

筑波大学大学院博士課程

システム情報工学研究科特定課題研究報告書

修士論文発表会

スケジューリングシステムの開発
ースケジューリングアルゴリズムの開発ー

成川 新吾

(コンピュータサイエンス専攻)

指導教員 三末 和男

2009年 3月

概要

本報告書は筑波大学システム情報工学研究科コンピュータサイエンス専攻「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」における特定課題研究の成果をまとめたものである。

コンピュータサイエンス専攻では毎年 2 月に修士論文発表会を行っている。この発表会のスケジュールはスケジュール担当者が手作業で作成しているため、非常に手間がかかっているのが現状である。報告者らのプロジェクトは 2008 年度のスケジュール担当者から依頼を受け、修士論文発表会スケジューリングシステムの開発を行った。開発したスケジューリングシステムにより、スケジュール作成の手間を軽減することができ、業務の効率化を図ることができた。

報告者はプロジェクトにおいて、スケジューリングアルゴリズムの開発を担当した。アルゴリズムを考案する上で、必ず満たす必要がある項目と可能な限り満たすべき項目があった。可能な限り満たすべき項目の中で、同一の審査員が連続して審査を担当するという項目を満たすために、グルーピングとグループレイアウトという手法をとった。主査グルーピング、内部グルーピング、親密グルーピングという 3 種類のグルーピングを行うことによって、同一の審査員の連続性を高めたグループを作成することができた。それらのグループをスケジュールテーブルに配置していくことで、連続性を高めたスケジュールにすることが可能となった。しかし、アルゴリズムを考案し開発を進めていく中で、2008 年度のデータでは親密グループのサイズが大きくなりすぎ、スケジューリングに失敗するという問題が発生した。その問題に対し、親密グルーピングを行うか否かを利用者を選択させるという対策を講じた。結果として、開発したスケジューリングアルゴリズムを実装したシステムで 2008 年度のスケジュールを実際に作成することができた。

目次

第1章	緒言	1
第2章	スケジューリングシステムの開発	2
2.1	開発体制と開発経過	2
2.2	業務内容の見直しとその改善	4
2.2.1	これまでの業務の流れ	4
2.2.2	これまでの業務の問題点	7
2.2.3	システム化の範囲	7
2.2.4	システム化後の業務の流れ	8
2.3	スケジューリングシステムの概要	9
2.3.1	システム要件	9
2.3.2	システム構成	13
2.3.3	開発したシステムの流れ	15
2.4	結果と考察	24
2.4.1	スケジュール担当者の作業負担の変化	24
2.4.2	審査員の作業負担の変化	25
2.4.3	今後の課題	26
第3章	スケジューリングアルゴリズムの開発	28
3.1	スケジュールの定義	28
3.2	スケジューリング時の考慮点	28
3.3	スケジューリングアルゴリズム	29
3.3.1	スケジューリング可否判定	30
3.3.2	グルーピング	31
3.3.3	グループレイアウト	34
3.4	アルゴリズムの問題点と改良	39
3.5	スケジューリングアルゴリズムにおける今後の課題	44
3.5.1	特定課題研究の学生の扱い	44
3.5.2	審査員の連続性	45
第4章	結言	48
謝辞		49

目次

図 2.1 開発スケジュール.....	3
図 2.2 これまでのスケジュール作成業務フロー	5
図 2.3 これまでのスケジュール調整業務フロー	6
図 2.4 改善後のスケジュール作成業務フロー	8
図 2.5 スケジュールリングシステムのユースケース図	10
図 2.6 ネットワーク構成概要.....	13
図 2.7 スケジュールリングシステムのアクセス認証.....	14
図 2.8 スケジュールリングシステムの画面遷移図	15
図 2.9 審査員用メニュー画面.....	16
図 2.10 主査情報入力画面.....	17
図 2.11 学生・副査情報入力画面	18
図 2.12 審査不可時間入力画面.....	19
図 2.13 スケジュール担当者用メニュー画面	20
図 2.14 審査情報照会画面.....	21
図 2.15 スケジュール条件入力画面.....	22
図 2.16 ダウンロード画面.....	23
図 3.1 スケジュールテーブル.....	31
図 3.2 主査グルーピング前	32
図 3.3 主査グルーピング後	32
図 3.4 内部グルーピング前	33
図 3.5 内部グルーピング後	33
図 3.6 親密グルーピング前	34
図 3.7 親密グルーピング後	34
図 3.8 親密グループ A 挿入位置探索	35
図 3.9 親密グループ B 挿入位置探索	35
図 3.10 主査グループ挿入位置探索.....	36
図 3.11 内部グループ挿入位置探索.....	36
図 3.12 内部グループ挿入位置探索（審査員重複確認 1）	37
図 3.13 内部グループ挿入位置探索（審査員重複確認 2）	37
図 3.14 スケジュールリング済みスケジュールテーブル	38
図 3.15 親密グループ.....	40
図 3.16 親密グループ挿入済みスケジュールテーブル	40
図 3.17 主査・内部グループ	41
図 3.18 主査・内部グループ挿入済みスケジュールテーブル.....	41
図 3.19 内部グループ作成例	46

表目次

表 2.1 開発体制.....	2
表 2.2 提供するサービスとレスポンスタイム.....	11
表 2.3 端末基準環境.....	11
表 2.4 保証する保持データ量.....	12
表 2.5 ハードウェア構成.....	14
表 2.6 ソフトウェア構成.....	14
表 2.7 スケジュールの初版作成までにかかった時間	24
表 2.8 審査情報登録時の負担.....	25
表 2.9 システムの使い方の直観的な分かりやすさ	26
表 2.10 システムの操作のしやすさ.....	26
表 3.1 スケジュールリングの入力情報と出力情報	28
表 3.2 スケジュール条件.....	30
表 3.3 親密グループ作成・未作成の利点と欠点	42

第1章 緒言

本報告書は、筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻(以下、CS 専攻とする)で実施している「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」(以下、高度 IT 専修プログラムとする)における特定課題研究の成果をまとめたものである。ここで特定課題研究とは、「高度 IT 専修プログラム」の科目“研究開発プロジェクト”において、カテゴリ 3 として分類されるもので、教員等からのリクエストに基づき、原則としてチームで、実際に動くシステムをプロジェクト形式で開発するものである。

この中で、報告者を含む 3 名のチームは「修士論文発表会スケジュール調整ツール」の開発リクエストを受け、システム開発プロジェクト(以下、本プロジェクトとする)に取り組んだ。このプロジェクトでは、最優先の目的を「スケジュール作成業務における人的作業を省力化すること」と設定し、スケジュール作成業務プロセスの見直しとスケジューリングシステムの開発を実施した。本プロジェクトの開発期間は、2008 年 9 月から 2009 年 2 月までの約半年間である。

報告者はこのプロジェクトにおいてスケジューリングアルゴリズムの開発を担当した。スケジューリングアルゴリズムはシステムの核であり、アルゴリズムを作成しなくてはスケジューリングができないためシステムが成り立たない。そこで、「スケジュールの条件を緩めさえすれば必ずスケジューリングが成功する」ということを、アルゴリズム作成において必ず達成しなければならない目標に設定することにした。そして、スケジューリングを行う上で考慮しなければならない点を洗い出し、その考慮点を満たし、修士論文発表会に適したアルゴリズムを考案、実現することに努めた。

本報告書は以下の構成をとる。第 2 章では、まず開発体制と開発経過について述べた後、スケジュール作成業務におけるこれまでの業務の流れを分析し、その問題点の抽出と解決方法について述べる。更に、開発したシステムの概要について説明し、システムに関する評価と今後の課題について述べる。第 3 章では本プロジェクトにおいて報告者が担当したスケジューリングアルゴリズムの開発の内容について述べる。ここではまず、第 3 章で扱う「スケジュール」を定義し、スケジューリング時に考慮しなければならない点について説明する。次に、開発したスケジューリングアルゴリズムについて説明する。アルゴリズムの流れに沿って、スケジューリング可否判定、グルーピングとグループレイアウトの方法について示す。そして、開発段階で発生したアルゴリズムの問題点とそれに対して実際にとった対策について述べる。章の最後にはスケジューリングアルゴリズムの今後の課題について検討する。第 4 章では本報告書のまとめを行う。

第2章 スケジューリングシステムの開発

本プロジェクトで修士論文発表会スケジューリングシステムを開発するに当たり、まず、スケジュール作成業務プロセスを見直し、システム化する範囲を限定した。その後、スケジューリングシステムの設計・開発を行い、今年度の CS 専攻修士論文発表会スケジュール作成時のシステム運用を担当した。本章では、本プロジェクトの開発体制と開発経過について述べた後、スケジュール作成業務プロセスの改善の過程を示し、これを実現するために開発したシステムの概要について説明する。そして、そのシステムを運用した結果をもとに、本システムの効果と課題について考察する。

2.1 開発体制と開発経過

本プロジェクトのチームメンバとその担当を表 2.1 に示す。プロジェクトマネージャの栗山は、プロジェクト推進の責任者としてプロジェクトをコントロールし、プロジェクトの目的を達成する。スケジューリングアルゴリズム開発責任者の成川は、スケジューリングアルゴリズムの開発において先導的な立場に立ち、アルゴリズムの考案、実装に責任を持つ。システム設計責任者の篠崎は、システム設計を主導し、拡張性と保守性を考慮したシステム設計に責任を持つ。

本プロジェクトでは、ウォーターフロー型の開発プロセスに沿ってスケジュールを立案した。ここで、同開発プロセスでは以下の工程が含まれる。まず、要求定義工程では、業務内容の見直しとその改善を行い、システム化する部分や開発するシステムが満たすべき要件を定義する。外部設計工程では、要求定義工程で決定した内容を詳細化し、実際に提供する機能やそれらを実現するための画面・データなどを定義する。内部設計工程では、外部設計工程で定義した機能を実装時にどう実現するかを検討し、これをもとに実装工程でシステムを実装する。そして、実装したシステムが要求定義工程・外部設計工程で定義した仕様を満たしているかどうかを、テスト工程で確認する。定義した品質が満たされていると確認できれば、開発したシステムを委託元教員に納品する。

表 2.1 開発体制

メンバ	担当
栗山 大	プロジェクトマネージャ
成川 新吾	スケジューリングアルゴリズム開発責任者
篠崎 龍夫	システム設計責任者

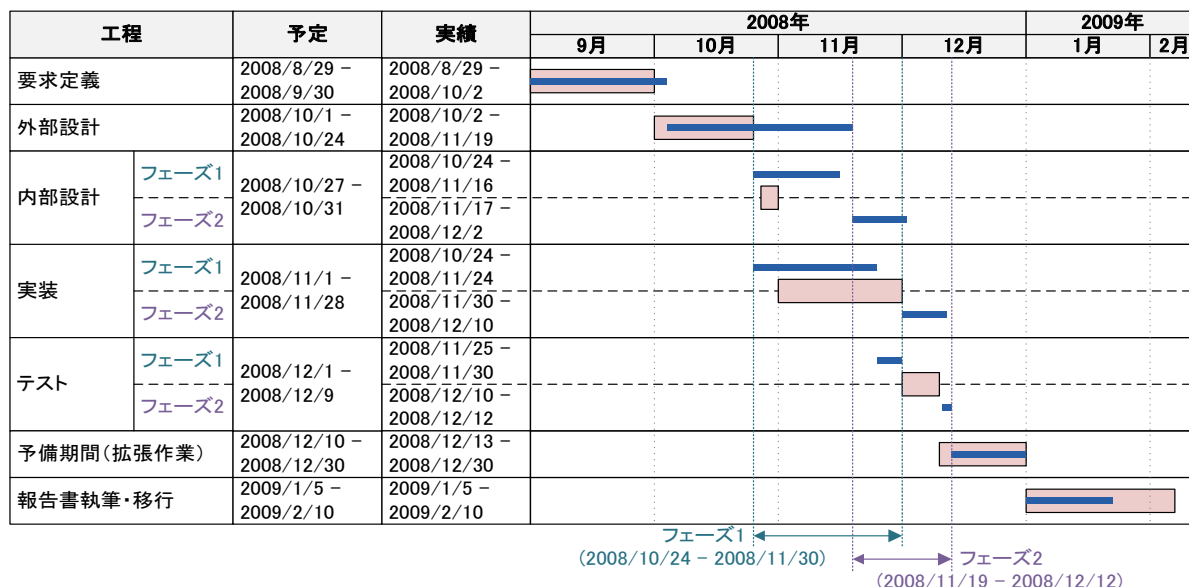


図 2.1 開発スケジュール

本プロジェクトの開発スケジュールを図 2.1 に示す。図中の赤線は当初に計画したスケジュールを、青線は実績を示している。この図に示す通り、開発したシステムを今年度の修士論文発表会のスケジュール作成に利用することを想定し、2008 年 12 月までに全ての開発が完了するスケジュールを立案した。このスケジュールの中には、テスト期間完了後に予備期間を 3 週間程度配置している。これは、要求変更やチームメンバーの経験不足といった要因によって開発スケジュールに遅延が発生することが想定されたため、それを回復するための予防措置である。

このスケジュールに従って開発を進めたところ、要求定義工程はほぼ予定通りに作業を進めることができた。しかし、外部設計工程でシステムの稼働開始時期が早まることと決定し、開発したシステムを今年度の修士論文発表会のスケジュール作成に利用してもらうためには、12 月 1 日までに機能の一部を、12 月 15 日までに全ての機能を実現する必要に迫られた。そこで、内部設計工程からテスト工程までを 2 つのフェーズに分割し、前半のフェーズ（以下、フェーズ 1 とする）で 12 月 1 日までに必要な機能を、後半のフェーズ（以下、フェーズ 2 とする）でそれ以外の機能を実装することにした。その結果、両方の稼働開始時期までに必要な機能の開発を間に合わせることができ、システム全体のテスト完了時期も予備期間内に収めることができた。その後の予備期間では、実際にシステムを使用して分かったシステムの問題点の改善や微修正を行い、12 月末の時点で修正したシステム全体のテストを完了している。

