

手のひらサイズの情報システム設計方法の一考察

—授業評価アンケート集計作業の支援システム—

A Study of Design Method of Small Scale Information System

槻木 公一*

要約

本学における授業評価アンケートの集計作業プロセスを事例として、小規模な組織体における部分的な業務を対象とした情報システムの設計手法を考察する。小規模な業務組織では、頻度が少なく少量で多様な業務を個々人が担当して非継続的、断片的に仕事を進めているので、全面的にプロセスを見直すことは現実的でない。むしろ、人との「なじみやすさ」をもって現行のワークフローの中に新しい情報システムを組み込んでいく必要がある。そのためには組み込む機能もこま切れにし、わずらわしい進捗管理をできるだけ避けるように設計することが肝要である。

1. はじめに

業務全体を対象としてビジネスプロセスを再構成して情報システムを構築する場合、担当組織、業務手順、業務内容、経費負担を全面的に見直すことができる。しかし、一部の業務のみを対象として小規模の情報システムを導入する場合、大幅な効率化や経費削減を狙うことはできない。むしろエンドユーザを単純作業や繰り返し作業から開放し、より創造的な業務に従事できる時間を確保して意欲の向上を狙うことが主たる導入目的となる。

小規模な業務組織においては、頻度が少なく少量で多様な業務を個々のユーザが担当しているのが現状である。情報システムを導入するとしても、対象となる業務はほんの一部であり、非継続的、断片的であることが多く、ビジネスプロセスの全面的な見直しは困難である。従って、情報システム導入後も、人の作業として考えて妥当で違和感の少ないプロセスとし、さらに断片的、非継続的に進められることが求められる。

ここでは、学生による授業評価の結果を集計して、冊子とWebページを作成する支援システムの実現を通して、小規模で使用頻度の少ない業務を対象とし、人となじみやすい情報システムの構築方法を考察する。

2. 授業評価の実施概要

本学では2004年度後期から2年間4学期に渡って、教育の質の向上および教授技能の向上に役立て、結果として学生の学習意欲を喚起することを目的として、学生による授業評価アンケートを実施した。今までは、迅速な集計処理やシステム変更の柔軟性、少ない運用経費などの長所があるWebによるアンケート収集システムを一部で実施していた。しかし、学生への協力要請の度合いによって大幅な回収率の差が生じ、加えて重複回答や類似回答とか未履修者からの回答などのノイズの大きさが問題となった。

的確な評価結果を得るためにはできるだけ多くの学生の協力が不可欠である。そのため、期末の授業時間内に、できるだけ学生の負担を軽減することを考慮して調査を実施することとした。必要最小限に絞り込んだ共通の評価項目に関しては、マークシート用紙によってデータを収集し、迅速な集計処理を行うようにした。また、教育改善のためには詳細な個別意見も大切であることから、書式を定めずに自由意見を収集して、担当教員が集計結果と併せてコメントとしてまとめ

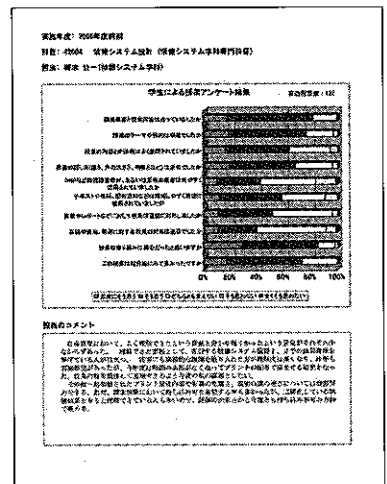


図2.1 評価結果の出力

*TSUKIGI, Koichi [情報システム学科]

ることとした。(図2.1) 協力した学生に伝えるために、これら評価結果とコメントはWeb上で学内公開している。

3. 支援システムの特徴とワークフロー構造

学生から2種類の評価シートを回収して集計する作業は、手作業としても複雑な作業ではなく、各科目毎に評価項目はPCで集計して結果のグラフ図を作成し、担当教員のコメントを添付して1ページにまとめる作業に過ぎない。ただ、半期に一度の作業であって百数十科目を対象として、非常勤講師を含む担当教員へ共通項目の集計結果を渡し、その後にコメントを回収しなければならない。すべてが揃ってから冊子出版のための製本作業と、公開のためのWebページ作成作業を行なうことになるが、本学のような小規模組織においては、一人の担当職員が日常の定例業務の合間にこま切りに作業を進めることになる。

