

OJL 事例報告：X 線画像を用いた自動検査の 可否判断支援システムの開発

石黒浩章[†] 白井靖人^{††*}

静岡大学大学院情報学研究科では、「IT スペシャリストプログラム」と称する OJL (On the Job Learning) を取り入れた教育プログラムを展開している。OJL では、修士学生が大学外の企業等との共同プロジェクトを実施する。OJL の目的は、受講者の「メタ技術展開力」を涵養することである。本稿では、当プログラムで実施された「X 線画像による自動検査の可否判断支援システムの開発」プロジェクトの事例紹介をおとして、広く情報システム人材育成における OJL の有効性に関して考察する。

An OJL Case Report: Developing a System for Feasibility Judgment of Automatic Diagnosis by X-ray Images

HIROAKI ISHIGURO[†] YASUTO SHIRAI^{††*}

At Shizuoka University, Graduate School of Informatics offers an educational program called "IT Specialist Program" which incorporates OJL (On-the-Job Learning) as its integral part. As a part of the program, a student takes on a project conducted jointly by the university and an external organization. Through the involvement in a project, the student is expected to gain meta-ability to deploy technology. In addition to describing one of such projects involving the development of a system for evaluating feasibility of automatic diagnosis using X-ray images, this paper discusses the effectiveness of OJL in developing IT human resources.

*[†] 静岡大学大学院情報学研究科
Shizuoka University Graduate School of Informatics
^{††} 静岡大学情報学部
Shizuoka University of Informatics

1. はじめに

静岡大学大学院情報学研究科では、「IT スペシャリストプログラム」と称する教育プログラムを展開している。本プログラムは、先導的 IT スペシャリスト養成プログラムの名古屋拠点「OJL による最先端技術適応能力を持つ IT 人材育成拠点の形成」の参加校として提案した修士課程のためのプログラムである。プログラムの特徴は、OJL (On the Job Learning) や PBL (Project-Based Learning) を取り入れて、ソフトウェア工学の基礎となる「ベース技術」のみでなく、「メタ技術」を育成することを目的としていることである。プログラムの詳しい説明は先行文献[1][2][3]を参照されたい。

本稿が報告するのは当該プログラム中の OJL というプロジェクトについてである。OJL では、学生が大学外の企業と連携してプロジェクトを実施する。OJL では、納期、予算などといった実社会の制約も真摯に受け止めながら、開発面、技術面では大学での研究成果の応用を重視して進められる。OJL の目的は、プロジェクトを通して受講者に「メタ技術」を身につけさせ、社会における即戦力となり得る人材を育成することである。

ここで「メタ技術」とは、蓄積された「ベース技術」を基に、それらを選択・合成し、新たに課題を解決するための技術を生み出す技術をさす。「ベース技術」の知識だけでは実際の開発に適用する際の様々な問題に対応できないため、近年産業界から大学へ寄せられる問題解決能力育成要請への対応である。

本稿では、上記 OJL を評価するための試みとして、地域の特色を活かした組込みシステム開発事例である「X 線画像による自動検査の可否判断支援システムの開発」を取り上げる。学生の視点から、その開発内容と経緯、及びそこから学んだことについて報告し、OJL の有効性に関して考察を行う。

2. プロジェクトの概要

2.1 本プロジェクトのテーマ

本プロジェクトのテーマは、X 線画像を用いた検査支援システムの開発である。本 OJL の協力企業は、インライン X 線検査装置を取り扱っているイメージテック株式会社（以下、イメージテック社）である。

2.2 プロジェクトの期間と構成

(1) 期間

2009/10 ~ 2011/2

(2) 構成

イメージテック株式会社（協力企業） 水野 俊博（PM: Project Manager）
静岡大学情報学部 白井 靖人（指導教員）
静岡大学情報学部 萩川 友宏（指導教員）
静岡大学大学院情報学研究科 石黒 浩章

2.3 先行プロジェクト

本プロジェクトに先立って先行プロジェクトが実施された。本プロジェクトはその問題を改善するために実施された。先行プロジェクトのテーマは「工業製品における全数検査支援システムの開発」である。

先行システムの概要を図1に示す。先行プロジェクトのシステムは、サンプル画像から教師データを作成し、それと検査装置から届く検査対象画像を比較して判定処理（スクリーニング）を行うシステムである。このシステムにより機械で判定できる製品の画像は人が目視で判断する必要がなくなるため、目視で確認する画像の数を減らすことができ、作業を効率化できる。