2.2 業務内容の見直しとその改善

業務内容を把握してその効率を改善するため、これまでの修士論文発表会スケジューリング業務の流れとその問題点を抽出した。本節では、これまでの業務の流れについて述べ、システム化による問題の解決方法とシステム化後の業務の流れについて説明する。

2.2.1 これまでの業務の流れ

これまでの修士論文発表会スケジューリング業務において、スケジュール担当者が実施する業務は 2 種類存在する。1 つは学生情報や審査員情報を収集し、それをもとにスケジュールを作成する「スケジュール作成業務」、もう 1 つはスケジュール作成後に発生した審査員からのスケジュール調整要望に対応する「スケジュール調整業務」である。

スケジュール作成業務の業務フローを図 2.2 に示す。まず、スケジュール担当者は審査員に対し、修士論文の審査情報を登録するよう依頼する。依頼を受けた審査員は担当事務員に対し、自身が担当する審査に関する審査情報を送付する。担当事務員は審査員から届く審査情報を取りまとめ、全ての審査情報を集計する。その後、担当事務員は集計済みの審査情報をスケジュール担当者へ送付する。この間、スケジュール担当者は発表会当日に使用可能な部屋を調査し、予約しておく。そして審査情報を受け取った後、スケジュール担当者はスケジュールを作成する。

次に、スケジュール調整の業務フローを図 2.3 に示す。スケジュール担当者から配布されるスケジュールを受け取った審査員は、スケジュールに不備がないか確認する。ここで不備が見つからなかった場合、業務は終了となる。一方、スケジュールに不備があった場合、審査員はスケジュール担当者にスケジュールの調整を依頼する。調整を依頼されたスケジュール担当者は、その内容をもとにスケジュールの調整を行う。スケジュール調整が完了した後、スケジュール担当者は審査員に再度スケジュールを配布し、スケジュールに不備がなくなるまで以上の調整を繰り返す。

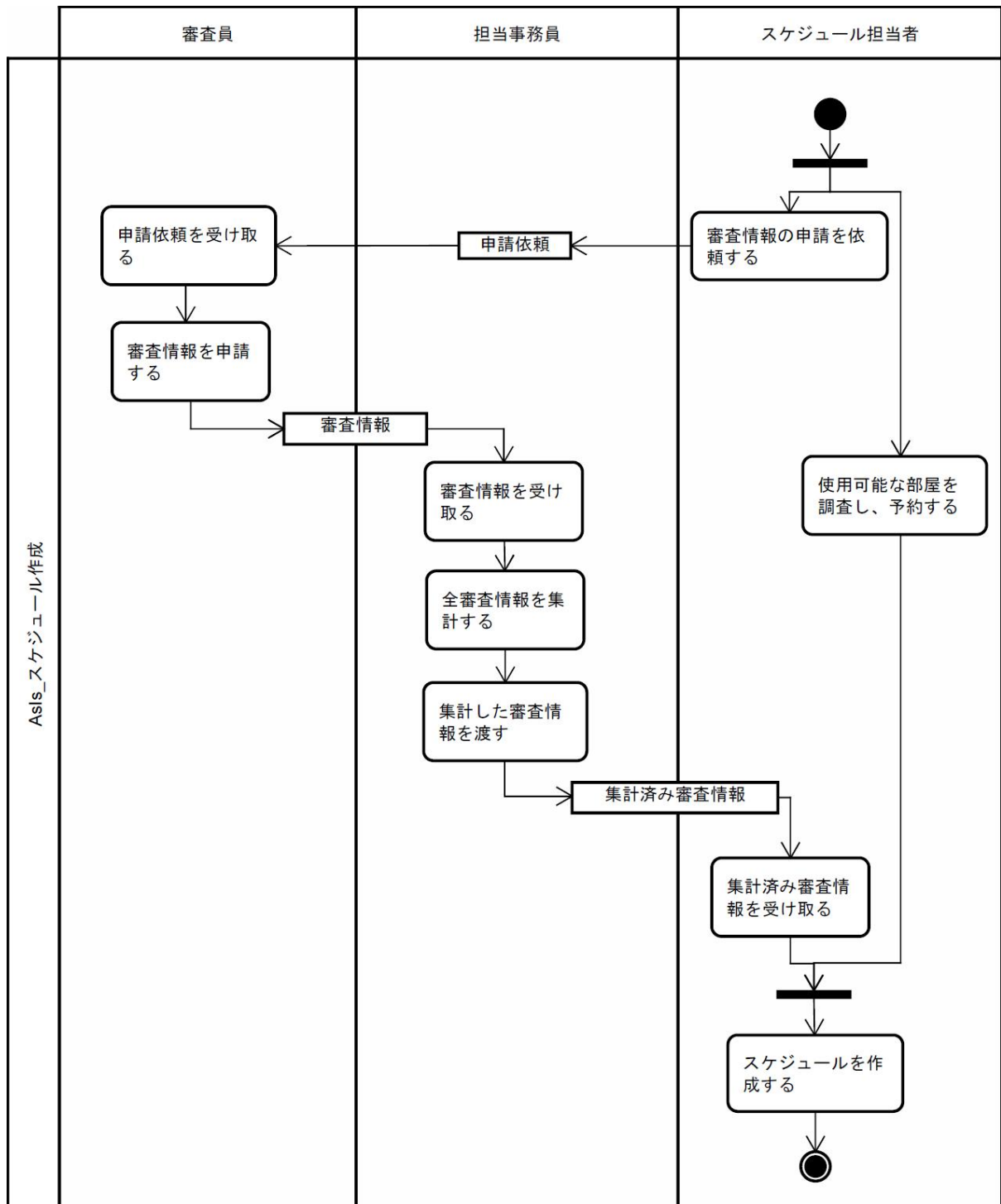


図 2.2 これまでのスケジュール作成業務フロー

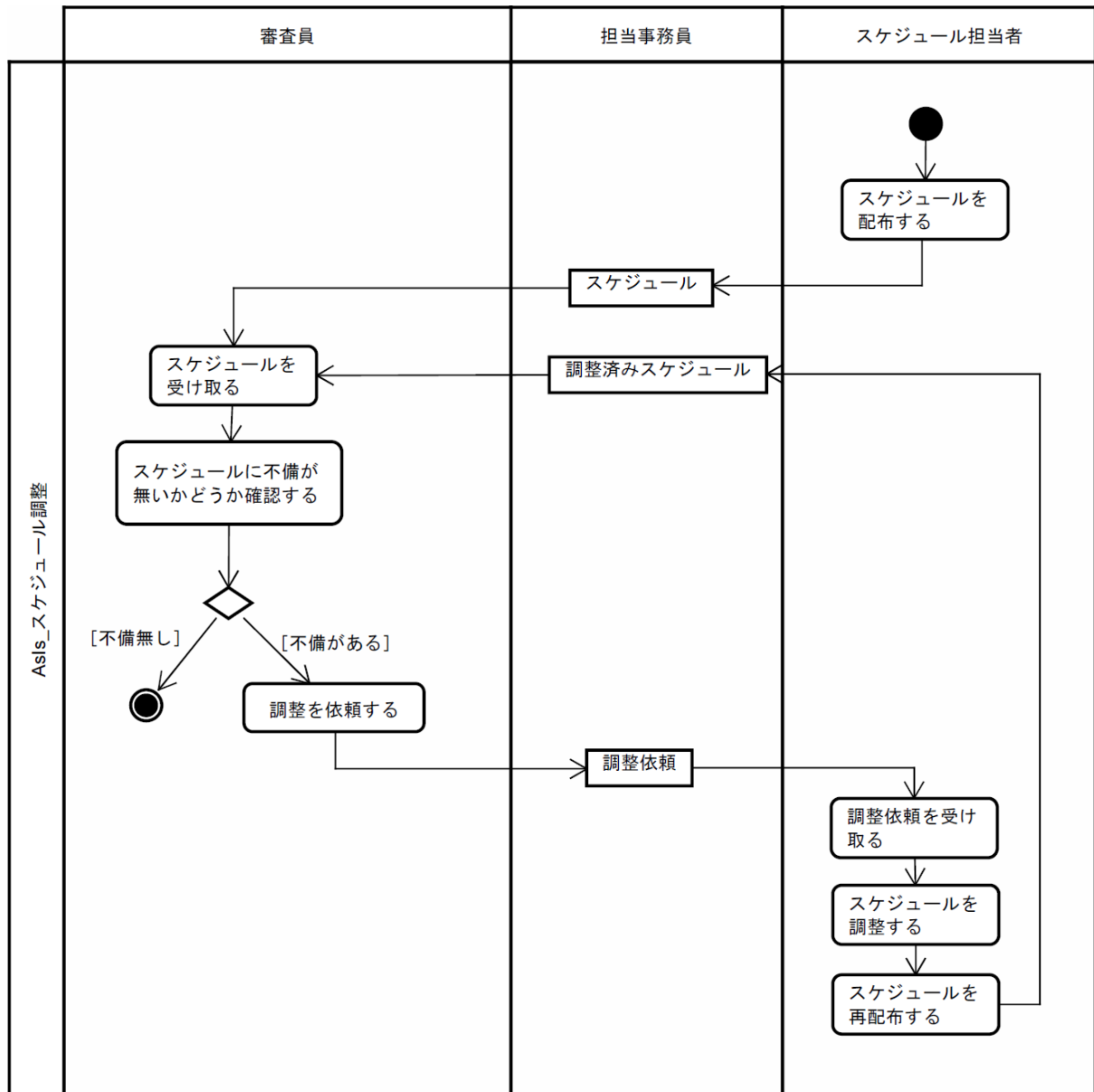


図 2.3 これまでのスケジュール調整業務フロー

2.2.2 これまでの業務の問題点

これまでの業務には大きく 2 つの問題があった。1 つ目の問題は、スケジュール作成業務の中で審査情報の集計が煩雑なことである。修士論文発表会では毎年約 80 の審査が行われる。これらの審査情報の取りまとめは担当事務員が行っていた。しかし、審査員が担当事務員へ送付する審査情報の内容は指定されていても、書式フォーマットは指定されていなかった。そのため、担当事務員は審査員ごとに表記が異なる審査情報から統一された形式に変換しなくてはならなかった。これと同様の問題が、スケジュール調整業務の中で審査員がスケジュール担当者へ送付する調整情報においても発生していた。

2 つ目の問題は、スケジュール作成の作業に時間がかかることである。CS 専攻では、毎年 2 月に修士論文発表会を行っている。この発表会では 80 名以上の CS 専攻の修士学生の審査が行われる。審査を担当するのは審査員である主査 1 名と副査 2 名以上の教員であり、1 つの審査を構成する学生 1 名とその審査員のことを審査委員会と呼ぶ。また、CS 専攻では全ての審査を 1 日で行うことが慣例となっている。これを実現するため、発表会当日は審査用の部屋を複数用意し、複数の審査を同時進行で実施する必要がある。このような状況で発表会のスケジュールを作成するためには、いくつか考慮すべき点がある。まず、複数の部屋で審査を同時に実施するため、同一時間帯に審査員が複数の審査を担当してはならない。また、発表会当日に審査不可能な時間帯（以下、審査不可時間とする）がある審査員は、その審査員が担当する審査をそれ以外の時間帯に配置する必要がある。さらに、審査員の便宜上、同一の審査員が複数の審査を担当する場合、可能な限り連続して審査を実施できるようにスケジュールを組むべきである。各部屋の審査に関しても、最終的に可能な限り早い時間に全審査が終了するようになっていくことが望ましい。これらの点を考慮して発表会のスケジュールを組むことが求められるため、スケジュール担当者はこの作業に多大な時間と労力を費やしている。

2.2.3 システム化の範囲

業務効率を高めるためには、上述の 2 つの業務全体をシステム化の範囲とすべきである。しかし、約半年間という短期間で、3 名のチームが双方の業務の設計・開発を行い、品質を確保するのは難しいと判断された。一方、委託元教員の依頼目的は、スケジュール作成にかかる時間を短縮することであった。そこで、スケジュール作成業務のみをシステム化の範囲とし、スケジュール調整業務についてはシステム化の範囲外とすることを委託元教員と確認した。スケジュール調整業務は、これまでと同様の方法で実施した。

2.2.4 システム化後の業務の流れ

これまでの業務フローで抽出された問題点を考慮してシステム化を行うことで、スケジュール作成業務は図 2.4 のように改善することができる。まずスケジュール担当者は、システム内に保存されている前年のスケジュールリングに使用したデータを削除する。その後、スケジュール担当者は審査員に審査情報の登録を依頼する。依頼を受けた審査員は、システム上で審査情報を登録する。これらの一連の流れと並行して、スケジュール担当者は発表会当日に使用可能な部屋を調査・予約する。そして全ての審査情報が登録された後で、スケジュール担当者はシステムにスケジュールリングの実行を指示する。

この業務フローではシステム上で審査情報の登録を受け付けるため、担当事務員の介在が無くなり、審査情報収集や集計の手間が省ける。また、スケジュールリングはシステム上で行うため、スケジュール担当者は簡単な操作でスケジュールリングを行うことができる。

