産学連携におけるオープンソース理論を活用したシステム構築

The Industry-University Cooperation Model appling Open Source Theory.

井　上　　明

1．はじめに

近年、TLO[1]をはじめとする産学連携の活動が活発化してきている。産学連携の大きな目的のひとつとして「大学に存在する知識・技術の社会移転」がある。しかしながら、現在の産学連携の多くは、企業から資金、人材、設備などの援助をしてもらい、商品やサービスを大学側で開発する「企業の下請け的連携」、または大学側が、連携先の企業や行政の問題点を、企画書や報告書の形でまとめただけの「『提案』どまりの連携」になっているところが多い。

本研究では、学生・教員自身が社会と連携を実施しながら、実社会の情報システムを開発する産学連携形態を提案する。また、その具体的方法論として、ソフトウェア開発手法のひとつである「オープンソース理論」を用い、社会への知識移転を行う為の方法論について考察をおこなう。

情報システムを構築するという行為は、「その企業の業務を知り」「実際に発生している問題を解決する」ことに他ならない。

情報システムの構築を、コンピュータを使った「業務効率化」「少力化」としてとらえるのではなく、企業や行政といった様々な分野における実社会の活動をより深く具体的に知り、多方面の分野の知識・技術を駆使しながら問題解決を図る「問題発見・解決プロセス」としてとらえる。その情報システム構築活動を、企業や行政といった当該組織の当事者だけで実施するのではなく、文系、理系、芸術といった様々な分野の専門知識を有する大学と連携しながら開発を実施することで、新たな成果やサービスの創出が期待できる。

---

[1] TLO：Technology Licensing Organization、大学や研究機関などの研究成果や新技術を発掘し、研究者に代わって特許等の形で権利を取得し、その事業化を希望する民間組織に対し技術移転を行う組織

2．産学連携とオープンソース理論

2．1．従来型システム開発における問題点

従来、産学連携において、「大学による企業の情報システムの開発」はほとんど行われていなかったといえる。

情報システムの開発が実施されにくかった大きな要因としては、1）開発環境への設備投資が必要、2）マンパワーの不足、3）対象業務の理解不足などが考えられる。

まず、第一点として、大学では年間の研究費が決められており、高価なソフトウェアやハードウェアを整備することは難しい。特に、情報システム開発の場合、客先（企業）の環境を開発側でも再現し、その環境のもとでシステム開発を実施する場合が多い。しかし、大学では企業のシステムを再現するといったような環境の構築は事実上不可能である。

次に、マンパワーの不足が挙げられる。従来型のシステム開発手法の主流であったウォーターフォール型では、要求定義、外部設計、内部設計、詳細設計、プログラミング、テスト、といった各フェーズを詳細な設計書と共に進めていった。このようなシステム開発手法では、莫大な人員と期間が必要であり、1研究室や個人では開発が実施できない。また、学生は授業単位や修学年度を過ぎるとシステム開発から離れてしまうので、人員の入れ替えがシステム開発へ大きな影響を与えるシステム開発手法では、継続的な作業が実施しにくいといえる。

三点目は、対象業務の理解不足である。企業などの業務に関わる情報システムを構築するには、「その業務を知る」ことが必要とされる。対象業務を知るには、実際にその業務を経験することが望ましいが、学生の身分では時間的・物理的制約が多く難しい。その場合、業務担当者と頻繁なコミュニケーションをとり、業務内容などを教えてもらう必要がある。しかし、実際に企業へ出向いての会議、または電話での連絡というのは、相手の都合などもあり実施しにくい。

つまりは、従来、大学で実施されていた情報システム開発とは、連携相手

や特別な環境を必要としない「アプリケーション・ソフトウェアの開発」や「プロトコルの実装」などが主たるシステム開発であったと考えられる。そして、作業量・時間、人員といった各種要因、要求定義や各種ドキュメント作成を必要とするシステム開発手法が「産学連携」という場面に適用しにくかったいえる。

