

部分理解のリスクを低減させるためのプロセス改善の提案

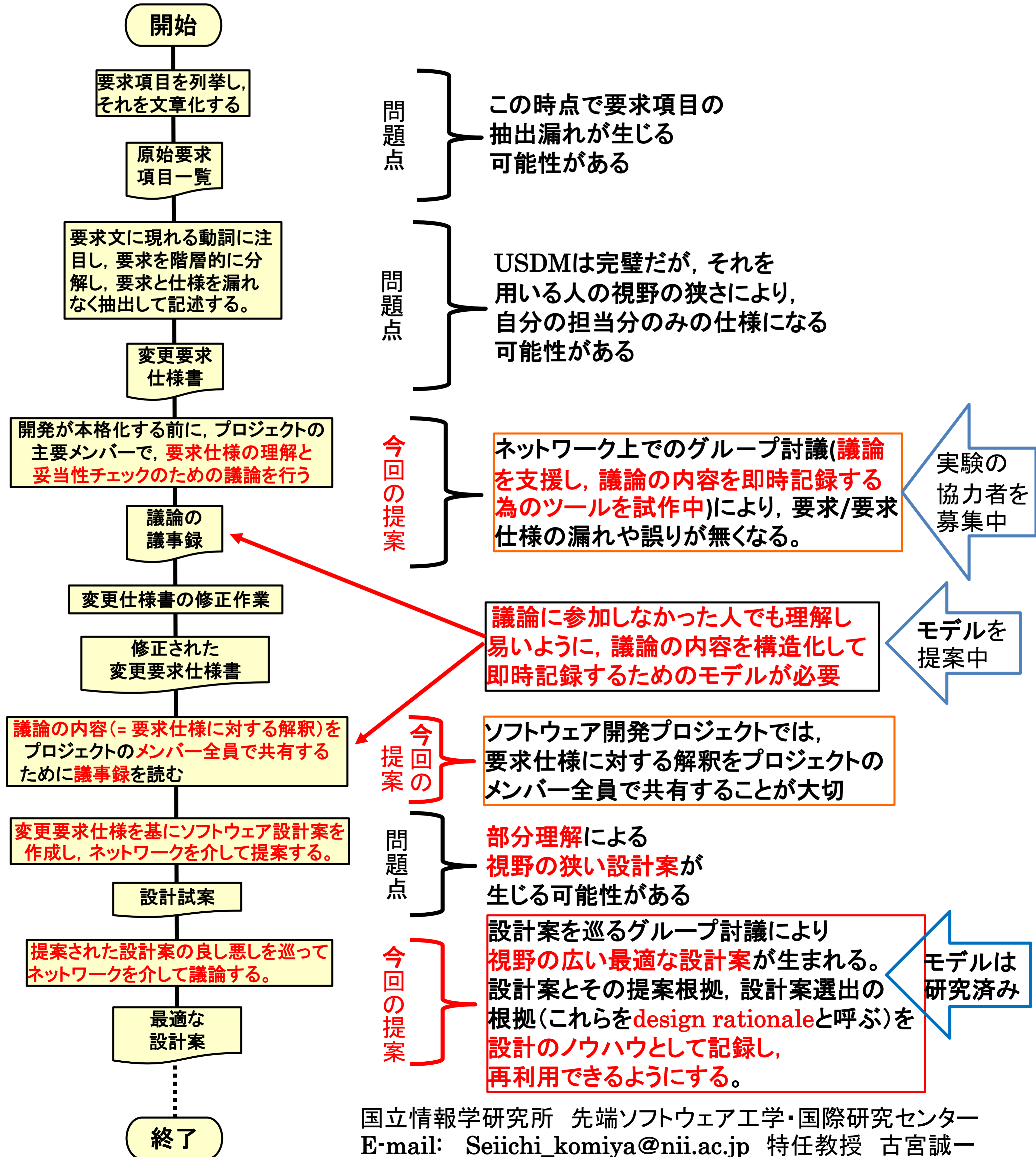
提案の背景: 派生開発では、短納期に起因する検討不足が原因で、部分理解という状況が生じ易い。部分理解が生じ易いのは、主に下記の2箇所だと考えられる。

(1) 要求を抽出し、要求仕様を定義する過程 ← 担当部分を個人で作業するから

(2) 要求仕様を満足する設計案を検討する過程 ← 担当部分を個人で作業するから

提案の骨子: 中間成果物(USDMや設計案)を生成する過程で、中間成果物の妥当性を巡って、関係者がネットワークを介して議論することにより、部分理解を解消する。
(議論の内容を第三者が理解し易い形式で即時記録し、後で再利用できるようにする。)

派生開発の変更プロセスにおける問題点とその対策(プロセス改善)



『要求仕様の内容』の理解と妥当性チェックのための議論の必要性 ～その議論の内容を理解し易い形で即時記録するためのモデルの提案～

1. 研究の背景と目的

(研究の背景: 現状の問題点)

