様として有効である。

【発明の効果】

【0019】

本発明によれば、情報システムのインターオペラビリティを保証し、ソフトウェア開発の効率化を図ることができる。

【発明を実施するための最良の形態】

【0020】

以下本発明を好適な実施の形態をもとに説明する。まず、本発明者が提唱するインクリメンタリモジュラな抽象階層 (incrementally modular abstraction hierarchy) を説明し、その後、インクリメンタリモジュラな抽象階層を実装した情報処理装置を説明する。

【0021】

[1] 集合論的設計

まず、サイバースペースに構築すべきオブジェクトの集まりを定義することからサイバースペースの設計を始める。そのような集まりに対してインテリジェントマシンとしてコンピュータを使った自動操作の実行が可能であるためには、各集まりは「集合 (set)」でなければならない。なぜなら、コンピュータは集合論的機械として構築されているからである。直観的に、集合  $X$  は、同一のプロパティ  $P(x)$  をもったすべてのオブジェクト  $x$  の集まりである。これを記号で表すなら、 $X = \{x \mid P(x)\}$  である。集合に含まれる任意のオブジェクトを元または要素 (element) という。要素をもたない集合を空集合という。すべての要素が内部 (interior) であるならば、その集合は開集合であるという。集合  $X$  と  $Y$  が与えられたとき、コンピュータは、和 (union)  $X \cup Y$ 、積または共通部分 (intersection)  $X \cap Y$ 、差 (difference)  $X - Y$  ( $x \setminus y$  とも書く)、否定 (negation)  $\neg X$  のような集合論的演算を施す。

【0022】

最初のサイバースペースとして集合  $X$  からサイバースペースアーキテクチャの設計を始めるとしよう。未知のサイバースペース  $U$  のすべての要素  $u$  が与えられたとき、もしそれらの要素がすべてサイバースペース  $X$  の要素であることが確かめられたならば、その未知のサイバースペースを  $X$  の部分集合、すなわち  $X$  の部分サイバースペースと呼び、 $U \subseteq X$  と表記する。部分集合であるかどうかは、 $(u \in U) \wedge (u \in X)$  を処理することで自動的に調べることができる。  $U$  の閉包 (closure)

【数1】

$\bar{U}$

は、 $U$  を含む  $X$  のすべての閉部分集合の積である。言い換えれば、 $U$  の閉包は、 $U$  の外部 (exterior) 要素でない  $X$  の要素である。  $X$  の部分集合  $\{U \mid U \subseteq X\}$  のすべてからなる集合を  $X$  の巾 (べき) 集合といい、 $2^X$  と表記する。これはまた  $X$  の離散位相 (discrete topology) とも呼ばれる。離散位相はサイバースペースを部分サイバースペースからなるものとして設計する上でたいへん有用である。なお、集合を要素とする集合 (いわゆる「集合の集合」) を「集合族」 (family of sets) ということがある。  $X$  の巾集合は、集

合  $X$  のすべての部分集合を要素とする集合族である。

【0023】

[2] 位相幾何学的設計

これから、サイバースペースを  $X$  の (複数の) 部分サイバースペースとその重複 (overlap) の和として設計する作業に入る。このように設計されるサイバースペースは一般に位相 (トポロジー) 空間  $(X, T)$  と呼ばれる。ここで  $T = 2^X$  である。位相空間の設計は次の仕様により自動化される。

- 1)  $X = T$  かつ  $T = \{ \emptyset, X \}$  ;
- 2) 任意の添え字の集合  $J$  に対して、  
 $\bigcup_{j \in J} (U_j \cap T) = \bigcup_{j \in J} U_j \cap T$  ;
- 3)  $U, V \in T \Rightarrow U \cup V \in T$  .

10

【0024】

上記の設計仕様の 1) は、空集合と全体集合が  $T$  の要素であること、2) は、 $T$  の要素をいくつもってきて和をとっても  $T$  の要素であること、3) は、 $T$  の 2 つの要素の積 (共通部分) も  $T$  の要素であることを示している。コンピュータは、 $X$  の巾集合  $2^X$  からこの設計仕様を満たすように  $T$  を設計する。

【0025】

$T$  を位相空間  $(X, T)$  の位相 (トポロジー) という。  $T_1 = T_2$  である 2 つの位相  $T_1$  と  $T_2$  が与えられたとき、  $T_1$  は  $T_2$  よりも弱いまたは小さいという。逆に、  $T_2$  は  $T_1$  よりも強いまたは大きいという。また  $T_2$  は  $T_1$  よりも細かい、あるいは  $T_1$  は  $T_2$  よりも粗いということもある。明らかなように最も強い位相は離散位相すなわち巾集合であり、最も弱い位相は空集合である。簡単のため、あいまいさが生じないならば、位相空間を表すために  $(X, T)$  の代わりに  $X$  をしばしば用いる。

20

【0026】

2 つの位相空間  $(X, T)$  と  $(Y, T')$  が与えられたとき、いかにして  $(X, T)$  と  $(Y, T')$  が同値であるということが出来るだろうか。コンピュータを使って、それらの空間が位相幾何学的に同値であることを自動的に検証 (validate) するための基準をここに与える。2 つの位相空間  $(X, T)$  と  $(Y, T')$  は、連続な関数  $f : (X, T) \rightarrow (Y, T')$  が存在し、かつその逆関数も存在して連続であるならば、トポロジー同値 (topologically equivalent)、すなわち同相、同位相または位相同形 (homeomorphic) であるという。  $(X, T)$  が  $(Y, T')$  と同相であることを

30

【数 2】

$$(X, T) \simeq (Y, T')$$

と書く。

【0027】

それでは、関数  $f$  の連続性はどのように立証することができるか。最初に、  $B \in T'$  ,  $f^{-1}(B) \in T$  であることを調べる。ここで  $f^{-1}(B)$  は  $f$  による  $B$  の逆像を意味する。次に、以下が成り立つことを調べる。

40

$B$  が開である  $f^{-1}(B)$  もまた  $X$  において開である。

【0028】

[3] 関数

関数  $f : X \rightarrow Y$  が与えられたとき、全域関数 (total function) と部分関数 (partial function) が存在する。関数  $f : X \rightarrow Y$  に対して、  $x \in X$  、  $f(x)$  であるとき、そしてそのときに限り、関数  $f$  は全域関数であるという。関数  $f : X' \rightarrow Y \mid X' \subseteq X$  を部分関数といい、このとき  $f(x)$  は必ずしもあらゆる  $x \in X$  に対して存在しているとは限らない。

【0029】

全域関数に対して、3 つの基本的なタイプの関係あるいは写像が存在する。

50



## 1. 単射 (injective または into)

$$x, y \in X, x \neq y \Rightarrow f(x) \neq f(y)$$

言い換えれば、

$$x, y \in X, f(x) = f(y) \Rightarrow x = y$$

## 2. 全射 (surjective または onto)

$$(\forall y \in Y) (\exists x \in X) [f(x) = y]$$

## 3. 全単射 (bijective)

単射かつ全射であること

【0030】

## [4] 同値関係

集合  $X$  上の任意の 2 項関係  $R \subset X \times X$  に対して、 $R$  は、

1)  $(\forall x \in X) [x R x]$  ならば、反射性があり、

2)  $(\forall x, y \in X) [x R y \Rightarrow y R x]$  ならば、対称性があり、

3)  $(\forall x, y, z \in X) [(x R y \wedge y R z) \Rightarrow x R z]$  ならば、推移性があるという。  
。1) ~ 3) の条件は、それぞれ反射律、対称律、推移律と呼ばれる。

【0031】

$R$  が反射性、対称性、推移性をもつとき、 $R$  を同値関係 (equivalence relation) と呼び、 $\sim$  で表記する。

【0032】

$x \in X$  が与えられたとき、 $x / \sim = \{ y \in X : x \sim y \}$  で定義される  $X$  の部分集合を  $x$  の同値類 (equivalent class) と呼ぶ。ここで類 (class) は実際には集合を意味するが、伝統的に類 (クラス) と呼ばれており、今となっては変更するのは難しい。すべての同値類の集合  $X / \sim$  を  $X$  の商空間 (quotient space) もしくは等化空間 (identification space) という。

$$X / \sim = \{ x / \sim \mid x \in X \}$$

推移律より、各  $x \in X$ 、 $x / \sim$  について、以下が成り立つ。

【数 3】

$$x \sim y \Leftrightarrow x / \sim = y / \sim, \text{ and}$$

$$x \not\sim y \Leftrightarrow x / \sim \cap y / \sim = \emptyset$$

これは集合  $X$  が空でない互いに素の (すなわち共通要素をもたない) 同値類に分割される (分解されるともいう) ことを意味する。

【0033】

一般に、集合  $X$  の上に同値関係  $\sim$  が定義されたとき、ある  $X$  の要素  $a$  に対して  $a$  に同値である要素をすべて集めた集合を考えることができる。この  $X$  の部分集合は、 $a$  を代表要素とする同値類であり、 $[a]$  と表記される。ここで、同値類に含まれる要素のうちどれを取っても、それを代表要素とする同値類は同じ集合になる。すなわち、同値類は代表要素の取り方によらない。同値類が反射律、対称律、および推移律を満たすことより、各同値類は空ではなく、 $a$  の同値類には  $a$  が属し、相異なる同値類には共通の要素がない。したがって、同値類をすべて集めると、それらは互いに交わず、また全体の和集合は  $X$  に等しくなる。つまり、集合  $X$  は同値類の直和 (disjoint union) に分割される。この直和分割を集合  $X$  の同値関係  $R$  に関する類別 (classification) といい、同値類全体の集合を集合  $X$  の同値関係  $\sim$  による商集合と呼ぶ。商集合に位相が定義されたものを上記のように商空間という。

【0034】

簡単な例で説明する。 $X$  を有理整数の集合とし、 $X$  の要素  $x, y$  の関係  $R$  として、「 $x - y$  が偶数である」を考えると、これは同値関係である。集合  $X$  をこの同値関係  $R$  で類別すれば、集合  $X$  は、奇数の同値類と偶数の同値類の直和に分割される。

【0035】

10

20

30

40

50

ユークリッド幾何学において、図形の集合が与えられたとき、合同関係は同値関係であり、図形の集合全体を、商空間として、合同な図形からなる部分集合の直和に分割する。また、相似関係も同値関係であり、図形の集合全体を、商空間として、相似な図形からなる部分集合の直和に分割する。合同関係と相似関係はアフィン変換の例である。群論における対称関係は、図形の集合全体を、商空間として、対称な図形からなる部分集合の直和に分割する。

#### 【0036】

電子商取引では、「電子商品であること」は同値関係であるが、「電子取引」は半順序 (partially ordered) 関係であることに留意する。集合  $X$  上の 2 項関係  $S$  は、次の 3 つの条件を満たすとき、半順序関係である。

10

1)  $(x, x) \in S$  (反射律)

2)  $(x, y) \in S, (y, x) \in S \Rightarrow x = y$  (反対称律)

3)  $(x, y) \in S, (y, z) \in S \Rightarrow (x, z) \in S$  (推移律)

#### 【0037】

半順序関係をもつ集合を半順序集合 (partially ordered set) と呼び、頭文字を取って、しばしばポセット (poset) ともいう。集合の 2 つの要素  $x, y$  が  $x \leq y$  か、 $y \leq x$  のいずれかであるとき、比較可能であるという。どの 2 つの要素も比較可能な半順序集合を全順序集合 (totally ordered set) という。

#### 【0038】

電子取引の関係は、売り手と買い手の関係が非対称であり、対称律を満たさず、反対称律を満たすため、半順序関係となる。一方、電子商品であるという関係は、売り手にとっての電子商品は買い手にとっての電子商品でもあるから、対称であり、同値関係となる。

20

#### 【0039】

同値関係について強弱を定義することができる。1 つの集合  $X$  における 2 つの同値関係  $R, S$  について、 $x R y \Rightarrow x S y$  であるとき、 $R$  は  $S$  より強い、 $S$  は  $R$  より弱いという。また、 $R$  による類別は  $S$  による類別より細かい、 $S$  による類別は  $R$  による類別より粗いという。最も強い同値関係は、「同じものである」という同値関係であり、最も弱い同値関係は、 $X$  の任意の 2 つの要素が同値であるとする同値関係である。