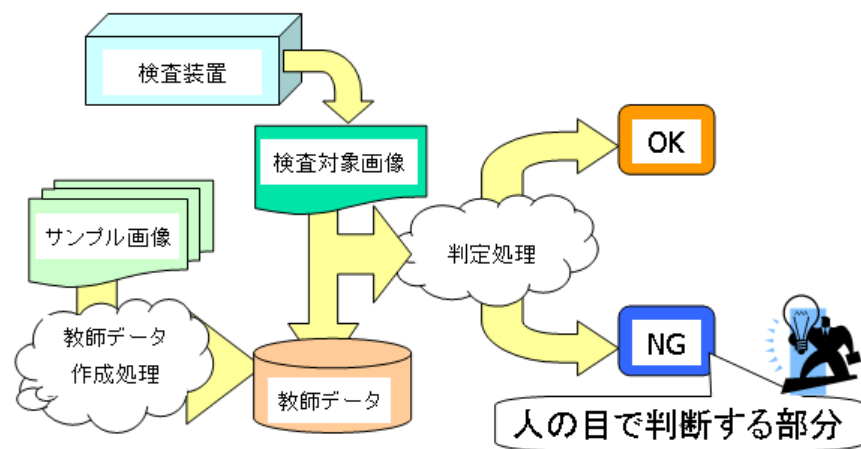


図1 先行システムの概要図
Figure 1 Outline of the Earlier System

2.4 プロジェクトの課題

協力企業の主たる要求は、X線画像を用いた製品検査の自動化である。先行プロジェクトで機械判定による全数検査支援システムが開発された。しかしながら、多様な製品に対応できない、という問題があった。

多様な製品に対応するためには製品や発生する異常によって有効な処理を選択する必要がある。これを解決するのが、本プロジェクトの目的である。具体的な解決課題は以下の2点である。

- (1) イメージテック社では画像処理に関する技術開発を外部企業に依頼しており、自社ではノウハウを蓄えておらず有効な処理の取捨選択ができない。
- (2) 外部企業に依頼している現状では、顧客からの要望に対して外部企業の返答を待つ必要があるため、対応に時間がかかってしまう。

この問題の背景にある現在の業務工程を図2に示す。自動化を要求する顧客に対し、サンプル品のX線画像を取得して、発生する異常の情報と共に外部企業へ渡す。外部企業は自動化が可能か判断し、それを受けて顧客へ結論を返すという工程である。この工程において、判断に至るまでの画像処理の試行錯誤は外部企業の中で行われており、イメージテック社にはその内容が分からない。また、判断が返ってくるまでの時間は一定でなく、長ければ2週間かかることもある。

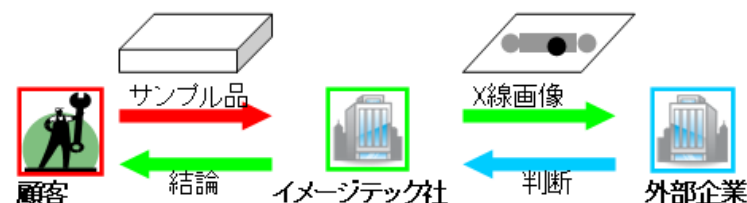


図2 現在の業務工程
Figure 2 Current Workflow

3. 本OJLのベース技術

本章では、本OJLで利用したベース技術について述べる

3.1 プロジェクト管理

本プロジェクトでは、進捗管理にEVM (Earned Value Management) を利用した。ただし、価値 (Value) への変換は難しかったため、単位を工数 (時間) で計上した。

開発プロセスには反復モデルを利用し、第1フェーズで基本システム、第2フェーズで画像処理機能の追加、第3フェーズで調整という工程で行った。

3.2 要求定義工程

要求定義の工程では、UML (Unified Modeling Language) を用いて機能やシステムの動作を説明した。また、業務フローの作成を通して必要な機能の考案を行った。

3.3 設計工程

設計工程に関しては、PAC (Presentation Abstract Control) アーキテクチャを利用し

た. PAC アーキテクチャは責務を担ったエージェントによる階層構造となっており, エージェントは Presentation, Abstraction, Control の要素を持っている. また, デザインパターンでは簡易的な Factory パターンを利用し, 生成と利用を区別した.

3.4 テスト工程

テスト工程では, ホワイトボックステストのカバレッジを利用した. カバレッジはシステムの制御フローにおける網羅率であり, 信頼性と労力を考えて C0~C7 のレベルから選択する必要がある. また, ユースケース記述に沿ったシナリオ作成による受入テストの設計などを利用した.

4. プロジェクトの内容

4.1 プロジェクトの目的

プロジェクトの目的は以下の二点である.

