

# 機能量に基づく機能表現 (第三報)

## －機能量による機能間交渉の実現－

○三田工業 下村芳樹, 谷川貞夫, 奈良先端大 武田英明, 東大工 大道憲哉, 梅田靖, 富山哲男

本報では、機能空間上の位相として対象の機能の発現状態を表現する機能モードに、機能量による量的な表現を導入することにより、より詳細に対象の機能の発現状態を表現する方法を提案する。またその結果に基づいて、機能間交渉と呼ばれる機能的な操作を実現する方法について考察を行なう。

### 1. 緒言

前報では、機能量の基本的な枠組について報告し、副詞的機能を機能量で近似することによって、機能量の設計評価への適用可能性を提示した。本報では機能を、目的を直接的に表現する機能本体と、それを修飾することで機能の表現力を高める機能修飾子に分類することで、より設計者の直観に沿った対象の機能の表現手法を提案する。またこの枠組を用いて、設計における機能展開の在り方と要求仕様の明確化の過程の関係について考察を行ない、実際の設計事例の分析を行なう。さらにこの結果から、機能間交渉と呼ばれる機能的な操作を実現する方法についても考察する。

### 2. 機能と対象表現

設計対象の機能をモデル化すると、一般にその表現は全体機能と部分機能という機能の階層構造を成す<sup>1)</sup>。この時全体機能から部分機能への展開は、設計者によるタスク分割により発生するものと物理的因果関係に基づく補間によるものの2つが考えられる<sup>2)</sup>。

しかし、設計対象をモデル化することの目的に、設計者の意図の保存を考えた場合、これら2つの機能間の関係のみを用いて対象を記述するだけでは不十分であり、機能に挙動を対応付ける際の選択基準<sup>3)</sup>、設計時に用いる機能の表現単位の変更という設計の過程をも同時に表現する必要がある。つまり、機能を修飾的な表現を含んだ形で考えるならば、全ての機能間の関係はタスク分割的な関係と物理的な因果関係によって説明可能な関係に還元でき、このとき、修飾的表現を含まない機能もオーバースペックであるが発現されるという関係になる。しかし、本来完全な機能知識というものが存在しない以上、上でいう「修飾的な表現を含む」が常に可能であるとは限らず、それが実際の設計に多く見受けられる本質的な振舞いであると考えられる。この問題は、設計における要求仕様の不完全性、変更可能性とも関連する。つまり、設計における仕様は外部から与えられるものではあるが完全ではなく、言い替えれば、機能の表現単位で説明される仕様明確化の過程は、曖昧である仕様を明確化していく過程も設計に含まれるという仮説そのものであるといえ

る。そして機能を仕様記述言語であるとする本研究の立場では、当然仕様の明確化の過程を機能によって表現する必要があり、この仕様の明確化の過程を設計者の意図に含まれるものとして扱う。

### 3. 機能修飾子

「美しく」や「同時に」は、共に機能的表現に見受けられる修飾的な表現である。しかし「同時に」という表現が用いられる経緯を考えると、この表現はいわば機能間(あるいは挙動間)の関係を制約するものである。つまりこの種の制約的な表現は、機能をより下位の機能に展開する時に発生し、機能の部分構造を制約するものであると考えられる。本研究では、このような部分機能間の関係に対する制約を機能関係(FR:Functional Relation)と呼ぶ。

一方、「美しく」といった表現は、主たる機能を修飾することで機能の仕様に対する記述力を高めることを目的とするものであるが、これによる制約は主たる機能にどの挙動を対応付けるかの選択基準、あるいは主たる機能の実現挙動との対応付けを含めた対象の物理的な基本構造が決定した後、状態(挙動)に対する従属的な制約として具体化するものである。そこで本研究では、このような制約的表現を機能修飾子(FM:Functional Modifier)と呼ぶ。つまり本研究においては、設計者は修飾子を用いることで、前節に述べた設計時に用いる機能、及び機能の表現単位を制御し、最終的に機能と修飾子を共に具体化することで要求仕様を明確化するものと仮定する。

#### 3.1. 機能修飾子による機能展開

ここでは機能の展開方法として、(1)タスク分割による機能展開、(2)因果関係による機能展開、(3)修飾子による機能展開の三つを仮定し、これらの機能展開の結果である機能間の関係を各々、**decomposed-into** 関係、**caused-by** 関係、**reinforced-with** 関係と定義する。この機能展開の様子を図1に示す。さらに以上の仮定に基づき、武田らの設計実験<sup>4)</sup>における機能展開の過程を表現したものを図2に示す。

図1において、まず機能Aは部分機能B、Cにタスク展開される。そして機能Cの実現挙動として挙