3．産学連携におけるオープンソース理論の適用

本研究では、産学連携における「大学・企業共同でのシステム開発」において、オープンソースの理論を適用する。

オープンソースとはソフトウェアの開発手法の一つであり、オペレーティング・システムからアプリケーション・ソフトウェアといった幅広い種類のソフトウェア開発に用いられている。近年、サーバ用途向けオペレーティング・システムの市場シェアで急速に伸びているLinuxもオープンソースによって開発されたソフトウェアである。

オープンソースの定義は、OpenSourceInitiative[2]によると、オープンソース・ソフトウェアの定義として、「再配布の自由」「ソースコードの公開「ライセンスの継承」などの特徴をもったソフトウェアと定義している。

さらに、オープンソースでは、成果物の配布・使用に関する特徴だけでなく、開発プロセスにも特徴がある。それら特徴を明らかにする。

（1）ユーザ参加での開発

オープンソースでは、「ソース公開、修正、機能追加」が許可されている為、ユーザがソフトウェアの機能追加や修正に参加できる。つまり「利用者が開発者」になれるわけである。

産学連携の形態で、企業におけるシステム開発へオープンソース理論を適用する場合、ターゲットとなる適用範囲の検討が重要課題になる。産学連携で開発する場合、学生からの新しい意見やアイデアを活用したいが、企業機密に関わる部分はオープンにしたくない。また、基幹システムのような大規

2　OpenSourceInitiative（OSI）：1998年設立。オープンソースの定義オープンソース・ライセンスの承認などを実施している任意団体。

模なものの構築は非常に難しい、といった問題がある。

つまり「既存システムの一部改変」や「全社的大規模システムの構築」ではなく、「新規システム」で「単体で独立可能」なシステム開発が、産学連携に適していると考える。利用者、学生、システム・エンジニアが対等な立場で、議論を重ね、様々な意見・要望を反映させながら、比較的小規模だけれども全く新しいシステムを作っていくようなアプローチである。

（2）インターネットの活用

オープンソースの代表作であるLinuxはインターネット上で開発されたといえる。世界各地の開発者が、メーリング・リストやウェブページ、FTP、NetNewsなどを活用し、情報やアウトプットの交換、蓄積を実施した。企業におけるソフトウェア開発では、ひとつの部屋へ、データベース、ネットワーク、ＯＳ、アプリケーションなどの専門のプログラマを集め、各自が決められた担当個所を構築していくという手法が多い。

しかし、オープンソースでは、インターネットを介して世界中からあらゆる分野の専門家が自由に参加した。24時間365日誰かがどこかでシステムの開発を行っていたわけである。また、バグや新しいアイデアがあればそれをインターネットで瞬時に連絡し、迅速な情報提供と共有を図っていた。

産学連携にもオープンソース・ソフトウェア開発で用いられたインターネットを活用した開発モデルを適用する。学生が企業へ出向き開発するのではなく、自分たちが好きな場所で好きな時間にメーリングリストやWebページを活用し、プログラムの配布、進捗状況の報告、意見交換を頻繁に実施しコミュニケーションを図りながら、システム構築を進めていく。企業側の担当者と直接会ってのコミュニケーションがあまり図れないデメリットを、インターネットを活用し補っていく。また、Telnetやエミュレータを使用して企業のコンピュータへ直接接続すれば、学生側では大規模な開発環境を構築しなくても済む。

メールや掲示板を活用すれば、そこで議論された内容やノウハウは直接デ

ータ化され「知識データベース」的な活用も可能である。卒業などでシステム開発を実施していた学生が変わっても、開発に関するノウハウは蓄積されており、知識の継承が望める。一般的なシステム開発では「どんなシステムを作るか」というドキュメントは詳細に書くが、「どうやって作っていったか」というプロセスについてのドキュメントはほとんど作成しない。メール・掲示板を利用し、プロセスに関する情報を蓄積できれば、システム開発の大きな参考になるであろう。

（3）報酬

オープンソースでの開発では、基本的にソフトウェア開発に携わっても金銭的報酬は受けられない。一方、ソフトウェア・メーカーの商用ソフトウェア開発者は、所属する企業から報酬を受け、仕事としてソフトウェアを開発している。オープンソースではなぜ金銭的報酬がないにもかかわらず世界中の人々がソフトウェアの開発に参加するのか。その理由として、エリック・レイモンドは「ノウアスフィアの開墾（Homesteading the Noosphere）」[ESR98]の中で、「仲間内での評判」だと述べている。仲間内での評判、つまり「名声」「名誉」を得ること自体が人間の本質的な欲求であり、評判を得ることで、そのグループ内での地位も上がり、自己の欲求を満たすことが「金銭に替わる動機」としている。