- 画像処理に関するノウハウを蓄積する
- 製品検査自動化の可否判断を迅速に行う

X 線画像を用いた製品検査の自動化が可能か不可能かの判断を支援し, その成果を蓄積するシステムを開発することが本プロジェクトの最終目的である.

今回開発したシステムの概要を図 3 に示す. まず, 本システムは取得した X 線画像に対し位置調整や領域抽出などの前処理を行い, 元画像を作成する. 次に, その元画像に画像処理パターンを適用して結果を表示する. そして, それらの結果を見てユーザはパラメータ調整や新しいパターンの生成を行う. 最終的に, ユーザは検査自動化が可能か不可能か判断を下し, その結果を元にレポートを生成する.

本システムは, 様々な画像処理の組み合わせをパターンとして登録することでノウハウの蓄積を支援し, その適用結果をユーザに表示することで自動検査の可否判断を支援する. また, 新規画像処理パターンの登録や, 登録したパターンにおける各画像処理のパラメータ調節機能によって, 可否判断を行うための試行錯誤を支援する. さらに, 行った画像処理と判断を入力してレポートを生成する機能によって, ノウハウの蓄積と顧客に対して判断を説明する際の根拠までを確保できる.

4.2 プロジェクトの開発プロセス

本プロジェクトの開発プロセスはインクリメンタルモデルを適用した. プロジェクトを 3 つのフェーズに分割し, 第 1 フェーズで基本システムの作成, 第 2 フェーズで画像処理など機能の拡充, 第 3 フェーズで最終調整という構成で実施した. 第 2 フェーズまででシステムを稼働させることはできるため, 第 2 フェーズの終了時に評価の一環として, JIMA2010 国際検査機器展 (以下, 展示会) でイメージテック社の出展に同行させていただき, 開発したシステムを稼働させた.

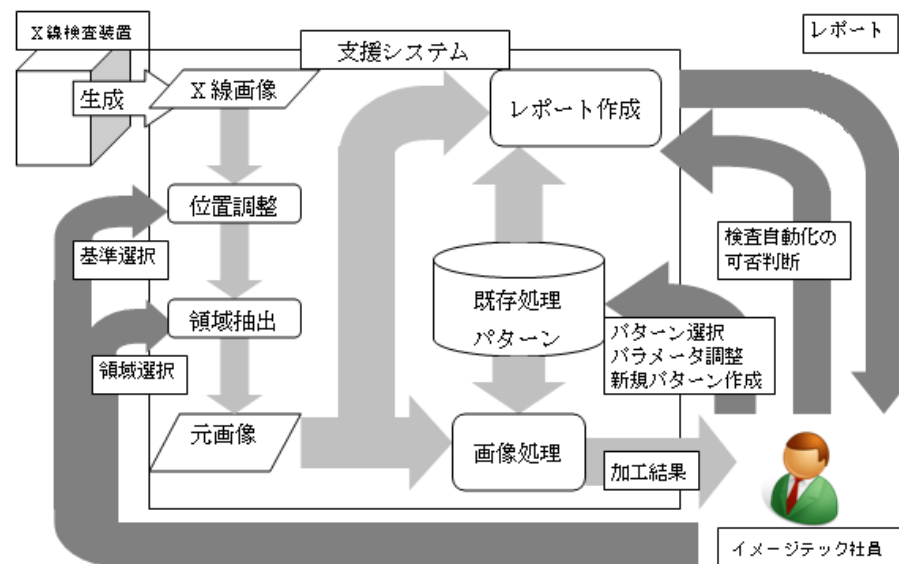


図 3 システム概要図

Figure 3 System Outline

4.3 プロジェクト管理

プロジェクトの進捗は, PBL と同様に単位を工数 (時間) とした EVM を用いて管理した. 全体の推移チャートを図 4 に示す. 図 4 中, PV (Planned Value) は計画上の工数, EV (Earned Value) は実際の成果, AC (Actual Cost) はかけた工数を示す. プロジェクト全体の PV は 498, AC は 540, EV は 498 であった. フェーズごとの結果は, 第 1 フェーズが PV201, AC181.5, EV149 である. 第 2 フェーズが PV187, AC232, EV239 である. 第 3 フェーズが PV110, AC126.5, EV110 である. 2009 年 10 月から 2010 年 3 月までは先行プロジェクトに参加し学習を行った期間であり, 開発の開始は 4 月である. 4 月から 8 月までが第 1 フェーズ, 9 月から 11 月までが第 2 フェーズ, 12 月から 2 月までが第 3 フェーズとなっている.