ソフトウェアの要求仕様に含まれる曖昧さやその要求内容に対する開発の担当者らの理解不足などによる設計不良が、プロジェクトマネジメント上の大きな問題となっている。

(ステップ1)

開発が本格化する前に、プロジェクトの主要メンバーで要求仕様の理解と妥当性確認のための議論を行う必要がある

支援

議論の内容を理解し易いように構造化して即時記録するためのモデルが必要

研究の対象(目的)

(ステップ2)

このときの議論の内容(=要求仕様に対する解釈)をプロジェクトのメンバー全員で共有する必要がある

対策(提案)

2. 開発の対象となるシステムの分類

開発の対象となるシステムは、要求仕様の決定権が誰にあるかで、次の2種類に分類できる。

(1) 要求仕様の決定権が顧客側にあるシステム

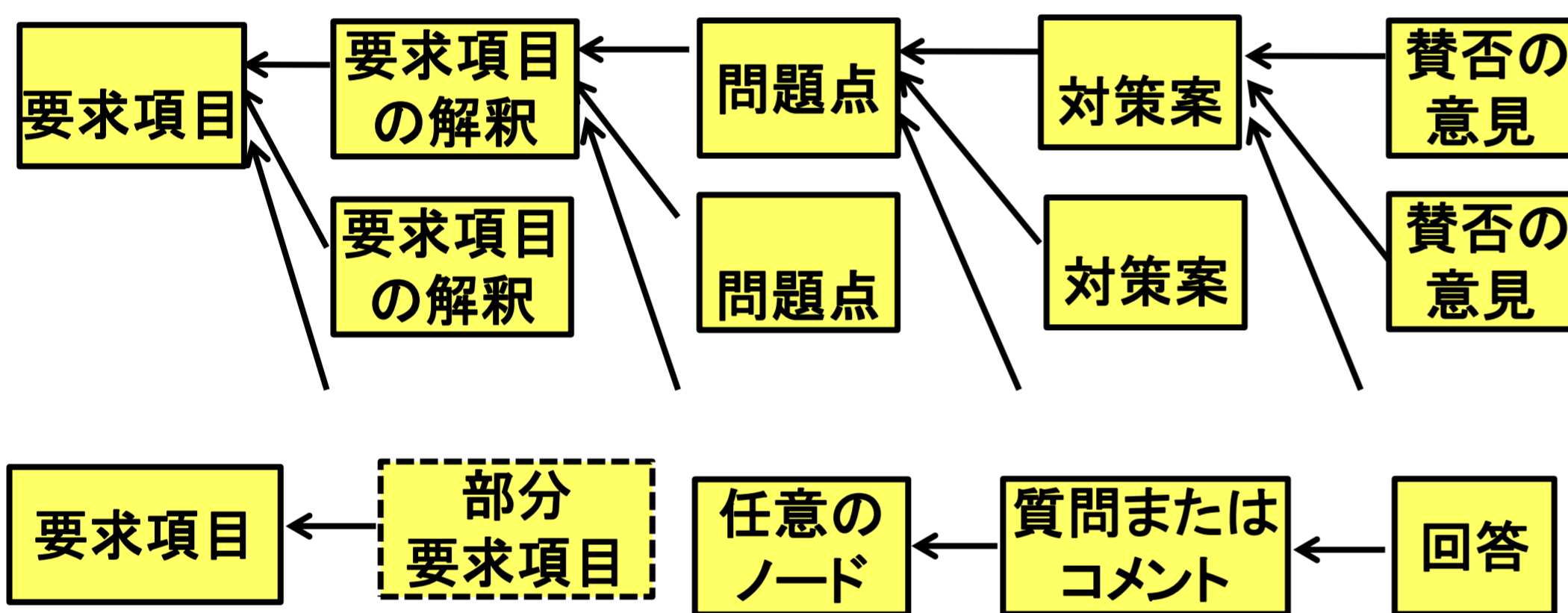
(顧客からの受注によって開発するシステム)

(2) 要求仕様の決定権が開発者側にあるシステム

(不特定顧客を対象とするシステム)