産学連携にオープンソースの考えを適用すれば、学生はシステム開発が「授業」の一環であるから「金銭的報酬」は無いが、知識やスキルの向上といった「報酬」は多く受けることができる。

（4）バザール方式

従来のソフトウェア開発手法の多くは、依頼者からのヒアリングからシステムの要求をまとめ、詳細な仕様書を莫大な時間と費用を費やして作成し、その後、厳密な仕様書に基づいてプログラミングを行い、テストを繰り返した後、出荷するという手法が取られていた。そこでは、システム設計、プロ

グラム・コーディング、テストといった各フェーズの核となる部分はそれぞれの専門家によって行われており、プログラミングの前には、全ての規則、要求、仕様が細心の注意を払い、慎重に決められていた。このような、「少数の専門家による厳密な設計に基づくシステム構築」における中央集権的システム開発を、エリック・レイモンドは「伽藍とバザール（The Cathedral and the Bazaar）」[ESR97]で「伽藍モデル」と表現している。伽藍のように、厳密な設計に基づき整然と構築されるものとしている。

一方、オープンソースでは、詳細なシステム設計書もなく、それを指揮し中央で統制・管理する人間もいない。Linuxの提案者であるリーナス・トーバルズも全てを統制しているのではなく、Linuxのカーネル（中心的部分）の変更の提案を採択するかどうかの最終的判断をしているだけである。

誰もが自由に参加し、厳密な規則もなくよってたかって開発を行う。このような開発方式を「バザール方式」とエリック・レイモンドは表現している。トップダウンのヒエラルキーではなく、参加者の自発性と相互信頼に基づき運営されている。実際には、全く無秩序に開発されているわけではなく、各担当分野にはそれぞれ主要メンバーが存在し、開発されたソフトウェアのバージョンも厳密に管理されている。

産学連携におけるシステム開発の場合も「バザール方式」が有効と考える。先に述べたように、詳細なドキュメントを作成した後、プログラミングを始めたのでは時間・労力ともに学生が担当できる範囲を超えてしまう。必要最小限のドキュメントと開発規則のみ決めておき、その都度、修正や追加を実施しながら開発を実施していく。また、バグが無くなるまでプログラムを公開しないのではなく、できあがった部分を頻繁にリリースし、動作確認、修正、機能追加を実施していく。リーナス・トーバルスのようなシステム全体の最終判断者は企業側のものが適しているが、大学側と企業とは対等な立場で作業を進めていく。「ここはどうすればいいのか」「何が必要なのか」のような疑問に関しては、大学側が企業へ確認しながら作業する。しかし企業側からの「こうしなさい」的なトップダウン的指示は適さない。

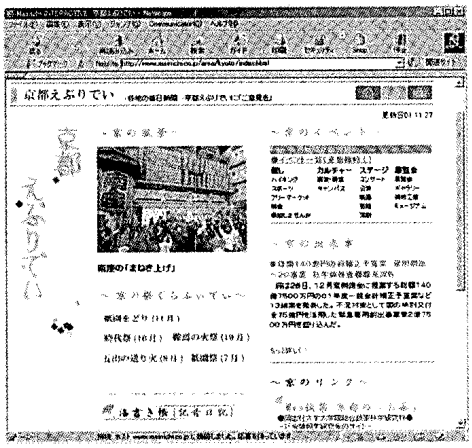

図1.「京都えぶりでぃ」トップページ
(http : //www.mainichi.co.jp/area/kyoto/)

4．毎日新聞社プロジェクト

4．1．プロジェクト概要

今回、産学連携におけるオープンソース理論の実践として、筆者、同志社大学大学院総合政策科学研究科、毎日新聞社京都支局の連携による地域情報集積・発信システム構築プロジェクトを実施した（図1）。