こま切れ作業には当然ながら何らかの形の進捗管理が必要となる。人手作業の進捗管理の目的は、労力の無駄となる同一作業の繰り返しを防ぐことと、副作用のある作業の重複を防ぐことにあるが、進捗管理そのものがユーザの負荷をさらに増大させる。人による処理内容やワークフローと違和感の少ない方式にするとしても、進捗管理の部分についてはユーザ負荷を軽減する工夫が不可欠となる。

集計作業支援システムにおいては、取り掛かった作業を継続しなければならない最小単位を、こま切れに実行が可能な1つの作業単位 (Activity) と捉える。(図3.1)

UNIXにおけるプロセスモデルと同様に、全体としてはこのActivityを連結してstep by stepに処理を進めるワークフロー構造とする。それぞれのActivityの入力と出力の独立性を保持しておけば、出力内容からActivityの完/未完を判断することができる。さらに、Activityが完了した時点で入力を消去すれば、人手による書類の扱いと同様の進捗管理も可能となるが、何らかの修正・再処理を組み込むと進捗管理がより複雑になる。人手作業と違って、PC処理では同一処理を繰り返しても無駄な処理負荷が若干増えるだけである。すなわち、繰り返しても副作用のないActivityに関しては、意図的に進捗管理の対象外とすることができる。一方、データベースを更新するとか、外部にメールを送信するような副作用のあるActivityは重複実行が許されないので、進捗管理を避けることはできない。従って、全体として副作用のあるActivityをできるだけ少なくすれば、こま切れ処理が容易になって間違いも少なくなる。図3.2にActivityを連結した支援システムのワークフロー構造を示す。

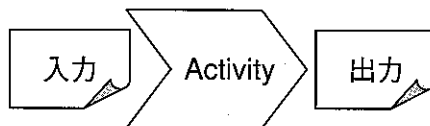


図3.1 作業単位 (Activity)

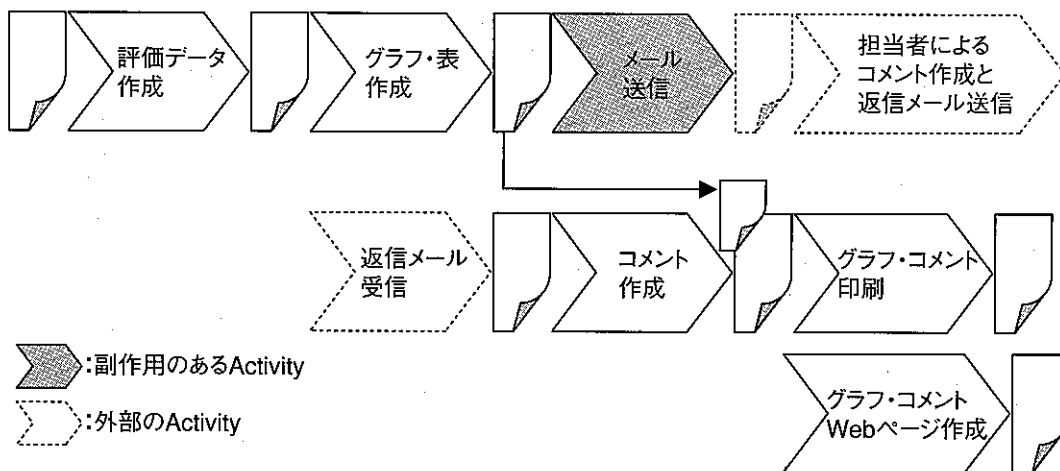


図3.2 Activityの連結

4. 支援システムの概要

図3.2に示したActivity単位で機能を作成し、入力と出力は原則として別のフォルダに格納することによって独立性を確保する。さらに、科目単位となる個々の入出力情報には科目番号と科目名を含む名称を付与することで、科目単位でそれぞれのActivityの完/未完を人が容易に判別できるようにして進捗管理を代替している。