#### 【0040】

[5] 商空間 (等化空間)

30

$X$  を位相空間とする。 $f$  を全射かつ連続な写像であって、各点  $x \in X$  を、 $x$  を含む部分集合である同値類  $x / \sim \in X / \sim$  に写像する商写像 (quotient map) (しばしば等化写像 (identification map) と呼ばれる) であるとする。

$f : X \rightarrow X / \sim$

ここで、既に説明したように、「写像  $f : X \rightarrow Y$  が全射である」とは、

$(y \in Y) \Rightarrow (x \in X) [f(x) = y]$

を意味する。

#### 【0041】

$X$  の部分集合  $X^0 \subset X$  に対して、

$X^0$  が開である  $f^{-1}(X^0) \cap Y = \emptyset$  が  $X$  において開である (これは  $f$  が連続であることを意味する)

40

が成り立つような全射写像  $f$  を考えた場合、 $X / \sim$  を商写像 (または等化写像)  $f$  による商空間 (または等化空間) と呼ぶ。商空間が等化空間とも呼ばれる理由がある。それは、既に述べたように、商空間は、各要素である同値類  $x / \sim \in X / \sim$  を、 $x / \sim$  に含まれる点  $x \in X$  と同一視することにより得られるからである。

#### 【0042】

[6] 接着空間

一般に、集合  $A$  と集合  $B$  に共通の要素がない、すなわち  $A \cap B = \emptyset$  のとき、 $A$  と  $B$  は互いに素 (mutually disjoint) であるという。互いに素である集合  $A$  と集合  $B$  の和集合  $C$  を特に直和 (disjoint union) (別名、「排他的論理和」) といい、

50

【数 4】

$$A \sqcup B$$

と表す。逆に集合  $C$  は互いに素である 2 つの集合  $A$  と集合  $B$  に直和分解されるということができる。

【0043】

さて、位相空間  $X$  から始めて、これに別の位相空間  $Y$  を接着する。図 1 は、2 つの互いに素である位相空間  $X$ 、 $Y$  から接着空間が生成される様子を説明する図である。

【数 5】

$$Y_f = Y \sqcup_f X = Y \sqcup X / \sim$$

10

は、接着写像 (attaching map)  $f$  によって  $Y$  を  $X$  に接着 (attach) することによって得られる接着空間 (attaching space) である。この接着写像  $f$  は、各点  $y \in Y_0 \mid Y_0$   $Y$  をその像  $f(y) \in X$  と同一視することにより同値関係を規定するものである。

【0044】

接着写像  $f$  は、 $f : Y_0 \rightarrow X$  である連続写像である。ただし  $Y_0 \subset Y$  である。このように、接着空間

【数 6】

$$Y_f = Y \sqcup X / \sim$$

20

は商空間の一例であり、

【数 7】

$$Y \sqcup X / \sim = Y \sqcup_f X = Y \sqcup X / (x \sim f(y) \mid \forall y \in Y_0)$$

である。この場合における等化写像 (identification map)  $g$  は、

【数 8】

$$g : Y \sqcup X \rightarrow Y \sqcup_f X = Y_f = Y \sqcup X / \sim = (Y \sqcup X - Y_0) \sqcup Y_0$$

である。すなわち、等化写像  $g$  は、互いに素である 2 つの位相空間  $X$ 、 $Y$  の直和

【数 9】

$$Y \sqcup X$$

30

を、接着写像  $f$  によって規定される同値関係による同値類の直和

【数 10】

$$(Y \sqcup X - Y_0) \sqcup Y_0$$

に写像する。

【0045】

[7] 制限と包含

任意の関数  $g : Y \rightarrow Z$  に対して、 $g$  の  $X$  ( $X \subset Y$ ) への制限 (restriction) とは、

40

【数 11】

$$g|X = g \circ i : X \rightarrow Z$$

である。ここで

$i : X \rightarrow Y$  は、包含 (inclusion) である。すなわち

$$x \in X, i(x) = x$$

である。

【0046】

[8] 連続写像のエクステンションとレトラクション

位相空間  $X$ 、 $Y$  と、部分空間  $A \subset X$  に対して、 $f|A : A \rightarrow Y$  を満たす連続写像  $f : X \rightarrow Y$

50

$Y$  を  $A$  から  $X$  への写像  $f|_A$  の連続エクステンション (または単にエクステンション) という。エクステンションは部分関数である。

【0047】

制限 (restriction)  $r$  は、 $r : X \rightarrow A$  を満たす恒等写像  $1_A : A \rightarrow A$  の連続エクステンションである。したがって、 $r|_A = 1_A$  である。

【0048】

$A \rightarrow X$  に対して、

【数12】

$$i \circ r \sim 1_X$$

10

を満たすレトラクション (retraction)  $r : X \rightarrow A$  があるならば、 $A$  を  $X$  の変形レトラクト (deformation retract) と呼び、

【数13】

$$X \rightsquigarrow A$$

で表す。

もし、 $A$  がただ一つの点  $A = \{a\} \subset X$  であるならば、 $A$  はレトラクト可能 (retractable) といい、

【数14】

$$X \rightsquigarrow *$$

20

で表す。

【0049】

[9] ホモトピー

ホモトピーはエクステンションの一例である。 $X$ 、 $Y$  を位相空間、 $f, g : X \rightarrow Y$  を連続写像、 $I = [0, 1]$  とする。ホモトピーは、次のように設計される。

$$H : X \times I \rightarrow Y$$

ただし、 $t \in I$  に対して、

$$t = 0 \text{ のとき、} H = f$$

$$t = 1 \text{ のとき、} H = g$$

30

が成り立つ。このとき、 $f$  は  $g$  にホモトピックであるという。

【0050】

ホモトピーは連続写像のエクステンションであり、

$$H|_{X \times \{0\}} = f \circ i_0$$

$$H|_{X \times \{1\}} = g \circ i_1$$

ここで

$$i_0 : X \rightarrow X \times \{0\}$$

$$i_1 : X \rightarrow X \times \{1\}$$

【0051】

さて、ホモトピーレベルにおける位相空間の設計について述べる。2つの位相空間  $X$  と  $Y$  がホモトピー同値 (homotopically equivalent) 40

【数15】

$$X \simeq Y$$

である、すなわち同じホモトピー型をもつように設計する。それは、次の条件が満たされるように設計することである。

【0052】

2つの連続写像  $f : X \rightarrow Y$  および  $h : Y \rightarrow X$  に対して、

【数 1 6】

$$h \circ f \simeq 1_X \text{ and } f \circ h \simeq 1_Y$$

である。ここで  $1_X$  および  $1_Y$  は恒等写像、すなわち  $1_X : X \rightarrow X$ 、 $1_Y : Y \rightarrow Y$  である。

【0 0 5 3】

ホモトピー同値も同値関係の一例であるが、ホモトピー同値はトポロジー同値よりも広い概念である。ホモトピー同値は、変化の後、位相幾何学的にはもはや同値ではなくなる情報の変化を同定することができる。情報はいろいろな操作や処理により変化するが、その操作および処理はホモトピーによって指定され、情報の変化はホモトピー同値によって検証される。事実、不変性 (invariance) という抽象の観点からは、ホモトピー同値は集合論的同値よりも抽象性が高い。なぜなら、与えられた集合を要素の追加や削除によって変化させるとき、追加や削除の操作手順と追加もしくは削除された要素を保存することにより、集合をホモトピー同値にすることができるからである。また、後述のセル分解もホモトピー同値になるように行うことにより、セル分解を逆にたどって分解前の状態に戻すことが可能になる。

10

【0 0 5 4】

[ 1 0 ] セル構造空間 (セル空間) 設計

セル (cell) は任意の次元 (たとえば  $n$  次元、ただし  $n$  は自然数) の閉球 (閉の  $n$  セルと呼ぶ) とトポロジー同値 (すなわち同相) である位相空間  $X$  である。

20

【0 0 5 5】

閉の  $n$  セル (closed  $n$ -cell) を

【数 1 7】

$$\mathcal{B}^n$$

と表記する。

【0 0 5 6】

開の  $n$  セル (open  $n$ -cell) を

【数 1 8】

$$\text{Int } \mathcal{B}^n = \overset{\circ}{B}^n$$

30

と表記する (しばしば  $e^n$  とも書かれる)。

【0 0 5 7】

閉の  $n$  セル

【数 1 9】

$$\mathcal{B}^n = \{x \in \mathbb{R}^n, \|x\| \leq 1\}$$

は、 $n$  次元閉球体 (closed  $n$ -dimensional ball) である。ここで

40

【数 2 0】

$$\mathbb{R}^n$$

は  $n$  次元の実数である。

【0 0 5 8】

開の  $n$  セル

【数 2 1】

$$\text{Int } \mathcal{B}^n = \overset{\circ}{B}^n = \{x \in \mathbb{R}^n, \|x\| < 1\}$$

50

は、 $n$ 次元開球体 (open  $n$ -dimensional ball) であり、 $n$ 次元閉球体

【数 2 2】

$$\mathcal{B}^n$$

の内部である。

【数 2 3】

$$\partial \mathcal{B}^n = \mathcal{B}^n - \overset{\circ}{\mathcal{B}}^n = S^{n-1}$$

は、閉の  $n$ セルの境界であり、これは  $(n - 1)$ 次元球面 (sphere)  $S^{n-1}$  である。

10

【0 0 5 9】

位相空間  $X$  に対して、特性写像 (characteristic map)

【数 2 4】

$$\mathcal{F}$$

は、連続関数

【数 2 5】

$$\mathcal{F}: \mathcal{B}^n \rightarrow X$$

であり、次を満たす。

20

【数 2 6】

$$\mathcal{F}: \overset{\circ}{\mathcal{B}}^n \rightarrow \mathcal{F}(\overset{\circ}{\mathcal{B}}^n), \text{ and}$$

$$\mathcal{F}(\partial \mathcal{B}^n) = \mathcal{F}(\mathcal{B}^n) - \mathcal{F}(\overset{\circ}{\mathcal{B}}^n)$$

【0 0 6 0】

【数 2 7】

$$e^n = \mathcal{F}(\overset{\circ}{\mathcal{B}}^n)$$

は、開の  $n$ セルであり、

30

【数 2 8】

$$\bar{e}^n = \mathcal{F}(\mathcal{B}^n)$$

は、閉の  $n$ セルである。

【0 0 6 1】

位相空間  $X$  から、整数集合  $\mathbb{Z}$  によってインデックスが与えられた、 $X$  の部分空間であるセル  $X^p$  の有限または無限の系列  $\{X^p \mid X^p \subset X, p \in \mathbb{Z}\}$  を構成することができる。これは、フィルトレーション (filtration) と呼ばれ、次の条件を満たす。 $X^p$  は  $X$  の被覆 (covering)、すなわち  $X = \bigcup_{p \in \mathbb{Z}} X^p$  であり、かつ、 $X^{p-1}$  は  $X^p$  の部分空間、すなわち  $X^0 \subset X^1 \subset X^2 \subset \dots \subset X^{p-1} \subset X^p \subset \dots \subset X$  (これはスケルトン (skeleton) と呼ばれる) である。最大  $p$  次元のスケルトンは  $p$ -スケルトンと呼ばれる。

40

【0 0 6 2】

$C = \{X^p \mid X^p \subset X, p \in \mathbb{Z}\}$  を位相空間  $X$  のセル分解 (cell decomposition)、あるいは、位相空間  $X$  の、閉セルである部分空間  $X^p$  への分割 (partition) という。 $(X, C)$  は CW 複体 (CW-complex) と呼ばれる。

【0 0 6 3】

セル接着写像を保存しながらセル分解を行うとき、セル空間を再利用可能な資源に変えることができる。そのような保存され、共有された情報をセルデータベース (cellular database) と名付け、セルデータベースを管理するシステムをセルデータベース管理シス