特徴として、企業と大学といった組織中心の産学連携ではなく、記者と学生・教員が主体になったボトムアップ型連携である。また、情報システムの構築といった理工系分野の課題であるが、参加した学生・教員は理工系出身者はもとより、法学・経済学・福祉といった様々な分野の出身者がメンバーとなり、現実社会で発生している問題解決を実践した。開発を通じて得られた種々の事柄を考察し、産学連携の場面でのオープンソース理論の可能性を検討する。

4．2．京都えぶりでぃ

本研究は、2000年7月、「毎日新聞京都支局独自の事業としてホームページを開設し、新聞としては一部地域でしか読むことのできない地域面の情報をウェブで提供してはどうか」という同志社大学大学院総合政策科学研究科博士前期課程に在籍していた、毎日新聞社京都支局記者猪狩（現サイバー編集部）の提案から始まった。

同志社大学大学院総合政策科学研究科の学生を中心に、2000年4月から社会情報学について自主的な勉強会を開いていた「社会情報研究会」のメンバーが協力することになった。社会情報研究会とは、「情報」の意義や活用を、法学・経済学・工学といった参加者それぞれの視点から、研究を行うことを目的としていた。

社会情報研究会では、猪狩の参加以前に「情報の価値と活用を考える」というテーマで自主的な勉強会を実施していたが、文献研究が中心で、参加者の学習そのものはほとんど進展していなかった。社会情報研究会では、猪狩からの「一緒に京都支局のWebページを作らないか」という提案に賛同し、共同作業がスタートした。また、その活動は一部、大学院博士前期課程の授業の一部として実施されることになった。参加者はメーリングリスト、掲示板などを駆使しながら、議論や意見交換、プログラム開発などを進めていった

当時、すでに多くの新聞社がWebページを公開し、各種情報提供を実施していた。しかし、その情報の多くは「全国版」の情報で、地域面に掲載されるような地方の情報はほとんどWebページに掲載されていなかった。

参加メンバーは、支局のWebページで提供するコンテンツについて議論をおこなった。「京都」が持つ様々な特徴を前面に押し出し、京都に住む人々はもちろん、それ以外の人々にも魅力的なページを目指すことが決まった。また支局Webページの名称を「京都えぶりでぃ」にすることも決まった。

参加者は、各自の視点から「どのようなコンテンツが必要か」のアイデアを出していった。紙面地域面に流れる情報、Web投票、高校野球速報、宗教家による法話、京都のイベント情報などが提案された。それぞれは「システム開発者」という立場ではなく、「一般読者」「利用者」の立場から、どんな情報であれば自分たちはそのWebページを見るかを考えた。

また、トップページのデザインもメンバーが担当することになった。トップページは紙面でいうと1面にあたる非常に重要な要素である。Webページ作成ツールなどを使用し、いくつかのサンプル・ページを作成し、それを

メンバー全員で検討していった。数種類のサンプルを同時並行的に作成し、頻繁に参加者へレビューし、そこでの意見・アイデアをすぐに反映しながらトップページを作成した。

4．3．イベントシステム開発

4．3．1．産学連携ゆえの問題

核となるコンテンツとして、京都の各種イベント情報を掲載することにした。京都のイベント情報はその数、内容ともに有益な情報が多い。しかし、問題は「情報の蓄積と加工をどうするか」であった。（図２）

支局の正式な業務としてのシステム開発であれば、専門のシステム開発者を雇い、ソフトウェア会社へ多額のシステム構築費を支払って、イベント情報を管理するシステムを構築させることも可能である。しかし、今回のシステム構築は、開発自体は毎日新聞社側から正式に認められてはいたが、あくまでも、自主的な立場での開発であり、金銭的、設備的援助はほぼ無いような状態であった。つまり、「企業から研究費をもらって開発する」といった従来型産学連携の方法では不可能であった。