(1) 副作用のないActivity群

(1-1) 評価データ作成

評価シートのマークから読み取った生データを評価項目・評価基準毎に集計し、科目単位のファイルに集約する処理である。入力データの修正変更は発生しないので、無条件に重複実行を可としている。

(1-2) グラフ・表作成

評価データから評価結果を図示するグラフと集計表を作成する。これも副作用のないActivityであり、無条件に重複実行を可としている。

(1-3) コメント作成

返信メールからコメント情報を作成する。送信メールに付加した持ち回り情報と、所定の区切り行の間に書かれたコメントを自動的に切り出して、科目毎のコメント情報を作成出力する。コメントの作成には人手作業が入るため、コメント内容の修正作業が必要となる。返信メールからコメントを切り出して、修正した結果をそのまま出力すると、このActivityは重複実行ができなくなって進捗管理が必要となる。この進捗管理を避けるため、コメントの修正は入力情報に対して行い、再度コメント作成を行なう方式とした。内容の修正以外にも、使われるメールの種類によって返信されたコメントの行内文字数が異なるケースが多く、コメントを整形する修正作業が予想以上に多発したので、進捗管理を避けたメリットは大きい。

さらに、通常のメールやメディア媒体、紙媒体などでコメントが提出されるケースもあり、いずれも擬似的な返信メールを作成して入力情報に加え、このActivityを副作用のないものとした。(図4.1)

(1-4) グラフ・コメント印刷およびグラフ・コメントWebページ作成

科目毎の表示グラフとコメントを編集し、印刷用のページとWebページを同一の書式で作成する。入力情報の修正は必要なく、重複実行が可能なActivityである。

(2) 副作用のあるActivity

(2-1) メール送信

集計結果のグラフと集計表を添付し、科目名、担当者情報などを持ち回り情報として付加したメールを自動作成して送信する。メールを受信した担当者

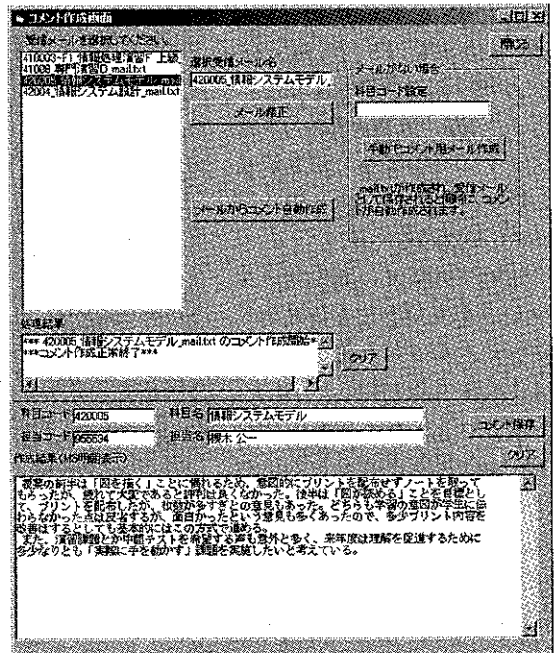


図4.1 コメント作成画面

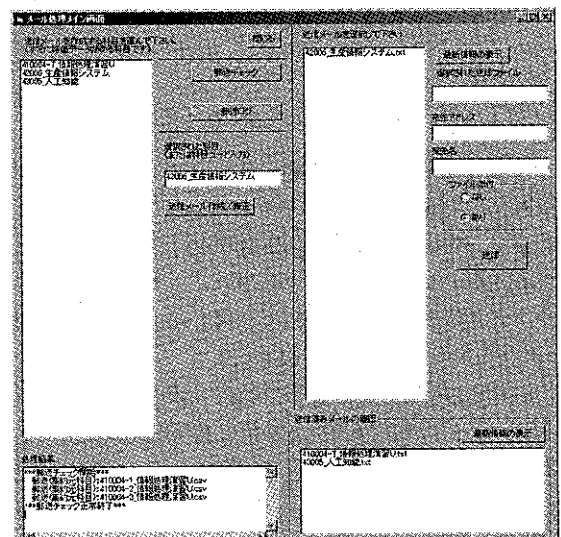


図4.2 メール送信画面

は、添付された評価結果を参照してコメントを作成して返信する。従って、一回の送信処理は外部に影響を与え、副作用のあるActivityとなるので、進捗管理が不可欠となる。