50

テム (cellular database management system または cellular DBMS) と名付ける。

【 0 0 6 4 】

より正確には、J. H. C. Whitehead, "Algebraic Homotopy Theory", Proceedings of International Congress of Mathematics, II, Harvard University Press, pp. 354-357, 1950によれば、位相空間  $X$  が与えられたとき、スケルトン  $X^0 \cup X^1 \cup X^2 \cup \dots \cup X^p \cup X^{p+1} \cup \dots$   $X$  をもったフィルトレーション  $X^p$  を以下の条件を満たす位相空間として帰納的に構成することができる。

( 1 )  $X^0 \cup X$  は、その要素が  $X$  の 0 次元セルである部分空間であり、かつ

( 2 )  $X^p$  は  $X^{p-1}$  から次のようにして作られる。すなわち、 $X^p$  は、接着写像と呼ばれる全射かつ連続な写像

10

【数 2 9】

$$F: \sqcup_i \partial \mathcal{B}_i^p \rightarrow X^{p-1}$$

によって、 $X^{p-1}$  に  $p$  次元閉球体の直和 (disjoint union)

【数 3 0】

$$\sqcup_i \mathcal{B}_i^p$$

を接着することによって作られる。

【 0 0 6 5 】

言い換えれば、 $X^p$  は、直和

20

【数 3 1】

$$X^{p-1} \sqcup (\sqcup_i \mathcal{B}_i^p)$$

を求め、

【数 3 2】

$$\partial \mathcal{B}_i^p$$

における各点  $x$ 、すなわち

【数 3 3】

$$x \in \partial \mathcal{B}_i^p$$

30

を、連続写像

【数 3 4】

$$F_i = F|_{\partial \mathcal{B}_i^p} : \partial \mathcal{B}_i^p \rightarrow X^{p-1}$$

(ただし、各インデックス  $i$  に対して  $x \sim f_i(x)$  である)

によって、その像

【数 3 5】

$$\mathcal{F}(x)$$

40

と同一視 (identify) することによって、 $X^{p-1}$  から構成される。

【 0 0 6 6 】

このように、 $X^p$  は商空間 (quotient space) あるいは等化空間 (identification space) であり、

【数 3 6】

$$\begin{aligned} X^p &= X^{p-1} \sqcup (\sqcup_i \mathcal{B}_i^p) / (x \sim F_i(x) \mid \forall x \in \partial \mathcal{B}_i^p) \\ &= X^{p-1} \sqcup_F (\sqcup_i \mathcal{B}_i^p) \end{aligned}$$

50

と書くことができ、これは、接着空間の一例である。写像  $F_i$  はセル

【数 3 7】

$$\mathcal{B}_i^p$$

の接着写像の一例である。

【0 0 6 7】

フィルトレーション空間 (filtration space) は、フィルトレーションとホモトピー同値な (homotopically equivalent) 空間である。位相空間  $X$  とスケルトン  $X^0 \quad X^1 \quad X^2 \quad \dots \quad X^{p-1} \quad X^p \quad \dots \quad X$  を合わせて  $CW$  空間 ( $CW$ -space) と呼ぶ。セル複体としては、先に説明したように、これは  $CW$  複体 ( $CW$ -complex) と呼ばれる。

10

【0 0 6 8】

このようにして、写像

【数 3 8】

$$\mathcal{F}$$

を等化写像 (identification map) の一例として得る。

【数 3 9】

$$\mathcal{F}: X^{p-1} \sqcup (\sqcup_i \mathcal{B}_i^p) \rightarrow X^{p-1} \sqcup_{\mathcal{F}} (\sqcup_i \mathcal{B}_i^p) = X^p$$

【0 0 6 9】

各  $n$  セル

【数 4 0】

$$\mathcal{B}_i^p = \mathcal{F}(\mathcal{B}_i^p) \in X^p$$

に対する特性写像 (characteristic map)

【数 4 1】

$$\mathcal{F}$$

は、

【数 4 2】

$$\mathcal{F}_i = \mathcal{F} | \partial \mathcal{B}_i^p : \partial \mathcal{B}_i^p \rightarrow X^{p-1}$$

と書ける。

【0 0 7 0】

$X^{p-1}$  を  $X^p$  の閉の部分空間として埋め込むことは

【数 4 3】

$$\mathcal{F} | X^{p-1} = X^{p-1} \rightarrow X^p$$

と書ける。

【0 0 7 1】

$CW$  空間が微分同相 (diffeomorphic) であるなら、これは多様体 (manifold) 空間と等価である。

【0 0 7 2】

サイバースペースに構築されるオブジェクトの属性をこのようなセルで表現した空間をセル構造空間 (cellular structured space) もしくは単にセル空間 (cellular space) という。セルモデリングでは、セルの次元を維持しながら、セル構築 (composition) とセル分解 (decomposition) が可能である。

【0 0 7 3】

セルの次元の例を述べる。たとえばサイバーワールドにおいて、ひとつの属性をもつオブジェクトは、ひとつの属性から他の属性へ移行することができないため、その自由度は

30

40

50



0 であり、したがってセルの次元は 0 である。このオブジェクトは、表現レベル (representation level) では点として表される。ここで、属性 (attribute) とは、オブジェクトが本来有する特質や特徴を指定するための互いに独立な集合をいう。

#### 【 0 0 7 4 】

属性をふたつ有するオブジェクトでは、一方の属性から他方の属性への移行が可能なため、その自由度は 1 であり、セルの次元も 1 である。このオブジェクトは、表現レベルにおいて直線として表すことができる。同様に、属性を 3 個または 4 個もつオブジェクトは、それぞれ自由度すなわち次元が 2 または 3 であり、それぞれ曲面または球として表現できる。一般に、 $n$  個の属性を有するオブジェクトは  $n - 1$  の自由度を有し、次元は  $n - 1$  である。これは、 $(n - 1)$  次元の球として表現できる。リレーショナルモデルは、 $n$  個の属性をもつオブジェクトをリレーショナルスキーマとして表現し、そのインスタンスとして  $n$  列のテーブルを生成する。リレーショナルモデルは集合の直積にもとづくものであり、したがってそれは集合論レベルにおける表現といえる。

10

#### 【 0 0 7 5 】

セル空間は、接着空間に次元の概念を加えたものであり、セル空間設計では、接着空間におけるオブジェクトの属性の次元が規定される。同値類の直和である接着空間において、各同値類に含まれるオブジェクトの属性の次元に不整合がある場合、オブジェクトの属性の次元を調整する処理が行われる。

#### 【 0 0 7 6 】

#### [ 1 1 ] 表現設計

20

セル空間レベルでセルの次元が設計された後、表現レベルにおいてセルの表現 (presentation) が設計される。表現設計では、オブジェクトの属性のデータ型や桁数などの表現形式が詳細化される。オブジェクトの属性の表現形式に不整合がある場合、表現形式を相容表現 (Admissible Representation) に合わせる処理が行われる。オブジェクトの属性の表現形式の不整合には、たとえば、データ型または桁数の違いがあり、データ型変換や桁数変換により、その違いを解消することができる。

#### 【 0 0 7 7 】

#### [ 1 2 ] ビュー設計

表現レベルでセルの属性の表現形式が詳細化された後、ビュー (view)、すなわちユーザが見ることのできる範囲が決められる。ユーザは、オブジェクトの集合のうち、関心のある部分だけを適当な形式で見たいという要求がある。また、セキュリティ上の理由などによりユーザが参照することのできる範囲を制限しなければならない。そのため、ビュー設計では、ユーザ向けに参照可能なオブジェクトの部分集合や、閲覧可能なオブジェクトの属性を制限したビューが決められる。

30

#### 【 0 0 7 8 】

#### [ 1 3 ] インクリメンタリモジュラな抽象階層

これまで述べてきたホモトピー、集合、位相空間、接着空間、セル空間、表現、ビューの各抽象レベルは、インクリメンタリモジュラな抽象階層 (incrementally modular abstraction hierarchy) を形成する。

- 1 . ホモトピーレベル
- 2 . 集合論レベル
- 3 . 位相空間レベル
- 4 . 接着空間レベル
- 5 . セル空間レベル
- 6 . 表現レベル
- 7 . ビューレベル

40

#### 【 0 0 7 9 】

これらの階層は、モジュール化されており、各階層においてインクリメンタルに情報を操作していくことが可能であり、サイバーワールドの不変量 (invariant) を示す同値関係を階層的に継承する上できわめて有用である。

50

## 【0080】

まず、複数のオブジェクト集合の特定の要素について同値にする対象が選ばれる。次に、オブジェクト集合に位相が規定され、位相空間において同値関係が規定されることにより、複数のオブジェクト集合は、同値類の直和である接着空間として設計される。さらに、セル空間において同値セルとそれ以外のセルにセル分解されることにより、セル空間レベルまで同値関係が継承される。

## 【0081】

さらには、受容表現にもとづいてセルの表現形式を規定することにより表現レベルまで同値関係が継承され、最後にユーザのビューが規定されることによりビューレベルまで同値関係が継承される。表現レベルの1つのインスタンスとして、受容表現を規定するための受容表現レベルを設けておけば、オブジェクトを表現レベルで同値にすることができる。

10

## 【0082】

このようにインクリメンタリモジュラな抽象階層モデルでは、同値関係を抽象階層間で継承しながら、オブジェクトの満たすべき仕様を各階層において設計するため、情報の追加、削除、修正をインクリメンタルに行いながら、抽象階層間を行き来してオブジェクトの仕様を詳細化できる。また、こうして設計されたオブジェクトは再利用可能な形で部品化することができる。

## 【0083】

集合 $X$ の部分集合からなる集合 $S$ において、 $S$ に属するすべての集合の和集合が $X$ に等しいとき、 $S$ を集合 $X$ の被覆 (covering) という。 $n$ 個の情報システムの統合を行うとき、「経営上の業務統合決定」は、インクリメンタリモジュラな抽象階層の上で、 $n$ 個の集合の被覆を作ることの決定がなされたことに相当する。ここで、業務は別々の機関のものであってもかまわない。 $n$ 個の集合は一般には重なりがある。たとえば、複数の業務に同じ社員が参加することなどがあるからである。この $n$ 個の集合の被覆が、インクリメンタリモジュラな抽象階層間で継承されることにより、業務統合が実装される。言い換えれば、ホモトピーレベル、位相空間レベル、接着空間レベル、セル空間レベル、表現レベル、およびビューレベルのそれぞれで業務の被覆が生成されることが、業務統合になる。いったん被覆が生成されると、 $n$ 個の情報システムの間の統合および操作は、インクリメンタリモジュラな抽象階層の上ですべて被覆を通して行われる。その結果、 $n$ 個の情報システムの間の統合および操作は、いずれも線形になる。このように、インクリメンタリモジュラな抽象階層によれば、 $n$ 個の情報システムの間で被覆という線形インタフェースを構成して線形インターオペラビリティを実現することができる。

20

30

## 【0084】

また、インクリメンタリモジュラな抽象階層モデルにおける位相空間は、ハウスドルフ空間 (Hausdorff space) すなわち $T_2$ 位相空間に限られない。 $T_2$ 位相空間では、分離公理により、異なる2点が開集合で分離できるが、異なる2点を開集合で分離することのできない $T_0$ 位相空間や $T_1$ 位相空間であっても、同様に接着空間が設計できることに留意する。したがって、抽象階層モデルは、非常に汎用性が高いものである。

## 【0085】

接着空間は、主要な企業や公的機関で用いられている有力な商用の情報システムに共通した特性を接着空間という形で異なる情報システム間で同値になるように抽象化し、モデル構築する。このように、接着空間は、異種情報システムを線形統合することのできる斬新なデータモデルであり、統合作業量の組み合わせ爆発を避けることに大いに貢献する。接着空間レベルにおける線形統合後の線形インタフェース生成の自動化のために、上記のインクリメンタリモジュラな抽象階層が用いられる。この抽象階層によれば、同値関係が下位の階層まで継承されるため、既存の情報システムに対するインタフェースを与えて、統合システム規模のタスクを実行するための線形インターオペラビリティを実現することができる。