本プロジェクトの PV は 500 程度である. 先行プロジェクトにおける PV が 1000 であったことを鑑みると, 比較的少ない数字である. これはシステムの開発と並行して別のプロジェクトに携わったためであり, 別のプロジェクトの作業時間を抜いた PV で計上したためである.

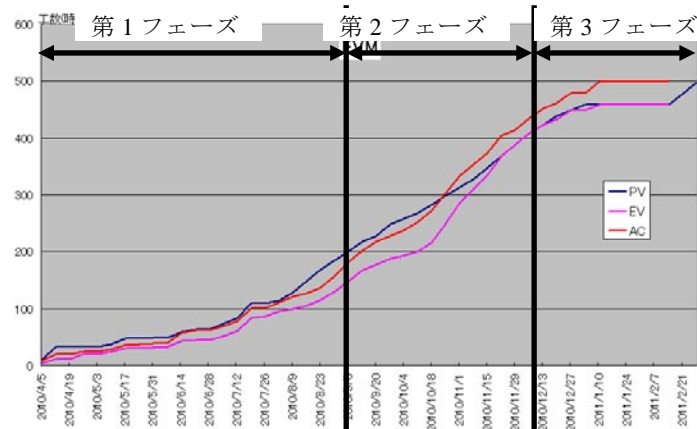


図4 EVMの推移チャート
Figure 4 EVM Chart

4.4 要求分析

4.4.1 基本システム要求分析

第1フェーズでは基本システムに関する要求を抽出した。まずはイメージテック社からの要望をヒアリングで集めた。しかし、今回開発するシステムの機能は今までイメージテック社が外部企業に依頼していた部分であり、必要な機能の分析が難しかった。そのため、まずはフリーのツールを用いてX線画像に画像処理を施し異常を抽出する作業を行った。そこから業務フローを作成し、そのフロー内で自分が行う作業とシステムにとって必要な機能を考え、ユースケース図とあわせて要求仕様書を作成した。

その際作成した業務フローを図5に示す。これはユーザが画像を入手してから可否判断を行うまでの作業を示した図である。この図から、システムの機能として画像の読込、領域のトリミング、多数の画像処理の機能が必要であることがわかった。また、ユーザ側の操作として、フォルダ選択、領域選択、画像処理の選択とパラメータ調整が必要であることが分かった。

4.4.2 画像処理の要求分析

第2フェーズでは、イメージテック社で得られる異常製品のサンプル画像から、これらの異常を抽出するための画像処理機能を考案した。サンプルは4種類あり、その中に埋め込まれた異常は異物混入、空洞の発生、製品の破損などが存在した。画像から有効と考えられる画像処理を選択し、設計を行った。

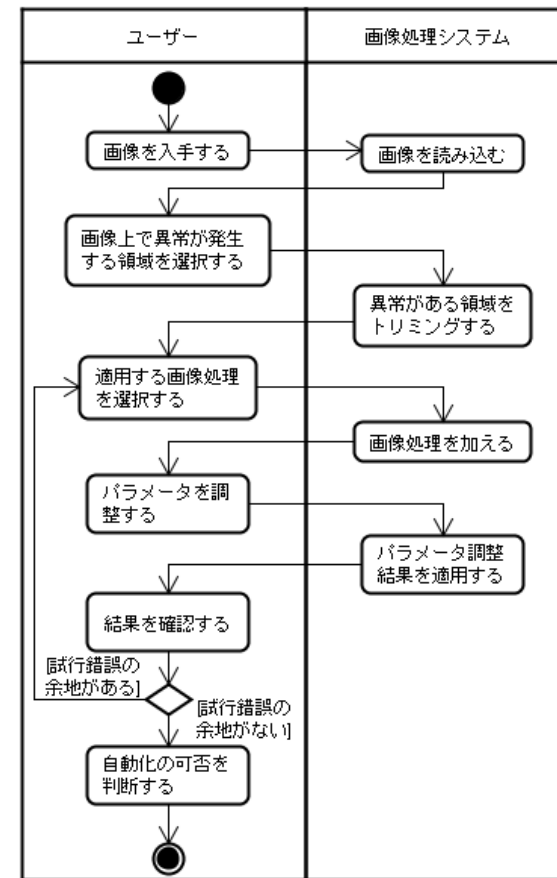


図5 業務フロー
Figure 5 Workflow

当初、ここで実装する画像処理機能は16bitの画像を対象にする予定であった。イメージテック社で用いているカメラの精度は10/12bitであり、16bitの画像にその値を保存することができた。しかし、第1フェーズの遅延対策のため、取得した画像を8bitの画像に変換しOpenCVを活用することにした。