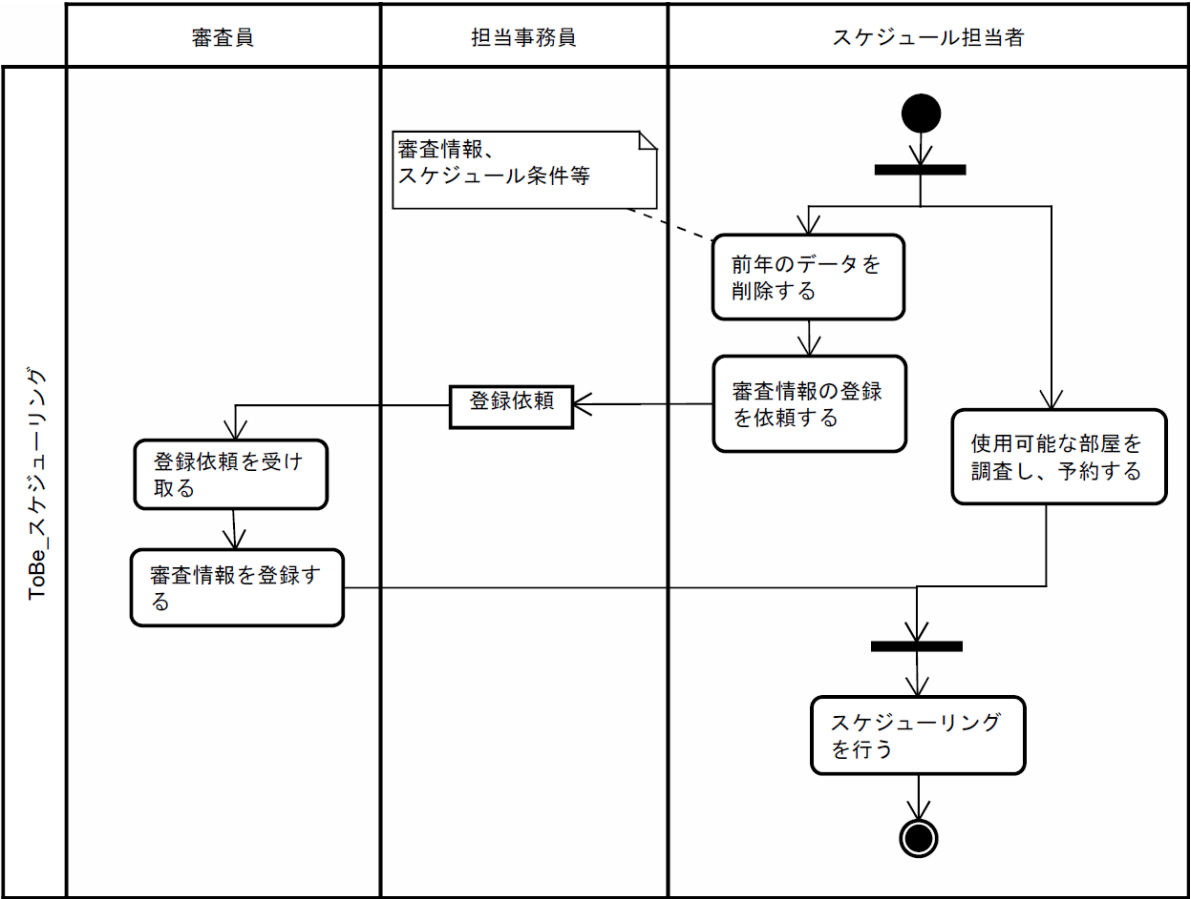


図 2.4 改善後のスケジュール作成業務フロー

2.3 スケジューリングシステムの概要

図 2.4 で示したスケジュール作成業務フローを実現するため、まずシステムに求められる要件を定義した。そして、そこで定めた要件を満たすシステムを設計し、その設計に基づいてシステムを実装した。本節では、要件定義、外部設計の工程で委託元教員と確認したシステム要件と、それを実現するためのシステム構成について説明する。その後、内部設計、実装、テスト工程を経て開発したシステムの効果や課題について検討する。

2.3.1 システム要件

以下では、本プロジェクトで開発したスケジューリングシステムの要件について述べる。

機能要件

スケジューリングシステムでは 4 つの機能を実装している。機能の概要を示すユースケース図を図 2.5 に示す。また、これらの機能の概要は以下の通りである。

- 審査情報を登録する
審査を構成する主査、副査および学生の情報を登録する。
- 審査情報を確認する
登録された主査、副査の情報および学生の情報を確認する。
- スケジューリングを行う
登録された情報とスケジューリング条件をもとにスケジュールを作成する。
- スケジュールをダウンロードする
「スケジューリングを行う」で作成したスケジュールをダウンロードする。

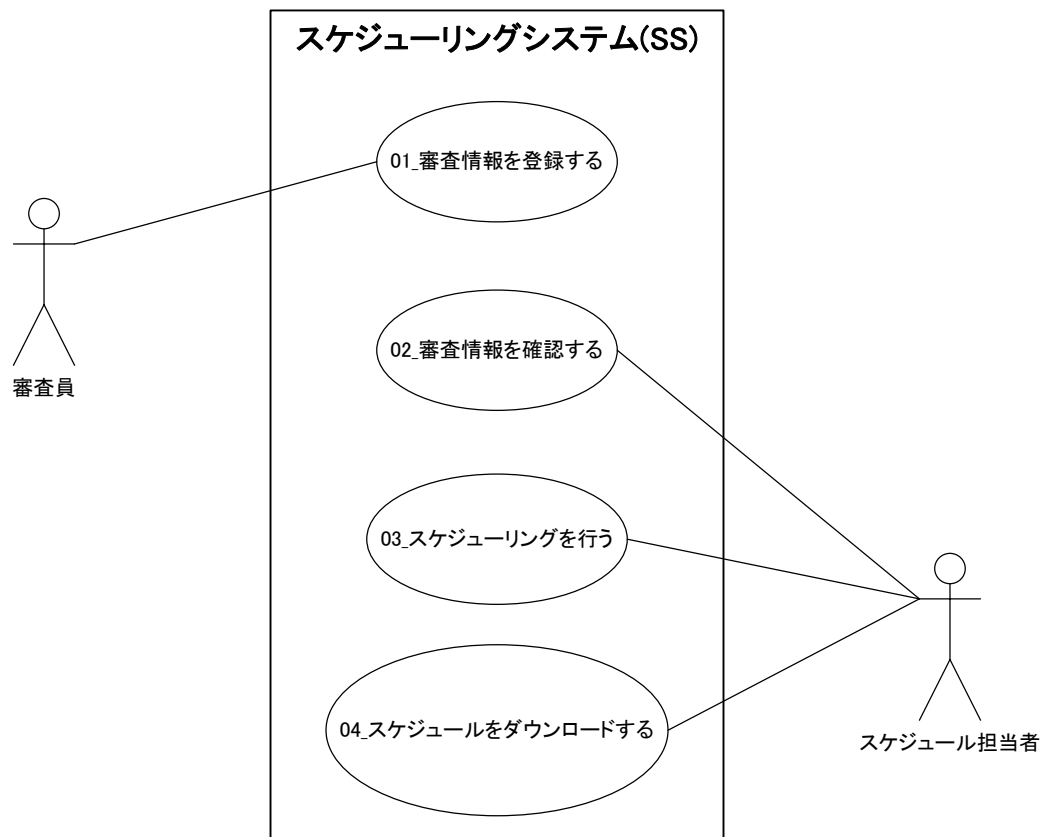


図 2.5 スケジュールリングシステムのユースケース図

データベース設計要件

スケジュール作成業務を実現するため、本システムではデータベース上に以下に示す4つのテーブルを作成している。

① 審査委員会情報

学生・審査員で構成される審査委員会に関する情報を格納するテーブル。審査委員会の構成要員のデータを保持する。

② 審査員情報

審査員に関する情報を格納するテーブル。審査員の氏名や職位、審査不可能な時間帯といったデータを保持する。

③ 学生情報

審査を受ける学生に関する情報を格納するテーブル。審査を受ける学生の学籍番号、氏名、論文題目に加え、その学生を審査する審査員に関する情報などを保持する。

④ スケジュール条件

スケジュールリングの初期値として設定した条件を格納するテーブル。使用する部屋数、部屋の使用可能時間、各審査の制限時間、審査間の休憩時間、スケジュールリングの結果といったデータを保持する。

サービスレベル要件

システムが提供するサービスに関する要件は以下の通りである。

- サービス提供時間

システムの稼働期間は毎年 12 月から翌年の 2 月までの 3 ヶ月間とする。この期間はサーバのメンテナンスや停電のような非常事態を除き、システムは 1 日 24 時間稼働し続けるものとする。

- レスポンスタイム

システムがサービスを提供する際に発生するレスポンスタイムの許容値を表 2.2 に示す。ただし、これらのレスポンスタイムは表 2.3 に示す端末基準環境を基準としたもので、それを下回る環境では保証外とする。

表 2.2 提供するサービスとレスポンスタイム

サービス	レスポンスタイム
画面遷移	3 秒以内
データベース照会表示	3 秒以内
スケジューリング実行時間	1 時間以内

表 2.3 端末基準環境

環境	基準
CPU	Pentium3 以上
搭載メモリ	500MB 以上
OS	WindowsXP 以上
ブラウザ	IE6.0 以上、FireFox2.0 以上

キャパシティ要件

システムが満たすキャパシティに関する要件を以下に示す。

● 保持データ量

システムでは表 2.4 に示すデータ量を保持し、システムが許容することを保証する。表中の「スケジュール」という項目は、本システムによって作成されたスケジュールのことを示している。参考として、2008 年度修士論文発表会のスケジューリングを行う際に必要となった実データ量を同表に併記する。ここで「スケジュール条件」が保証データ量を超過しているが、システムは正常に動作した。

表 2.4 保証する保持データ量

対象データ	保証データ量	2008 年度実データ量
審査員情報	80 名分	70 名分
学生情報	120 名分	98 名分
スケジュール条件	10 件	32 件
スケジュール	10 件	8 件

● インプット数

システムへインプットされる情報は、審査員が入力する審査情報およびスケジュール担当者が入力するスケジュール条件のみである。システムは審査情報の入力を 1 日 100 件まで、スケジュール条件の入力を 1 日 10 件まで保証する。

セキュリティ要件

システムの実装に当たり、利用者の誤操作や誤入力を防ぐための仕組みや審査員の機能を制限するための機能を実装する。また、システムは教員専用のサーバ上で運営される予定であり、そのサーバには基本認証機能によって教員のみがアクセスできるようになっている。そのため、悪意ある第三者の攻撃や、他のシステムが介在したデータの流出等に対する特別なセキュリティ対策は講じない。

2.3.2 システム構成

以下では、スケジューリングシステムを構成するネットワーク、ハードウェアおよびソフトウェアの詳細を示す。

ネットワーク構成

スケジューリングシステムの利用におけるネットワーク構成の概要を図 2.6 に示す。審査員およびスケジュール担当者は、ネットワークを通じて CS 専攻教員専用サーバへアクセスする。図 2.7 に示すように、サーバへのアクセスは、筑波大学の内外を問わず、既存の筑波大学教員専用サーバの使用ポリシーに準じた共通認証を行う他、スケジュール担当者専用ページへアクセスする際は、専用の認証を行うことでアクセスを制限する。

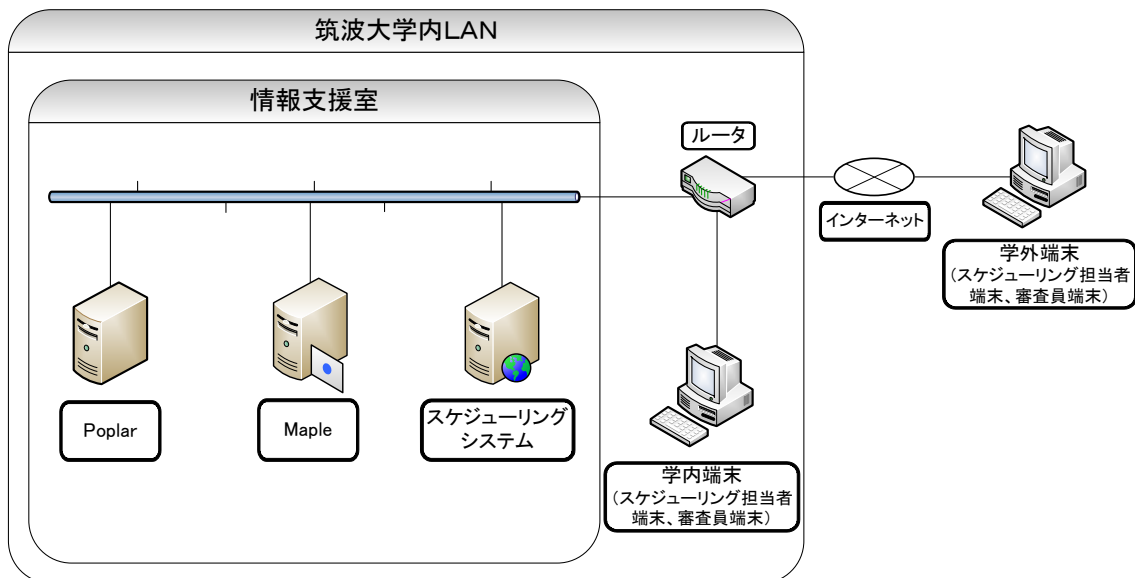


図 2.6 ネットワーク構成概要

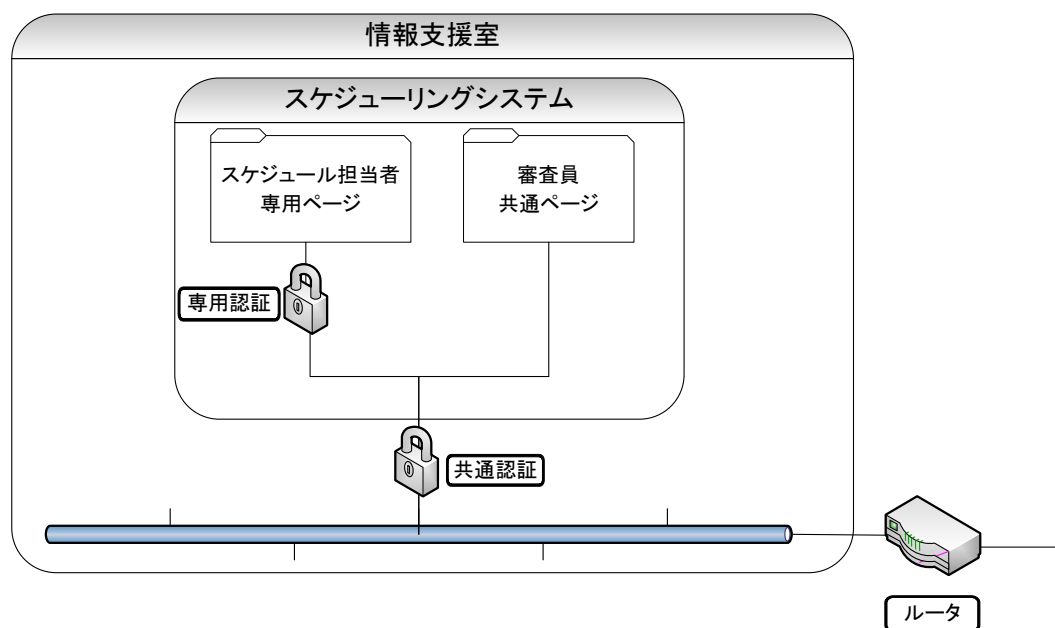


図 2.7 スケジューリングシステムのアクセス認証

ハードウェア構成

ハードウェア構成の詳細は表 2.5 の通りである。スケジューリングシステムの導入に当たり、サーバコンピュータ 1 台を新規に購入する。その他のクライアントコンピュータやネットワークインフラについては、大学内外に設置されているものを利用する。