ここでは、(1)に分類されるシステムを対象にして、要求仕様の内容の理解とその妥当性をチェックするために議論する場合を想定する。

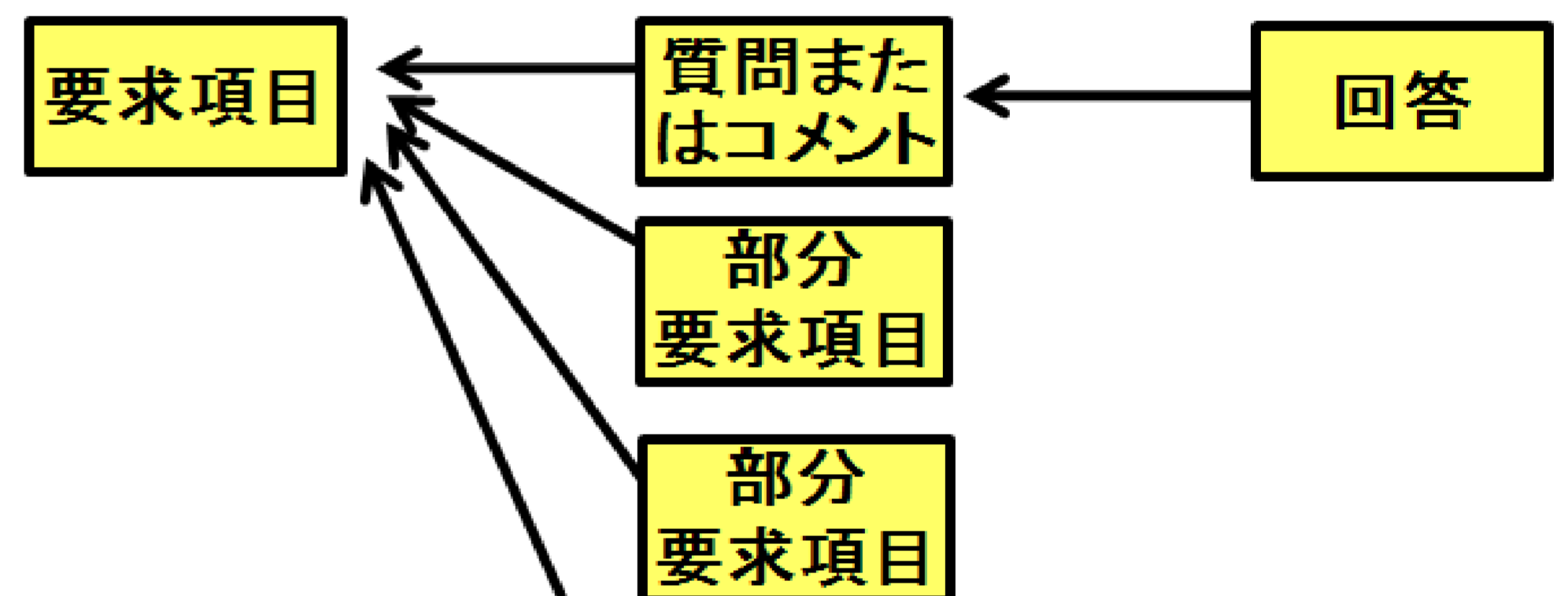
3. 要求仕様の理解と妥当性確認のための議論において議論の内容を構造化して即時記録するためのモデル



(注意)

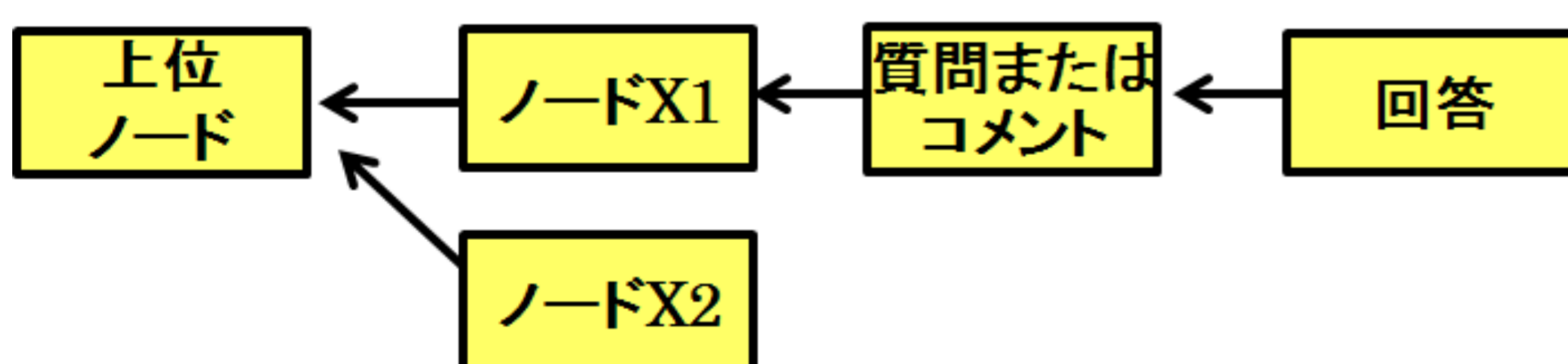
『部分要求項目』というノードは、『要求項目』または『部分要求項目』というノードの要求内容が複数の要求項目に分解できる場合にだけを使用する。部分要求項目を使用する場合の記述場所は、『要求項目』の下位ノードで、かつ、『問題点』の上位ノードとして使用する。