40

## 【0086】

50

比較のために、リレーショナルデータベースシステムを例に挙げて説明する。リレーショナルデータベースシステムにおいて、リレーションはタプルの集合である。リレーションに新たなタプルが挿入されたり、既にあるタプルが削除されたり、タプルの属性値が修正されることにより、リレーションは時間とともに変化していく。たとえば、リレーションとして「社員（社員番号，氏名，所属，入社年度）」を考える。新しい社員が入社した場合、リレーション「社員」にその社員を表すタプルを挿入しなければならない。一方、ある社員が退職すると、リレーション「社員」にその社員を表しているタプルを削除しなければならない。また、配置換えにより、ある社員の配属先が変わった場合、その社員のタプルにおいて属性「所属」の値を修正しなければならない。

【0087】

10

このように、リレーショナルデータベースでは、タプルの挿入、削除、修正により、リレーションが時間とともに変化していく。しかしながら、リレーションには、社員の社員番号、氏名、所属、入社年度のデータを表しているという固有の性質があり、これは時間に対して不変である。リレーションにはこのように時間に対して不変な構造があり、これをリレーションスキーマと呼んでいる。

【0088】

リレーショナルデータベースモデルは、集合論にもとづくものであるから、位相空間のレベルで同値関係を定義することができない。したがって、複数の異なるデータベースを統合しようとする、IDのつけなおしなど、データベースの再設計を行う必要があり、工数爆発の問題が起こる。それに対して、インクリメンタリモジュラな抽象階層モデルでは、上位概念で規定された同値関係を表現レベルまで落とし込むことができるため、複数の異なる情報システムの統合が線形操作で可能となる。すなわち情報システムの各要素のインタフェイスは、接着空間に一回変換するだけでとることができ、 $n$ 要素で $n$ 回の変換でインターオペラビリティが保証され、インタフェイスの数は線形である。

20

【0089】

このように、本発明の抽象階層モデルに対して、従来のリレーショナルモデルは、同値関係をモデル化していないため、情報の統一的な扱いができない。せいぜい、テーブル間の関係を示す別テーブルを、パッチを当てる要領で設けてインタフェイスをとろうとしているが、これではインタフェイスの数が指数関数的に増える。一方、ER (Entity Relation) モデルを基本とするSAP、UML (Unified Modeling Language)なども、これらはすべて直観的なグラフ理論モデルであり、そのモデル自身が同値関係を扱うことができないため、同様の問題が起こる。これが社会基盤情報システムの現状であり、システムが明確なインターオペラビリティをもたないことが大きな問題となっている。

30

【0090】

[14] 応用例

[14.1] Web情報モデリング

通常、Web情報マネジメントシステムのやるべき仕事は、Webグラフィックスを用いて情報を可視化することを通して、人間の認識との間で非常に近い相互作用をもちながら、Web上の情報を管理することである。Webグラフィックスの仕事は、人間の理解のためにグラフィックススクリーンにいろいろなイメージを投影することである。人間は、ディスプレイ空間に表示されたイメージを人間の認識空間における認識対象物にリンクすることによって、表示されたイメージを理解している。しばしば、幾何学的に正確な表示であっても、サイバーワールドでは通常は本質的な情報ではない、優先度の低い幾何学的形状によって、人間の認識を誤りに導くことがある。Web情報マネジメントに対してWebグラフィックスは、サイバーワールドの変化に合った速度で、人間が即座に認識できるようにスクリーン上に重要なメッセージを伝えることができない。以下、簡単な例を挙げて説明する。

40

【0091】

[14.2] エレクトロニックファイナンスのWeb情報モデリング

エレクトロニックファイナンスにおいて、顧客Xが、Webサーフィンしているときに

50

、金融取引会社 Y のホームページの Web ページに投稿されている収益の期待できるファンド  $Y_0$  を見つけたとする。この時点では、顧客 X は Web で「ウインドウショッピング」をしている段階であるから、顧客 X と取引会社 Y はファンドをまだ共有していない。したがって、X と Y は互いに素であり、共通要素を持たない。すなわち、

【数 4 4】

$Y \sqcup X$

と表記することができる。一般化のために、X と Y を位相空間とする。ファンド  $Y_0$  は金融取引会社 Y の属性の一部であるから、 $Y_0 \subset Y$  が成り立つ。図 2 は、Web 上のエレクトロニックファイナンスのプロセスを Web グラフィックスにおいて図示したものである。次に、ファンドが取引のために特定された後、顧客 X は取引会社 Y とどのようにして関係づけられるかを述べる。ここで提案する Web 情報モデルは、顧客 X と取引会社 Y の関係を接着写像  $f$  によって正確に表現し、また、「ファンド  $Y_0$  が取引のために特定されている」という状況を 2 つの互いに素である位相空間 X (顧客) と位相空間 Y (金融取引会社) の接着空間として表現する。この接着空間は、顧客 X から始めて、金融取引会社 Y を顧客 X に接着することにより得られる。ここで、Y の X への接着は、連続写像  $f$  を介して、各点  $y \in Y_0 \subset Y$  をその像  $f(y) \in X$  と同一視することによって行われ、その結果、 $x \sim f(y) \mid y \in Y_0$  である。このように、 $\sim$  で表記される同値は、Web 情報の接着空間モデルとして接着空間を作成するために、Web 情報モデリングにおいて中心的な役割を果たす。

【0 0 9 2】

上記の接着空間モデルは、エレクトロニックマニュファクチャリングにおいてもまったく本質的に等しく適用することができる。エレクトロニックマニュファクチャリングプロセスを管理し、表示するために正確に同一の技術が要求される。エレクトロニックマニュファクチャリングが市場の需要に効果的に即応するためには、いかにしているいろいろなサイズのコンポーネントを統一したデザインで組み立てるかを指定しなければならない。先に提示したエレクトロニックファイナンスは、顧客と取引会社をエレクトロニックマニュファクチャリングのコンポーネントとして組み立てることであると考えることにより、エレクトロニックファイナンスのための Web 情報マネジメントシステムと Web グラフィックスは、実際にエレクトロニックマニュファクチャリングに適用可能となる。もし、異なるアプリケーションに異なる技術を使うとなると、急成長しているサイバーワールドは、タイムリーに Web 情報マネジメントシステムで管理することも、Web グラフィックスで表示することもできなくなる。

【0 0 9 3】

[ 1 4 . 3 ] エレクトロニックマニュファクチャリングの Web 情報モデリング

接着空間モデルによるエレクトロニックマニュファクチャリングの Web 情報モデリングはまったく簡単である。基本的には、エレクトロニックマニュファクチャリングと呼ばれる Web 上のマニュファクチャリングは、エレクトロニックマニュファクチャリングの工程に関する次の情報からなる Web 情報としてモデル化される。

- 1) 製品仕様
- 2) 組み立て仕様
- 3) Web 上での部品購入
- 4) Web 上の組み立てサイトでの購入

【0 0 9 4】

各ステップは、必要に応じてより細かいサブステップに分解される。たとえば、ステップ 1 は次のサブステップに分解できる。

- 1 . 1) 電子市場での製品のマーケット調査
- 1 . 2) マーケット調査から導き出される製品の要求
- 1 . 3) 要求を満たすための製品の仕様

【0 0 9 5】

10

20

30

40

50

エレクトロニックマニファクチャリングのためのWeb技術全体の中で核となる部分は、Web情報モデリングとしてのWeb上での製品と組み立てのモデリングである。図3は、座席と支持部という2つのコンポーネントをもつ椅子の単純な組み立て例を用いて、最も初歩的な組み立てのモデリングを明確に図示したものである。発展したマニファクチャリングとしてのエレクトロニックマニファクチャリングにおいて、製品のコンポーネントは、モジュール化されており、より品質の高いコンポーネントを買うため、最も有効なアップグレードを行うため、あるいは修理するために置き換え可能に構成されている。

【0096】

エレクトロニックファイナンスとエレクトロニックマニファクチャリングが接着空間と同値にもとづいて同一の情報モデリングを共有することは明らかである。

【0097】

[15] 実施例

図4は、実施の形態に係るソフトウェア開発支援装置100の構成を示す。オブジェクト集合記憶部190は、ソフトウェア開発に利用されるオブジェクトの集まりを保持する。ソフトウェア開発に利用されるオブジェクトすなわちソフトウェア部品は、データ構造体やプログラムモジュールである。ソフトウェア開発支援装置100は、オブジェクト集合記憶部190に格納されたオブジェクトの集まりを参照する。ソフトウェア開発支援装置100は、オブジェクト集合記憶部190を介さずに、ウェブ上の情報システムからオブジェクト集合を取得して参照してもよい。以下、実施の形態では、説明の便宜上、2つのオブジェクト集合の間に所定の対応関係またはインタフェースを定義する場合を説明するが、これはより一般的に複数のオブジェクト集合を参照してそれらの間にインタフェースを定義する場合の一例に過ぎない。

【0098】

ソフトウェア開発支援装置100は、ホモトピーレベル処理ブロック110、集合レベル処理ブロック120、位相空間レベル処理ブロック130、接着空間レベル処理ブロック140、セル空間レベル処理ブロック150、表現レベル処理ブロック160、およびビューレベル処理ブロック170を含み、それぞれ既に述べたインクリメンタリモジュールな抽象階層モデルにおける各階層レベルの処理を行う。

【0099】

集合レベル処理ブロック120は、オブジェクト集合記憶部190に格納されたオブジェクトの集まりに集合演算を適宜施して、設計対象となる2つのオブジェクト集合を特定する。位相空間レベル処理ブロック130は、集合レベル処理ブロック120が特定した2つのオブジェクト集合に対して位相を規定することで、2つのオブジェクト集合はそれぞれ位相空間X、Yとして設計する。この時点で2つの位相空間X、Yは互いに素であり、これらの位相空間X、Yの直和あるいは排他的論理和である

【数45】

$Y \sqcup X$

が確定する。

【0100】

接着空間レベル処理ブロック140は、図5にその詳細な構成を示すように、同値関係設定部44、同値関係保持部52、対応要素抽出部20、接着写像取得部22、等化写像取得部24、手順取得部26、および手順記録部28を含む。

【0101】

同値関係設定部44は、2つのオブジェクト集合に適用したい同値関係の入力をユーザから受け付ける他、ユーザの入力をもとにして自動的に同値関係を生成する。ユーザが同値関係を設定しやすいように、同値関係保持部52は同値関係の候補があらかじめ格納されている。また、ユーザがオブジェクト集合の要素を指定して同値関係を記述することのできるインタフェースを設けてもよい。

10

20

30

40

50

## 【 0 1 0 2 】

同値関係設定部 4 4 により設定された同値関係（以下、目的同値関係ともいう）は、対応要素抽出部 2 0 へ伝えられる。対応要素抽出部 2 0 は、2 つのオブジェクト集合において対応関係を定義すべき構成要素（以下、「対象要素」ともいう）を、目的同値関係をもとに抽出する。その際、ユーザが 1 つのオブジェクト集合において対象要素の一方を指定し、対応要素抽出部 2 0 が目的同値関係をもとに他方のオブジェクト集合から対応する対象要素を抽出するようにしてもよい。

## 【 0 1 0 3 】

対応要素抽出部 2 0 により対応しあうことが判明した対象要素は、ひとつの部分位相空間、すなわち同値類を形成し、接着空間モデルの理論的保証により、同値類の排他的論理和として接着空間が形成される。同値類は同値関係で規定される不変量をもとに分類されるため、オブジェクト集合を客観的に分類することができ、分類の結果にはシステム設計者による個人差が含まれない。また、一方の対象要素を決めると、他方の対象要素が決まるため、前述のごとく、対応関係またはインタフェースの数が線形になり、インターオペラビリティが保証されることになる。

## 【 0 1 0 4 】

目的同値関係が決まると、対象要素間がどのような意味において対応するか、その関係も一意に決まる。そのため、接着写像取得部 2 2 は対象要素間の対応を示す接着写像  $f$  を特定することができる。接着写像  $f$  が決まると、接着空間も、