表 2.5 ハードウェア構成

ハードウェア	必要数	備考
サーバコンピュータ	1 台	新規購入
クライアントコンピュータ	1 台以上	既存のものを利用

ソフトウェア構成

サーバコンピュータ上で使用しているソフトウェアの詳細は表 2.6 の通りである。

表 2.6 ソフトウェア構成

ソフトウェア	概要	備考
Solaris 10 5/08	OS	
MySQL 5	DBMS	※1
Apache 2	Web サーバ	※1
PHP 5	開発言語	※1
phpMyAdmin 3	DB 操作アプリケーション	※1

※1：オープンソースソフトウェア

2.3.3 開発したシステムの流れ

2.3.1 で述べた 4 つの機能を実現したスケジューリングシステムの画面の遷移を図 2.8 に示す。図上部にある丸は遷移の開始を、矢印に付加されている文は遷移する条件を、矢印の先にある楕円は画面を表している。また、審査員とスケジュール担当者で利用する機能が異なるため、それぞれのメニュー画面が存在する。以下ではそれらのメニュー画面と、4 つの機能についての詳細を説明する。

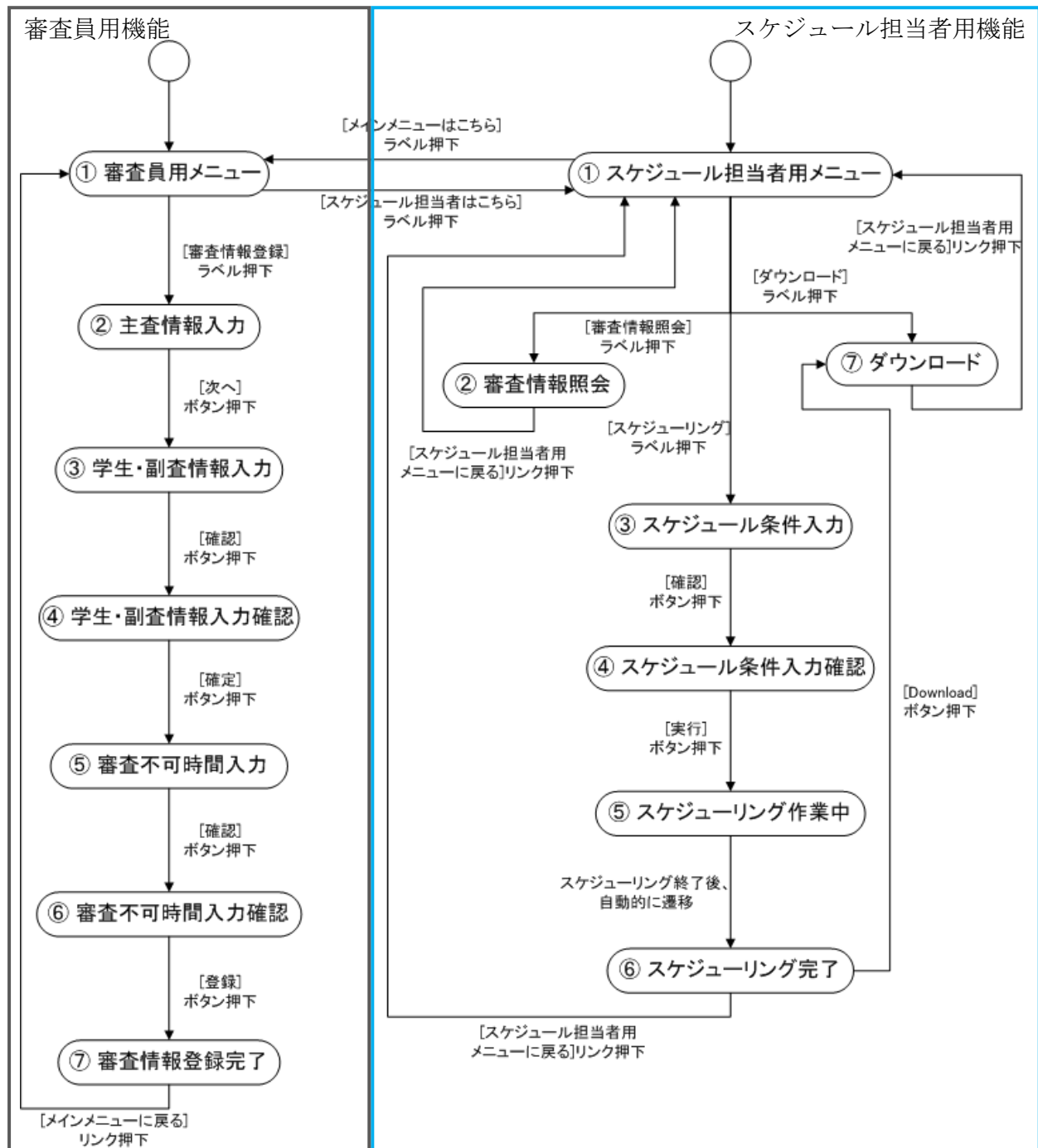


図 2.8 スケジューリングシステムの画面遷移図

審査員用機能

以下では、審査員が利用する機能である、審査情報登録機能と、そこで表示される各画面について説明する。

● 審査情報登録機能

審査情報登録機能は、審査委員会の主査が審査情報を登録する際に利用する機能である。まず、主査自身の氏名、主査が担当する審査委員会の副査および学生の氏名などを入力する。その後、主査自身と副査の審査不可時間を入力し、審査情報の登録を行う。

① 審査員用メニュー

審査員用メニュー画面（図 2.9）には、審査員が利用する機能である、審査情報を登録する画面へ遷移するためのラベルが存在する。このラベルへマウスカーソルを合わせることで、説明用ラベルが表示される。

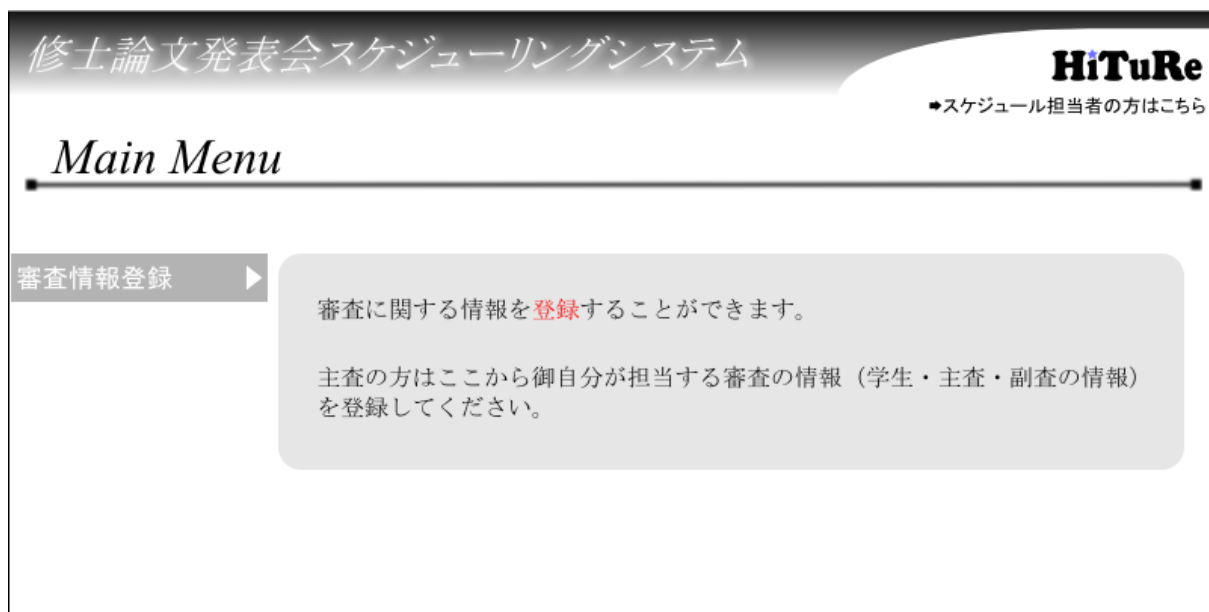


図 2.9 審査員用メニュー画面

② 主査情報入力

主査情報入力画面（図 2.10）では、主査が自身の氏名、職位、所属と、主査を担当する学生の人数を入力する。職位と学生人数は簡略化のため選択式になっており、職位は「教授」もしくは「准教授」、学生人数は「1」から「15」までとしている。また、本システムの対象は CS 専攻の修士論文発表会であるため、審査員のほとんどが CS 専攻に所属している教員である。そのため、所属は「コンピュータサイエンス専攻」もしくは「その他」を選択し、「その他」を選択した場合には直接入力するようになっている。

修士論文発表会スケジュールリングシステム

HiTuRe

主査情報入力

主査情報入力 > 学生・副査情報入力 > 審査不可時間入力

○主査である御自身の情報と主査を務める学生の人数を入力し、次の画面に進んでください。
※既に登録済みの場合は新規に追加登録する人数を選択してください。
※人数が15人を超える場合は、数回に分けて登録してください。

主査 [必須]	氏		名		教授 ▼
	所属	☒ コンピュータサイエンス専攻 ☐ その他			
学生人数 [必須]	1 ▼				

図 2.10 主査情報入力画面

③ 学生・副査情報入力

学生・副査情報入力画面（図 2.11）では、主査を担当する学生の人数分だけ、学生の学籍番号、氏名、論文題目、副査の氏名、職位、所属を入力する。職位は主査情報の入力と同様に選択式になっているが、副査は講師も務めることができるため、「教授」、「准教授」、「講師」の3つから選択するようになっている。また、副査は2人以上という規定があるため3人目の副査が存在する場合がある。その場合には副査3のチェックボックスにチェックを付けることで、3人目の副査情報が入力可能となる。

修士論文発表会スケジューリングシステム

HiTuRe

学生・副査情報入力

主査情報入力 > 学生・副査情報入力 > 審査不可時間入力

主査

筑波 太郎 教授 (コンピュータサイエンス専攻)

○ 主査を務める学生の修士論文に関して以下の情報を入力し、確認画面へ進んでください。(1人分)

※ 1人の学生に4人以上の教員が副査を担当する場合には、3人分の副査情報を入力しスケジュール担当者にご連絡ください。

※ 3人目の副査を入力する場合には、チェックボックスにチェックを付けてください。

※ 論文題目が英文の場合、英文の後に()付きで和訳を入れてください。

※ 既に登録済みの学生を登録しようとした場合は上書きされるため、学籍番号に間違いのないよう十分注意してください。

1人目

学生 [必須]	学籍番号				
	氏		名		
	論文題目				
副査1 [必須]	氏		名		教授 ▼
	所属	☒ コンピュータサイエンス専攻 ☐ その他			
副査2 [必須]	氏		名		教授 ▼
	所属	☒ コンピュータサイエンス専攻 ☐ その他			
副査3 ☐	氏		名		教授 ▼
	所属	☐ コンピュータサイエンス専攻 ☐ その他			

1

戻る

確認

▲Page Top

図 2.11 学生・副査情報入力画面

④ 学生・副査情報入力確認

③で入力した学生・副査情報が表示される。主査は入力内容を確認する。

⑤ 審査不可時間入力

審査不可時間入力画面（図 2.12）では、②と③で入力した主査と副査の審査不可時間をを入力する。時と分それぞれが分かれて選択式になっており、時は「09」から「21」まで1刻みに、分は「00」から「55」まで5刻みになっている。入力可能時間は9:00 から21:00 までである。また既にデータベースに登録されている審査員で、審査不可時間が設定されている場合には、その情報が初期値として表示される。

修士論文発表会スケジューリングシステム **HiTuRe**

審査不可時間入力

主査情報入力 > 学生・副査情報入力 > 審査不可時間入力

主査 筑波 太郎 教授 (コンピュータサイエンス専攻)

○続いて、発表会に参加することができない時間帯(9:00～21:00の間)がある教員がいる場合には、以下の「審査参加不可時間」を設定してください。
○設定完了後、確認画面へ進んでください。

