

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

スライドノートを活用した
講演字幕システムの実現
—外部システム連携機能の設計開発と品質管理—

楊暄妍

修士（工学）

（コンピュータサイエンス専攻）

指導教員 田中 二郎

2015 年 3 月

概要

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研究開発プロジェクトの成果をまとめたものである。

近年の高等教育の普及や裁判員制度の開始、高齢化の進展に伴う難聴者の増加などの社会環境の変化によって、より多様で専門的な情報保障が求められてきている。現在、研究発表会における情報保障は要約筆記である場合が多い。しかし、人的コスト・特殊な機材が必要といった制約が存在し、特に予算が限られる小規模の発表会などでは導入が難しい。

本プロジェクトは、研究発表会において発表内容がある程度決まっているという特徴を利用し、PowerPoint のノート機能を活用した講演字幕システムを開発する。講演者が講演原稿を PowerPoint のノート機能に書き込み、即興発言や質疑応答のみを現場で補助者が入力する。これより、特殊な機材を利用せず、現場での字幕入力の時間を縮小し、人的コスト・特殊な機材の制約の問題を解決でき、小規模な発表会などに要約筆記を導入できるという目的を実現できる。

本プロジェクトで開発される講演字幕システムには、スライドノートの抽出や字幕の表示といった基本機能とより多くの環境で利用するための拡張機能がある。筆者は、拡張機能の 1 つである、外部システムとの連携を担当した。

外部システムとの連携では、IPtalk との連携機能と音声合成との連携機能を実現した。IPtalk との連携機能は、講演大会の質疑応答時に字幕入力が追いつかない問題を解決するため、聴覚障害者への情報保障である IPtalk との連携である。音声合成との連携機能では、発話が困難な障害者が講演を行いやすいため、字幕を読み上げる音声合成エンジンとの連携である。

また、成果物の品質を確保するため、必要な機能が満たされるかどうかの確認のための機能テストとユーザ視点から問題点を発見するためのユーザビリティ評価実験を担当した。

結果として、聴覚障害者が実際に来場するのであれば、本システムを利用したいという評価が約 9 割の情報保障の実施者から得られて、字幕の見やすさが従来手法とそれほど変わらないという評価が過半数以上の聴覚障害者から得られた。以上の評価で、情報保障の実施のための運用コストを抑える目的を達成できることを判断した。ただし、操作について不便なところがあるので、将来的により使いやすいシステムの実現を目指している。

目次

1	序論	1
1.1	背景	1
1.2	目的	2
1.3	体制	2
1.4	報告書の構成.....	3
2	関連研究	4
2.1	筆談要約筆記.....	4
2.2	OHP/ OHC 要約筆記	4
2.3	パソコン要約筆記	4
2.3.1	ステノタイプ	5
2.3.2	IPtalk	5
2.3.3	ドラゴンスピーチ	6
2.3.4	UD トーク	6
2.4	まとめ	6
3	講演字幕システム「CaPPtitioner」	8
3.1	現状の課題に対する解決策.....	8
3.2	「CaPPtitioner」の概要.....	9
3.2.1	基本形態以外の 3 つ動作モード	10
3.2.2	聴講者への字幕配信	12
3.2.3	補助者の入力支援	12
3.3	「CaPPtitioner」の要件定義	12
3.4	機能要件に対する設計	13
3.5	非機能要件に対する設計	14
3.6	開発体制.....	16
3.6.1	開発環境.....	16
3.6.2	メンバの開発担当	16
4	IPtalk との連携機能	18
4.1	IPtalk との連携機能の概要	18
4.2	要件の詳細化.....	19
4.3	IPtalk との連携機能の設計	20
4.4	IPtalk 通信プロトコルの確認	20
4.4.1	公開されている情報	21
4.4.2	公開されていない情報.....	23

4.5	IPtalk との連携機能の実装	28
4.5.1	IPtalk 入力者との接続	28
4.5.2	IPtalk 入力の受信および表示	29
4.5.3	IPtalk からの取消	29
4.6	まとめ	30
5	音声合成との連携機能	31
5.1	要件の詳細化	31
5.2	音声合成との連携機能の設計	31
5.2.1	表示する文を音声合成で発声できる	32
5.2.2	発声終了が講演者に知覚できる	32
5.3	音声合成との連携機能の実装	34
5.4	まとめ	35
6	品質管理	37
6.1	品質とは	37
6.2	本プロジェクトの品質管理	38
6.2.1	基本方針	39
6.2.2	実施内容	39
6.3	機能テスト	40
6.3.1	機能テスト方針	40
6.3.2	機能テスト項目の作成	40
6.3.3	機能テストの実施	43
6.3.4	機能テストの振り返り	43
6.4	ユーザビリティ評価実験	44
6.4.1	代表的な評価方法	44
6.4.2	評価実験の計画	44
6.4.3	評価実験の実施	47
6.4.4	評価実験の考察	53
6.4.5	今後の展望	54
7	終わりに	55
	謝辞	56
	参考文献	57

図目次

図 2-1 : CART システム	5
図 2-2 : IPtalk イメージ図	6
図 3-1 : 各研究発表の構成	8
図 3-2 : 「CaPPTioner」の基本形態	10
図 3-3 : 補助者なし 2 画面	10
図 3-4 : 補助者あり 1 画面	11
図 3-5 : 補助者なし 1 画面	11
図 3-6 : 「CaPPTioner」の機能構成	14
図 3-7 : デスクトップ OS のシェア	15
図 4-1 : IPtalk 単体と IPtalk との連携イメージ図	18
図 4-2 : IPtalk との連携機能の全体イメージ図	19
図 4-3 : IPtalk 間の補助方法	19
図 4-4 : IPtalk に関するチャンネル情報	21
図 4-5 : Packet List	23
図 4-6 : IPtalk 入力者間の接続仕組み	24
図 4-7 : メンバ探索リクエストの packets	25
図 4-8 : リクエスト応答の packets	25
図 4-9 : メンバ確立の packets	26
図 4-10 : IPtalk の入力画面	26
図 4-11 : IPtalk 字幕受信および表示の packets	27
図 4-12 : IPtalk からの取消の packets	27
図 4-13 : UDP 実装のフロー図	29
図 4-14 : IPtalk 字幕 ID の保存のソースコード	29
図 4-15 : IPtalk 字幕 ID 管理のソースコード	30
図 5-1 : 部分文字列で字幕を出すイメージ図	33
図 5-2 : 実装説明図	34
図 5-3 : SpeakProgress のソースコード	35
図 5-4 : StateChanged のソースコード	35
図 6-1 : 品質の 4 つ種類	37
図 6-2 : 機能テストについての手順	40
図 6-3 : CFD 技法で抽出したテスト項目表例	42
図 6-4 : Redmine のテスト項目図例	42
図 6-5 : 1 回目のテスト記録	43

図 6-6 : 2 回目のテスト記録.....	43
図 6-7 : GQM 手法.....	45
図 6-8 : GQM 手法によって抽出したアンケート質問	46
図 6-9 : GQM によって抽出した質問とメトリクスの例.....	46

表目次

表 1-1 : チーム構成	2
表 3-1 : 機能・非機能要件	12
表 3-2 : OS と.NET Framework の対応状況図	15
表 3-3 : 開発環境.....	16
表 3-4 : 責任範囲の一覧.....	17
表 4-1 : CaptionManager の字幕操作メソッド	20
表 4-2 : UDP/IP を使用しているポート番号	22
表 4-3 : Packet List	23
表 4-4 : IPtalk との連携機能に関するクラス	28
表 5-1 : 音声合成機能に関するクラス	32
表 5-2 : Speech.cs の役割	34
表 6-1 : ソフトウェア品質特性・副特性(ISO/IEC25000)	38
表 6-2 : 聴講者向けの実験のタスク	47
表 6-3 : 講演者・補助者向けの実験のタスク	50

1 序論

本プロジェクトは、筑波大学大学院「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、専修プログラム）」における特定課題研究として、「スライドノートを活用した講演字幕システムの実現」というテーマのもと、システムの開発を行ったものである。プロジェクトは、著者を含む専修プログラム所属の学生 5 人のチームで進行した。本報告書はこのプロジェクトの実施内容、および著者のプロジェクトへの貢献についてまとめたものである。