4. 『要求項目』というノードを『部分要求項目』へ分解する過程



(注意) 『質問またはコメント』というノードが無くても、幾つかの『部分要求項目』を『要求項目』や『部分要求項目』の下位ノードとして追加することができる。

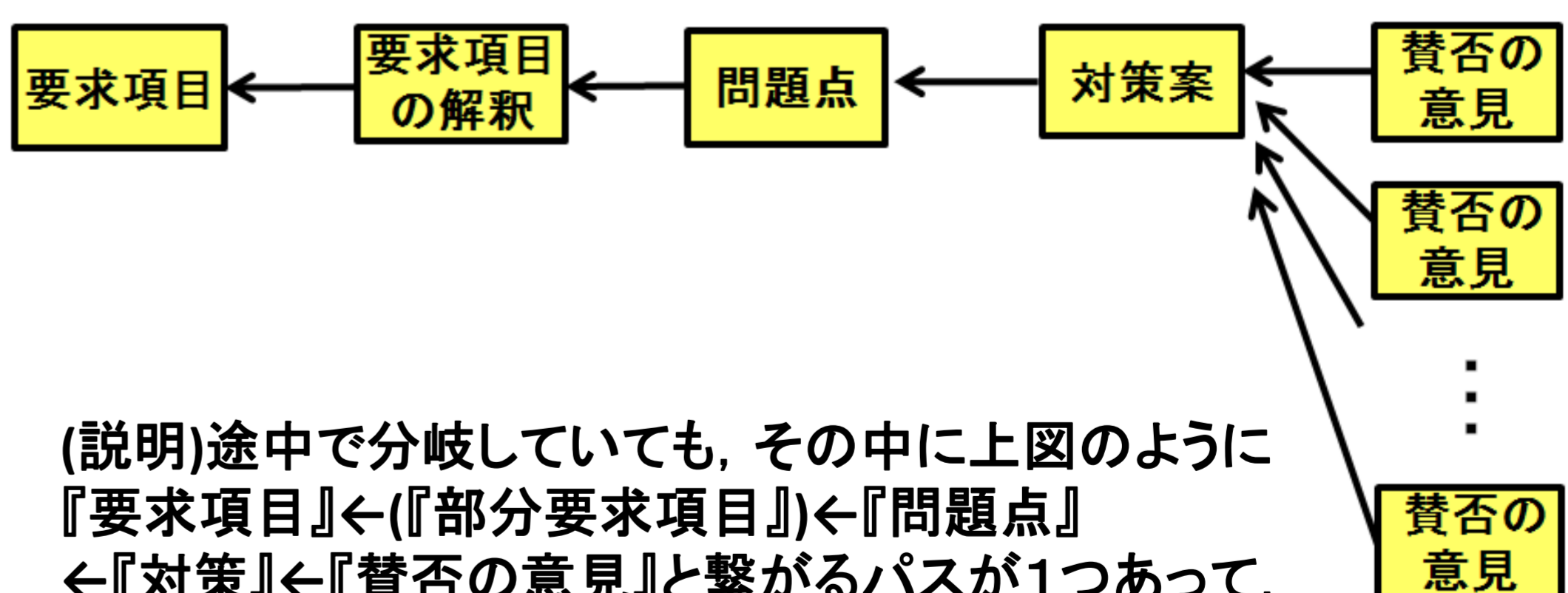
記述済みノードの記述内容を修正する方法



(説明) 上図は、『質問またはコメント』の記述が契機となって『ノードX1』の記述内容が『ノードX2』に書き改められる過程を示している。

『ノードX1』そのものを修正することはできないので、『ノードX1』の修正版として『ノードX2』を作成し、その中の修正対象というフィールドにノードX1と記入することにより、『ノードX2』が、『ノードX1』の修正版であることを明記する。

議論の参加者全員から賛成意見が得られていることの確認



(説明)途中で分岐していても、その中に上図のように『要求項目』←(『部分要求項目』)←『問題点』←『対策』←『賛否の意見』と繋がるパスが1つあって、かつ、**全ての立場の人からの賛成意見が得られていれば、上記の要求項目に対する解釈に対して、全員の合意が得られた、何らかの結論が出ていると見なしてよい。**