そこで、多人数が効率的にシステムを開発し、さらに、可能な限り開発費がかからない方式での開発をスタートした。

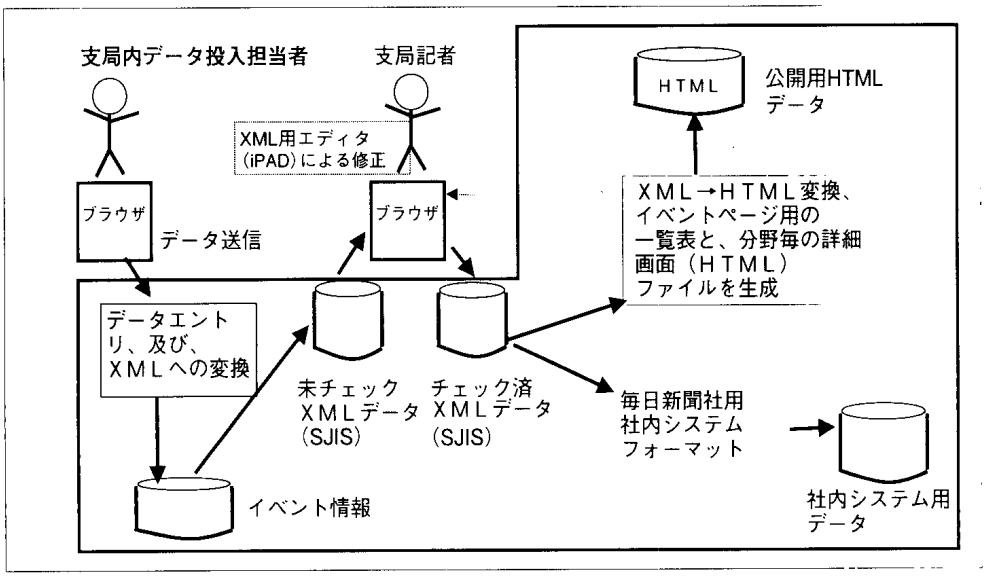

図２．イベント・システム・フロー図

4．3．2．オープンソースソフトウェアの利用

まず、使用するソフトウェアはほとんど全てを、オープンソース・ソフトウェアまたはフリーウェア、シェアウェアで整備した。ハードウェア環境は各自のパソコンがメインであった。

大量のデータを効率的に管理し、しかも、紙面とWebの双方へ出力したい。そのような要望を満たすためにデータ・フォーマットには「XML（eXtensible Markup Language）」を採用した。XMLでは、データ構造と出力形式が独立して管理できるので、一つのイベント・データを紙面用フォーマット、Web用フォーマットへと容易に出力可能である。（図3）

さらに、専門的プログラミング知識のない文系学生でもXMLを扱えるように、基盤となる開発環境にBXS（Baykit XML Server以下BXS）とXi（Extend it!以下Xi）を採用した。BXSは、「横浜ベイキット」の手によるオープンソースのXMLサーバであり、CocoonやFOP、SOAPなどを実装している。その特長は、「インストール・設定が容易」「XMLパーサやXSLTプロセッサを実装し、すぐにXMLを扱える」「マルチプラットフォーム」といったものになっている。また、「日本発」のソフトウェアであり、システム処理上での日本語が扱いやすいといった特徴を有していた。

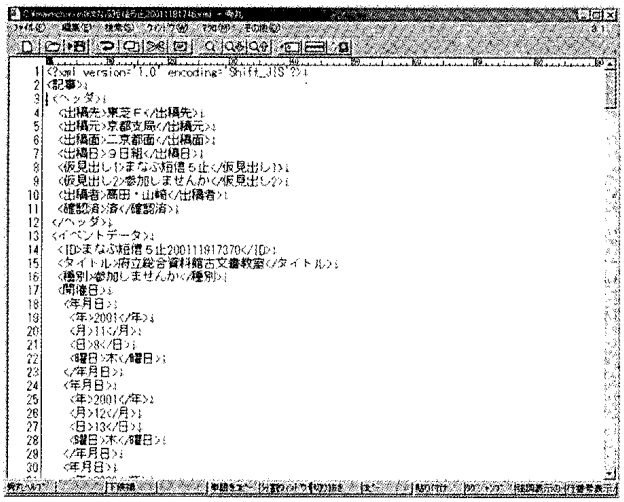

図3．XML化されたイベント・データ

一方 Xi は BXS サーバにおける XML を扱う為のプログラミング言語である。スクリプト言語であり、DOM などに比べ、プログラミング初心者にも理解しやすい構造になっている。