1.1 背景

聴覚障害者とは、聴覚に障害のある（耳が不自由な）人のことである。この聴覚障害者には聾者、軽度難聴から高度難聴などの難聴者、成長してから聴覚を失った中途失聴者、加齢より聴力が衰える老人性難聴者が含まれる。平成 18 年度の日本国内の調査 [1] で分かっている日本の聴覚障害者数は、18 歳以上で 34 万 3000 人、18 歳未満で 15800 人である。全体では、およそ 1000 人に 3 人が聴覚障害者である。この数字は身体障害者手帳を取得した人と、「身体障害者福祉法」が掲げる障害に該当する人を対象に調査した結果である。加齢による衰えで耳が遠くなった「老人性難聴」を抱える人々は上記の調査対象外である。また、「人と比べて聞こえづらい」という判断は本人には難しいため、自身の聴覚機能の低さに気づかず手帳を取得していない人もいる。そのため、実際にはもっと多くの人が聴覚障害に悩まされている可能性が高い。聴覚障害者の主なコミュニケーション方法には、腕から指先までを使って表す手話、指で文字を表す指文字、唇の動きで表す口話法、筆談・要約筆記などが挙げられる。特に、補聴器と併用して音声言語に基づいて行う口話法あるいは訓練する必要がある手話法は、中途失聴・難聴者は幼少期に習得していないため扱うことが難しい。また、聴覚障害者が必要とするコミュニケーション手段は、筆談・要約筆記を必要とする方が約 3 割（約 10 万人）となっていて、口話・手話通訳を必要とする方の約 2 割（約 7 万人）を上回っている [2]。これは、日常のコミュニケーション手段に文字を伝達方法として使用する人が多いことを表している。発話内容を短い文でテキスト化する要約筆記が、手話同様聴覚障害者のコミュニケーション支援として機能している状況を示しているといえる。

しかし、コミュニケーション支援事業を担う派遣登録者の状況（厚生労働省調べ：平成 19 年度末実登録者数）は、手話通訳派遣の実登録者数が約 2 万人であるのに対し、要約筆記派遣の実登録者数は約 1 万人となっていて、手話通訳派遣者の半数に留まって

いる。また、近年の高等教育の普及や裁判員制度の開始、高齢化の進展に伴う難聴者の増加などの社会環境の変化によって、聴覚障害者が要約筆記に係る場面も増加してきている。こうした新しいニーズの増加によって、難聴者等のコミュニケーション支援（要約筆記）には、より多様で専門的な対応が求められてきているので、要約筆記支援の現状を改善することが急務となっている。

1.2 目的

本プロジェクトでは、様々な状況に対応して運用可能することができる要約筆記システムの提案・開発を行い、要約筆記支援の現状を改善することが本プロジェクトの目的である。

この目的を達成するため、既存の要約筆記が広く実施できない原因を抽出し（第2章で詳しく述べる）、この原因について検討した上で、開発システムを提案する。また、開発システムに対してシステム利用者・聴覚障害者の評価実験を実施し、得られた結果から将来的な対策についても議論する。

1.3 体制

本プロジェクトのチーム構成は次の通りである。

表 1-1：チーム構成

顧客
秋川友宏准教授
課題担当教員
秋川友宏准教授
チーム名
LOVEPPT
チームメンバ名
顧毅捷
楊暄妍
落合遥堂
後藤慎也
横山快

1.4 報告書の構成

各章の概要を述べる。2 章では聴覚障害者に対する情報保障に関連した研究について記載する。3 章では提案システム「CaPPTioner Ver.UT1」(以下、CaPPTioner)について述べる。4 章、5 章では筆者の担当範囲である外部システムとの連携に関する設計・実装に関して説明する。6 章では筆者の担当したプロジェクトの品質管理について述べる。

2 関連研究

本章では、要約筆記の関連研究について述べる。要約筆記の実施形態 [3]は、情報保障を必要とする人々の条件や、場所や機材上の条件により、最善の実施形態が採用される。主な実施形態は、次の通り。

2.1 筆談要約筆記

情報保障上としての筆談は、ノートやホワイトボードに文字を書いて情報を伝える。筆記具のみを用いるため、場所を選ばないオーソドックスな要約筆記として広く利用されている。現在では、ノートの代わりにノートパソコンや特殊なキーボードとワープロソフトを使って、1人の入力者が入力作業を行う。手書きのノートテイクと違い、字幕出力機器によっては、多数の人に見せることが出来るが、入力者の負担が大きいので単独での長時間の実施は困難である。

2.2 OHP/ OHC 要約筆記

OHP 要約筆記は、オーバーヘッドプロジェクタを用いる。屋内でスクリーンが使用できる場合、少ない機材で運用できる。

ロールと呼ばれる巻物状の OHP シートに、フェルトペンで文字を書き、それをスクリーンに映し出す。筆記者は光源を注視するため、専用のサングラスが必要となる。この他に滑りのよい手袋を着用する。少なくとも筆記者のほかに、ロールを進める補助者、その他の作業を行う補助者が必要となる。

OHC 要約筆記は、屋内でオーバーヘッドカメラ（書画カメラ）を用い、紙に筆記したものをスクリーンに映し出す。OHP 要約筆記と比べて、強烈な光源がないため、筆記者の身体的負担が少なく、紙などによる既存の資料をそのまま映すことができる。また、資料の多い講演会では融通が利く。半面、機材が高価である。

ロール状の OHP シートに原稿を事前に書いておき、講演に合わせて巻き取って投映することもある、これは前ロールと言われる。

2.3 パソコン要約筆記

パソコン要約筆記は、パーソナルコンピュータをプロジェクタに接続し、音声情報をパソコンにテキスト入力することで、テキストをスクリーン上に提供する要約筆記である。入力システムの単語登録機能により作業効率が向上し、他の要約筆記に比べ圧倒的

に多い情報提供量が特徴である。文章の入力方法には、単独で入力する「1 人入力」のほか、複数の人で 1 文を完成させてゆく「連係入力」、音声認識による入力といった方法がある。以下では、代表的な既存システムについて述べる。

2.3.1 ステノタイプ

ステノタイプ [3]は、ステノグラフ（＝速記）をするための特別なタイプライターで、24 のキーがついているキーボードである。ステノタイプの入力装置とパソコンを組み合わせたものが CART システム(図 2-1)と呼ばれる。生放送番組向けの字幕制作において多くの場合、タイプによる付与を行っている。この手法では、早いスピードで入力でき、生放送と同時に字幕を生成できるという利点がある。一方、熟練するには専門の教育を受ける必要があり、また、ステノタイプは特殊な機械なので購入価格も高いという問題がある。



図 2-1 : CART システム [4]

2.3.2 IPtalk

IPtalk [5]は、パソコンを使い、リアルタイムに文字を入力し、事前に準備した文章を表示することで、聴覚障害者のコミュニケーションを助ける情報保障（パソコン文字通訳、パソコン要約筆記、PC テイクなど）用のソフトである。

IPtalk の仕組みは図 2-2 を示した通り。2 人のパソコン要約筆記者がパソコン要約筆記を行っているところである。しかし、2 人だけで長時間の情報保障を続けることは負担が大きいため、実際のパソコン要約筆記の現場では、少なくとも 2 組がそれぞれの入力用パソコンを LAN で接続し、交替で入力にあたるのが通例である。

OHP と同じく要約筆記としての用途が主で、前ロール原稿をテキストファイルで準備し読み込ませる手法を用いて、字幕を表示することも可能である。



図 2-2 : IPtalk イメージ図

2.3.3 ドラゴンスピーチ

ドラゴンスピーチ [6]は、ASCII 社において開発された、約 100 万語の音声認識辞書を備え文章を声で読み上げるだけで素早く入力できるというソフトウェアである。音声認識による文字入力は、特殊な技術が無くても話速に近い速度で入力できるというメリットがある。しかし、音声認識は誤認識が生じる可能性が高く、要約筆記では 100%に近い高精度の認識が求められるので、話者の声を聞き取った入力者が、別室でマイクに入力していくという方式が主である。