## 【 数 4 6 】

$$Y \sqcup_f X = Y \sqcup X / \sim$$

と定まる。接着空間は、もとは共通部分をもたない位相空間  $X$  と  $Y$  の間に、接着写像  $f$  を媒介として対応関係を持たせた位相空間である。

## 【 0 1 0 5 】

等化写像取得部 2 4 は、位相空間  $X$  と  $Y$  の単なる排他的論理和の空間から、接着写像  $f$  によって定まる接着空間への写像である等化写像  $g$  を取得する。等化写像  $g$  は、現実には、その定義から位相空間  $X$ 、 $Y$  および接着写像  $f$  によって記述することができる。

## 【 0 1 0 6 】

手順取得部 2 6 は、同値関係設定部 4 4 から目的同値関係、対応要素抽出部 2 0 から対象要素、接着写像取得部 2 2 から接着写像  $f$ 、等化写像取得部 2 4 から等化写像  $g$  の入力を受け、ユーザにより指定された対象要素および目的同値関係（以下、これらを「注目手順」ともいう）をホモトピーとして、一連の処理の結果で特定される、

- ・ 位相空間  $X$ 、 $Y$
- ・ 目的同値関係
- ・ 接着写像  $f$
- ・ 等化写像  $g$

と関連づけて手順記録部 2 8 へ記録する。

## 【 0 1 0 7 】

セル空間レベル処理ブロック 1 5 0 は、図 6 にその詳細な構成を示すように、セル接合部 6 2、セル次元調整部 6 4、および対応テーブル保持部 6 6 を含む。セル空間レベル処理ブロック 1 5 0 は、接着空間レベル処理ブロック 1 4 0 において目的同値関係をもとに接着空間として設計された複数のオブジェクト集合の仕様を継承して、セル空間レベルでオブジェクト集合の仕様を詳細化する。セル構造空間モデルは、接着空間モデルに次元を導入したものであり、オブジェクトの属性の ID や属性の個数に関する仕様がセル空間レベルで詳細化されることになる。

## 【 0 1 0 8 】

セル接合部 6 2 は、2 つのオブジェクト集合間で目的同値関係により対応関係にある 2 つの対象要素を接合し、各オブジェクト集合に属するオブジェクトの属性をセルで表現す

10

20

30

40

50

る。セル次元調整部 6 4 は、対応する 2 つの対象要素間に属性の次元に関する不整合がある場合、その不整合を解消するための処理を行う。セルの次元に関する不整合として、対象要素の属性の ID や個数の違いなどがある。セル次元調整部 6 4 は、必要に応じて、2 つの対象要素間で属性の ID や属性の個数の不揃いを調整し、2 つの対象要素間の属性の対応関係を示すテーブルを対応テーブル保持部 6 6 に格納する。

#### 【0109】

表現レベル処理ブロック 1 6 0 は、図 7 にその詳細な構成を示すように、表現生成部 7 2、表現形式調整部 7 4、および変換テーブル保持部 7 6 を含む。表現レベル処理ブロック 1 6 0 は、セル空間レベル処理ブロック 1 5 0 においてセル空間レベルで詳細化されたオブジェクト集合の仕様を継承して、表現レベルでオブジェクト集合の仕様をさらに詳細化する。表現レベルでは、セルの具体的な表現形式を決めるため、オブジェクトの属性のデータ型や桁数などに関する仕様が詳細化されることになる。

10

#### 【0110】

表現生成部 7 2 は、2 つのオブジェクト集合のそれぞれに属するオブジェクトの属性の表現を生成する。表現形式調整部 7 4 は、目的同値関係により対応関係にある 2 つの対象要素間に表現レベルの不整合がある場合に、その不整合を調整するための処理を行う。表現レベルの不整合として、対象要素の属性のデータ型や桁数の違いなどがある。データ型には、ブール値、符号付き整数、正の整数、実数、倍精度実数、文字型などがあり、表現形式調整部 7 4 はこれらのデータ型を変換して対象要素のデータ型を揃えてもよい。また、データ型が同じでも文字列の桁数や数値の有効桁数などの桁数が異なる場合には、表現形式調整部 7 4 は 2 つの対象要素間で属性の桁数を調整して揃えてもよい。その他の不整合の例として、計量や貨幣などの単位の違いがあり、変換式を用いて、一方の単位系が他方の単位系に変換される。

20

#### 【0111】

表現形式調整部 7 4 は、位相空間 Y を位相空間 X に接着する場合、表現レベルにおいて、位相空間 X のオブジェクト集合の対象要素と位相空間 Y のオブジェクト集合の対象要素の属性のデータ型や桁数を比較して、相対的に精度が高い方の属性に揃えて、データ型や桁数を変換する。3 つ以上のオブジェクト集合の対象要素間で属性の表現レベルの不整合を調整する場合も同じであり、表現形式調整部 7 4 は、複数のオブジェクト集合の対応する対象要素の属性表現の内、最も精度の高い属性表現を選んで、他の精度の低い対象要素の属性表現をその精度の高い対象要素の属性表現に合わせるように調整する。これにより、精度の異なる対象要素の接着後に、データ型変換や桁落ちなどにより元のデータが失われるのを防ぐことができる。

30

#### 【0112】

表現形式調整部 7 4 は、最大の文字数や桁数をもった受容表現をもった表現フィルタとして作用し、当該表現フィルタを通過した表現は、表現レベルで同値になる。表現形式調整部 7 4 は、表現の不整合の調整をした場合の変換規則を記述したテーブルを変換テーブル保持部 7 6 に格納する。

#### 【0113】

ビューレベル処理ブロック 1 7 0 は、ユーザに合わせたビューを設定し、ビューにしたがってオブジェクト集合をディスプレイなどに表示する。

40

#### 【0114】

ホモトピーレベル処理ブロック 1 1 0 は、ソフトウェア開発支援装置 1 0 0 の各処理ブロックがオブジェクト集合に対して操作を行った際、その操作によって生じるオブジェクト集合の変化がホモトピー同値になるように保証する。ホモトピーレベル処理ブロック 1 1 0 は、手順記録部 2 8 に記録された接着空間生成に係る注目手順、対応テーブル保持部 6 6 に格納されたセルの次元の不整合の対応関係を示すテーブル、変換テーブル保持部 7 6 に格納されたセルの表現の変換規則を示すテーブルを利用し、各操作によってオブジェクト集合にもたらされた変化を逆にたどることにより、オブジェクト集合を変化前の状態に戻すことができる。

50

## 【 0 1 1 5 】

このようにして、集合レベル処理ブロック 1 2 0 から始めて、位相空間レベル処理ブロック 1 3 0、接着空間レベル処理ブロック 1 4 0、セル空間レベル処理ブロック 1 5 0、表現レベル処理ブロック 1 6 0、およびビューレベル処理ブロック 1 7 0 へと、同値関係を継承しつつ、2つのオブジェクト集合の仕様を段階的に詳細化していくことで、2つのオブジェクト集合が表現レベルまで同値となって出力される。このようにインクリメンタリモジュラな抽象階層モデルにもとづくソフトウェア開発支援装置 1 0 0 は、同値関係を、集合レベルから位相空間レベル、接着空間レベル、セル空間レベル、表現レベル、およびビューレベルまで保証する一貫性のあるフィルタとして機能し、フィルタを通過したオブジェクト集合は同値関係が数学的に保証されたインスタンスとなって出力される。

10

## 【 0 1 1 6 】

対応づけたい構成要素が複数あれば、それぞれの構成要素について目的同値関係を設定することによって、同様に接着写像が規定され、最終的に複数のオブジェクト集合間で対応する要素が接着された結果が出力される。一方のオブジェクト集合が注目する構成要素を  $n$  個もてば、その  $n$  個のそれぞれについて、他方のオブジェクト集合のいずれかの構成要素との対応関係が特定されるため、インタフェイス数は線形であり、工数爆発の問題が解決する。

## 【 0 1 1 7 】

本実施の形態によれば、同値関係を設定すれば、複数のオブジェクト集合の構成要素間で対応づけができ、対応していない構成要素はそのまま残しつつ、同値関係にある構成要素については常に同じ情報をもつことができ、不変量を切り口に分類や対応づけができるので、データ利用性が向上し、ソフトウェア開発の効率を各段に高めることができる。

20

## 【 0 1 1 8 】

## [ 1 6 ] 具体例

以上の構成のソフトウェア開発支援装置 1 0 0 によるソフトウェア開発支援手順の具体例を説明する。ソフトウェア開発支援装置 1 0 0 が各種プログラムモジュールをもとにソフトウェアを開発する際、プログラムモジュール間のインタフェイスを確定し、各プログラムモジュールで利用されるデータの仕様を詳細化する。[ 1 6 . 1 ] でデータの設計手順を述べ、[ 1 6 . 2 ] でプログラムモジュールの設計手順を述べる。

## 【 0 1 1 9 】

30

## [ 1 6 . 1 ] データ設計

ソフトウェアにおいて利用される複数のオブジェクト集合の一例として、シューズショップ X のオブジェクト集合 3 0 0 と、シューズメーカー Y のオブジェクト集合 2 0 0 を考える。これらのオブジェクト集合は、集合レベル処理ブロック 1 2 0 によって設計され、オブジェクト集合記憶部 1 9 0 に格納されている。これらのオブジェクト集合はデータベースとして管理されているものであってもよい。位相空間レベル処理ブロック 1 3 0 は、シューズショップ X のオブジェクト集合 3 0 0 を位相空間 X として、シューズメーカー Y のオブジェクト集合 2 0 0 を位相空間 Y として設計する。この時点で、2つの位相空間 X、Y は互いに素である。

## 【 0 1 2 0 】

40

接着空間レベル処理ブロック 1 4 0 は、オブジェクト集合記憶部 1 9 0 に格納されたシューズショップ X のオブジェクト集合 3 0 0 を位相空間 X として、シューズメーカー Y のオブジェクト集合 2 0 0 を位相空間 Y として参照し、同値関係をもとに2つの位相空間 X、Y から接着空間を生成する。図 8 に、接着空間レベル処理ブロック 1 4 0 により2つの位相空間が接着される様子を示す。

## 【 0 1 2 1 】

いま、シューズメーカー Y の製造する特定の運動靴 Y<sub>0</sub> をシューズショップ X が仕入れて、販売しているとする。このとき、ソフトウェアの開発者は、シューズショップ X のオブジェクト集合、シューズメーカー Y のオブジェクト集合との間でこの運動靴 Y<sub>0</sub> をプログラム上では同一のオブジェクトとして扱いたい。同値関係設定部 4 4 は、シューズメー

50



カー Y とシューズショップ X の位相空間の間で、注目すべき対象要素である運動靴  $Y_0$  を同値関係にあるものとして設定する。

【0122】

対応要素抽出部 20 は、シューズメーカー Y の運動靴  $Y_0$  (符号 202) に対応するシューズショップ X の運動靴  $f(Y_0)$  (符号 302) を抽出する。接着写像取得部 22 は、この対応関係をもとに、接着写像  $f: Y_0 \rightarrow X \mid Y_0 \rightarrow Y$  を取得する。

【0123】

等化写像取得部 24 は、接着写像取得部 22 が取得した接着写像  $f$  によって、シューズショップ X とシューズメーカー Y の 2 つの位相空間の排他的論理和を接着空間

【数 47】

$$Y_f = Y \sqcup_f X$$

に写像する等化写像  $g$  を取得する。すなわち、等化写像取得部 24 は、等化写像

【数 48】

$$g: Y \sqcup X \rightarrow Y \sqcup_f X$$

を取得する。ここで、

【数 49】

$$Y_f = Y \sqcup_f X = Y \sqcup X / \sim = Y \sqcup X / (x \sim f(y) \mid \forall y \in Y_0)$$

であり、同図に示すように、運動靴  $f(Y_0)$  (符号 402) において 2 つの位相空間 X、Y が接着する。

【0124】

図 9 は、シューズメーカー Y の位相空間における運動靴 202 のセル構造を説明する図である。運動靴 202 は、セルの属性として、靴のブランド名 210、会社名 212、種類 214、サイズ 216、色 218、素材 220、希望小売価格 222、製造原価 224、および原産国 226 を含む。