企業でのシステム開発では、このような「商品でない」ソフトウェアや、一般にまだ普及していないようなプログラム言語を利用することは少ない。その理由は、信頼性、使用できる人材数、参考マニュアル数などが少なければ、それだけ開発コストが大きくなるからである。つまり「無難な環境」で構築する。しかし、逆に産学連携だからこそ、まだ一般に普及していないツールやソフトウェアにチャレンジし、その有効性や性能を検証し、社会へ広く公開する役割を担わなければならない。

4．3．3．オープンソース的開発

イベントシステムの開発は、週数回程度のミーティングと、メーリングリストによる議論を中心として、業務分析、XML の仕様策定、などを決めながら随時プログラムを作成していった。各自の担当プログラムは決めていたが、あくまでも暫定的なもので、修正・変更は誰でもおこなえるようにした。

基本的な XML タグ構造、システムフローなどは参加者が実際に顔を合わせながら決めていったが、プログラミングはそれぞれ各自の環境で進めていった。パソコンへ BXS サーバの環境を構築し、ある程度できたプログラムから開発者全員へ配布し、機能追加、バグ修正を実施した。

作業を実施するなかでのシステムに対する各自のアイデアは、可能な限り採用していった。提案を採用するかどうかは基本的には全員のコンセンサスをとった。

プログラム開発でも、トップページ作成と同様に、作成できた段階までを頻繁に公開した。特にデータ入力ページのユーザインターフェースは、実際に使用する記者とメールや掲示板などを通じて頻繁に調整をとった。DTD 構造についても最初から厳密な仕様は決めず、ある程度システムが動作するようになった段階で、データ投入・テストを繰り返しながら必要・不必要タ

グを検討し構築した。

システム構築の間、ドキュメントはほとんど作成しなかった。ドキュメントの代わりとなったのが、頻繁なメール・掲示板での情報交換であった。疑問に思えば直ぐに問い合わせる。それに対して誰かが答える。それがインターネット上に蓄積され、作業指示書、システムフロー、マニュアルの役割をしていた。

以上のような開発プロセスの中から、Web インターフェースからイベント情報を入力し、データが XML 形式で蓄積。XML 化されたイベント情報を、紙面用フォーマットのテキスト・データ、Web ページ用の HTML データの両方を出力するシステムを構築した。（図４）

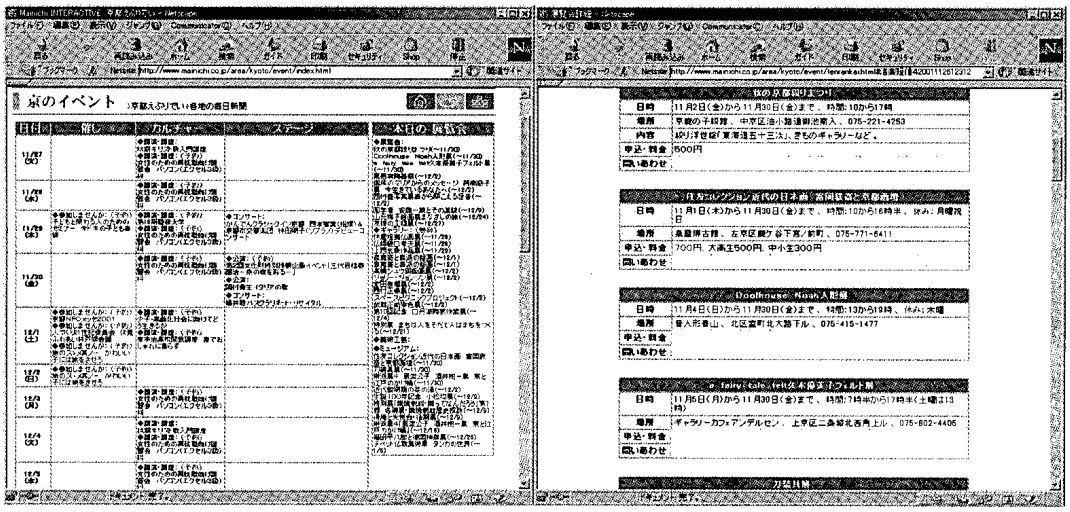

図４.「京のイベント」一覧ページと詳細ページ

５．考察

５．１．企業側からの考察

本研究では産学連携へオープンソース理論を適用し、毎日新聞社京都支局ホームページとそのメイン・コンテンツであるイベント情報を管理するシステムを構築した。本研究の成果を企業側の視点と大学側の視点の両面から考察する。