2.3.4 UD トーク

UD トーク [7]とは、Shamrock Records 株式会社より提供されている、「会議のユニバーサルデザイン」をスムーズに取り入れるための iOS アプリである。5 人までのユーザが連携し、発言は音声認識技術で自動的に文字化して表示、キーボードによる文字入力もサポートできる。入力内容は参加者の端末に自動的に配信され、手元で内容を確認することができ、誰でも参加しやすい会議を実現し、ユニバーサルデザイン化を図っている。特に、キーボードで入力している過程がお互いに把握できるので、簡易的な要約筆記を行うこともできる。しかし、お互いの入力している内容を確認しづらいので、「速さ」を求めるリアルタイムの要約筆記は困難である。

2.4 まとめ

本節では、本プロジェクトによって要約筆記が広く実施することが可能であるかについて考察する。

要約筆記は、聞いてから文字にするという作業なので時間的な遅れが出る。従って、少しでも遅れないために「速さ」が求められている。そして、字幕を読みやすくため、正確な文字、箇条書き、まとめ方、改行の位置などの書き方も訓練する必要がある。ま

た、ステノタイプなど特殊な機材の入力方法は複雑であるため、マスターするためには特別な訓練が必要である。それ故、熟練者の養成が難しく、その人数は少ない。

また、要約筆記の現状から見ると、訓練した熟練者の存在だけではなく特殊な機材も補助のために必要条件になる入力法もある。しかし、小規模な講演会では、コストの問題によってこの条件を満たすことが困難である。

上記の原因が、要約筆記のより広い実施を妨げていると考えた。そのため、本プロジェクトでは「いつでも」「どこでも」利用可能な字幕情報保障を実施するために、運用環境に依存せず一般人でも使用可能な要約筆記システムを提案する。

3 講演字幕システム「CaPPtioner」

3.1 現状の課題に対する解決策

前章で述べた通り既存の要約筆記システムでは、多数の熟練した補助者や特殊な機材が必要である。

小さな研究発表会やグループ発表会ではコストの制約があるので、なかなか情報保障が実施できない。しかし、これらの機材や人材の問題が解決すれば、より多くの環境で聴覚情報保障が実施できると考えている。

本プロジェクトは、小さな研究発表会で利用することを前提として、簡易的に要約筆記が実施できるシステムの開発を行う。

ここで、まず研究発表会の特徴について述べる。

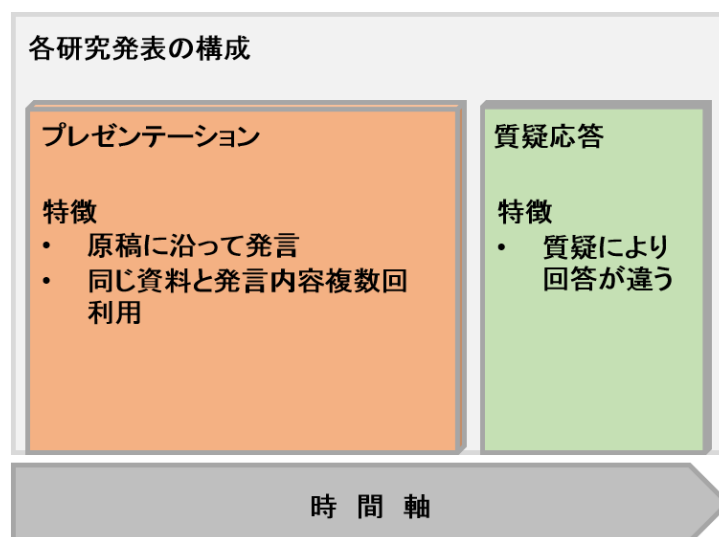


図 3-1：各研究発表の構成

図 3-1 は研究発表の簡単な構成である。研究に関するプレゼンテーションでは、同じ講演資料が複数回の研究発表会で使い回され、講演内容も事前準備した原稿に沿って発言されることがほとんどである。

既存の要約筆記は、現場の発言にリアルタイムの文字入力追いつかないので熟練した入力者や特殊な入力機材で補っている。しかし、発表原稿を利用して入力を補う事ができれば、よりコストを削減した情報保障につながると考えた。

発表原稿を準備する場所は以下の3つが考えられる。

(1) 外部ファイル

従来の字幕システムで前ロールを利用する方法では、外部ファイルを利用していた。しかし、外部ファイルでは原稿と発表資料が同期しないので、補助者が発言している内容に合わせて字幕を表示することが非常に大変であった。

(2) 講演資料の本編内

発表用スライドなどの講演資料の中に、講演原稿を追加する方法である。しかし、スライド1枚あたりの講演原稿は一言ではない。スライドを多数コピーし、1文ずつ追加するために、スライドの枚数が非常に多くなってしまう。また、スライドの内容を修正する場合、コピーしたページを全て修正しなければならず原稿を準備するための負担が大きい。

(3) スライドノート

スライドノートとは、Microsoft PowerPoint のメモ機能であり、スライドショー中に発表者の端末に表示される文章である。このメモ機能を利用して講演原稿を入力する方式にする事で、発表資料と並行して講演原稿を入力できるというメリットがある。

また、講演補助者も、少ないノート内容の中から、話者が発言している内容を発見することが簡単にできるようになる。

以上の理由で、本プロジェクトでは、発表原稿として発表資料のノートが適切であると考えている。

また、スライドを使うので、特殊な機材を利用せず、PC 上で行える。さらに、PC、スクリーン、プロジェクタの数を調整できるよう設計する事で、より広く情報保障が実施できると考えている。

3.2 「CaPPTioner」の概要

前節の方針に沿って、講演字幕システム「CaPPTioner」を提案する。既存の要約筆記と同じ形式で、「CaPPTioner」も講演会場にスクリーンが2枚あり、字幕表示の補助をする人(補助者)がいる場合を基本形態として動作する。「CaPPTioner」の基本形態は図 3-2 である。