また、当初はサンプル画像もイメージテック社の顧客から得られる様々な画像を想定していたが、時間の都合によりイメージテック社で用意できる異常製品と、仮想的に再現できる異常製品からサンプルを得ることにした。

4.4.3 最終調整

第3フェーズでは、展示会や実際の稼働を通して得られた情報からシステムに対する要求をまとめた。最終的なユースケース図を図6に示す。ユーザの操作が絡む機能が記述されておりそれぞれの機能の関係が分かるようになっている。また、「画像を拡大表示する」機能は様々な場面での使用を想定し、対象機能の拡張で表現されている。

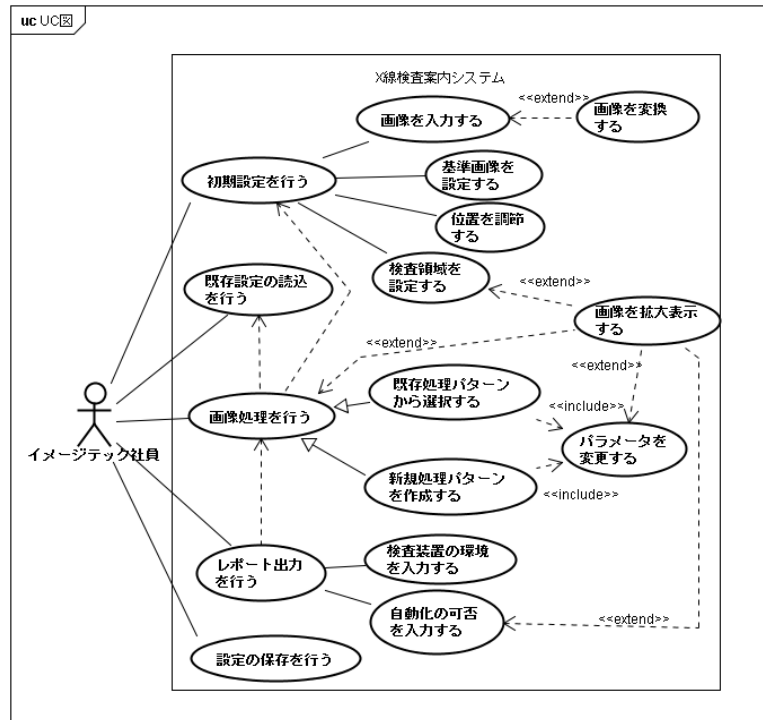


図6 ユースケース図
 Figure 6 Use Case Diagram

4.5 システムの設計

4.5.1 全体の構成

開発したシステムのパッケージ図を図7に示す。「メインシステム」が全体を統括しており、各機能呼び出ししている。「初期設定」は画像の読み込みから領域設定まで、画像処理の前段階の作業を担っている。「処理パターン」は既存パターンのデータと、新規パターンの作成機能を持っている。また、「画像処理」は画像処理機能とパラメータ調

整機能を持っており、画像処理機能の追加や改良時にはこのパッケージを変更する。「位置補正」は、先行プロジェクトのパッケージを流用している。「共通データ」にはシステム全体で利用するデータや、様々な場面で利用される拡大機能を担っている。