まず、企業側の成果であるが、１）新しいユーザ参加型システムでの構築、

2）ソフトウェア構築におけるセル生産方式の試み、が考えられる。

失敗するシステム開発の一つに、ユーザの意見を取り入れず、システム開発者だけの考えで構築されたものがある。完成はしたが、実際にユーザが利用しようとすると、日常業務とかけ離れた処理や複雑すぎる手順により、システム導入前よりも非効率的になってしまうことも多々ある。

そういった事柄を是正する為に、ユーザ参加型システム構築手法が採用される場合がある。一般的なユーザ参加型システム開発では、システム構築に必要な情報や要望を、そのシステムの利用者から吸い上げ、システムの機能へ反映させるものが多い。そのメリットとしては、よりユーザの希望に近いアウトプットが望めることである。しかし、その反面、ユーザの意見を全て反映させようとするとコスト高や開発期間延長になったりする。

一方、オープンソース的開発手法では、利用者が直接開発組織に所属する。つまり、「ユーザであり開発者でもある」ことを求められる。全員がプログラムを組むのではなく、他のユーザとの意見交換に関するコーディネート、多方面との折衝などを担当する。従来のシステム開発ではあまり重視されていない、「プログラム開発以外」の作業を担当することで、ユーザからの意見が妥当なものか、開発期間内に完了する要望なのか、をより詳細に把握できる。その結果、利用者の要望に近い、機能的にも妥当なアウトプットが期待できる。

また、産学連携であるため、学生・教員などの参加者は実際に商品やサービスを購入・利用する可能性がある「顧客」でもある。その意見や要望をシステムへフィードバックさせるメリットもある。

オープンソース的開発では、１人の人間が、ユーザとの意見交換、プログラミング、テスト、メンテナンスなどをいくつもの役割を担う。従来のシステム開発では、上位の要件定義はシステム・エンジニア、実際のコーディングは仕様書を見ながらプログラマ、テストは運用部門、メンテナンスは保守部門でおこなうといった分業制が主流であった。小規模の個人で開発をおこなうような場合では、全て１人が担当する場合もあるが、これは人的・資金

的に「やむをえない場合」であった。確かに、システム全体を１人で構築するのは無理がある。そこで、オープンソース的開発では、「ひとつのシステムを多数の人間で最初から最後まで作成する」システム開発の可能性を提示していると考えられる。多数の人間が共同作業で、仕様の決定から、プログラミング、テストまでを実施していく。その情報交換・共有の為のツールとしてインターネットの各種サービスを利用する。

従来のシステム開発においても、「多人数で最初から最後まで開発」という形態になっているといえよう。しかし、それはあくまでも全体を見た場合で、それぞれの担当個所に関しては、基本的には「１プログラム(システム)１人」である。サブルーチン化で担当割をしても、「自分の個所が終われば完了」である。システム全体の様々な「ソース（情報）」を参加者全員にオープンにし、複数の人間が同時並行的に開発していく。もし競合しても、その機能・性能を比較し良い方が正式に採用される。

「ひとつのプログラム（システム）を複数人で開発」するという形態は、工業製品を作成する方式のセル生産方式と似ている。セル生産方式とは、１人あるいは数人のチームで部品の装着から組み立て、検査まで製造の全工程を受け持つ生産方法のことである。生産体制の柔軟性が高く、多品種少量生産に適したものとして近年、ベルトコンベア方式に替わり多くの生産現場で採用されつつある。オープンソース的開発手法は、柔軟な仕様変更へ即座に対応し、高品質なソフトウェアを構築するといった、システム分野への「セル生産方式」の可能性も考えられる。