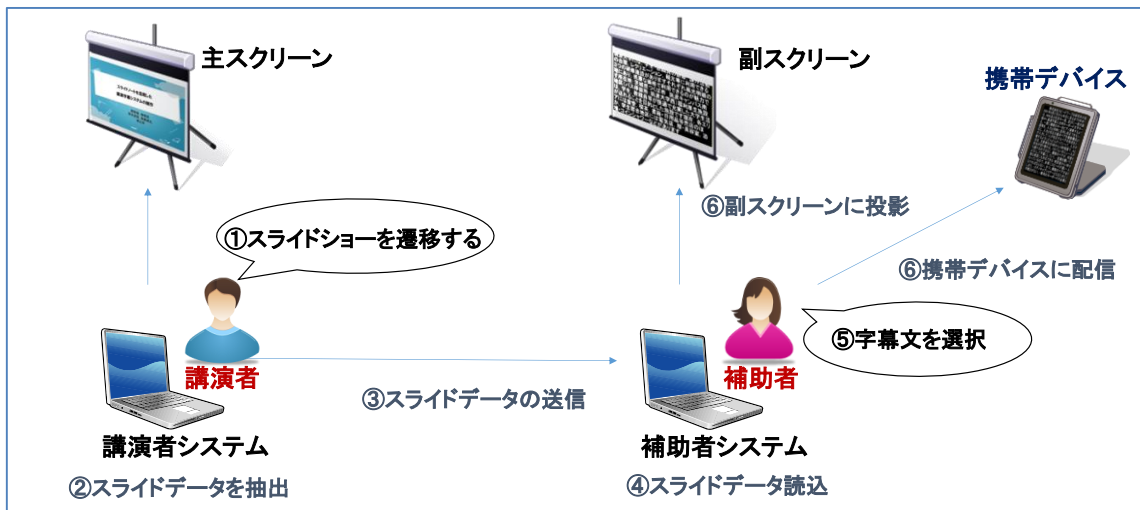


図 3-2 : 「CaPPTioner」の基本形態

3.2.1 基本形態以外の3つ動作モード

補助者の人数とスクリーンの数が調整できるように、基本形態以外に、補助者なし2画面、補助者あり1画面、補助者なし1画面の3つのモードを実装する。

(1) 補助者なし2画面

図 3-3 に示すように、補助者いない時、講演者システムから受信したスライドノートを補助者システムが自動でスクリーンに映し出す。

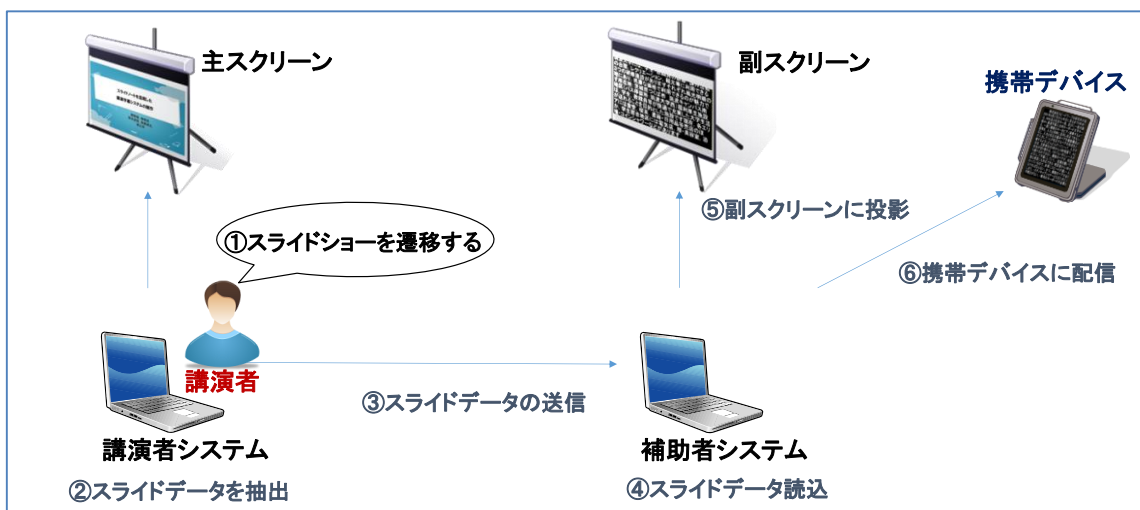


図 3-3 : 補助者なし2画面

(2) 補助者あり1画面

このときは、補助者システムでスライドデータを受信した後、字幕データを一度講演者システムに返送する。それを講演者システムが受け取り、1枚のスクリーンにスライドショーと字幕を同時に表示する。その構成は図 3-4 に示す。

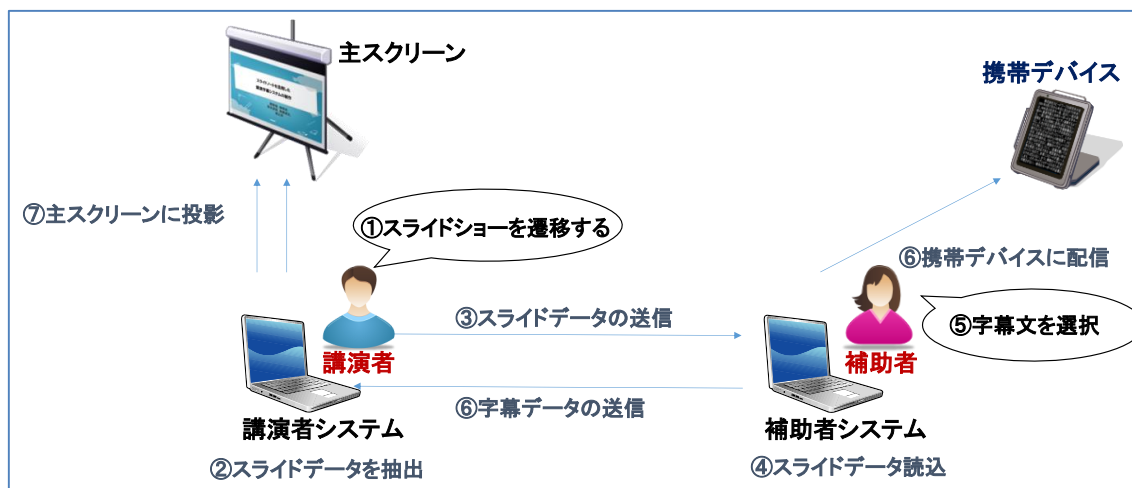


図 3-4：補助者あり 1 画面

(3) 補助者なし 1 画面

補助者がいない、かつ、副スクリーンがない場合を補助者なし 1 画面モードとし、その構成を図 3-5 に示す。

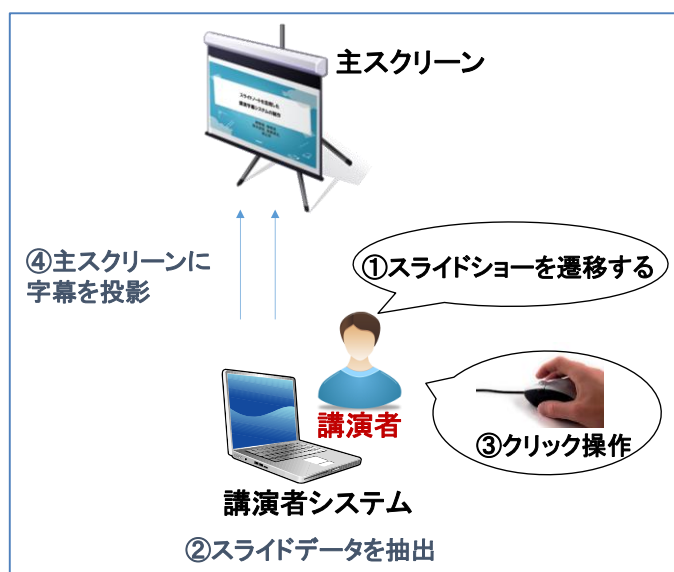


図 3-5：補助者なし 1 画面

3.2.2 聴講者への字幕配信

研究発表会を行う会場に、常に十分な大きさの字幕スクリーンがあるとは限らない。例えば広い会場に小さなスクリーンが1台しか無い場合、後ろの方に座った人は字幕が見えない。また、字幕用スクリーンは講演に合わせて字幕を表示する事になるが、読むスピードが字幕に追いつかない人も居る。見逃した部分のバックログを読みたい場合もある。そこで、聴講者の持つ情報端末に字幕を配信できるようにする。

3.2.3 補助者の入力支援

図 3-1 を示した通り、研究発表会では質疑応答がある。

質疑応答では、原稿が事前準備できないので、即興発言への対応ができるよう、2つの入力支援機能を実装する。

(1) 予測変換

講演内容に基づいた予測変換を行う。スライドに含まれる文とスライドノートの文を利用して、補助者に候補を提示する。これは、発言内容は講演本編中の語句が使われる可能性が高いためである。

(2) IPtalk との連携

IPtalk は各種学会や障害者スポーツ大会などで導入実績がある字幕システムである。2人連携して補助する方法である。2人だけで長時間の情報保障を続けることはできないので、多く人が交代補助することが通例である。そのため、IPtalk のみで利用する場合、コストが高い。

そこで CaPPTioner と連携し、質疑応答の時間だけ IPtalk を利用すれば、同じ補助者の人数で2倍の講演に対して情報保障が実施できる。

3.3 「CaPPTioner」の要件定義

本節では、「CaPPTioner」の要件定義について説明する。「CaPPTioner」の機能要件は4つ、非機能要件は3つである。表3-1でまとめる。

表 3-1 : 機能・非機能要件

機能要件		
番号	機能要件内容	説明
K1	スライドノートを字幕として表示できる	講演者が予め用意した、スライドノートを活用して、字幕として表示ができる。また、字幕の表示は、