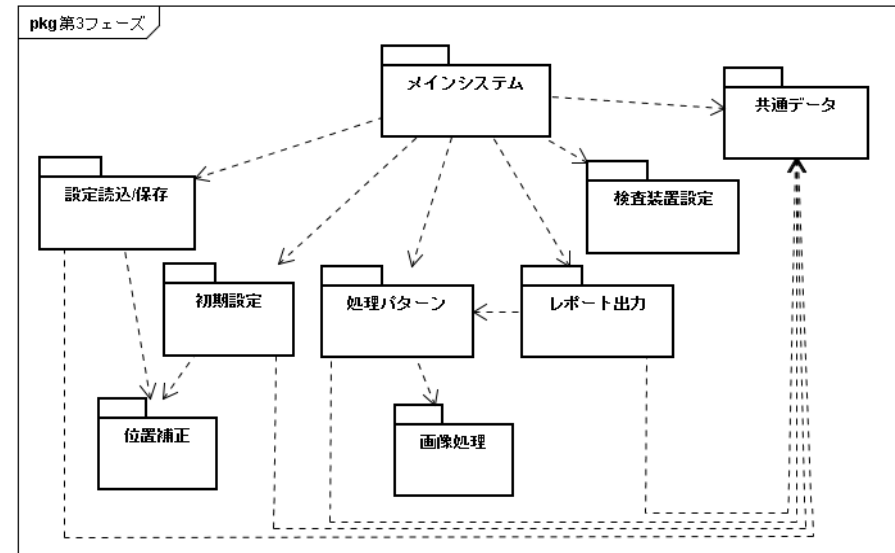


図7 パッケージ図
 Figure 7 Package Diagram

システムの構造は先行プロジェクトで用いられていた PAC アーキテクチャを採用した。先行プロジェクトは PAC アーキテクチャに則り、PAC を内包するエージェントの階層構造で表現されていた。また、エージェントが持つデータは保持されており、Composite Message パターンを利用した連絡を通してデータを呼び出していた。現プロジェクトでも PAC に相当するように分割され、パッケージ間で階層構造で構成されている。

このアーキテクチャは、開発言語であった C#や.NET フレームワークと相性がよく、エージェントごとに責務を負うため役割分担が容易であることが利点といわれている[4]。先行プロジェクトで用いられていたため資産の流用や再利用が容易であった。

4.5.2 抽象化による拡張性の確保

OJL の初期段階で画像処理機能についての詳細を設計することが困難であり、システム完成後も画像処理技術の発達に伴い画像処理機能の追加を行いたいという要求があったため、画像処理機能については拡張性を持たせる必要があった。そのため、

各画像処理フィルタの抽象クラスと、生成用の Factory クラスを作成した。画像処理部分のクラス図を図 8 に示す。この形にすることで、抽象クラスを継承して作成した画像処理クラスは、Factory 内のメソッドを変更することで追加可能となった。

画像処理の中には画像同士の差分を取るなど、入力画像が二つとなる処理も存在する。特に今回の場合は、正常品の画像と異常品の画像の比較が必要となる可能性が高い。そのため、抽象クラスには入力画像が一つの場合と二つの場合について抽象メソッドを作成した。第 2 フェーズの画像処理機能を追加していく前にその可能性を考慮することができたため、抽象クラスの変更による作業を発生させることなく画像処理機能を追加することができた。

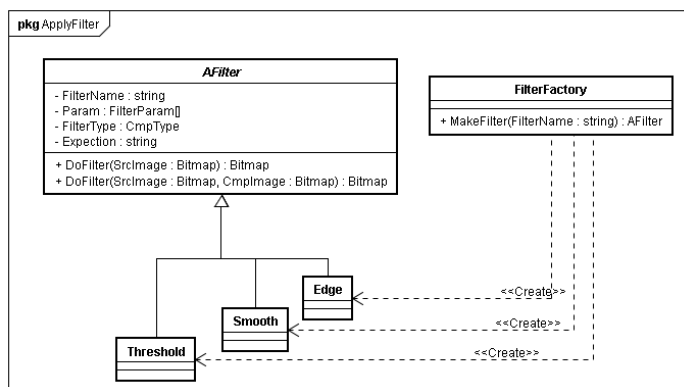


図 8 画像処理機能のクラス図

Figure 8 Class Diagram of Image Processing

4.6 システムの評価

JIMA2010 国際検査機器展において、OJL 協力企業は検査装置による画像取得の実演で展示を行った。その際、その場で持ち込んだ製品のサンプル画像取得も行っており、検査の自動化を尋ねてきた顧客も存在した。それらの顧客に、図 9 のようにサンプルに対して本システムを用いて異常を抽出し、レポートの出力まで行った。

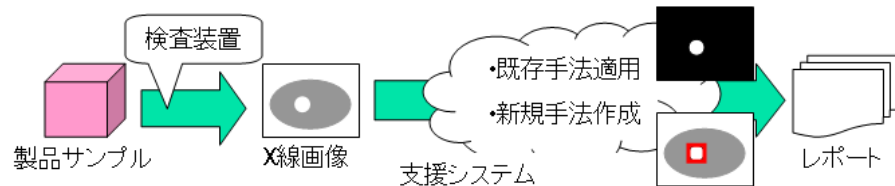


図 9 展示会におけるシステムの稼働

Figure 9 The operation of the system in the exhibition

実際の実行画面を図 10 に示す。画像は空洞入りのボールジョイントで、後ろの画面は異常部を矩形で囲む画像処理パターンであり、手前の表示は画像処理のパラメータ調整画面である。パラメータ調整は、変化を確認しながら調整を行うことができる。