『設計案と提案理由』を巡る賛否の意見とその根拠に関する議論の必要性 ～その議論の内容を理解し易い形で即時記録するためのモデルの提案～

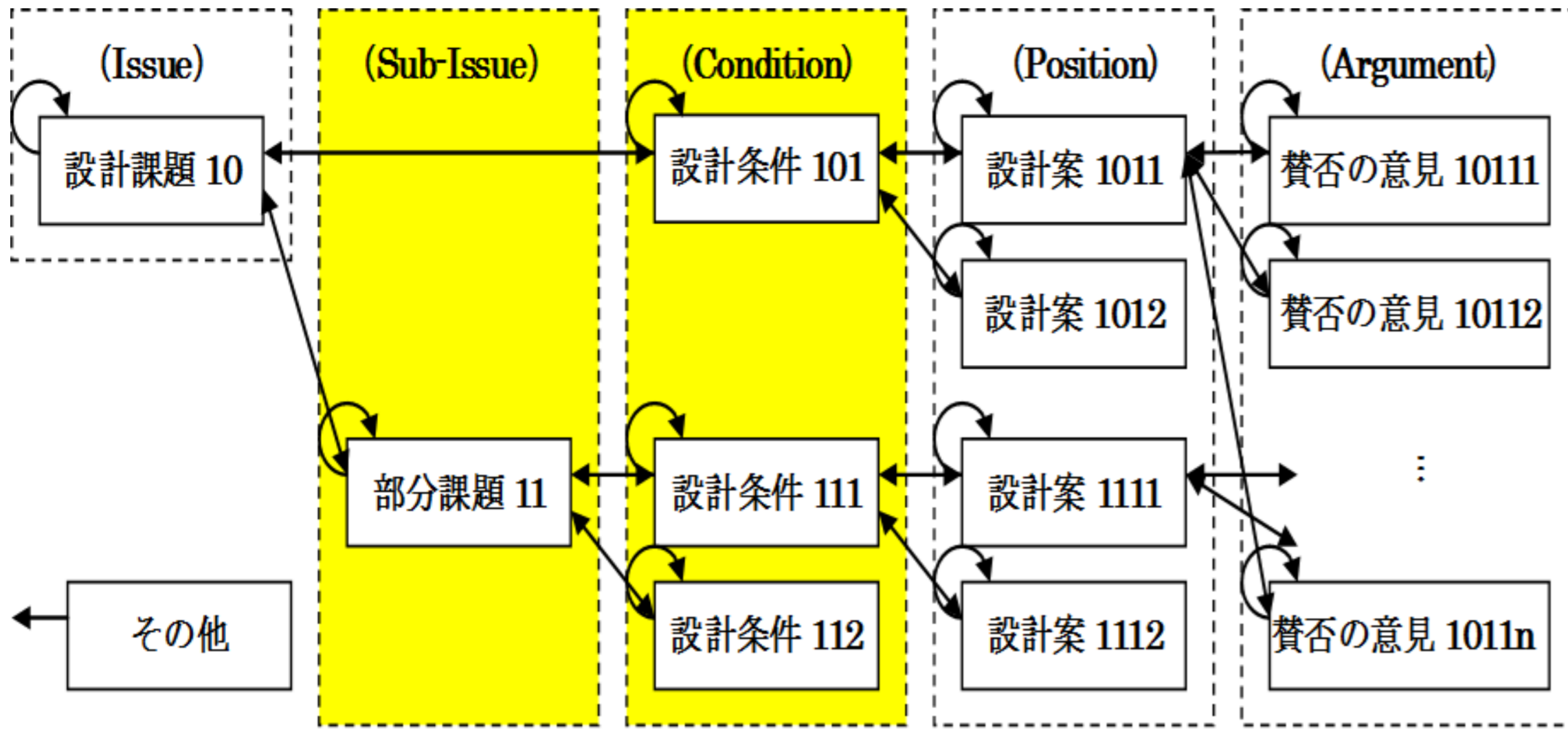


図 改良型IBISモデル [Komiya (1993)]