		講演者や補助者の好みにより自由に設定することができる
K2	補助者の有無、副スクリーンの有無に関わらず運用できる	補助者がいない場合でも最低限の情報保障が求められる。また字幕表示専用の副スクリーンがなくても、主スクリーンのみで情報保障ができる
K3	聴講者の携帯情報端末で字幕を閲覧できる	表示専用の副スクリーンがない場合や、副スクリーンに表示された字幕が見つからない場合に対応するため、聴講者の携帯情報端末で字幕を閲覧できる。また、スクリーンの字幕を見逃した場合でもその部分のバックログを閲覧することができる
K4	原稿のない発言を字幕として表示できる	講演中の発言の大部分は原稿通りの発言であるが、それ以外の即興発言にも対応し、字幕として表示することができる
非機能要件		
番号	非機能要件内容	説明
h1	より多くの環境で動作することができる	システムが利用される環境は様々である。そのため、「CaPPTioner」はより多くの環境で動作することを可能とする
h2	システムのインストール不要	他者の端末を利用する場合もある。インストール作業による時間ロスが発生したり、レジストリの変更を行わず済むよう、実行ファイルを USB フラッシュメモリに入れ、そのまま起動できる、といった使い方も可能とする
h3	講演に副作用を及ぼさない	システムの操作や、システムの問題によって、スライドショーを止めないようにする。例えばシステムが強制終了しても、スライドショーそのものまでは停止せず、最悪の場合でも講演を続けることができるよう設計するものとする

3.4 機能要件に対する設計

機能要件に対する設計に対応している機能構成は図 3-6 のようにまとめた。

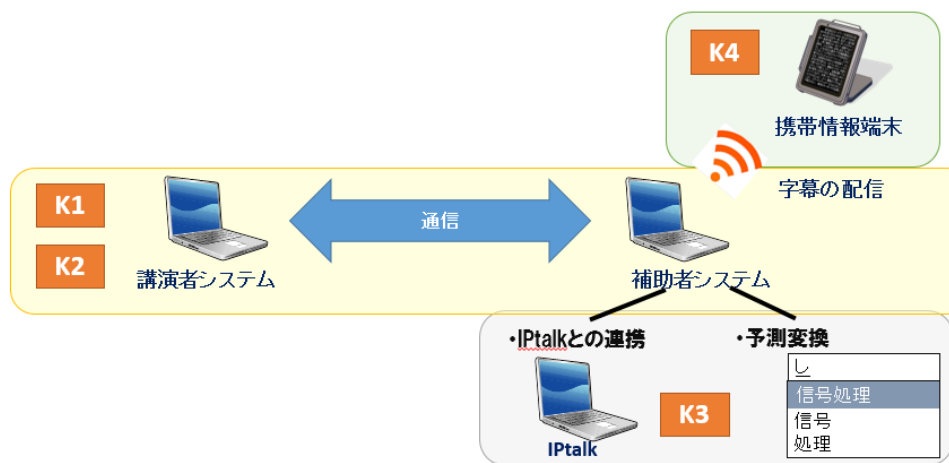


図 3-6 : 「CaPPTioner」の機能構成

「CaPPTioner」は、講演者システムと補助者システムの2つのシステムで構成する。そして、携帯情報端末への字幕配信、入力支援である予測変換と IPtalk との連携は補助者システムの一部として構成する。

3.5 非機能要件に対する設計

- ・ h1 より多くの環境で動作することができる

Net Applications の調査による 2014 年 11 月時点でのデスクトップ OS のシェアは図 3-7 に示す通りとなっている。Window Vista 以降のすべての Windows OS のバージョンを合わせたシェアは、約 78%であることがわかる。Windows OS 以外の OS にも対応することが望ましいが、今回のプロジェクトでは Windows Vista 以降の Windows OS で動作するように設計した。

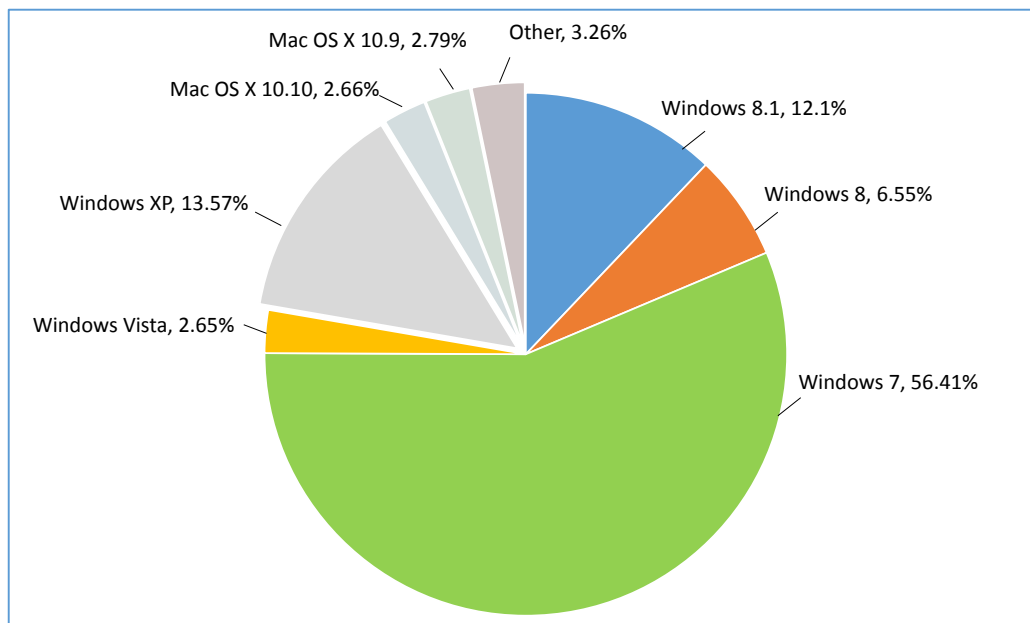


図 3-7 : デスクトップ OS のシェア

また、表 3-2 に示すように、Window Vista 以降のすべての Windows OS のバージョンで動作するのは、Net Framework2.0～4.0 であることがわかる。Windows Vista においてプリインストールされているのが.NET Framework3.0 であるため、対象となる全ての動作環境で新たに.NET Framework をインストールする必要が無いよう、.NET Framework 3.0 を使用するものとした。

表 3-2 : OS と.NET Framework の対応状況図

	XP	Vista	7	8	8.1
.Net Framework 1.0	○				
.Net Framework 1.1	○	○			
.Net Framework 2.0	○	○	○	○	○
.Net Framework 3.0	○	○	○	○	○
.Net Framework 3.5	○	○	○	○	○
.Net Framework 4.0	○	○	○	○	○
.Net Framework 4.5		○	○	○	○

また、Windows XP で動作することがサポートされている中で最も古い PowerPoint のバージョンは PowerPoint 2000 である。よって、PowerPoint2000 以降のバージョンで運用可能とする。

- ・ **h2 インストール不要**

h2 の非機能要件を満たすため、CaPPTioner は exe ファイルの形式で動作するように設計した。また、前項で述べた入力補助のための形態素解析機能も、外部のシステムを端末にインストールするのではなく、CaPPTioner 内に実装する。これによってシステム本体を携帯可能な記憶デバイスで持ち歩き、そのまま起動することも可能とする。

- ・ **h3 講演を止めない**

h3 の要件を満たすため、CaPPTioner はマルチスレッドで動作するように設計した。お互いに影響を与えないように、システムの制御によってスライドショーが動作するのではなく、スライドショーは通常通りの動作をする事になる。講演とは別スレッドで CaPPTioner が動作すれば、講演に影響を与えることはない。

3.6 開発体制

3.6.1 開発環境

開発環境は表 3-3 に示した通りである。

表 3-3 : 開発環境

OS	Microsoft Windows8.1
プログラミング言語	C#
統合開発環境	Visual Studio 2012
フレームワーク	.NET Framework 3.0
Office 開発者ツール	Visual Studio Tools for Office
バージョン管理システム	Git
プロジェクト管理ソフトウェア	Redmine

3.6.2 メンバの開発担当

このシステムの開発において、本プロジェクトでは開発担当・マネジメントの責任範囲を分け、それぞれで設計・実装・プロジェクト管理を進めることとした。表 3-4 にそれぞれの機能の主な開発担当をまとめる。