【0125】

図 10 は、シューズショップ X の位相空間における運動靴 302 のセル構造を説明する図である。運動靴 302 は、セルの属性として、靴のブランド名 310、メーカー名 312、種類 314、サイズ 316、色 318、素材 320、売価 322、および販売店 324 を含む。

【0126】

セル空間レベル処理ブロック 150 のセル接合部 62 は、図 9 に示すシューズメーカー Y の位相空間における運動靴 202 と、図 10 に示すシューズショップ X の位相空間における運動靴 302 をセルレベルで接合するため、共通する属性を互いに対応づける。靴のブランド名 210、310、種類 214、314、サイズ 216、316、色 218、318、素材 220、320 は属性名が一致するため、共通する属性として対応づけられる。

【0127】

シューズメーカー Y の運動靴 202 の会社名 212 と、シューズショップ X の運動靴 302 のメーカー名 312 は、属性名は異なるが同一の要素である。そこで、セル空間レベル処理ブロック 150 のセル次元調整部 64 は、シューズメーカー Y 側の属性名をシューズショップ X 側の属性名に対応づけ、接合セルにおける属性名を「メーカー名」に統一することで不整合を調整し、シューズメーカー Y 側の「会社名」をシューズショップ X 側の「メーカー名」に対応づける関係を記述したテーブルを対応テーブル保持部 66 に保持する。

【0128】

シューズメーカー Y の運動靴 202 とシューズショップ X の運動靴 302 をセルレベルで接合する際、一方の位相空間に属性が存在し、他方の位相空間にはそれに対応する属性

10

20

30

40

50

が存在しないことがある。たとえば、シューズメーカーYの運動靴202には、原産国226という属性があるが、シューズショップXの運動靴302には、それに対応するものがない。逆に、シューズショップXの運動靴302には、販売店324という属性があるが、シューズメーカーYの運動靴202には、それに対応するものがない。このように、2つの対象要素間で属性の対応付けをする際、第1の位相空間に属性aが存在し、第2の位相空間にはそれに対応する属性が存在しない場合、第1の位相空間の属性aには対応づけられる属性がないとものして処理するか、もしくは第2の位相空間において空の属性bを追加して第1の位相空間の属性aと対応づける処理を行う。たとえば、セル次元調整部64は、シューズメーカーYの運動靴202の原産国226については対応づけられる属性がないとして処理し、一方、シューズショップXの運動靴302の販売店324につい

10

#### 【0129】

属性の個数の調整の他の例として、一方の位相空間に「名前」という1つの属性があり、他方の位相空間には、「姓」と「名」という2つの属性がある場合、実質的には両者は同一の属性であるから、セル次元調整部64は、「名前」という1つの属性と、「姓」と「名」という2つの属性を対応づけることで、属性の個数の不整合を調整することができる。

#### 【0130】

20

図11(a)~(c)は表現レベルでの不整合の調整を説明する図である。シューズメーカーY側の会社名212は、図11(a)に示すように、M桁の文字列で表現されている。一方、シューズショップX側のメーカー名312は、図11(b)に示すように、L桁の文字列で表現されている。ここで、 $L < M$ である。表現レベル処理ブロック70の表現形式調整部74は、桁数の大きいM桁に合わせて、図11(c)に示すように、接合セルの属性であるメーカー名412の表現形式を決めてもよい。

#### 【0131】

対象要素の属性の表現形式において桁数の違いがあるときに、桁数の大きい方に合わせて表現することは、「受容表現レベル」の一例である。表現形式調整部74は、所定の受容表現レベルをフィルタとして用いることにより、対象要素の属性の表現形式の不整合を調整する。

30

#### 【0132】

図12(a)~(d)も表現レベルでの不整合の調整を説明する図である。シューズメーカーY側のサイズ216は、図12(a)に示すように、inchを単位とした実数値で表現されている。一方、シューズショップX側のサイズ316は、図12(b)に示すように、cmを単位とした実数値で表現されている。

#### 【0133】

ソフトウェアの設計仕様として、cmの単位を採用する場合、表現レベル処理ブロック160の表現形式調整部74は、図12(c)に示すように、接合セルの属性であるサイズ416の表現形式はcmを単位とすることにし、シューズショップX側のサイズ316の属性値を用いる。このとき、表現形式調整部74は、図12(d)に示す変換式516を変換テーブル保持部76に格納しておき、属性値の互換性を保つ。このようにいずれかの単位系に合わせることも受容表現レベルの一例である。

40

#### 【0134】

図13(a)~(d)も表現レベルでの不整合の調整を説明する図である。シューズメーカーY側の色218は、図13(a)に示すように整数値で表現されている。一方、シューズショップX側の色318は、図13(b)に示すように文字列で表現されている。これは、シューズメーカーYでは靴の色を細かく識別するための番号体系を利用するのに対して、シューズショップXでは靴の色を顧客が理解しやすい「紺」「紫」などの名称で記述するからである。この違いは、シューズメーカーY、シューズショップXのそれぞれ

50

のシステムにおける設計者の設計意図や顧客のニーズにより生じたものである。

【 0 1 3 5 】

ソフトウェアの設計仕様として、色の表現形式を文字列とする場合、表現レベル処理ブロック 1 6 0 の表現形式調整部 7 4 は、図 1 3 ( c ) に示すように、接合セルの属性である色 4 1 8 の表現形式を文字列として、シューズショップ X 側の色 3 1 8 の属性値を用いる。このとき、表現形式調整部 7 4 は、図 1 3 ( d ) に示すように、シューズメーカー Y 側の色を識別する整数値とシューズショップ X 側の色を識別する文字列との間の変換規則を示した変換テーブルを作成して、変換テーブル保持部 7 6 に格納し、属性値の互換性を保つ。このように文字列の表現形式に変換することも受容表現レベルの一例である。

【 0 1 3 6 】

上記の例では、ソフトウェアの開発者がシューズショップ X とシューズメーカー Y の間で同値関係にある運動靴を指定したが、シューズショップ X と複数のシューズメーカー Y との間で同値関係を指定してもよい。たとえば、「素材が革である靴」という同値関係をもとに、シューズショップ X の位相空間と、複数のシューズメーカー Y 1 , Y 2 , ... , Y n のそれぞれの位相空間との間に対応する構成要素を抽出し、接着空間を生成し、オブジェクトのデータ設計を行ってもよい。

【 0 1 3 7 】

[ 1 6 . 2 ] プログラムモジュールの設計

オブジェクト集合の一例として、プログラムを考え、本実施の形態に係るソフトウェア開発支援装置 1 0 0 により、複数のプログラムをもとにソフトウェアを開発する手順を説明する。

【 0 1 3 8 】

図 1 4 ( a )、( b ) は 2 つのプログラム X とプログラム Y が統合される様子を説明する図である。これらのプログラムは、いずれも画像処理を行うものであるが、プログラムの開発者が異なるなどの理由で、プログラム構成が異なっているとす。これらのプログラムは、いずれも画像の符号化および復号の機能をもっており、たとえば、電子商取引で売買される商品の画像の符号化および復号を行うことができる。

【 0 1 3 9 】

集合レベル処理ブロック 1 2 0 は、複数のモジュールからなるプログラム X、Y を集合として設計し、位相空間レベル処理ブロック 1 3 0 は、プログラム X、Y を位相空間として設計する。

【 0 1 4 0 】

接着空間レベル処理ブロック 1 4 0 の同値関係設定部 4 4 は、プログラム X、Y を構成するモジュールに対して、「符号化処理機能をもつ」および「復号処理機能をもつ」という 2 つの同値関係を設定する。対応要素抽出部 2 0 は、同値関係設定部 4 4 により設定された 2 つの同値関係をもとにプログラム X、Y 間の対応するモジュールを特定し、接着画像取得部 2 2 は、各同値関係による対応関係を与える接着画像  $f_1$ 、 $f_2$  を取得する。

【 0 1 4 1 】

これにより、プログラム Y の位相空間は、符号化処理機能をもつモジュール  $Y_0$  と、圧縮処理機能をもつモジュール  $Y_1$  と、それ以外の機能をもつモジュールの直和

【 数 5 0 】

$$Y_0 \sqcup Y_1 \sqcup (Y - Y_0 - Y_1)$$

に分割され、一方、プログラム X の位相空間は、符号化処理機能をもつモジュールで  $Y_0$  に対応する  $f(Y_0)$  と、圧縮処理機能をもつモジュールで  $Y_1$  に対応する  $f(Y_1)$  と、それ以外の機能をもつモジュールの直和

【 数 5 1 】

$$f_0(Y_0) \sqcup f_1(Y_1) \sqcup (X - f_0(Y_0) - f_1(Y_1))$$

に分割される。

10

20

30

40

50

## 【 0 1 4 2 】

等化写像取得部 2 4 は、図 1 4 ( a ) に示すプログラム X、Y の 2 つの位相空間の直和を図 1 4 ( b ) に示す接着空間

## 【 数 5 2 】

$$Y \sqcup X / \sim = (Y \sqcup X - Y_0 - Y_1) \sqcup Y_0 \sqcup Y_1$$

に写像する等化写像 g を取得する。

## 【 0 1 4 3 】

図 1 5 ( a )、( b ) は、接着写像により対応づけられた 2 つのモジュール A、B の入出力インタフェースを説明する図である。モジュール A の入力に使われている変数は整数型の a、b の 2 つであり、出力に使われている変数は実数型の c である。一方、モジュール B の入力に使われている変数は実数型の s、t の 2 つであり、出力に使われている変数は実数型の u である。接着空間レベル処理ブロック 1 4 0 において対応づけられた 2 つのモジュール A、B 間で入力変数、出力変数の名前に違いがあるが、セル空間レベル処理ブロック 1 5 0 のセル次元調整部 6 4 は、2 つのモジュール A、B の入力変数、出力変数の名前を対応が取れるように調整する。また、2 つのモジュール A、B 間で入力変数、出力変数の個数に違いがあれば、セル次元調整部 6 4 は、2 つのモジュール A、B の入力変数、出力変数の個数を対応が取れるように調整する。

## 【 0 1 4 4 】

表現レベル処理ブロック 1 6 0 の表現形式調整部 7 4 は、2 つのモジュール A、B の入力変数、出力変数のデータ型に違いがあれば、データ型を調整する。モジュール A の入力変数は整数型であり、モジュール B の入力変数は実数型であるから、たとえば、モジュール A の入力変数を実数型に変換する。また、変数が文字型の場合は、表現形式調整部 7 4 は、文字列の桁数の調整を行う。

## 【 0 1 4 5 】

図 1 6 ~ 図 1 9 において、ソフトウェア開発支援装置 1 0 0 によって、同値関係をもとにモジュールもしくはデータが対応づけられてソフトウェアが開発される手順を説明する。

## 【 0 1 4 6 】

図 1 6 ( a )、( b ) は、プログラムモジュールが新しいバージョンに更新される様子を説明する図である。図 1 6 ( a ) のように、新バージョンのモジュール  $A_{new}$  と旧バージョンのモジュール  $A_{old}$  があるとすると、それぞれ別の開発者が設計したものであってもよい。同値関係によって新バージョンのモジュール  $A_{new}$  が旧バージョンのモジュール  $A_{old}$  に対応づけられる。また、それに伴い、新バージョンのモジュール  $A_{new}$  の入力変数 s、t が、旧バージョンのモジュール  $A_{old}$  の入力変数 a、b にそれぞれ対応づけられ、新バージョンのモジュール  $A_{new}$  の出力変数 u が旧バージョンのモジュール  $A_{old}$  の出力変数 c に対応づけられる。

## 【 0 1 4 7 】

このように同値関係で新旧バージョンのモジュールを対応づけることにより、ソフトウェア開発支援装置 1 0 0 は、図 1 6 ( b ) のように、旧バージョンのモジュール  $A_{old}$  を新バージョンのモジュール  $A_{new}$  に置き換えて、更新することができる。ソフトウェア開発支援装置 1 0 0 は、モジュールの置き換えをホモトピー同値になるように行うため、必要に応じて新バージョンのモジュール  $A_{new}$  を旧バージョンのモジュール  $A_{old}$  に戻すこともでき、ソフトウェアのデバッグを効率良く行うことができる。