通番	氏名	審査参加不可時間1	審査参加不可時間2
1	筑波 太郎 教授	- : - ~ - : -	- : - ~ - : -
2	鈴木 一郎 教授	- : - ~ - : -	- : - ~ - : -
3	佐藤 二郎 講師	- : - ~ - : -	- : - ~ - : -
4	田中 三郎 教授	- : - ~ - : -	- : - ~ - : -

[▲Page Top](#)

戻る 確認

図 2.12 審査不可時間入力画面

⑥ 審査不可時間入力確認

⑤で入力した審査不可時間が表示される。主査は入力内容を確認する。

⑦ 審査情報登録完了

審査情報の登録を完了した旨が表示される。

スケジュール担当者用機能

以下では、スケジュール担当者が利用する機能である、審査情報照会機能、スケジューリング機能、ダウンロード機能と、そこで表示される各画面について説明する。

- **審査情報照会機能**

データベースに登録されている審査情報を照会するための機能である。

- **スケジューリング機能**

スケジューリングを行うための機能である。まず、スケジュール担当者はスケジュール条件入力し、その後、システムにスケジューリングを実行させる。スケジューリングの成否にかかわらず、最後にスケジューリング完了画面が表示される。

- **ダウンロード機能**

スケジューリングによって出力されたスケジュールファイルをダウンロードするための機能である。過去にスケジューリングを行った際のスケジュール条件を確認することもできる。

① スケジュール担当者用メニュー

スケジュール担当者用メニュー画面（図 2.13）では、スケジュール担当者が利用する機能である、審査情報を確認する画面、スケジューリングを行う画面、スケジュールをダウンロードする画面へ遷移するためのラベルが存在する。審査員用メニュー同様、ラベルへマウスカーソルを合わせることでこれらの説明用ラベルが表示される。



図 2.13 スケジュール担当者用メニュー画面

② 審査情報照会

審査情報照会画面（図 2.14）では、審査員が登録した審査情報のうち、審査委員会情報と審査員情報を閲覧することができる。審査委員会情報と審査員情報でテーブルが分かれており、審査委員会情報テーブルには学生の学籍番号、氏名、論文題目とその学生の審査員（主査、副査）が掲載され、審査員情報テーブルには審査員の氏名と審査不可時間が掲載されている。

修士論文発表会スケジューリングシステム
HiTuRe

審査情報照会

審査委員会情報テーブル

学籍番号	学生 氏名	主査 氏名	副査1 氏名	副査2 氏名	副査3 氏名	論文題目
200720897	筑波 太郎	鈴木 一郎 教授	佐藤 二郎 教授	田中 三郎 教授		修士論文発表会自動スケジューリングシステム
200720000	筑波 二郎	鈴木 二郎 教授	佐藤 三郎 准教授	鈴木 一郎 教授	鈴木 四郎 講師	マーキングジェスチャを利用したフリーハンド作図インタフェース
200720001	筑波 光太郎	鈴木 二郎 教授	佐藤 三郎 准教授	田中 四郎 講師		SunSPOTを用いた研究

審査員情報テーブル

氏 名	所 属	審査参加不可時間1	審査参加不可時間2
鈴木 一郎 教授	コンピュータサイエンス専攻	9:00 ～ 10:00	11:00 ～ 12:00
佐藤 二郎 教授	コンピュータサイエンス専攻	－	－
田中 三郎 教授	リスク工学専攻	9:00 ～ 12:00	－
鈴木 二郎 教授	連携大学院	－	－
鈴木 三郎 准教授	コンピュータサイエンス専攻	－	－
田中 四郎 講師	図情メディア研究科	17:00 ～ 20:00	－

[スケジュール担当者用メニューへ戻る](#)

図 2.14 審査情報照会画面

③ スケジュール条件入力

スケジュール条件入力画面（図 2.15）では、スケジューリングを実行するための条件を入力する。入力はすべて選択式で、使用部屋数は「1」から「20」の 1 刻み、部屋使用可能時間は 9:00 から 24:00、一人あたりの発表時間は「5」から「60」の 5 刻み、1 セッションあたりの発表人数は「1」から「10」の 1 刻み、1 セッションあたりの休憩時間は「5」から「20」の 5 刻み、昼休み時間は「30」から「60」の 5 刻みである。アルゴリズム選択では 2 種類から選択することができる。「主査グルーピング」は、スケジューリングが成功しやすい、全審査終了時刻が早くなりやすいという特徴がある。一方、「親密グルーピング」では、同一の審査員の審査が、同部屋・連続で配置される可能性が高くなるという特徴がある。

修士論文発表会スケジューリングシステム

HiTuRe

スケジュール条件入力

○スケジューリングを実行するために以下の情報を入力してください。

使用部屋数

:

6

部屋使用可能時間

:

09

:

00

~

18

:

00

発表時間／人
(質疑応答含む)

:

30

分

発表人数／セッション

:

3

人

休憩時間／セッション

:

10

分

昼休み時間

:

50

分

アルゴリズム選択

:

☒ 主査グルーピング(標準)

☐ 親密グルーピング

戻る

確認

図 2.15 スケジュール条件入力画面

④ スケジュール条件入力確認

③で入力したスケジュール条件が表示される。スケジュール担当者は入力内容を確認する。

⑤ スケジューリング作業中

スケジューリングを実行中である旨が表示される。

⑥ スケジューリング完了

スケジューリングを完了した旨が表示される。

⑦ ダウンロード

ダウンロード画面（図 2.16）では、これまでに行ったスケジューリングのスケジュール条件とスケジューリング結果が表示されており、成功した場合にはスケジュールファイルへのアドレスにリンクが張られている。成功した時と失敗した時のスケジュール条件を参照することで、どのような条件であればスケジューリングに成功するのか確認することができる。

修士論文発表会スケジューリングシステム

HiTuRe

Download

ID	作成日時	条件	成否	リンク
1	2009/01/09 14:42:53	使用部屋数：11 部屋使用可能時間：09:00～23:00 発表時間/人：30分 発表人数/セッション：1人 休憩時間/セッション：5分 昼休み時間：30分 アルゴリズム：親密グルーピング	×	-
2	2009/01/09 14:41:55	使用部屋数：10 部屋使用可能時間：09:00～00:00 発表時間/人：30分 発表人数/セッション：3人 休憩時間/セッション：10分 昼休み時間：50分 アルゴリズム：主査グルーピング	○	http://hiture.cs.tsukuba.ac.jp/schedule/table/090109_144155.csv
3	2009/01/09 14:41:12	使用部屋数：10 部屋使用可能時間：09:00～22:00 発表時間/人：30分 発表人数/セッション：3人 休憩時間/セッション：10分 昼休み時間：50分 アルゴリズム：主査グルーピング	×	-

[▲Page Top](#)

[スケジュール担当者用メニューへ戻る](#)

図 2.16 ダウンロード画面

2.4 結果と考察

2008 年度の筑波大学修士論文発表会のスケジュール作成に当たって、実際にシステムを導入・運用した。そこで、スケジュール作成業務における人的作業を省力化できたか検討するため、システムの利用者であるスケジュール担当者および審査員に、アンケート調査を実施した。本節ではその結果を示し、システム化の効果と課題について考察する。

2.4.1 スケジュール担当者の作業負担の変化

昨年度の方式と今年度の方式とで、収集された審査情報をもとにスケジュール担当者がスケジュールの初版の作成にかかった時間は表 2.7 の通りである。昨年度の方式では、スケジュール担当者はスケジュール作成の準備に 2 時間かかっていた。ここには、担当事務員から受け取った審査情報を、スケジュールテーブル上に配置しやすいように変形したり、スケジュールテーブルを作成したりする時間が含まれる。その後、審査の配置を考えてスケジュールの初版を作成するには 6 時間かかり、合計で 8 時間かかった。一方、本システムを利用した今年度の方式では、審査情報の変形やスケジュールテーブルの作成は不要であり、時間は全くかからない。スケジュールの初版は、スケジュール担当者がシステム上でスケジュール条件を設定すると自動的に作成され、0.2 時間（12 分）程度で完了した。よって、スケジュール担当者のスケジュール初版作成にかかる労力を 7.8 時間削減することができた。

表 2.7 スケジュールの初版作成までにかかった時間

	スケジュール作成の 準備 [時間]	スケジュールの 初版作成 [時間]
昨年度の方式	2	6
今年度の方式	0	0.2

さて、本システムの主な拡張先として、筑波大学大学院の博士前期課程および修士課程の各専攻における修士論文発表会での使用が挙げられる。以下では、本システムを利用して発表会のスケジュールを作成した場合に削減される作業時間を示す。

本システムを利用した場合、スケジュールの初版作成には、1 専攻当たり約 8 時間分の労力が削減される。ただし、システム導入初年度に関してはシステムの設置や Web ページの修正などの保守作業が発生するため、システムの効果が多少異なってくる。この保守作業にかかる時間を約 3 時間程度とすると、システム導入初年度に 1 専攻当たりで削減される労力は 5 時間となる。なお、導入 2 年目以降にも保守作業は発生するが、前年度の審査情報やスケジュールを削除するといった 5 分程度で完了する作業のみであるため、これに要する時間は無視できる。

筑波大学大学院の博士前期課程および修士課程の専攻は 30 以上ある。そこで全専攻数を 30 とし、全専攻で本システムを利用した場合、大学全体で導入初年度は 150 時間、次年度

以降は 240 時間ずつ作業時間が削減される計算になる。

$$\text{導入初年度} \quad 150[\text{時間}] = (8 - 3) \left[\frac{\text{時間}}{\text{専攻数}} \right] \times 30[\text{専攻数}]$$

$$\text{次年度以降} \quad 240[\text{時間}] = 8 \left[\frac{\text{時間}}{\text{専攻数}} \right] \times 30[\text{専攻数}]$$

2.4.2 審査員の作業負担の変化

審査員の作業負担の変化を調査するため、システムの運用後に、システムを利用した主査に対してアンケート調査を行った。アンケート調査は、CS 専攻の教員全員が登録されているメーリングリストを通じてアンケート用紙を配布し、記入したアンケート用紙を電子メールで返信してもらうことで実施した。アンケートの配布先には、スケジュールリングシステムを利用した主査だけでなく、副査も含まれているが、アンケートの回答と返信は主査にのみ依頼した。その結果、今年度主査を担当する教員 35 名のうち、8 名からの回答を受け取ることができた。

審査情報登録方法

審査情報登録時の負担に関して調査した結果を表 2.8 に示す。昨年度までの方式と今年度のスケジュールリングシステムによる方式を比較したところ、今年度の方式の方が楽であるという回答が 5 件、昨年度と今年度の方式とでは差がないという回答が 2 件、昨年度の方式の方が楽であるという回答が 1 件得られた。よって、回答があった 8 件の中では、今年度の方式の方が楽という意見と、昨年度と今年度の方式では手間は変わらないという意見が多数を占めた。一方、「昨年度の方が楽」と回答した理由としては、主査が審査情報を登録する際、副査の審査員情報も登録しなくてはならないことが面倒であったことが挙げられていた。

表 2.8 審査情報登録時の負担

選択肢	回答数
今年の方が楽	5
同程度	2
昨年の方が楽	1

システムの使用性

次に、本システムの使い方が直感的に分かりやすいものか、操作しやすいものであったかを 1（悪い）から 5（良い）の 5 段階評価で尋ねたところ、表 2.9 および表 2.10 の回答が得られた。これらの結果から、アンケート回答者のほとんどが、システムの使い方が分かりやすい、システムが操作しやすいと回答していることが分かった。

表 2.9 システムの使い方の直観的な分かりやすさ

選択肢	分かりにくい		分かりやすい		
	1	2	3	4	5
回答数	0	1	0	2	5

表 2.10 システムの操作のしやすさ

選択肢	操作しづらい		操作しやすい		
	1	2	3	4	5
回答数	0	1	0	5	2

2.4.3 今後の課題

2.4.1 の結果から、スケジューリングシステムを利用することで、スケジュール担当者の作業量は 8 時間程度削減できた。また、2.4.2 で示した主査に対するアンケート調査の結果、アンケート回答者から審査情報収集やシステムの使用性について高い評価を得ることができた。これらは本プロジェクトの成果である。一方、スケジューリングシステムを利用することで生じた課題もあった。以下では、スケジューリングシステムの課題とその対策について述べる。

スケジュール担当者から見たシステムの課題： スケジュール修正支援機能の欠如

本システムのシステム化範囲は、スケジュール担当者が審査情報を受け取ってから、スケジュールの初版を作成するまでである。しかし、実際はその後、スケジュール担当者は各審査員からスケジュールの調整依頼を受け、依頼内容に沿ってスケジュール調整を行う。この調整作業はスケジュールの最終版が完成するまで数回繰り返される。スケジュール担当者へのヒアリングの結果、スケジュールの初版作成後、最終版が完成するまでの調整作業に平均して約 15 時間かかることが分かった。

2.2.2 節で述べたように、スケジューリングを行う際にはいくつかの留意点があり、それらを考慮しながらスケジュールを作成するのは非常に時間がかかる。これと同様の問題が調整作業でも発生し、スケジュール担当者の大きな負担となっていた。

この問題の解決策として、以下に示すようなスケジュール調整支援機能の実装が考えられる。スケジュール担当者はシステム上で、調整したい審査委員会をドラッグ&ドロップでスケジュールテーブル上に再配置する。その際、システムは再配置したスケジュールに対して、前述の留意点を満たしているかをリアルタイムで評価し、留意点を満たしていない場合には画面上に警告表示する。このように、スケジューリングにおける留意点を満たしているかど

うかの評価をシステムが自動的に行うことで、スケジュール担当者の負担を減らすことができると考えられる。

審査員から見たシステムの課題： 審査不可時間の確認作業の軽減

本システムで審査情報を登録する際、副査の審査不可時間も同時に登録しなければならない。このため、主査は副査の審査不可時間を予め聞いておく手間がかかった。また、主査が異なる審査委員会の副査を複数担当する教員は、複数の主査から審査不可時間を尋ねられるため、それら全てに回答することも手間であった。そのため、これらの確認作業を軽減することが課題となっている。