表 3-4：責任範囲の一覧

メンバ名	責任範囲	機能要件				非機能要件		
		K1	K2	K3	K4	h1	h2	h3
顧毅捷	補助者システム、通信プロトコル	○	○	○	○	○	○	
楊暄妍	外部システムとの連携、品質管理				○	○	○	
落合遥堂	講演者システム、進捗管理	○	○			○	○	○
後藤慎也	即興発言の入力支援				○	○	○	
横山快	字幕配信システム			○		○	○	

4 IPtalk との連携機能

4.1 IPtalk との連携機能の概要

3.3 節において定義された要件のうち、「K4. 即興発言に対応できる」を実現するために、「CaPPTioner」には予測変換を揃えた即興発言入力機能をもたせており、他のメンバーが担当している。これに対し、IPtalk による即興発言の入力を可能とする方法も考えられる。筆者は IPtalk との連携機能を担当した。

本節では、まず IPtalk との連携機能の目的と全体構成イメージを述べる。

3.1 節の図 3-1 に示した通り、研究発表会の各研究発表ではプレゼンテーションが終わった後、質疑応答がある。質疑応答は、原稿が事前準備できないので、入力が話速に追いつかない時もあると考えている。そのため、質疑応答に対応できるよう、IPtalk 熟練者が IPtalk を使用して入力補助する方法を考えた。

IPtalk は、2.3.2 節で説明したように、通常 4 人 1 組で入力にあたるのが通例である。従って、実際の現場補助では、人的・予算的に広く継続的な実施が難しい。もし、図 4-1 のように、字幕システムと連携すれば、発表部分は字幕システムを使い、質疑の時だけ IPtalk を使うことで、同じ人数で倍の補助できる。そうすることで、人的・予算的に IPtalk 単体より広く実施できると考えている。

以上の 2 つの理由で、「CaPPTioner」と IPtalk を連携することにした。

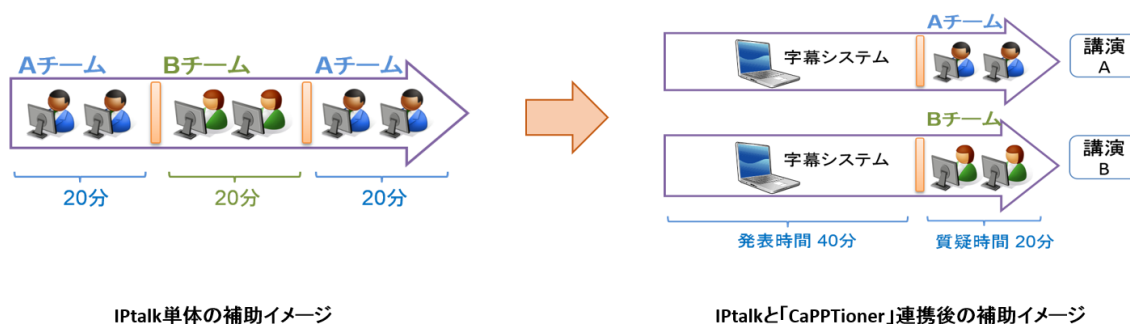


図 4-1 : IPtalk 単体と IPtalk との連携イメージ図

IPtalk との連携機能の全体構成イメージを図 4-2 に示す。

IPtalk と「CaPPTioner」は、同じ LAN 上で接続する。接続後、IPtalk を利用する補助者が入力する字幕を、字幕システムが中継してスクリーンに表示される。

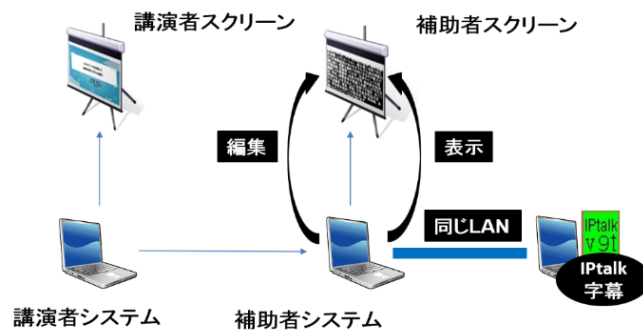


図 4-2 : IPtalk との連携機能の全体イメージ図

4.2 要件の詳細化

K4 要件について、IPtalk との連携機能を実現するために以下の要素が必要である。

- (1) IPtalk 入力者との接続
- (2) IPtalk 入力を受信および表示
- (3) IPtalk からの取消

各要素に関して、以下に解説する。

(1) IPtalk 入力者との接続

IPtalk 間の補助方法を、図 4-3 に示す。IPtalk では、同じ入力班に入っている入力者をメンバと呼ぶ。同じ入力班の中の 2 人の入力者がパートナーになり、文の前半と後半を分担して連携入力を行うことで、入力スピードを高めている。同じ入力班のメンバがお互いの入力内容を見ることができる。

図 4-3 中、入力者 A さんと入力者 B さんがパートナーになり、連携入力を行う。この場合、A さん、B さん、C さんが入力内容を見る事ができる。

従って、IPtalk の入力内容を受け取って字幕スクリーンに表示するためには、IPtalk 入力者と同じ入力班のメンバになる必要がある。

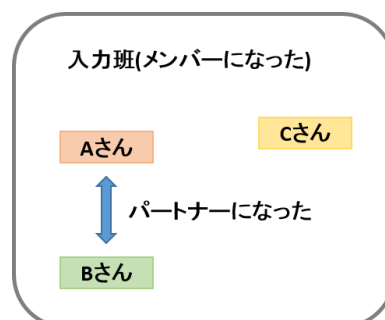


図 4-3 : IPtalk 間の補助方法

(2) IPtalk 入力の受信および表示

IPtalk の入力内容がスクリーンに表示できることが重要である。従って、IPtalk の入力を受信し、「CaPPTioner」への入力と同じように扱えるようにする。

(3) IPtalk からの取消

IPtalk は、便利な字幕補助を可能にするため、非常に多くの機能が盛り込まれている。その中、送り出した最後の文を取り消す undo 機能がある。undo 機能では、IPtalk で一度「Enter」で送ってしまった内容が間違いだった場合、「F9 キー」で表示部に流した文を入力部などに戻せる。「F9 キー」の undo は、直前の 5 度の操作まで有効である。「F9 キー」を押すと、表示部から直前に表示された文が消え入力部の行頭に入る。他のパソコンの表示部からも文は消える。

undo 機能を必要と考え、操作が字幕に反映されるようにした。

4.3 IPtalk との連携機能の設計

「CaPPTioner」では、表示される字幕を CaptionManager で管理している。字幕に対する、「字幕の追加」、「字幕の編集」、「字幕の削除」などの処理がこのクラスで行われている。CaptionManager の字幕操作は表 4-1 の通りである。

表 4-1 : CaptionManager の字幕操作メソッド

メソッド名	内容
Append	字幕を追加する。字幕 ID を自動で割り当て、その ID を返す
Remove	引数で与えられた ID の字幕を削除する
Update	引数で与えられた ID の字幕を更新する。与えられた ID が未登録であれば、新しく字幕を追加する

従って、IPtalk に成りすまして通信を待ち受け、これらのメソッドを呼ぶことで IPtalk との連携は実現できる。しかし、IPtalk に関する情報の一部は IPtalk9i のマニュアルで公開されているが、公開されている情報だけでは不明な点があった、そのた

「生パケット」を確認可能な、パケットキャプチャツール「Wireshark」 [8] [9]を導入し、IPtalk の通信プロトコルを解析することにした。

これについては次の節で詳細に述べる。

4.4 IPtalk 通信プロトコルの確認

本節では、IPtalk の仕組みについて公開されている情報と、パケットキャプチャツール「Wireshark」を用いて公開されていない情報に対する確認する方法および解析の結果を述べる。