５．２．大学側からの考察

大学側からの成果としては、１）システム構築を通じての業務理解、２）専門知識の社会活用実践、が考えられる。

先にも述べたように、企業におけるシステム構築を実施するには、その企業における「物」「情報」「人」などの流れ、蓄積、活用方法を理解しなければならない。文献研究、ヒアリング・アンケートによる調査、などが企業

調査の主流であるが、さらに「システム開発」という企業調査・理解の方法論としての可能性も考えられる。小規模なシステムでも、対象とする企業の現状と課題を正確に把握し、それを解決する機能が求められる。そのようなプロセス全てが企業を理解する作業になっている。また、最近では、大学におけるシステム開発を「基礎研究」のひとつとして採用しようとする動きもでてきている[崎尾01]。企業研究に対する基礎研究のひとつとしても、システム開発が利用できると思われる。

産学連携では、企業の求める製品やサービスを、大学側の知識・技術を駆使して形にしていくことが求められる。オープンソース手法の場合、企業・大学は基本的には対等な立場である。学生や教員が今までに学んできた知識・技術を全て注ぎ込み、何かを作り上げていく過程に「知識の活用」方法を実践する。

本システムは当初、プロトタイプの構築で終了する予定であったが、2001年4月から毎日新聞社の正式システムとして採用された。

ただ「TOYプログラム」を作成するのではなく、「本物」のシステムを構築する。通常の授業ではなかなか受けることができない、自分達が作成したものが、社会の一部として機能しているという、責任感、満足感から多くを得ることができたといえよう。

6．まとめ

本研究では、産学連携の場面へ、ソフトウェア開発手法のひとつである「オープンソース理論」を用い、企業でのシステム開発を企業・大学の協同でおこなう手法を試みた。その具体的事例として、毎日新聞社京都支局ホームページ、イベント管理システム構築において、1）ユーザ参加型開発、2）インターネットの活用、3）報酬、4）バザール方式、といったオープンソースの特徴を適用した。

その結果、企業・大学協同でのシステム開発において、1）ユーザ参加型システムの一形態、2）ソフトウェア構築におけるセル生産方式の可能性、

3）システム構築を通じての業務理解、4）専門知識の社会活用実践、などが明らかになった。

今回は産学連携の場面にオープンソース理論を適用したが、まだ検討すべき項目は多数存在している。開発したシステムのメンテナンスの問題、著作権、特許、などに対する対応が必要になってくる。また、より具体的な産学連携での開発手法の検討も必要であろう。例えばオープンソース理論における「ユーザ参加」「柔軟な仕様変更」「頻繁に成果公開」などは、エクストリーム・プログラミング（eXtreme Programming : XP）の理論にきわめて近い。XP プログラミングは、近年注目されている小中規模のソフトウェア工学の方法である。XP で提唱されている「12のプラクティス」と呼ばれるものの中には，「常に２人がペアになってプログラミングを行う」「試行錯誤とテストを繰り返しながら制作していく」「顧客に開発チームにフルタイムで参加してもらう」「週40時間以上仕事をしてはいけない」といった開発手法が提唱されている。このような特徴は、産学連携連携におけるシステム開発にも適用可能ではないか。学生と企業の開発者がペアでプログラミングをする、試行錯誤を頻繁におこなう、といった開発パターンが考えられる。今後の展望としては、産学連携におけるシステム開発の場面における XP プログラミングの実践も試みていきたい。

引用・参考文献

[ESR98] Eric S.Raymond, "Homesteading the Noosphere" , http : //www. tuxedo. org /~esr/writings/homesteading/

[ESR97]Eric S.Raymond, "The Cathedral and the Bazaar" , http : //www. tuxedo. org /~esr/writings/homesteading/cathedral-bazaar/

[崎尾01]崎尾　健他「システム開発を媒介にした文科系の産学連携の可能性と範囲」，日本社会情報学会第16回全国大会研究発表論文集,2001年

[inoue01]井上明、猪狩淳一、金田重郎「文系研究科における実プロジェクト参画によるＩＴ教育」』，平成13年度情報処理教育研究集会講演論文集,2001

年。

[baykit]横浜ベイキット，http : //www.baykit.org

[XP]Ｒ・ジェフリーズ他「ＸＰエクストリーム・プログラミング導入編」，ピアソン・エデュケーション,2001年