この課題には、審査不可時間を含めた副査自身の審査員情報は、副査本人がシステム上に登録できるようにするという改善案が考えられる。このような環境を実現するためには、審査情報の登録と審査不可時間の登録を分離すればよい。具体的には、主査はこれまで通りの手順に沿って審査情報の登録を行う。ただし、そこでは審査不可時間は登録せず、それ以外の情報だけを入力する。また、各審査員は個別に自身の審査不可時間を登録する。このような業務プロセスに変更することで、主査と副査の間で審査不可時間の確認作業を減らすことができる。

第3章 スケジューリングアルゴリズムの開発

本章より、報告者が担当したスケジューリングアルゴリズムの開発について説明する。まず、開発したスケジューリングアルゴリズムの説明と行い、そこから生じた問題と解決策について述べる。最後にアルゴリズムにおける今後の課題について検討する。

3.1 スケジュールの定義

本章で使用する「スケジュール」とは、本システムで対象としている修士論文発表会の発表日時、発表順番のことを表す。また、スケジューリングを行うとはどの審査委員会がどの部屋のどの時間になるかを決定するもので、表 3.1 に示すような入力情報から、審査委員会の部屋と時間を決定したスケジュールテーブルを導くことである。そのスケジューリングの時に考慮しなければならない点があり、詳細について 3.2 で説明する。また、スケジュール条件、スケジュールテーブルについても 3.3 で説明する。

表 3.1 スケジューリングの入力情報と出力情報

入力情報	出力情報
● スケジュール条件 ➤ 使用部屋数 ➤ 部屋使用可能時間 ➤ 発表時間／人 ➤ 発表人数／セッション ➤ 休憩時間／セッション ➤ 昼休み時間 ● 審査委員会情報 ➤ 学生情報 ➤ 審査員情報	● スケジュールテーブル ➤ 審査委員会の位置（部屋、時間）

3.2 スケジューリング時の考慮点

2.2.2 でも述べたとおり、スケジューリングを行う上で考慮しなければならない点がある。それらを整理すると以下のようになる。

- 必ず満たす必要がある項目
 - A) 同一時間帯に審査員が複数の審査を担当しないようにする
 - B) 審査員の審査不可時間を考慮する
- 可能な限り考慮する項目
 - C) 同一の審査員は連続して審査を実施する
 - C-1) 同部屋で審査を実施する
 - C-2) 異なる部屋にして移動時間を確保する
 - D) 全審査終了時刻を早くする

一人の審査員が同時刻に二つ以上の審査を担当することは物理的に不可能である。審査員によっては授業や会議などがあり、どうしても審査を行うことができない時間が存在する場合がある。それらの理由から、A)、B)は必ず満たす必要がある。

同一の審査員が連続して審査を担当し、その審査員の担当する審査がまとまって行われた場合、その審査員の拘束時間が減少するという利点がある。また、全審査終了時刻が早くなることも委託元教員からの要求であったが、C)、D)共に満たさなくてもスケジュールとしては成り立つため、可能な限り考慮する項目である。

C)において、同一の審査員の審査を連続して行うようにする場合、さらに C-1)もしくは C-2)どちらかに考慮しなければならない。なぜなら、同一の審査員の審査が異なる部屋で連続して行われ、前の審査の終了時刻と次の審査の開始時刻が同じであった場合、移動することができず審査を行うことができないためである。開発したアルゴリズムでは、C-1)に重点を置き、C-1)を可能な限り満たすように努めた。

D)においては、全ての部屋の終了時刻を早い時間かつ、同程度になるようにすることで対応する。例えば、ある一部屋のみ全審査の終了時刻が非常に遅くなるような状況が生じてしまうと、その部屋の審査員や学生に余計な負担がかかってしまうと同時に、結局のところ発表会全体の終了時刻が遅くなってしまうことになる。また、審査員や学生に負担がかかることによって、スケジュールに対する不満も出てき得るため、D)を可能な限り満たすように努めた。

開発したスケジューリングアルゴリズムでは、これらのスケジューリングの考慮点を満たすようなスケジュールを出力することを目指した。

3.3 スケジューリングアルゴリズム

ここでは 3.2 で説明した考慮点に基づいて考えたスケジューリングアルゴリズムについて説明する。

スケジューリングを行う際の入力情報は、データベースに登録されている審査委員会の情報（学生情報、審査員情報も含む）とスケジュール条件（スケジューリングを行う際に、スケジュール担当者が指定するもの）で、入力情報を用いてスケジューリングを行い、スケジュールを出力する。スケジューリングではまずスケジューリング可否判定を行う。可否判定でスケジューリング可能であると判定された場合に、審査委員会のグルーピングを行い、続いて、作成されたグループをスケジュールテーブルに挿入していく（グループレイアウト）。

全ての審査委員会をスケジュールテーブルに挿入し終わった時点でファイルに出力しスケジュールリング完了とする。以下ではスケジュールのファイル出力以外について説明する。

3.3.1 スケジュールリング可否判定

入力情報からスケジュールリングできるか否かを判定する。確実にスケジュールリングでスケジュールが完成しない場合に、スケジュールリングの最初に不可であると判定することによって、システムに 3.3.2 以降の処理を実行させ、余計な時間をかけないために行う。表 3.2 のようなスケジュール条件で行った場合を考える。

表 3.2 スケジュール条件

使用部屋数	6 部屋
部屋使用可能時間	9:00 ～ 18:00
発表時間／人	30 分
発表人数／セッション	3 人
休憩時間／セッション	10 分
昼休み時間	50 分

① スケジュールテーブル作成

スケジュール条件を用いて、スケジュールテーブルを作成する（図 3.1）。部屋使用可能開始時間から始まり、発表時間分時間を確保する。それが一人分の発表時間となり、セッションの人数分繰り返す。セッションの人数分繰り返したらセッション間の休憩時間を挟み、同様の処理を行う。昼休みは 12:00 を過ぎて最初にセッション区切りがきた段階で、通常の休憩を設定せず昼休みを設定する。確保しようとした発表時間が部屋使用可能終了時間と重なってしまった時点でそれを発表時間と設定せず、処理を終了する。

	部屋 1	部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30						
9:30～10:00						
10:00～10:30						
10:40～11:10						
11:10～11:40						
11:40～12:10						
13:00～13:30						
13:30～14:00						
14:00～14:30						
14:40～15:10						
15:10～15:40						
15:40～16:10						
16:20～16:50						
16:50～17:20						
17:20～17:50						

図 3.1 スケジュールテーブル

② 審査委員会数とスケジュール枠の比較

データベースに登録されている審査委員会の総数と、①で作成したスケジュールテーブルの枠数（1 部屋当たりの審査委員会数×部屋数）を比較し、スケジュールテーブルの枠数の方が多ければスケジューリング可能であるとする。表 3.1 スケジュール条件と図 3.1 スケジュール条件の例で考えると、スケジュールテーブルの枠数が $15 \times 6 = 90$ あるので、データベースに登録されている審査委員会の総数が 90 以下であればスケジューリング可能であるとなる。

3.3.2 グルーピング

「グルーピング」とは、複数の審査員会を一つのまとまりとみなし、「グループ」を作成することを言う。グルーピングを行うことで複数のグループが作成されるが、ここではそれらのグループの並び替えもグルーピングの一部とする。3.2 で述べたスケジューリング時の考慮点 C)を満たすことを目的に、本アルゴリズムではグルーピングを導入する。グルーピングには 3 種類あり、①主査グルーピング、②内部グルーピング、③親密グルーピング、と呼ぶことにする。これらのグルーピングは番号の順にグルーピングを行う。また、グループは③ → ② → ①の順に階層構造になっている。

① 主査グルーピング

主査グルーピングでは審査委員会の主査に注目し、主査が同一の審査委員会を一つのグループとし、主査グループと呼ぶ。同一の主査でグループを作成することによって、

スケジューリング時の考慮点 A)と C)を主査において同時に満たすことができるため、主査グルーピングを行う。図 3.2 では主査グルーピングを行う前の状態を示し、図 3.3 では主査グルーピングを行った後の状態を示している。また、行が 1 つの審査委員会を表している。

図 3.2 において、主査 A の審査委員会が 3 つ、主査 H の審査委員会が 2 つ、主査 B の審査委員会が 2 つ、主査 I と主査 C の審査委員会がそれぞれ 1 つある。それぞれをグループとし図 3.3 のような形となり、それぞれが「主査グループ」となる。主査グループ作成後、審査員会数の多い順に並び替えを行い、主査グルーピングが完了となる。

主査	副査	副査
A	B	C
A	D	E
H	I	J
I	H	J
H	J	K
A	B	C
A	C	E
B	C	D
H	K	J
A	D	E
C	E	A
B	C	E

図 3.2 主査グルーピング前

主査	副査	副査
A	B	C
A	D	E
A	B	C
A	C	E
A	D	E
H	I	J
H	J	K
H	K	J
B	C	D
B	C	E
I	H	J
C	E	A

図 3.3 主査グルーピング後

② 内部グルーピング

内部グルーピングでは主査グルーピングで作られた主査グループに属している、審査委員会の審査員に注目する。審査委員会によっては、主査だけでなく副査もまったく同じという状況の、審査員が完全に一致しているものが存在する。そこで、主査グループの中で審査員が完全一致している審査委員会とそうでないものの 2 つに分け、それぞれをグループとし、内部グループと呼ぶ。主査と副査が完全に一致している審査委員会を連続にすることによって、考慮点 C)を満たすことができるようになるため、内部グルーピングを導入する。図 3.4 では内部グルーピングを行う前の状態を示し、図 3.5 では内部グルーピングを行った後の状態を示している。

主査 A の主査グループでは、審査員 A・B・C の審査委員会と審査員 A・D・E の審

査委員会において完全一致の審査委員会が存在する。それらをそれぞれ完全一致内部グループとし、残りの審査委員会を1つの内部グループとする。同様に主査 H の主査グループにおいても完全一致とそうでないものの内部グループを作成する。他の主査グループでは完全一致の審査委員会が存在しないため、主査グループ=内部グループとする。

主査	副査	副査
A	B	C
A	D	E
A	B	C
A	C	E
A	D	E

H	I	J
H	J	K
H	K	J

B	C	D
B	C	E

I	H	J
---	---	---

C	E	A
---	---	---

図 3.4 内部グルーピング前

主査	副査	副査
A	B	C
A	B	C
A	D	E
A	D	E
A	C	E

H	J	K
H	K	J
H	I	J

B	C	D
B	C	E

I	H	J
---	---	---

C	E	A
---	---	---

図 3.5 内部グルーピング後

③ 親密グルーピング

親密グルーピングでは主査グルーピングで作られた主査グループに属している、審査委員会の主査と副査の関連に注目する。ある審査委員会の「主査ア」が「主査イ」の審査委員会の副査を担当し、主査イの審査委員会の主査イが主査アの審査委員会の副査を担当するという場合がある。そのように審査委員会同士において審査員が交差しているような関係を持つとき、それをグループとし、「親密グループ」と呼ぶ。図 3.6 では親密グルーピングを行う前の状態を示し、図 3.7 では親密グルーピングを行った後の状態を示している。

主査 A の主査グループに属する副査は B、C、D、E である。この副査が主査を担当している主査グループがないかどうか見ると、主査 B の主査グループを主査 C の主査グループがあることがわかる。次にその主査グループに対し A が副査を担当しているかどうかをみると、主査 C の主査グループでは A が副査を担当している。そこで、この主査 C の主査グループを主査 A のグループと親密であるとみなし、親密グループとする。同様に、主査 H のグループと主査 I のグループも親密関係にあり、親密グループを作成する。

主査	副査	副査
A	B	C
A	B	C
A	D	E
A	D	E
A	C	E

H	J	K
H	K	J
H	I	J

I	H	J
---	---	---

B	C	D
B	C	E

C	E	A
---	---	---

図 3.6 親密グルーピング前

主査	副査	副査
A	B	C
A	B	C
A	D	E
A	D	E
A	C	E
C	E	A

H	J	K
H	K	J
H	I	J
I	H	J

B	C	D
B	C	E

図 3.7 親密グルーピング後

3.3.3 グループレイアウト

3.3.2 のグルーピングで作成したグループを、3.3.1 で作成したスケジュールテーブルに挿入していく。グループレイアウトではまず、親密グループに属する審査委員会数（以下、親密グループのサイズとする）から、スケジュールテーブルの挿入位置を確保する。その上で内部グループを順番に挿入していき、挿入が完了した時点でまた挿入位置を探すところに戻るといった手順を踏む。これからその詳細を説明する。

① 親密グループ挿入位置探索

スケジュールテーブルから親密グループの挿入可能な位置を探す。スケジュールテーブルの早い時間から順番に、その時間帯が空かどうかを見ていき、空であった場合は親密グループのサイズ分だけ後ろの時間帯全てが空いているかどうかを確認する。もし挿入不可能であると判断された場合には、同じ時間の別の部屋に挿入できないかどうか探す。そうすることによって、考慮すべき項目 D を可能な限り満たすことができる。スケジュールテーブルに何も挿入されていない初期の状態であるとき、必ず部屋 1 の最も早い時間に挿入しようとする。

親密グループ A を挿入しようとしたとき、部屋 1 の 9:00 のところから親密グループ A のサイズ分だけ後ろの時間帯を調べる。空であった場合、ここを親密グループの挿入位置の候補とする（図 3.8）。挿入位置の候補が決定した時点で、② 主査・内部グループ挿入位置探索を行う。