4.4.1 公開されている情報

「IPtalk」の仕組みについて公開されている情報が 2 つある。以下は、それぞれについて述べる。

A) チャンネル

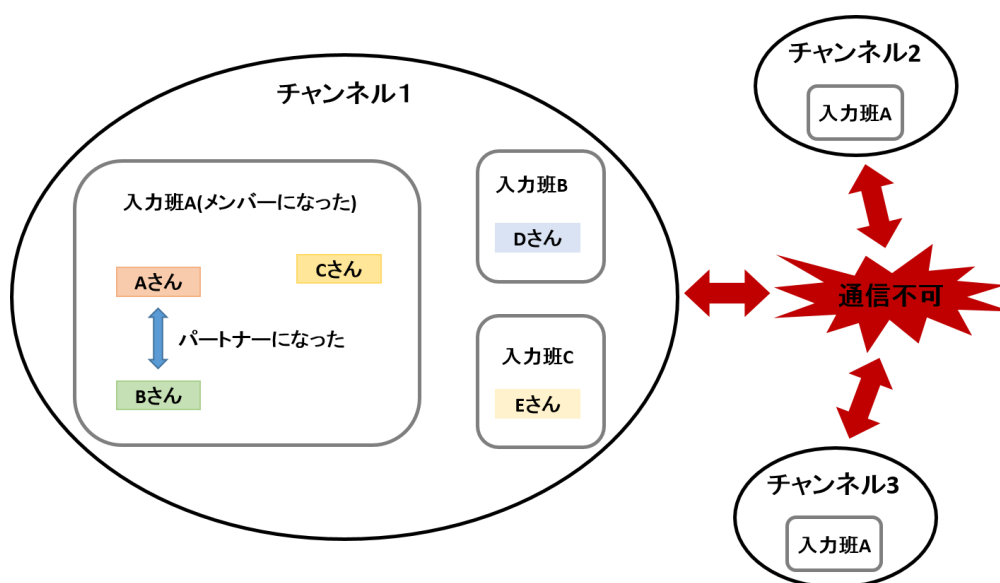


図 4-4 : IPtalk に関するチャンネル情報

図 4-4 は IPtalk 間の通信イメージ図である。同じ LAN に接続している IPtalk 入力者は、1~12 チャンネルの内のどれかのチャンネルを使って通信する。違うチャンネルは全く通信できない。従って、「CaPPTioner」が IPtalk 入力者と接続する時、同じチャンネルに入らなければならない。IPtalk 入力者との接続する前、チャンネルを選択できるように設計および実装した。

B) UDP/IP に関するポート番号

IPtalk は、UDP/IP を使用している。表 4-2 は、IPtalk9i の電子マニュアルで公開されているチャンネル 1 のポート番号である。チャンネル 2 以後、チャンネルが 1 増えることに 100 ずつ増やした。

表 4-2 : UDP/IP を使用しているポート番号

ポート	ウィンド	機能
6711	メインウィンド	表示部
6712	メインウィンド	モニター部
6713	メインウィンド	修正通信
6714	I Pアドレスリスト	IP 探索、遅延
6715	I Pアドレスリスト	教えて、遅延
6716	新カラオケリモコン	カラオケ
6717	確認修正パレット	同期
6718	メインウィンド	M 探索
6719	メインウィンド	P 通信
6720	temp roll/main(9g)	ロール連動
6721	メインウィンド	8 人モニター
6722	メインウィンド	BC
6723	メインウィンド	undo
6724	メインウィンド	離脱
6725	練習ウィンド	お手本
6726	練習ウィンド	残り時間
6727	連絡窓	連絡用
6728	カラオケリモコン	カラオケ
6729	メインウィンド	リレー
6730	確認修正パレット	確認修正
6731	お知らせウィンド	お知らせ
6732	確認修正パレット	モニター
6733	スライド	ページ送信
6734	テンプレート前ロール	モニター(下)
6735	テンプレート前ロール	モニター(上)
6736	ルビ送信	ルビ付け
6737	手書き	ペンの x,y の座標
6738	I Pアドレスリスト	ネットワーク遅延
6739	訂正送信ウィンド	訂正送信

電子マニュアルに「M 探索」と記載されているのが、メンバ探索である。「CaPPTioner」では、ポート番号 6711、6718、6722、6723 を利用する。

4.4.2 公開されていない情報

IPtalk の通信プロトコルに関する不明な部分は、「Wireshark」を利用して解析した。

A) 「Wireshark」によるバースエンジニアリング

「Wireshark」は UNIX/Linux や Mac OS X、Windows で動作するネットワークプロトコルアナライザである。コンピュータがネットワークを介して通信するパケットを収集し、その内容や送信先などを解析することができる。図 4-5 は Packet List 画面である。



The image shows a screenshot of the Wireshark Packet List pane. A red rectangular box highlights the header row of the packet list. The header row contains the following columns: No., Time, Source, Destination, Protocol, and Info. Below the header, several packets are listed, including DB-LSP-IDropbox LAN sync Discovery Protocol and SSDP NOTIFY * HTTP/1.1.

No.	Time	Source	Destination	Protocol	Info
18	1.721495	192.168.1.102	255.255.255.255	DB-LSP-IDropbox LAN sync Discovery Protocol	
19	1.721785	192.168.1.102	192.168.1.255	DB-LSP-IDropbox LAN sync Discovery Protocol	
20	1.721991	192.168.1.102	255.255.255.255	DB-LSP-IDropbox LAN sync Discovery Protocol	
97	15.407229	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
98	15.423598	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
99	15.437610	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
103	15.454282	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
104	15.471797	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
105	15.487447	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
106	15.504295	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
107	15.521551	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
108	15.537590	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
109	15.554352	192.168.1.1	239.255.255.250	SSDP	NOTIFY * HTTP/1.1
154	23.409936	192.168.1.102	192.168.1.255	UDP	Source port: 6722. Destination port: 6722.

図 4-5 : Packet List

図 4-5 赤枠で囲った部分の説明は以下の表 4-3 の通り：

表 4-3 : Packet List [10]

項目	説明
No	採取したパケットの順番を示す
Time	1 番目のパケットから経過した時間を秒で示す。「View」から「Time Display Format」を選択すると、絶対時間や、1 つ前のパケットからの間隔等に表示を切り替える事が出来る
Source	送信元の IP アドレスを示す。IP アドレスがない場合は Mac アドレスが表示される
Destination	送信先の IP アドレスを示す。IP アドレスがない場合は Mac アドレスが表示される

Protocol	送信先の IP アドレスを示す。IP アドレスがない場合は Mac アドレスが表示される
Length	送信先の IP アドレスを示す。IP アドレスがない場合は Mac アドレスが表示される
Info	送信先の IP アドレスを示す。IP アドレスがない場合は Mac アドレスが表示される

B) IPtalk 入力者との接続

図 4-6 は、IPtalk 入力者間の接続の仕組みを示す。

送信元の IPtalk が 6722 ポートでブロードキャストをして、受信元の IPtalk が情報を取った後 6718 ポートで返信する。そして、送信元の IPtalk がこの返信を受けた後 6718 ポートでもう 1 回返事をする。このように、3 回の通信を経て、IPtalk 入力者間の接続を確立する。