このシステムでは、送信は1件ごとに目視確認して送信することに加え、同一担当者へ再度送信する場合にはさらなる確認動作を求めるようにした。受信メールを喪失した担当者へ再送信するという強制的な重複実行を可能とするために、送信時刻を付加した送信メールを保存している。(図4.2)

(3) その他

(3-1) 作業環境設定

評価は学期毎に複数回実施するために、評価対象学期の選択や各学期ごとに各Activityの入出力情報を格納するフォルダの生成(初回起動時のみ)、プリンタの選択など作業環境を設定する。

(3-2) 一括処理

個別の科目毎に副作用のないActivityを実行する個別処理に加え、揃った入力情報すべてについて繰り返してActivityを実行する一括処理を実装した。副作用のあるメール送信は一括処理からは除外してある。(図4.3)

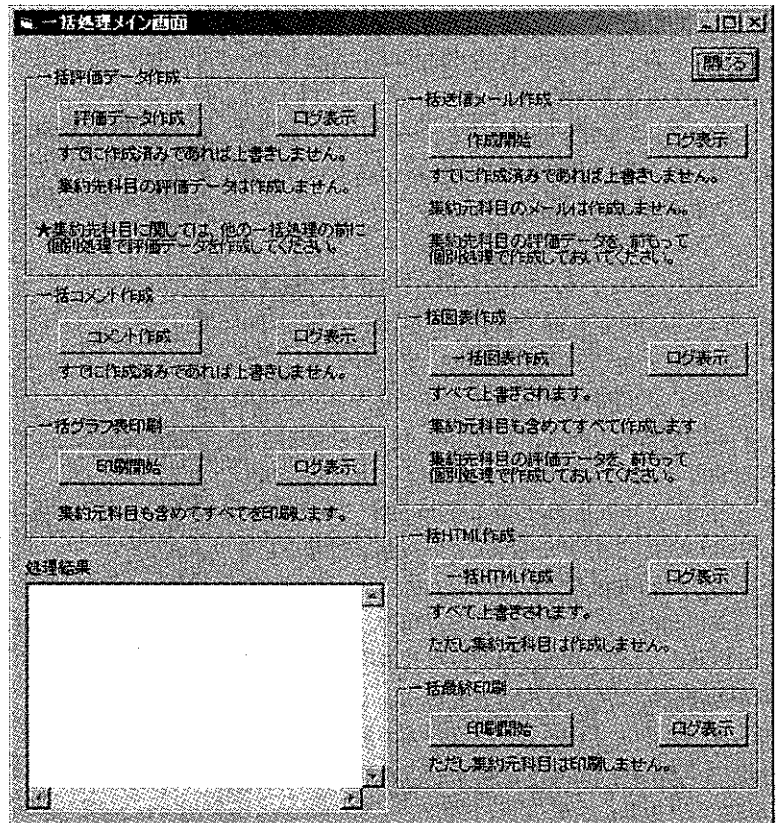


図4.3 一括処理画面

5. おわりに

この支援システムはVisual Basic、Excel、Notepadを利用して作成し、2年間4回の集計作業に活用された。演習科目や語学科目では、複数の教員による同一科目の授業、あるいは同一教員による複数クラスの担当などがあり、複数の評価結果を集約する科目もある。また、講義科目とは異なった評価項目の表記が求められる科目もある。これらに関連するシステムデータはすべてExcelで作成し、そのまま利用することでデータメンテナンスを容易にした。さらに、図4.3に示すように、操作画面に注意事項を直接記載することによって操作マニュアルの類を不要としている。

小規模の組織において一時的に人手を増やせば対処できるような場合は、PCを利用するとしても市販のOAソフトウェアを部分的に利用する形態が多い。汎用品ゆえに市販パッケージをそのまま多様な業務プロセスに適合させることには無理もあり、適合する部分だけ利用して後は人手作業で補うことになる。これでは他とは異なる特色ある業務は、いつまでもたっても情報化の恩恵を受け難く取り残されてしまうことになる。

従って、小規模な情報システムにおいては、より一層人の作業となじみやすく円滑に導入できる設計手法を確立し、既存のソフトウェア部品を活用して低コストで迅速な開発を実現していく必要がある。