親密グループ A に対し②の処理を行ってこのまま挿入完了したとし、その上で新たに親密グループ B を挿入する場合、まず親密グループ A のときと同様に部屋 1 の 9:00 に挿入でき

ないか調べる。すると既に親密グループ A が挿入されており挿入することができないことがわかる。挿入できないことが分かった場合、同じ時間帯の別の部屋に挿入できないか調べるので、部屋 2 の 9:00 に挿入できないかどうかを確認する（図 3.9）。

このように親密グループの挿入位置探索を繰り返し、全ての親密グループに対して操作が終了した時点で、親密グループ挿入位置探索が完了する。

	部屋 1	部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30	親密 A					
9:30～10:00						
10:00～10:30						
10:40～11:10						
11:10～11:40						
11:40～12:10						
13:00～13:30						

図 3.8 親密グループ A 挿入位置探索

	部屋 1	部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30	親密 A	親密 B				
9:30～10:00						
10:00～10:30						
10:40～11:10						
11:10～11:40						
11:40～12:10						
13:00～13:30						

図 3.9 親密グループ B 挿入位置探索

② 主査・内部グループ挿入位置探索

①で親密グループの挿入位置候補が決定しているので、まずその挿入位置候補の中で主査グループが挿入可能な位置を探す。親密グループ A（サイズ 6）が部屋 1 の 9:00 のところに挿入可能であるとなっていた場合、その中の主査グループに注目する。1 つ目の主査グループ A（サイズ 5）を親密グループの挿入位置候補の中で早い時間帯（ここでは 9:00）から挿入可能かどうか調べる（図 3.10）。ここでの挿入可能かどうかの判定は、空であるかどうか、主査グループのサイズ分後ろの時間帯に空きがあるかどうかという 2 つについて調べる。

主査グループの挿入位置の候補が決定したのち、以下で説明する内部グループの挿入位置探索が行われる。もしその段階で挿入不可能であると判断された場合には、主査グループの挿入位置候補を変更する必要がある。そこで、主査グループの挿入位置について一つ後ろの

時間帯に挿入可能であるかどうか調べる。この処理を繰り返しでも挿入不可であった場合にはこの親密グループ挿入位置候補を取り消し、①の親密グループ挿入位置探索に戻る。

	部屋 1	部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30	主査 A					
9:30～10:00						
10:00～10:30						
10:40～11:10						
11:10～11:40						
11:40～12:10						
13:00～13:30						

図 3.10 主査グループ挿入位置探索

主査グループが挿入可能であるとなった場合には、主査グループの挿入可能位置の中でその主査グループに属する内部グループの挿入位置を探す。図 3.5 の通り主査グループ A には 3 つの内部グループがあるとする。これらの内部グループを順番に挿入可能かどうか、早い時間帯（ここでは 9:00）から調べる（図 3.11）。ここでも主査グループ探索同様に、空であるかどうか、内部グループのサイズ分後ろの時間帯に空きがあるかどうかという 2 点で、挿入可能かどうか判定する。

挿入不可能であると判断された場合には、一つ後ろの時間帯に挿入可能であるかどうか調べる。この処理を繰り返しでも挿入不可であった場合にはこの主査グループ挿入位置候補を取り消し、主査グループ挿入位置探索に戻る。

	部屋 1	部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30	内部 A					
9:30～10:00						
10:00～10:30						
10:40～11:10						
11:10～11:40						
11:40～12:10						
13:00～13:30						

図 3.11 内部グループ挿入位置探索

内部グループが挿入可能であるとなった場合には、内部グループの挿入可能位置の中でその内部グループに属する審査委員会の挿入が可能かどうかを調べる。ここで行う挿入可能かどうかの判定において、3.2 で述べたスケジューリングの考慮点のうち必ず満たさなければ

ならない項目 A、B を満たしているかという判定を行う。内部グループに属している審査委員会の審査員（主査、副査全て）に対し、挿入しようとしている審査の時間と審査不可時間とが被っていないか確認する。審査不可時間に問題がなかったら、続いて同じ時間の別の部屋に、その審査員が審査を担当していないかどうか確認する。

内部グループ A に属している審査委員会を A、B とする。そしてそれぞれの審査員を A、B、C とする。まず審査不可時間の確認を行うために審査員 A の審査不可時間を確認する。審査不可時間がない場合や審査不可時間が 9:00～9:30 と被っていなければ審査不可時間は問題ないと判断する。審査不可時間に問題がなければ、次に別の部屋に同じ審査員がいないかどうかの確認を行う。内部グループ A を部屋 1 に挿入しようとしているため、部屋 2 → 部屋 3 → 部屋 4 → …… と全ての部屋を確認する（図 3.12）。

審査不可時間もしくは審査員の重複で内部グループを挿入しようとした位置に挿入できなかった場合、その内部グループの挿入位置を一つ後ろの時間帯にずらし、同様に審査不可時間と審査員の重複の確認を行う（図 3.13）。

このように内部グループの挿入位置探索を繰り返し、全ての内部グループに対して操作が終了した時点で、主査グループの挿入位置探索に戻る。主査グループ全てに対し探索が終了した時点で主査・内部グループ挿入位置探索が完了となる。

	部屋 1			部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30	A	B	C					
9:30～10:00	A	B	C					
10:00～10:30								
10:40～11:10								
11:10～11:40								
11:40～12:10								
13:00～13:30								

図 3.12 内部グループ挿入位置探索（審査員重複確認 1）

	部屋 1			部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30								
9:30～10:00	A	B	C					
10:00～10:30	A	B	C					
10:40～11:10								
11:10～11:40								
11:40～12:10								
13:00～13:30								

図 3.13 内部グループ挿入位置探索（審査員重複確認 2）

最後に図 3.7 の親密グループ 3 つに対しグループレイアウトを行った場合の例を図 3.14 に示す。但し、審査員 B の審査不可時間が 9:00 ～ 10:15、審査員 I の審査不可時間が 9:00 ～ 11:30 であったとする。

	部屋 1			部屋 2			部屋 3			部屋 4			部屋 5			部屋 6		
9:00～9:30	A	D	E															
9:30～10:00	A	D	E															
10:00～10:30	A	C	E															
10:40～11:10	A	B	C		H	J	K											
11:10～11:40	A	B	C		H	K	J											
11:40～12:10	C	E	A		H	I	J		B	F	G							
13:00～13:30					I	H	J		B	C	F							
13:30～14:00																		
14:00～14:30																		
14:40～15:10																		
15:10～15:40																		
15:40～16:10																		
16:20～16:50																		
16:50～17:20																		
17:20～17:50																		

図 3.14 スケジューリング済みスケジュールテーブル

3.4 アルゴリズムの問題点と改良

3.3 で説明したスケジューリングアルゴリズムを用いてシステムを開発したところ、新たな問題が発生した。アルゴリズムを考えていく段階では 2007 年度の修士論文発表会のデータを使用していた。その時点では主査グループ、親密グループ共に最大のサイズが 12 であり、グループ化してもサイズが 1 のグループも多々あったため、グループレイアウトがスムーズに行えていた。しかし、2008 年度のデータを用いてグルーピングを行ってみたところ、親密グループの最大サイズが 20 を超える想定より遥かに大きいサイズの親密グループが多々できてしまった。その結果、グループレイアウトを行った際にスケジュールテーブルにグループを挿入することができず、スケジューリングに失敗してしまうという問題が発生してしまった。問題が発生した原因として考えられる事項は、「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」が 2007 年度から新設され、コンピュータサイエンス専攻の学生が 20~30 名程度増加したことが挙げられる。学生が増加することで審査員の関連で作成している親密グループにおいては、サイズが大きくなる可能性が高くなる。

そこで、主査グループに注目してみると主査グループは最大でサイズが 8 であるため、親密グループを作成せずに主査グループの状態（内部グルーピング後の状態）でグループレイアウトを行うことでスケジューリングが成功するのではないかと考えた。実際に親密グループまで作成した場合と作成しなかった場合にスケジュールテーブルに挿入したものの例を以下の①の図 3.15 と図 3.16、②の図 3.17 と図 3.18 に示す。但し、審査不可時間はないとする。

① 親密グループを作成した場合

主査	副査	副査
A	B	C
A	B	C
A	B	C
A	D	E
A	D	E
A	D	F
A	D	F
F	N	O
F	N	Q
F	R	S
F	A	O
S	F	U
S	F	Q

G	H	F
G	H	F
G	H	F
G	H	I
G	H	I
G	H	J
I	T	Q
I	U	G

K	L	M
K	L	M
K	N	O
K	P	Q
K	G	O

図 3.15 親密グループ

	部屋 1	部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30	A B C	G H F				
9:30～10:00	A B C	G H F				
10:00～10:30	A B C	G H F	K L N			
10:40～11:10	A D E	G H I	K L N			
11:10～11:40	A D E	G H I	K N O			
11:40～12:10	A D F	G H J	K P Q			
13:00～13:30	A D F	I T Q	K G O			
13:30～14:00	F N O	I U G				
14:00～14:30	F N Q					
14:40～15:10	F R S					
15:10～15:40	F A O					
15:40～16:10	S F U					
16:20～16:50	S F Q					
16:50～17:20						
17:20～17:50						

図 3.16 親密グループ挿入済みスケジュールテーブル

② 親密グループを作成しなかった場合

主査	副査	副査
A	B	C
A	B	C
A	B	C
A	D	E
A	D	E
A	D	F
A	D	F

G	H	F
G	H	F
G	H	F
G	H	I
G	H	I
G	H	J

K	L	M
K	L	M
K	N	O
K	P	Q
K	G	O

F	N	O
F	N	Q
F	R	S
F	A	O

I	T	G
I	U	O

S	F	U
S	F	Q

図 3.17 主査・内部グループ

	部屋 1	部屋 2	部屋 3	部屋 4	部屋 5	部屋 6
9:00～9:30	A B C	G H F	I T G			
9:30～10:00	A B C	G H F	I U Q			
10:00～10:30	A B C	G H F	K L N			
10:40～11:10	A D E	G H I	K L N	S F U		
11:10～11:40	A D E	G H I	K N O	S F Q		
11:40～12:10	A D F	G H J	K P Q			
13:00～13:30	A D F		K G O			
13:30～14:00	F N O					
14:00～14:30	F N Q					
14:40～15:10	F R S					
15:10～15:40	F A O					
15:40～16:10						
16:20～16:50						
16:50～17:20						
17:20～17:50						

図 3.18 主査・内部グループ挿入済みスケジュールテーブル

ここで、①親密グループを作成した場合と②親密グループを作成しなかった場合の利点と欠点について考える。まとめたものが以下の表 3.3 である。

表 3.3 親密グループ作成・未作成の利点と欠点

	利点	欠点
①親密グループ作成	A) 審査員の関わりを見るため、同一の審査員が連続して異なる部屋で審査を担当しなければならない可能性が少なくなる	B) 親密グループのサイズが大きくなりすぎると、グループレイアウトで失敗する（スケジューリングに失敗する）可能性が増す C) サイズの小さいグループが少ないと、部屋毎の全審査終了時間の均一化が図りづらい D) サイズの小さいグループが少ないと、スケジュールテーブルの中の空き時間が多くなりやすい
②親密グループ未作成	B) グループのサイズが大きくなりすぎないため、スケジューリングに成功しやすくなる C) 部屋毎の全審査終了時間の均一化が図りやすい D) スケジュールテーブルの中の空き時間が少なくなる可能性が高い	A) 同一の審査員が連続して異なる部屋で審査を担当しなければならない可能性が増す

親密グループの欠点の特徴としてみられるものは、親密グループのサイズが大きかった場合に発生する欠点であるということである。つまり、サイズが大きくなりすぎたり、サイズが大きいグループが多かったりしなければ、親密グループを作成したときの欠点がなくなることができる。しかし、親密グループのサイズが大きくなりすぎたり、サイズが大きかったりするグループが多くなってしまう場合には、親密グループを作成しないほうが良いということができる。

スケジューリングを行う上で最も重要なことは、スケジューリングに成功することであると考えている。もし親密グルーピングを行うことによって成功の確率が下がってしまうのであれば、親密グルーピングを行わないほうが良いと考えられる。しかし、①親密グループ作成の利点 A は特に重要であるため、ここでは親密グルーピングを活かすために 2 つの対策を考え、どちらかを選択することにした。

- 対策 1 :

親密グルーピングによるグループレイアウトに失敗した場合、親密グルーピングを行わずグループレイアウトを試す

まず初めに今まで通り親密グルーピングを行い、グループレイアウトを行う。もしグループレイアウトに失敗した場合、グルーピングのところまで戻り親密グルーピングを行わずグループレイアウトに進み、グループレイアウトを実行する。

対策 1 の利点は必ず親密グルーピングを試すため、親密グルーピングを使ってスケジューリングが成功すれば親密グルーピングの利点を活かすことができることである。もし親密グルーピングをしてスケジューリングに成功したのであれば、親密グルーピングの利点を活かしたスケジュールができたことになる。

しかし、対策 1 では親密グルーピングに失敗しなければ、親密グルーピングを行わない方法を試せないため、スケジュール担当者が親密グルーピングを行わない方法を自分の意思で試すことができない。例えばできる限り審査員が同一の部屋で連続したスケジュールを出力させたい場合には親密グループを作成するアルゴリズムを選択すべきであるし、連続性よりもスケジュールテーブルの中の空き時間が少なくなることや終了時間の均一化を重視したスケジュールを出力したい場合には親密グループを作成しないアルゴリズムを選択すべきである。そのため、スケジュール担当者の意思で選択できないことが欠点である。

- 対策 2 :

スケジュール条件に、親密グルーピングを行うか否かを含める

本システムではスケジュール担当者がスケジュール条件を入力するようになっている。その入力の 1 に親密グルーピングを行うかどうかの選択肢を設け、親密グルーピングを行うかどうかを決定するという作りにする。