図 10 システム画面

Figure 10 Screenshot

検査の自動化を求める顧客が持ってきたサンプル製品から、X線画像を得て、本システムを稼働し画像処理を行って異常が抽出できることを確認できた。そのため、暫定的にはあるが自動化が可能であると判断したレポートを作成し、その日のうちに顧客に対応を行うことができた。

今までの業務からすれば、判断を下し対応できるのは1日以上過ぎた後であったが、システムを導入することでその日のうちに対応することができた。また、異常抽出の画像処理パターンを作成できたこと、レポートという形で履歴を残せたことから、イメージテック社のノウハウとして蓄積することができた。このことから、プロジェクトの目的を達成するシステムを開発することができたと判断した。

5. OJL で学んだこと

5.1 メタ技術

5.1.1 開発プロセスの選択

本システムの開発プロセスには反復型の開発モデルを適用した。当初、開発モデルについてはウォーターフォール型を予定していたが、企画当初から仕様が明確に決まっていることが少なく、実践でウォーターフォール型を適用させるのは難しいと実感した。今回のようなプロジェクトでは、要求定義の段階でシステムの大きな流れが決定していたことから、第1フェーズにて大きな流れを作成し、第2フェーズ以降に機能を追加していく反復モデルの流れで行うことができた。要求定義段階で大きな流れが見えず、フェーズ毎に変化の可能性を感じられたならば、スパイラルモデルを適用するのが好ましいと考えられた。

5.1.2 要求分析の反省

本システムの要求定義工程で、機能の分析が不足していたことから手戻りが発生した。要求仕様書は作成したが、確認をPMと開発者のみで行ったため、専門の知識を持つアドバイザーの観点から問題点を指摘してもらうことができなかった。

また、システムが持つ機能の説明にユースケース図を用いたが、顧客側と開発側の双方がUMLについて理解していなければ、図の意味を理解するために時間がかかってしまうことが問題として起こった。顧客側がユースケース図やユースケース記述の表記に対して意味を理解することができず、「その機能で本当に要求を満たすことができるのか」ということを議論できなかった。そのため、再分析を行なった際には限定的な機能を動かせるプロトタイプを作成し、実際の大まかな動作を顧客と共に確認しながら必要な機能を詳細に分析することができた。早期の段階でプロトタイプを作成し動かすことは手戻りを防ぐ上で有効な手段であり、顧客の理解度に左右されず説明ができるとわかった。

5.1.3 アーキテクチャの選択

本プロジェクトではアーキテクチャとしてPACアーキテクチャを適用した。このアーキテクチャは先行プロジェクトで決められており、アーキテクチャの利点を生かし第2フェーズ以降で抽出された画像拡大表示機能、画像変換機能などを新しいエージェントにすることで、機能の拡張を容易に行えた。また、先行プロジェクトの位置補正機能などをエージェントごとと流用することができた。

他に候補として考えられたのはレイヤー型やMVC (Model View Control) 型があった。その中で、本プロジェクトにおける要求に画像処理機能の拡張性があり、拡張性に優れたアーキテクチャであるPACとMVCが考えられたが、先行プロジェクトやその開発環境との親和性を考え、PACアーキテクチャを選択した。

5.2 その他

5.2.1 開発の途中参加に有効な資料

本システムの開発に移る前に、先行プロジェクトの開発に携わった。その際、システムの目的を知るために「要求定義書」が有効であった。何を目的にしており、何ができるようにすれば良いのか、という方針を知ることができた。ただし、版管理がされていないこともあった。メンバーに確認を取ったところ内容が異なっているということもあった。

また、システムの構成に「PACアーキテクチャ」を適用しているという情報から、役割がエージェントとして独立しており、データがどこに存在するか、エージェント間の連絡はこのメソッドを見ればよいかなど、開発中のシステムについて学ぶことに役立った。

これらのことから、システムの方針と構造の情報について、要求定義書やシステムの構成図としてパッケージ図とそのパッケージが担う責務などが資料に残してあると有効であることが分かった。また、版管理と一貫性の確保の大切さを実感した。

5.2.2 ツールの有用性

先行プロジェクトの単体テストに携わった際、ツールを使うことの有用性が分かった。また、その際コメントを残すことによって何を調べようとしてそのテストを組んだのか把握することができた。開発と単体テストがほぼ同時に行われていた状況もあり、何度も回帰テストすることでバグ潰しをしながら明確に行うことができた。また、当時はモジュールの目的や入力・出力に詳しくなかったこともあり、テストの設計に苦労したが、C2カバレッジ (分岐網羅) という明確な目標を持つことで、モジュールの内容を理解しながらテストを組むことができた。