## 【 0 1 4 8 】

図 1 7 ( a )、( b ) は、2 つのプログラムモジュールが合成される様子を説明する図である。図 1 7 ( a ) のように、モジュール A は 2 つのサブモジュール  $A_1$ 、 $A_2$  をもち、モジュール B は 2 つのサブモジュール  $B_1$ 、 $B_2$  をもつ。同値関係によってモジュール B のサブモジュール  $B_1$  がモジュール A のサブモジュール  $A_1$  に対応づけられる。また、それに伴い、サブモジュール  $B_1$  の入力変数 s とサブモジュール  $A_1$  の入力変数 a が対応

づけられる。

【0149】

図17(b)のように、同値関係により定まる接着写像によって、モジュールBのサブモジュールB<sub>1</sub>がモジュールAのサブモジュールA<sub>1</sub>と同一視されることにより、2つのモジュールA、Bが合成され、1つのプログラムが生成される。このとき、接着写像で対応づけられたサブモジュールB<sub>1</sub>とサブモジュールA<sub>1</sub>の入出力変数のデータ型や桁数などが調整される。ソフトウェア開発支援装置100は、モジュールの合成をホモトピー同値になるように行うため、必要に応じて合成されたプログラムから元のモジュールA、Bを分離して取り出すこともできる。

【0150】

このように、ソフトウェア開発支援装置100は、同値関係をもとにして複数のモジュールを合成したり、合成されたプログラムからモジュールを取り出したりすることができ、ソフトウェアの部品化を進め、再利用性を高めることができる。

【0151】

図18(a)、(b)は、2つのプログラムモジュール間でデータが共有される様子を説明する図である。図18(a)のように、2つの変数a、bを入力して変数cを出力するモジュールAと、2つの変数s、tを入力して変数uを出力するモジュールBとがあり、モジュールBの入力変数sとモジュールAの入力変数aが同値関係によって対応づけられ、それ以外の変数の間には対応関係がないとする。

【0152】

図18(b)のように、同値関係により定まる接着写像によって、モジュールBの入力変数sがモジュールAの入力変数aと同一視されることにより、2つのモジュールA、B間で入力変数の1つが共有される。このように、ソフトウェア開発支援装置100は、プログラムモジュール間で同値類として対応づけられるデータを共有し、データの利用率を高めることができる。

【0153】

図19(a)、(b)は、2つのプログラムモジュールが結合される様子を説明する図である。図19(a)のように、2つの変数a、bを入力して変数cを出力するモジュールAと、2つの変数s、tを入力して変数uを出力するモジュールBとがあり、モジュールBの出力変数uとモジュールAの入力変数aが同値関係によって対応づけられ、それ以外の変数の間には対応関係がないとする。

【0154】

図19(b)のように、同値関係により定まる接着写像によって、モジュールBの出力変数sがモジュールAの入力変数aと同一視されることにより、モジュールBとモジュールAがこの順で結合され、1つのプログラムが生成される。このとき、モジュールBの出力変数とモジュールAの入力変数の間でデータ型や桁数などが調整される。このプログラムの結合は、ホモトピー同値になるように行われるため、必要に応じて結合を解除して、モジュールAとモジュールBを分離することもできる。このように、ソフトウェア開発支援装置100は、同値関係をもとにして複数のプログラムモジュールを結合したり、分離したりすることができ、プログラムモジュールの利用率を高めることができる。

【0155】

このように、ソフトウェア開発支援装置100によれば、同値関係にもとづいてソフトウェア部品の合成、置換、連結などの開発工程を線形操作で行うことができるため、ソフトウェア部品の個数が増えても、開発工数爆発が起きることなく、ソフトウェアを効率良く開発することができる。

【0156】

上記の具体例では、プログラムモジュールの仕様を特徴づける属性として、入出力インタフェースを例に挙げ、対応するプログラムモジュール間で入出力インタフェースの次元や表現形式の不整合を調整することを説明したが、これは属性の一例に過ぎない。プログラムモジュールの仕様を特徴づける他の属性の例として、そのプログラムモジュールが参

10

20

30

40

50

照する変数を考えてもよい。その場合、対応するプログラムモジュール間で参照変数の次元や表現形式の不整合が調整される。また、プログラムモジュールが利用する関数を属性として考えてもよい。その場合、対応するプログラムモジュール間で関数の次元、すなわち関数名や関数の個数などの不整合が調整される。さらに、その関数の引数や戻り値の次元や表現形式の不整合が調整されてもよい。

#### 【0157】

ソフトウェア開発支援装置100のビューレベル処理ブロック170は、ソフトウェア開発者毎に異なるビューを設計してもよい。たとえば、上級のソフトウェア開発者にはプログラムモジュールの内部変数の設計変更ができるように、プログラムモジュール内部のデータ操作が可能なビューを設定する。一方、初級のソフトウェア開発者にはプログラムモジュールの外部インタフェイスの設計変更だけを許すように、プログラムモジュールの内部のデータ操作が不可能なビューを設定する。このように、開発者のプログラミングの経験や技術などのレベルに応じてビューを異ならせてもよい。また、ソフトウェアを複数の開発者が共同開発する場合、各開発者が担当するプログラムエリアに限って設計変更ができるように、開発者毎にビューを異ならせてもよい。

10

#### 【0158】

以上、実施の形態を説明した。実施の形態は例示であり、さまざまな変形例が可能であり、そうした変形例も本発明に含まれることは当業者に理解されるところである。

#### 【図面の簡単な説明】

#### 【0159】

20

【図1】2つの互いに素である位相空間から接着空間が生成される様子を説明する図である。

【図2】顧客と取引会社の位相空間が接着される様子を説明する図である。

【図3】座席と支持部の位相空間が接着される様子を説明する図である。

【図4】実施の形態に係るソフトウェア開発支援装置の構成図である。

【図5】図4の接着空間レベル処理ブロックの構成図である。

【図6】図4のセル空間レベル処理ブロックの構成図である。

【図7】図4の表現レベル処理ブロックの構成図である。

【図8】図5の接着空間レベル処理ブロックにより、2つの位相空間が接着されて接着空間が生成される様子を説明する図である。

30

【図9】オブジェクトのセル構造を説明する図である。

【図10】オブジェクトのセル構造を説明する図である。

【図11】図7の表現レベル処理ブロックにより、表現レベルでの不整合が調整される様子を説明する図である。

【図12】図7の表現レベル処理ブロックにより、表現レベルでの不整合が調整される様子を説明する図である。

【図13】図7の表現レベル処理ブロックにより、表現レベルでの不整合が調整される様子を説明する図である。

【図14】2つのプログラムが統合される様子を説明する図である。

【図15】接着写像により対応づけられた2つのモジュールの入出力インタフェイスを説明する図である。

40

【図16】プログラムモジュールが新しいバージョンに更新される様子を説明する図である。

【図17】2つのプログラムモジュールが合成される様子を説明する図である。

【図18】2つのプログラムモジュール間でデータが共有される様子を説明する図である。

【図19】2つのプログラムモジュールが結合される様子を説明する図である。

#### 【符号の説明】

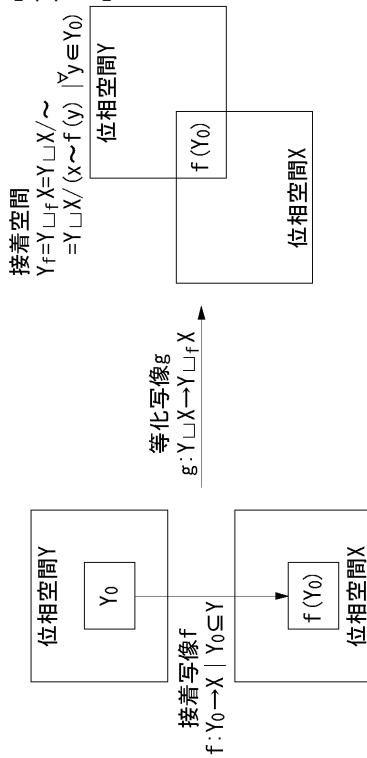
#### 【0160】

20 対応要素抽出部、 22 接着写像取得部、 24 等化写像取得部、 26

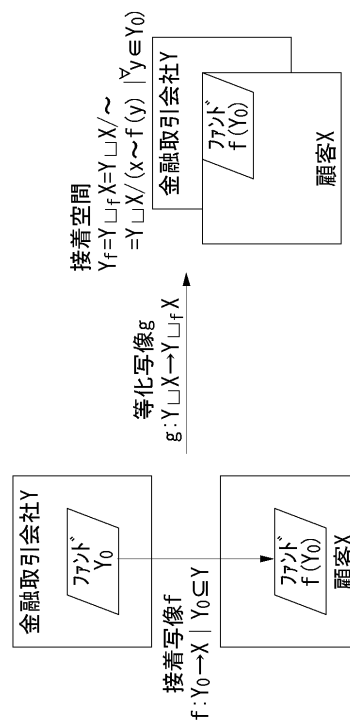
50

手順取得部、 28 手順記録部、 44 同値関係設定部、 52 同値関係保持部、  
 62 セル接合部、 64 セル次元調整部、 66 対応テーブル保持部、 72  
 表現生成部、 74 表現形式調整部、 76 変換テーブル保持部、 100 ソフト  
 ウェア開発支援装置、 110 ホモトピーレベル処理ブロック、 120 集合レベル  
 処理ブロック、 130 位相空間レベル処理ブロック、 140 接着空間レベル処理  
 ブロック、 150 セル空間レベル処理ブロック、 160 表現レベル処理ブロック  
 、 170 ビューレベル処理ブロック、 190 オブジェクト集合記憶部。