対策 2 の利点はスケジュール担当者が自分の意思で親密グルーピングを行うかどうかを選ぶことができるという点である。しかし、親密グルーピングを行うか否かをそのまま表示させてしまった場合、スケジューリング担当者が親密グルーピングを行った場合と行わなかった場合の利点と欠点を理解している必要がある。そのため、親密グループを作成した場合としなかった場合のそれぞれの利点を表示させ、スケジュール担当者が期待するスケジュールが出力されるほうを選択できるようにすべきである。

親密グルーピングを行った場合に比べて、親密グルーピングを行わなかった場合に出力されたスケジュールのほうが、考慮すべき項目が満たされたスケジュールになっているという可能性もある。また、対策 2 で述べたように、選択式にすることで得られる利点は大きいと考えられる。そのため、スケジュール担当者が自分で選択できるようにしたほうがよいのではないかと考え、今回のスケジューリングでは対策 2 を選択することにした。改良後のスケジューリングアルゴリズムの流れをまとめると以下ようになる。

- 親密グルーピングを行う
 1. 主査グルーピング
 2. 内部グルーピング
 3. 親密グルーピング
 4. 親密グループ挿入位置探索
 5. 主査・内部グループ挿入位置探索

- 親密グルーピングを行わない
 1. 主査グルーピング
 2. 内部グルーピング
 3. 主査・内部グループ挿入位置探索

3.5 スケジューリングアルゴリズムにおける今後の課題

開発したスケジューリングアルゴリズムでスケジューリングを行うことはできた。しかし、出力されたスケジュールがそのまま使えるわけではなく、手作業で調整をしなければならない。それは人間の都合がのちに発生してしまうことも一つの要因ではあるが、アルゴリズムがまだ考慮すべき項目を満たせていない部分があることも要因である。本節では、考慮すべき項目を満たせていない点をスケジューリングアルゴリズムにおける今後の課題として述べる。

3.5.1 特定課題研究の学生の扱い

2007 年度の修士論文発表会までは全ての学生が修士論文についての発表を行っていた。しかし我々2008 年度の学生からは高度 IT 人材育成ための実践的ソフトウェア開発専修プログラムが開始されたために修士論文以外に特定課題研究の発表も同日に行われ、スケジュールに組み込む必要がある。特定課題研究は複数人のチームで一つの課題に取り組んでいる場合があり、そのような人たちは連続して審査を行うようにしたいということと、一人で取り組んでいる場合でも、できれば他の特定課題研究を行っている学生と連続して審査が行われるようにしたいという要望があった。ここで問題になるのが、原則では特定課題研究を行う学生の主査も修士論文同様に指導教員が担当することになっていることと、委託元教員は副査を担当することになっていることである。指導教員は学生が所属している研究室の教員で、特定課題研究の委託元教員とは別の場合が多い。現在のアルゴリズムでは主査グループを作るため、グループを作った際に特定課題研究を行う同じチームの学生は別々のグループになり、修士論文の学生と特定課題研究の学生が一つのグループになってしまう。

この問題を解決するためには特定課題研究を行う学生に、特定課題研究であるという情報を付加することと、副査を担当する審査員のうち委託元教員にはそれを示すための情報を付加する必要がある。そして、その学生に対し何らかの処置をしなければならない。その対策について検討する。

- 対策 1 :
特定課題研究の学生は 1 部屋で審査を行う

特定課題研究の発表をする学生は 1 部屋で審査を行うようにし、修士論文の学生は今まで通り複数の部屋に分かれて審査を行うようにする。そのためにはまず特定課題研究の学生のみで委託元教員毎にグルーピングを行い、グループプレイアウトを行ってスケジュールを組む。その上で修士論文の学生に対して開発したアルゴリズム通りグルーピングを行い、特定課題研究の学生のスケジュールも考慮に入れてグループプレイアウトを行う。そうすることで課題を解決することができる。しかし、2008 年度に特定課題研究を行った学生は 17 名である。そのため、特定課題研究の学生の部屋は確実に終了時間が 19:00 を過ぎてしまう。審査不可時間があつた場合には更に終了時刻が遅くなる恐れがある。人数が少なかった場合には有効な手段ではあるが、多くなってしまうと対策 1 では現実には厳しいと言わざるを得ない。

- 対策 2 :
特定課題研究の学生の中で同一の委託元教員である学生を 1 つのグループにまとめる

対策 1 とは違い全ての学生に対しグルーピングを行うが、特定課題研究の学生のみ委託元教員でグルーピングを行ってからグループプレイアウトを行う。そうすることで委託元教員が同一の学生は連続して審査が行われる。しかしこの対策の問題は、一人で特定課題研究を行っている学生である。一人で行っている学生はグルーピングをしても一人であるため、修士論文の学生の間に入ってしまい特定課題研究を行っている学生が連続しないことになる。一人の学生は一人の学生同士再びグルーピングしてグループ化するという対策も考えられるが、そうすると同一の審査員ができるだけ連続していたほうが良いという要求を無視してしまうことになる。結果、対策 2 も課題に対する最適な対策とは言い難い。

現在この 2 つの方法が考えられるが、どちらにも利点と欠点が存在する。そのため、どちらかを選択する場合、ステークホルダーや状況などをよく考え選択する必要があると考えられる。また、どちらの対策もうまくいかない場合も考えられるため、そのときは新たな対策を講じる必要があることも今後の課題である。

3.5.2 審査員の連続性

3.2 においてスケジューリング時に考慮すべき項目として、C)同一の審査員が同部屋で連続して審査を実施する（以下、「審査員（主査、副査）の連続性」と呼ぶ）という項目があることを述べた。これを考慮してグルーピングとグループプレイアウトを行うアルゴリズムを考案した。このアルゴリズムによって主査の連続性は確保されたが、副査についての連続性を考慮できていない部分がある。内部グループを作成した際、審査員が完全に一致した場合のグループとそれ以外のグループに分けている。完全一致のグループは副査の連続性も確保されているが、それ以外のグループはただグループ化しているだけで、グループ内の並び順ま

で考慮していない。また、グループレイアウトの際に考慮していることは、同じ時間帯に同一の審査員がいないかどうかと審査不可時間に問題がないかという 2 点であるため、別々の部屋で連続して審査が行われる、休憩時間がないにもかかわらず部屋を移動しなければならないということが起こり得る。実際にシステムを使用して出力したスケジュールでは、同一の審査員が連続して別の部屋で審査を担当しなければならない状況が起きていた。

それらの問題を解決し審査員の連続性を確保する手法を以下で検討する。

● 対策 1：

内部グループ作成時に審査員の連続性を確保する

完全一致でない内部グループ作成時に、副査の連続性を確保することでグループ内において審査員の連続性が確保されるため、全体を見ても連続性が確保されるようになると考えられる。ここで問題になるのが、1 つの審査委員会に 2 人以上の副査がいるということである。例えば図 3.19 左のような審査委員会があったとする。今まで通り内部グループを作成すると①のように上から順番に並ぶという解のみ存在する。しかし、副査の連続性を確保して作成しようとするすると図 3.19 右のように 3 通りの解が見つかってしまう。解が 3 つのままグルーピングを終了することはできないため、どれか選択しなければならない。そのときの選択の基準を何にするかということが大きな問題になる。

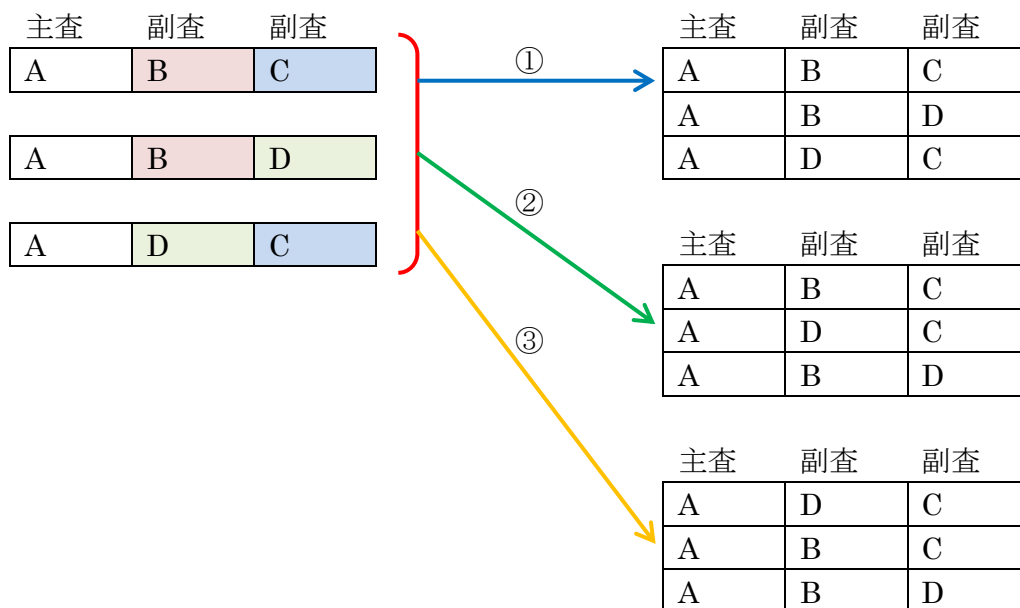


図 3.19 内部グループ作成例

上記のような問題は発生するが、この対策 1 で内部グループ内の連続性は以前より確保される。しかし、これだけでは別々の部屋で連続して審査を担当しなければならないという問題は解決していない。この問題の解決について、更なる対策 2 を考えていく。

- 対策 2 :
グループレイアウト時に審査員の連続性を確保する

グループレイアウトで考慮することとして、審査員の連続性を追加する。まず、スケジュールテーブルにグループを挿入するとき、挿入する枠の前後の少なくともどちらか一方に他の審査委員会があるかどうか確認する。ある場合にはその審査委員会の審査員と挿入しようとしているグループの先頭（もしくは末尾）の審査員を確認し、審査員が連続していたら挿入する。連続していなかったら挿入位置の候補として、更に他を探してみる。とすることで、審査員の連続性が確保されと考えられる。

また、挿入しようとしているグループと、既にスケジュールテーブルに挿入されている審査委員会の審査員を比較し、別々の部屋で連続して審査を担当しなければならない状況になっていた場合には挿入位置として認めないようにすることで、課題の解決が期待できる。

対策 1 と対策 2 を述べてきたが、この対策ではまだ不足している部分がある。それはグループ同士の審査員の連続性と、グループ同士の連続性の確保のためのグループ内の並び順の考慮である。ここまで述べてきた対策 1、2 の実現と、不足部分の新たな対策を考えていくことが今後の課題である。

第4章 結言

本報告書ではまず、本プロジェクトの開発体制として個人の担当領域と、開発経過について説明した。続いて、システムの開発にあたり、業務の洗い出しやシステム化後の業務改善について整理し、開発したシステムの概要について説明した。スケジュール担当者に行ったシステムについてのアンケート結果に対して考察をすることで、スケジュール作成の手間を軽減できたことがわかり、業務の効率化を図ることができたという結論に至った。

次に、報告者の担当部分であるスケジューリングアルゴリズムの開発について説明した。アルゴリズムを考案する上で、必ず満たす必要がある項目と可能な限り満たすべき項目があった。可能な限り満たすべき項目の中で、同一の審査員が連続して審査を担当するという項目を満たすために、グルーピングとグループレイアウトという手法をとった。主査グルーピング、内部グルーピング、親密グルーピングという3種類のグルーピングを行うことによって、同一の審査員の連続性を高めたグループを作成することができた。それらのグループをスケジュールテーブルに配置していくことで、連続性を高めたスケジュールにすることが可能となった。

アルゴリズムでは初めにスケジュール可否判定を行い、必ずスケジューリングに失敗してしまう条件でスケジューリングを行おうとした場合を取り除く。スケジューリング可能である場合には初めにグルーピングを行う。次にグルーピングによって作成されたグループを可否判定時に可能であると判定された場合に作成される、空のスケジュールテーブルに挿入していく。この挿入していく処理をグループレイアウトと呼ぶ。ここまでがスケジューリングアルゴリズムの動作である。アルゴリズムを考案し開発を進めていく中で、2008年度のデータでは親密グループのサイズが大きくなりすぎ、スケジューリングに失敗するという問題が発生した。その問題に対し、親密グルーピングによるグループレイアウトに失敗した場合、親密グルーピングを行わずグループレイアウトを試すという対策と、親密グルーピングを行うか否かをスケジュール担当者を選択させるという対策を考案し、後者の対策に取り組んだことについて述べた。

最後に現在のスケジューリングアルゴリズムで実現することができなかった、特定課題研究の学生の扱いと審査員の連続性について述べ、報告者が考える対策を説明した。そして、その対策を実現することを今後の課題とした。

謝辞

本プロジェクトの遂行と本報告書の執筆にあたり、委託元教員である河辺徹准教授には、大変貴重なご意見やご指導、激励をいただきました。指導教員である三末和男准教授をはじめ、報告者が所属している研究室の教員であります、田中二郎教授、志築文太郎講師、高橋伸講師には研究室のゼミを通してご指導いただきました。高度 IT 人材育成のための実践的ソフトウェア開発専修プログラムの専任講師でおられる、菊池純男教授、駒谷昇一教授には、大学院生活 2 年間でシステム開発にかかわる多くのことを教えていただきました。この場を借りて深く御礼を申し上げます。

また、本プロジェクトを共に遂行したチーム HiTuRe の栗山大君、篠崎龍夫君にも深く御礼を申し上げます。この二人とプロジェクトを行い、3 人で協力し合えたからこそ本システムが完成しプロジェクトが成功したのだと実感しております。

最後に、24 年間報告者を支え育ててくれた両親に深く感謝し、全ての友人にも感謝申し上げます。本当にありがとうございました。