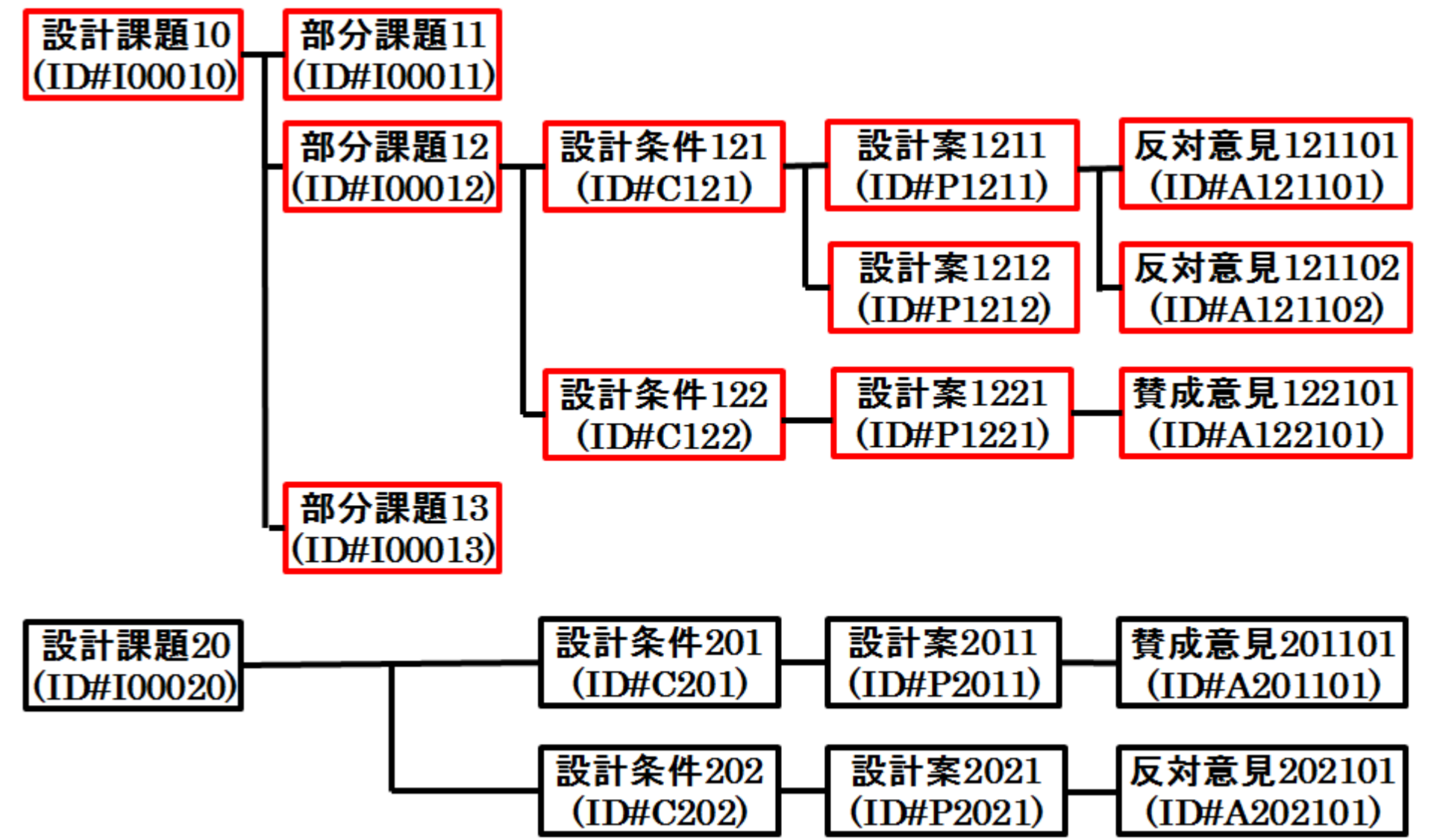


図 記述例における議論の構造(改良型IBISによる表現)

改良型IBISによる記述例

設計課題10 (ID#I00010)

過去の設計事例を検索して
再利用する処理を設計せよ。

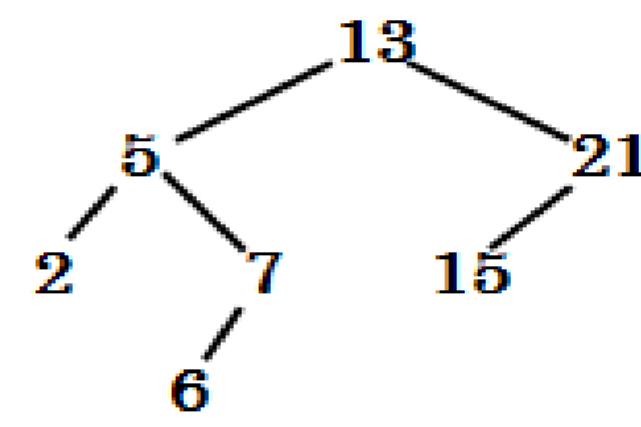
A氏の考え方:

設計事例 = 設計課題の記述部分
+ 設計条件の記述部分 } この部分に
+ 設計案の記述部分 } キーワードを与え
← この部分を取り出す

検索対象	与えるキーワード		処置
	設計課題	設計条件	
設計課題 条設計件	一致	一致	両方のキーワードが一致した 設計案を検索して再利用する
設計課題 設計条件	一致	不一致	設計課題が一致した設計条 件の近い設計案を再利用

137

設計条件121 (ID#C121) の下での設計案1211 (ID#P1211)
(設計条件の集合を包含関係に基づく全順序集合とするとき)



凡例 $y \begin{matrix} x \\ z \end{matrix}$ のとき $y < x < z$ を
満たす木

図 二分探索木
(全順序木の一種)の例

全順序集合の特徴:

全順序集合に属する任意の要素
同士の間では必ず順序(ここでは
包含関係)が決定する。

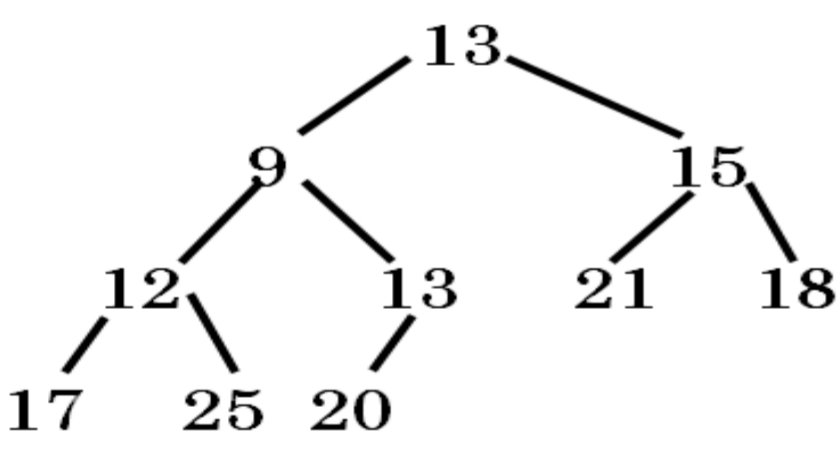
従って、全順序木を辿れば、求める
設計案またはその代替案が、どの
設計案であるか(具体的には、その
設計案の番号)が求まる。

この設計案の欠点:

1. 全順序木が複数個必要になるの
で、検索時間が大きくなる。
2. 求める設計案と代替案との差分を
求めることが困難。

140

設計条件122 (ID#C122) の下での設計案1221 (ID#P1221)
(設計条件の集合を包含関係に基づく半順序集合とするとき)



凡例 $y \begin{matrix} x \\ z \end{matrix}$ のとき
 $x < y, x < z$ を
満たす木

図 半順序木
(二分木の場合)の例

半順序集合の特徴:

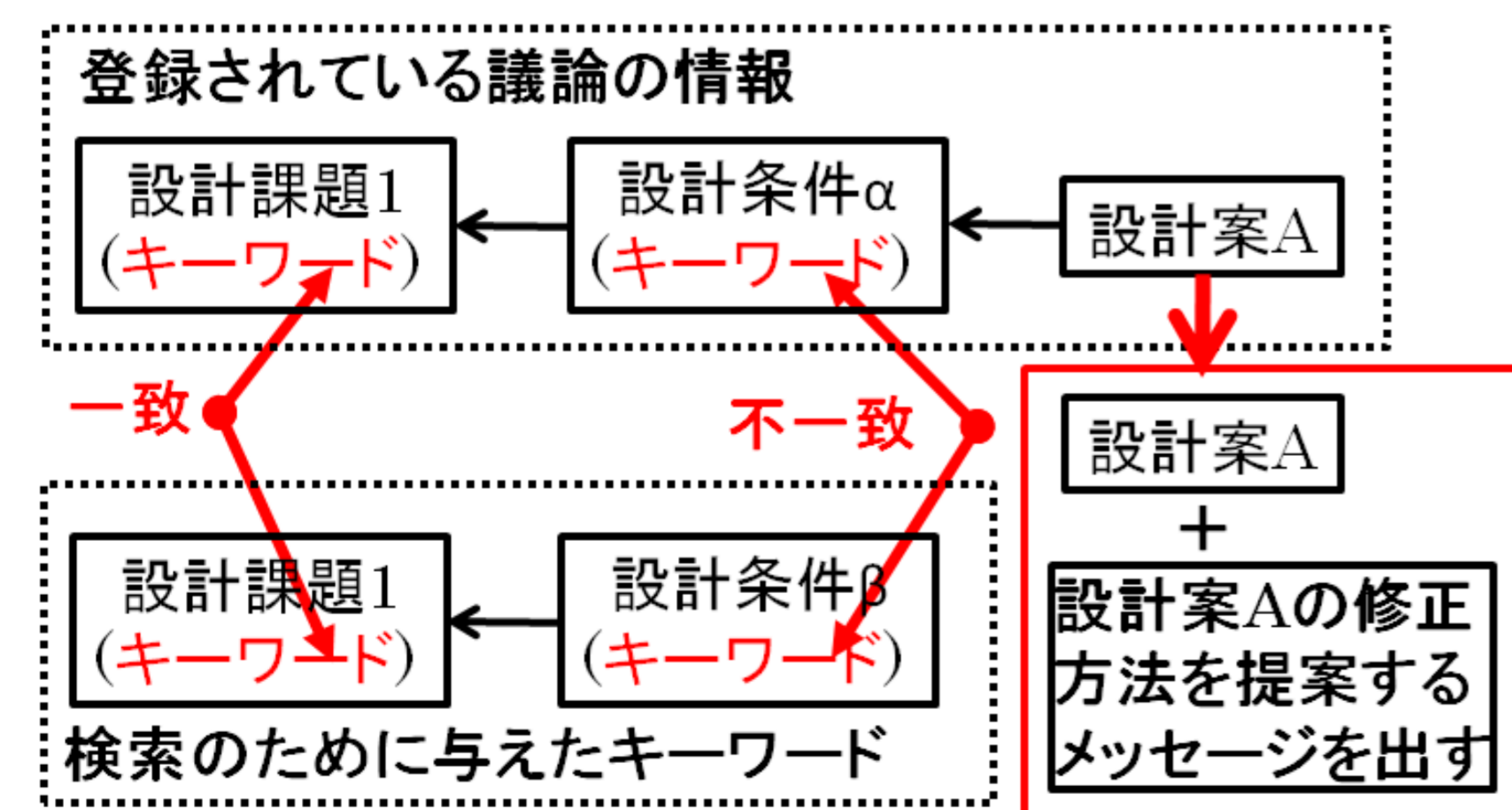
半順序集合に属する要素同士の間では、順序に関して、反射律、反対称律、推移律の3法則が成立するので、半順序木を辿れば、求める設計案またはその代替案が、どの設計案であるか(具体的には、その設計案の番号)が求まる。

この設計案の長所:

1. 包含関係に基づく半順序木の階層は、IS-A階層に対応するので、
求める設計案と代替案との差分を
求めることは容易である。

142

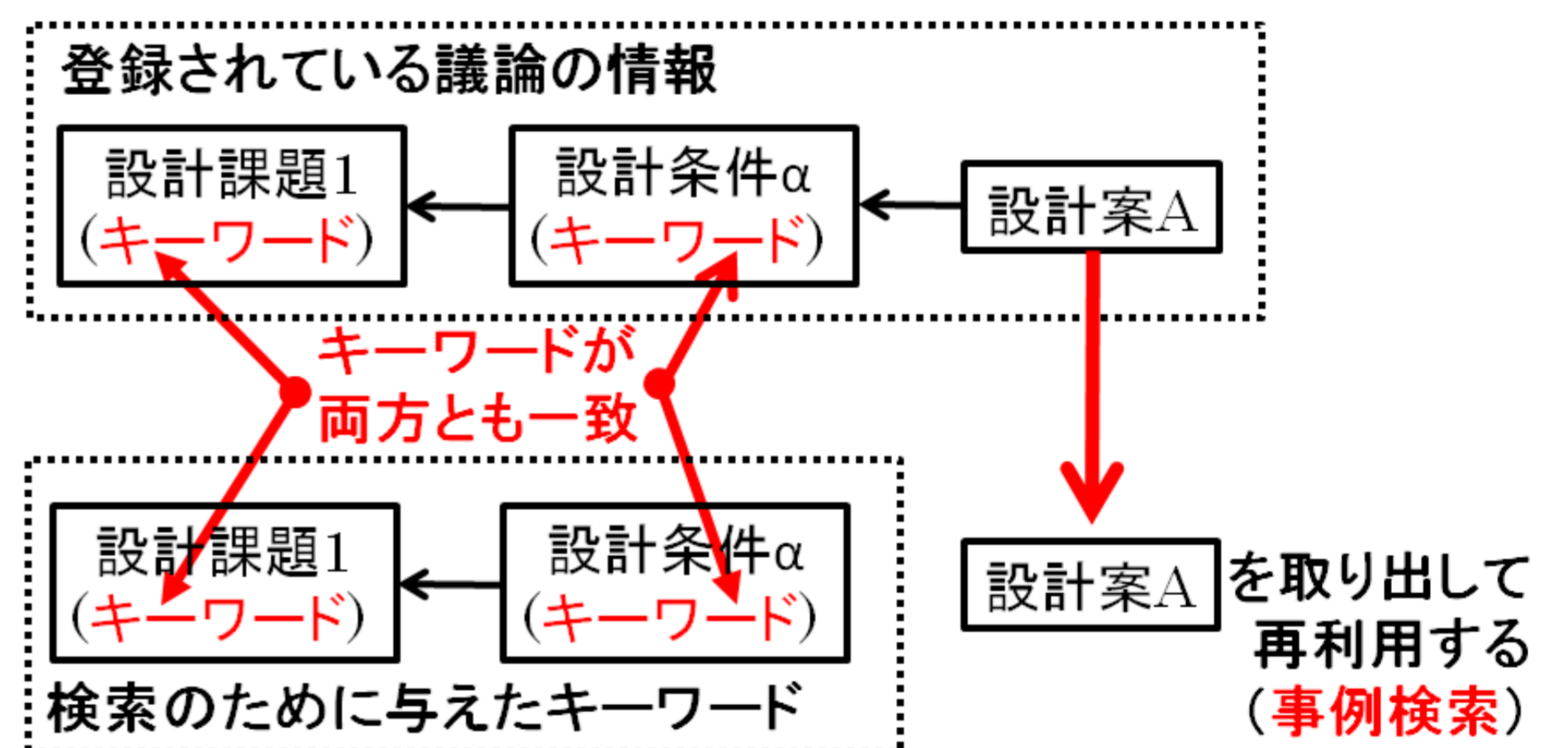
設計課題10 (ID#I00010) に対して最終的に採択された設計案(2/2)



設計課題1で設計条件がβであるような設計案が見つからなかったとき、同一設計課題の下で、設計条件がβと類似するαであるような設計案Aを検索・提示するとともに、Aに対する修正方法をαとβの差分から推論してメッセージの形で提案する(事例ベース推論)。

145

設計課題10 (ID#I00010) に対して最終的に採択された設計案(1/2)



設計課題と設計条件のそれぞれに対して与えたキーワードの両方が、登録されている議論の情報のキーワードとそれぞれ一致した場合には、その下にある設計案Aを取り出す。

Design Rationale (設計案の根拠)
について議論し記録することの効用

1. 狭い視野から設計案を検討することによる思い込みや勘違いが原因で起こる設計不良を低減させることができる。
2. その設計案がどのような案になるかで影響を受ける人達の間でレビューをしながら設計していることになるので、設計上のインタフェース不良を低減させることができる。
3. チームの中に経験やスキルの乏しい人がいても、そのチームに属する人達全員で支援することになるので、その人への指導が特定の個人の負担にはならない。
4. 設計案の良し悪しとその根拠について議論し、記録することは、設計のノウハウを残すことになる。
5. 設計案の良し悪しとその根拠について議論し、記録する過程は、設計担当者の脳裏だけにある検討の深さを吐き出すことになるので、設計作業の進捗を把握できることを意味する。

147