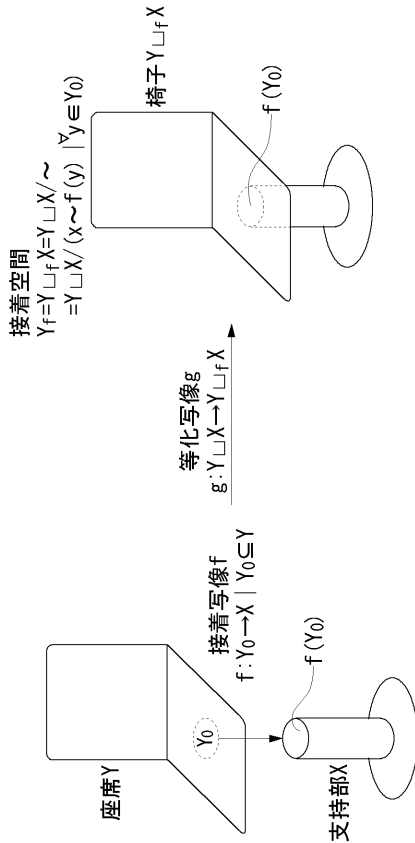
【図 1】



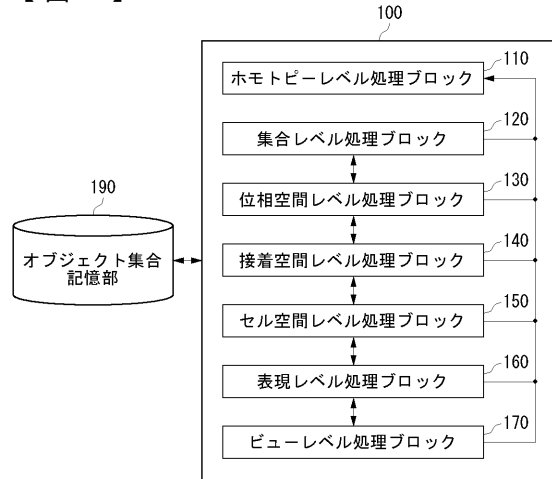
【図 2】



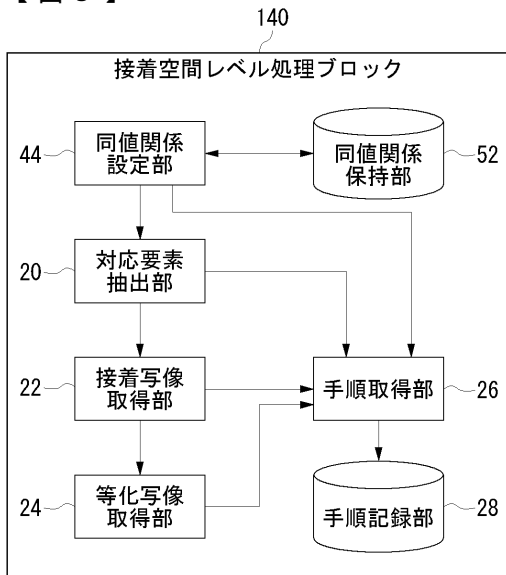
【図 3】



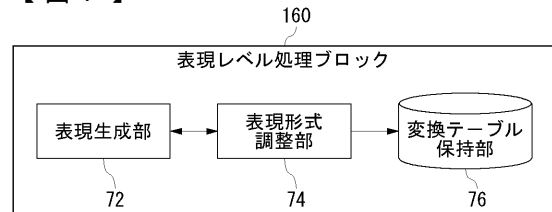
【図 4】



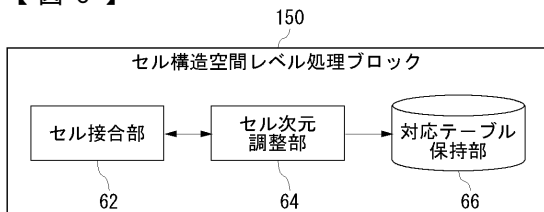
【図 5】



【図 7】

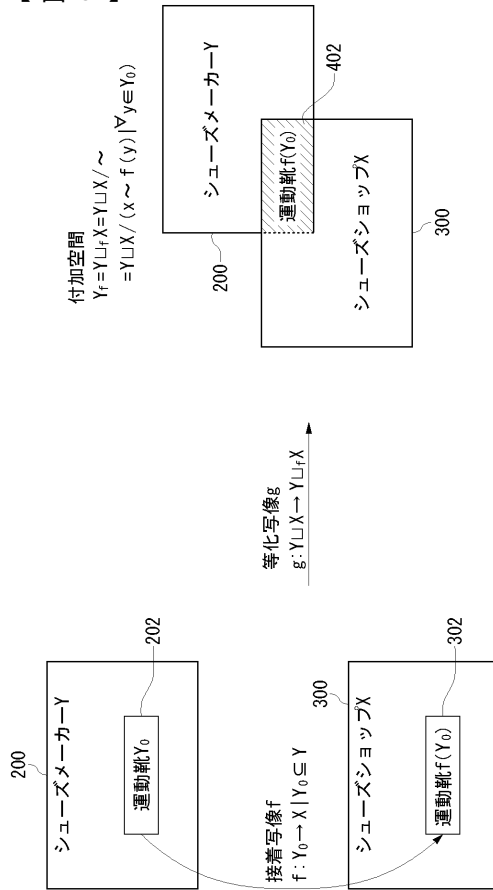


【図 6】

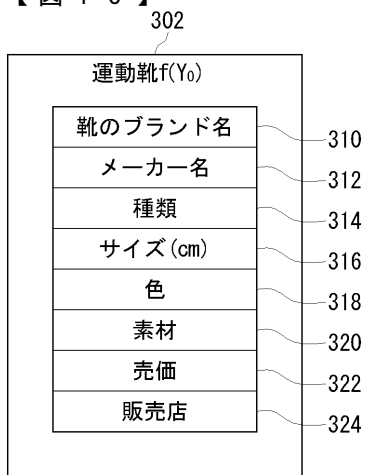




【図 8】



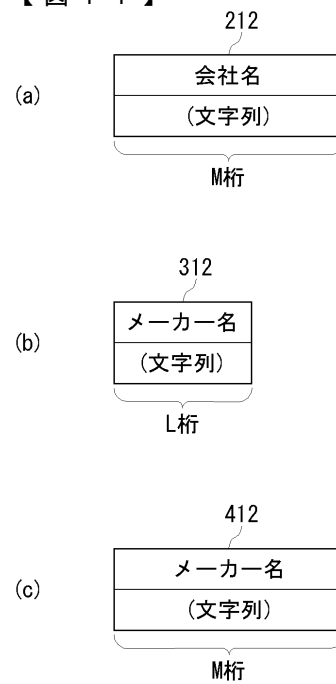
【図 10】



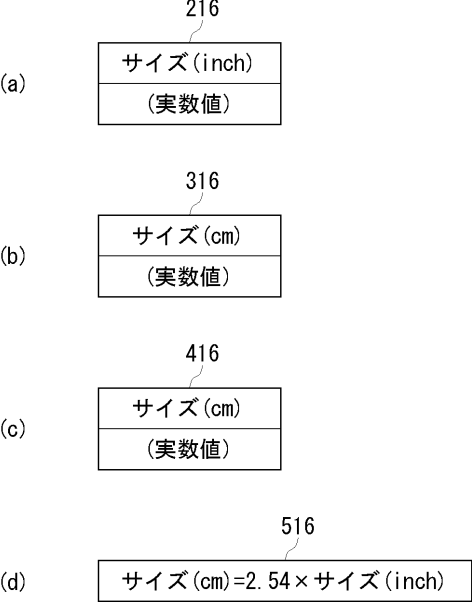
【図 9】



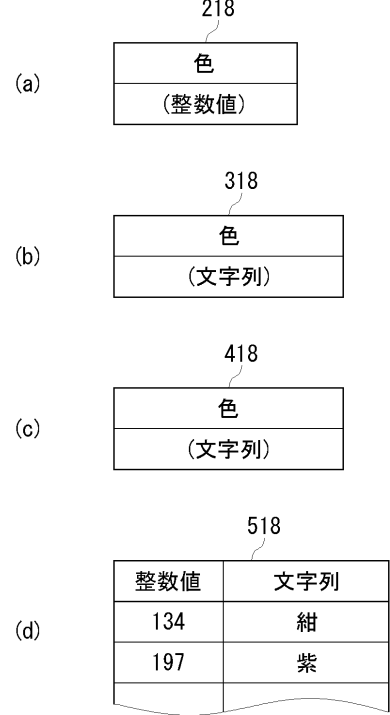
【図 11】



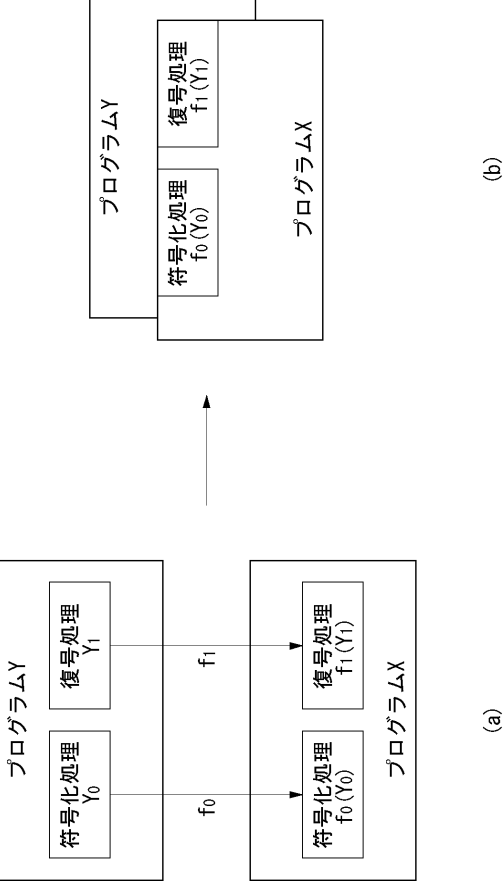
【 図 1 2 】



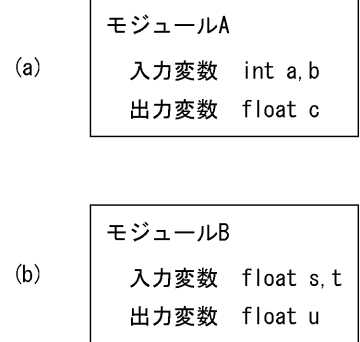
【 図 1 3 】



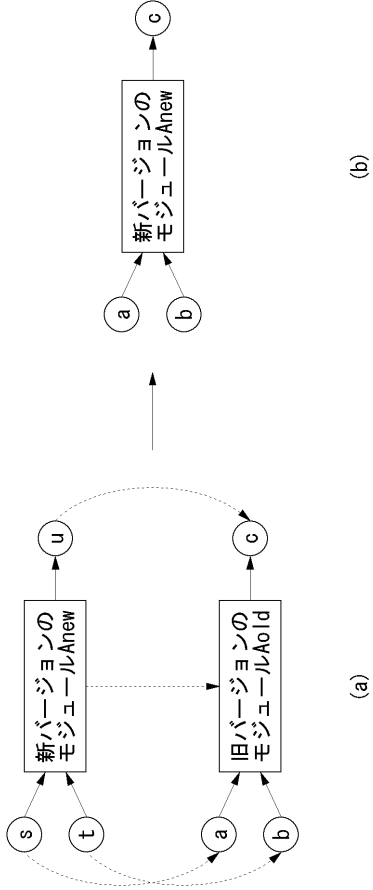
【 図 1 4 】



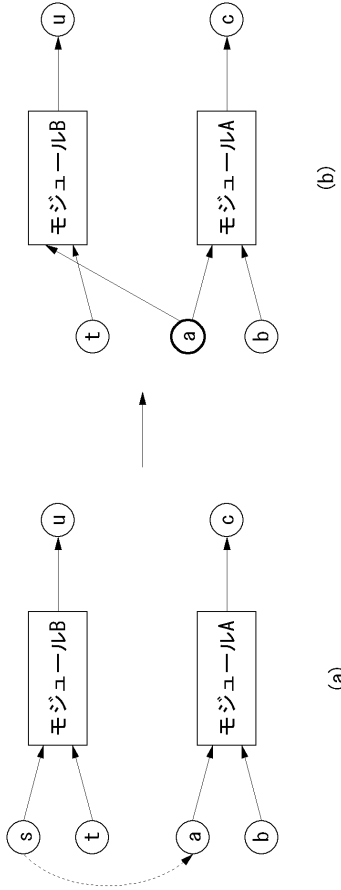
【 図 1 5 】



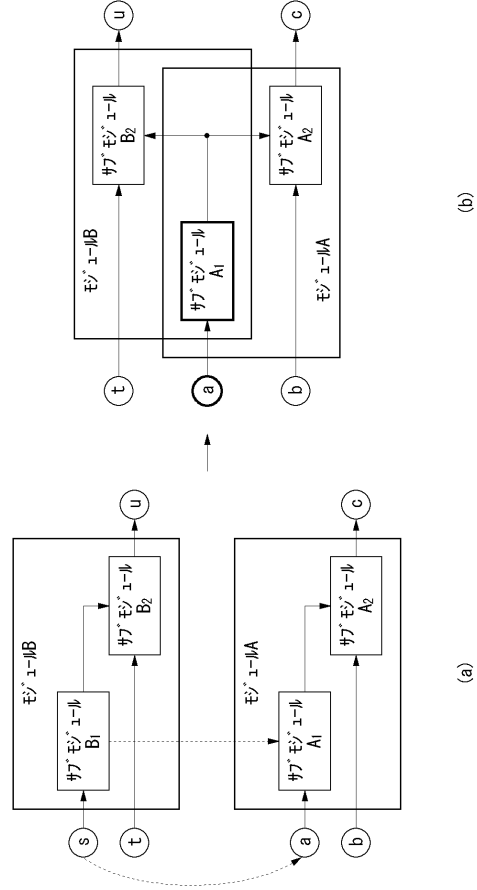
【図 16】



【図 18】



【図 17】



【図 19】

