

## Analysis

### 1 目次

### 2 はじめに

解析によって、正しく完璧で一貫した明白なシステムのモデルが得られる。

開発者は要求獲得および、より詳細な境界条件、例外事例の調査の間に作られた要求仕様を形式化する。

たとえ誤りや曖昧な部分が見つかったとしても、開発者は要求仕様を正しく、明瞭に確認する。

要求仕様を変更しなければならない場合や、付加情報を収集しなければならない場合に、顧客とユーザはたいていこのアクティビティに関係付けられる

オブジェクト指向における解析では、開発者はモデルを構築する。アプリケーションの概念を説明するモデル

例えば、時計の解析モデルは時計がどのように現在時刻を表すか説明する。:すなわち、時計はうるう年を知っているのか？曜日は？月の満ち欠けは？

解析モデルはその時どのようにしてアクターやシステムが相互作用してアプリケーションの概念モデルを操作するかを説明するために拡張される。:時計の所有者はどのようにして曜日をリセットするのか？

開発者は、非機能要件とともに上位レベルの設計(6章)の間に開発されたシステムの構成に備えて解析モデルを使う。

この章では、より詳細は解析について議論する。

オブジェクトの同一性、振る舞い、関係、分類、構成に注目する。

複数チームでの開発プロジェクトという状況における解析に関連した管理問題について説明する。

最後に、より詳細な解析問題や、ARENA ケーススタディを使用した際のトレードオフについて議論する。

#### 2.1 はじめに:錯視

1915 年、Rubin は図 5-1 のような図を示した。あなたには何が見えるだろうか。見つめ合う二つの顔だろうか。

白い部分により注目すれば、代わりに花瓶が見えるかもしれない。

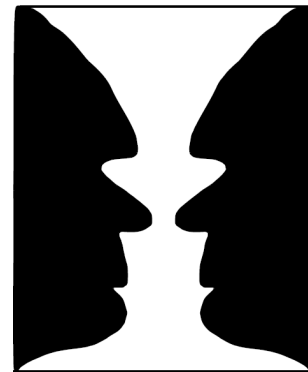


図 1: 曖昧さ:何が見えますか？

一旦両方の形を個別に認識できれば、花瓶と顔の間を簡単に行ったり来たりできる。

もし図 5-1 の絵が要求仕様であれば、どちらのモデルを作成すべきだろうか？

どちらも画像のような仕様は、自然言語に固有の不正確さや仕様を記述した人の仮定による曖昧さを含む。

例えば、単位の明記されていない量は曖昧であり、時間帯のない時刻は曖昧である。

形式化は要求仕様で矛盾した部分や省略された部分と同様に、曖昧な部分を確認する助けになる。

一旦開発者が仕様書の問題を確認すれば、更なる情報をユーザや顧客から引き出すことによってそれら进行处理する。

要求の抽出や解析は同時に起こる反復型の作業である。

#### 2.2 分析の概要

解析では、解析モデルと呼ばれる、正確で完全で一貫性のある検証可能なシステムのモデルを作成することに注目する。

解析は、開発者がユーザから引き出された要求の構造化や形式化に注目するという点で、要求抽出とは異なる。この形式化は、要求物における新たな視点やエラーの発見につながる。

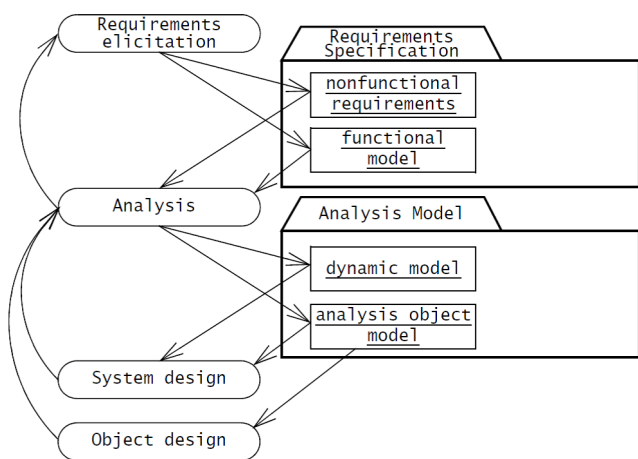


図 2: 要求抽出と解析の結果 (UML のアクティビティ図)