5.2.3 EVMによる進捗管理

今回のプロジェクトはEVMを用いて進捗を管理した。工程を時間で管理したため、厳密なEVMというわけではなかったが、予定した作業時間、予定していた進捗を考えながら作業することで、安定した作業量をこなすことができると理解した。

しかし、プロジェクトの当初における要求定義工程では、何を基準に計上すればよいのか判断できなかった。設計以降の工程では、作成する機能や設計図などで具体的な進捗を判断することができる。また、要求定義工程でも要求仕様書を作成している段階であれば、章毎の完成度で計上することも可能だと考えられる。しかし、要求を抽出している段階などでは、要求を抽出したといっても全てできたと判断することはできないし、全要求のうち何割と示すこともできない。また、必要な機能もシステム作成中や作成後に新しく発生することも考えられる。

今回のOJLで要求定義工程を終えるきっかけにできたのは、プロトタイプによる合意形成だった。抽出した機能を簡易的に実装し、顧客と共に確認を行い必要な機能を決定することができた。このことから「プロトタイプを作成できるほど分析を行えた」

という進捗や区切り目を入れることができるのではないかと考えた。

EVM において PV や EV を計上する際は、「何ができたならその工程が終了なのか」、「どこまでできたら全体の何%進んだのか」を考える必要がある。ほとんどの工程では WBS を明確に作成することで判断できるが、要求定義工程は成果物の数が少ないこともあり難しいと思った。

5.2.4 拡張性を持たせる構成

本システムでは画像処理に関するモジュールに拡張性が求められ、抽象化の有効性を体感することができた。しかし、これは抽象化したモジュール自体に変更を加える必要ができてしまったとき、多大な変更作業が必要となる。抽象化を利用するためには、具象化先で起こりうる変更や操作、パラメータなどを把握した上で、可能な限り変更が起らないように留意する必要があると分かった。

6. おわりに

本稿では、「IT スペシャリストコース」の一環として実施された OJL の事例を報告した。OJL とは大学と企業が共同でプロジェクトに取り組み教育するカリキュラムで、学生のメタ技術の養成が目的である。本 OJL プロジェクトのテーマは「X 線画像を用いた自動検査の可否判断支援システムの開発」だった。先行プロジェクトの多様な製品に対応できないという問題点と、業務工程の都合で自動検査の可否判断に時間がかかってしまうという問題点を改良するため、「画像処理のノウハウの蓄積」「迅速な自動検査の可否判断」を目的として本プロジェクトは実施された。ベース技術はプロジェクト管理に EVM、要求定義や設計工程で UML、テスト工程でカバレッジなどを利用した。プロジェクトは PV500 程度であり、ほぼスケジュールどおり実施され、展示会の使用に耐えるソフトウェアが納品された。本 OJL で学習した成果として、「開発プロセスの選択基準」、「顧客との合意形成方法の選択」、「PAC アーキテクチャの活用」をメタ技術として提案した。

謝辞 本プロジェクトの全過程において、親切丁寧なご指導を頂きました静岡大学情報学部秋川友宏准教授、塩見彰睦教授、松澤芳昭助教に厚く御礼申し上げます。また、本 OJL の機会及び素晴らしい実験環境を与えてくださり、貴重な時間を割いて本プロジェクトにご協力頂きましたイメージテック株式会社の菅圭司殿、水野俊博殿に心から感謝いたします。そして、多大なご協力と指導を頂きました先行プロジェクトのメンバーに感謝します。

参考文献

- 1) 沢田篤史, 小林隆志, 金子伸幸, 中道上, 大久保弘崇, 山本晋一郎: 飛行船制御を題材としたプロジェクト型ソフトウェア開発実習, 情報処理学会論文誌 Vol.50, No.11, pp.2677-2689 (2009)
- 2) 松澤芳昭, 塩見彰睦, 秋川友宏, 酒井三四郎: “ソフトウェア開発の教員主導型 PBL における反復プロセスと EVM 導入の効果”, 情報処理学会研究報告(CE-99-9), pp.1-8(2009)
- 3) 小林隆志, 沢田篤史, 山本晋一郎, 野呂昌満, 阿草清滋. "On the Job Learning: 産学連携による新しいソフトウェア工学教育手法", 情報システム学会誌, Vol.5, No.2, pp.44-56 (2010)
- 4) F.ブッシュマン, R.ムニエ, H.ローネルト, P.ゾンメルラード, M.スタル. “アーキテクチャパターン”. ソフトウェアアーキテクチャーソフトウェア開発のためのパターン体系. 近代科学社, 2000 年, p25-215