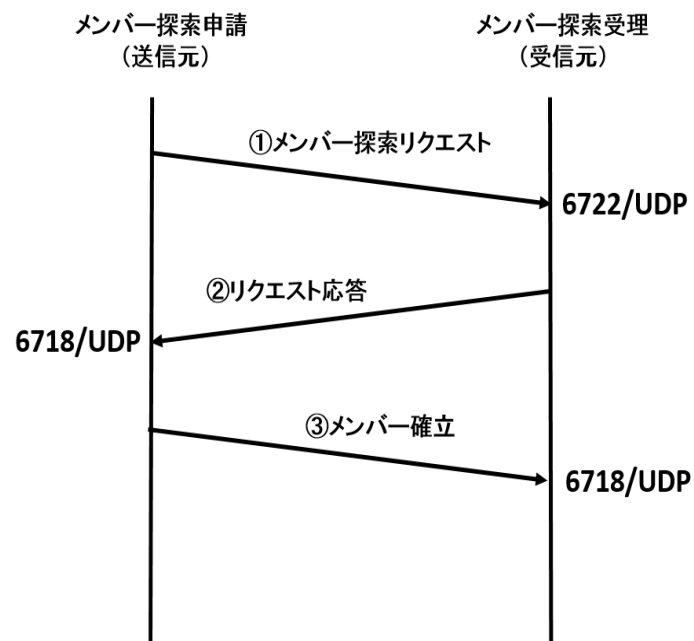


図 4-6 : IPtalk 入力者間の接続仕組み

「Wireshark」による解析の結果、図の中の①～③の通信でメンバを確立していることが分かった。それぞれのパケットの内容は以下の通りである。

①メンバ探索リクエスト (ブロードキャスト)

```

⊕ Frame 5949: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
⊖ Ethernet II, Src: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊖ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... 1 ..... = IG bit: Group address (multicast/broadcast)
      .... 1 ..... = LG bit: Locally administered address (this is NOT the factory default)
  ⊕ Source: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
    Type: IP (0x0800)
⊕ Internet Protocol, Src: 192.168.11.8 (192.168.11.8), Dst: 192.168.11.255 (192.168.11.255)
⊖ User Datagram Protocol, Src Port: 6722 (6722), Dst Port: 6722 (6722)
  Source port: 6722 (6722)
  Destination port: 6722 (6722)
  Length: 36
  ⊕ Checksum: 0x6d72 [validation disabled]
⊖ Data (28 bytes)
  Data: 3139322e3136382e31312e382c796f756b656e6b656e2c52...
  [Length: 28]

0000  ff ff ff ff ff ff 48 d2 24 44 a0 75 08 00 45 00  .....H.$D.u..E.
0010  00 38 33 0f 00 00 80 11 6f 4e c0 a8 0b 08 c0 a8  .83.....oN.....
0020  0b ff 1a 42 1a 42 00 24 6d 72 31 39 32 2e 31 36  ...B.B.$ mr192.16
0030  38 2e 31 31 2e 38 2c 79 6f 75 6b 65 6e 6b 65 6e  8.11.8,y oukenken
0040  2c 52 65 71 2c 31                               ,Req,1

```

図 4-7 : メンバ探索リクエストのパケット

② リクエスト応答

```

⊕ Frame 5952: 85 bytes on wire (680 bits), 85 bytes captured (680 bits)
⊖ Ethernet II, Src: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1), Dst: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
  ⊖ Destination: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
    Address: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
      .... 0 ..... = IG bit: Individual address (unicast)
      .... 0 ..... = LG bit: Globally unique address (factory default)
  ⊕ Source: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1)
    Type: IP (0x0800)
⊕ Internet Protocol, Src: 192.168.11.15 (192.168.11.15), Dst: 192.168.11.8 (192.168.11.8)
⊖ User Datagram Protocol, Src Port: 6718 (6718), Dst Port: 6718 (6718)
  Source port: 6718 (6718)
  Destination port: 6718 (6718)
  Length: 51
  ⊕ Checksum: 0x30bb [validation disabled]
⊖ Data (43 bytes)
  Data: 3139322e3136382e31312e31352c3139322e3136382e3131...
  [Length: 43]

0000  48 d2 24 44 a0 75 b4 82 fe 6d 03 d1 08 00 45 00  H.$D.u..m....E.
0010  00 47 04 0e 00 00 80 11 9f 30 c0 a8 0b 0f c0 a8  .G.....0.....
0020  0b 08 1a 3e 1a 3e 00 33 30 bb 31 39 32 2e 31 36  ...>.>.3 0.192.16
0030  38 2e 31 31 2e 31 35 2c 31 39 32 2e 31 36 38 2e  8.11.15, 192.168.
0040  31 31 2e 38 2c 46 4d 56 2d 42 49 42 4c 4f 2d 4e  11.8,FMV -BIBLO-N
0050  57 2c 41 6e 73                               W,Ans

```

図 4-8 : リクエスト応答のパケット

③ メンバ確立

+	Frame 5953: 82 bytes on wire (656 bits), 82 bytes captured (656 bits)
[-]	Ethernet II, Src: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75), Dst: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1)
[-]	Destination: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1)
	Address: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1)
	0.... = IG bit: Individual address (unicast)
	0.... = LG bit: Globally unique address (factory default)
+	Source: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
	Type: IP (0x0800)
+	Internet Protocol, Src: 192.168.11.8 (192.168.11.8), Dst: 192.168.11.15 (192.168.11.15)
[-]	User Datagram Protocol, Src Port: 6718 (6718), Dst Port: 6718 (6718)
	Source port: 6718 (6718)
	Destination port: 6718 (6718)
	Length: 48
+	Checksum: 0x3731 [validation disabled]
[-]	Data (40 bytes)
	Data: 3139322e3136382e31312e382c3139322e3136382e31312e...
	[Length: 40]
0000	b4 82 fe 6d 03 d1 48 d2 24 44 a0 75 08 00 45 00 ...m..H. \$D.u..E.
0010	00 44 39 4a 00 00 80 11 69 f7 c0 a8 0b 08 c0 a8 .D9J.... i.....
0020	0b 0f 1a 3e 1a 3e 00 30 37 31 31 39 32 2e 31 36 ...>.>.0 71192.16
0030	38 2e 31 31 2e 38 2c 31 39 32 2e 31 36 38 2e 31 8.11.8,1 92.168.1
0040	31 2e 31 35 2c 79 6f 75 6b 65 6e 6b 65 6e 2c 41 1.15,you kenken,A
0050	6e 73 ns

図 4-9：メンバ確立のパケット

C) IPtalk 入力の受信および表示

図 4-10 は、IPtalk の入力画面である。図 4-10 に示した「入力部」が字幕を入力するスペース、「表示部」が送り出した字幕が表示されるスペースである。



図 4-10：IPtalk の入力画面

同じ入力班の他のメンバの字幕は、6711/UDP ポートに送信されてくる。このポートに到達した字幕データが IPtalk の画面に表示される。「Wireshark」による解析のパケットの内容は以下の通りである。

+	Frame 10129: 50 bytes on wire (400 bits), 50 bytes captured (400 bits)
+	Ethernet II, Src: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1), Dst: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
+	Destination: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
+	Source: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1)
	Type: IP (0x0800)
+	Internet Protocol, Src: 192.168.11.15 (192.168.11.15), Dst: 192.168.11.8 (192.168.11.8)
+	User Datagram Protocol, Src Port: 6711 (6711), Dst Port: 6711 (6711)
	Source port: 6711 (6711)
	Destination port: 6711 (6711)
	Length: 16
+	Checksum: 0xf140 [validation disabled]
+	Data (8 bytes)
	Data: 955c8ea68a6d9446
	[Length: 8]
0000	48 d2 24 44 a0 75 b4 82 fe 6d 03 d1 08 00 45 00 H.\$D.u.. .m....E.
0010	00 24 08 f8 00 00 80 11 9a 69 c0 a8 0b 0f c0 a8 .\$...... .i.....
0020	0b 08 1a 37 1a 37 00 10 f1 40 95 5c 8e a6 8a 6d ...7.7.. .@.\...m
0030	94 46 .F

図 4-11 : IPtalk 字幕受信および表示のパケット

D) IPtalk からの取消

「F9 キー」で表示部に流した文を入力部などに戻せるために、IPtalk で「F9 キー」をクリックする時の仕組み確認して、「Wireshark」のパケット情報が図 4-12 である。

+	Frame 10277: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)
+	Ethernet II, Src: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1), Dst: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
+	Destination: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
	Address: 48:d2:24:44:a0:75 (48:d2:24:44:a0:75)
	0.... = IG bit: Individual address (unicast)
	0.... = LG bit: Globally unique address (factory default)
+	Source: AskeyCom_6d:03:d1 (b4:82:fe:6d:03:d1)
	Type: IP (0x0800)
+	Internet Protocol, Src: 192.168.11.15 (192.168.11.15), Dst: 192.168.11.8 (192.168.11.8)
+	User Datagram Protocol, Src Port: 6723 (6723), Dst Port: 6723 (6723)
	Source port: 6723 (6723)
	Destination port: 6723 (6723)
	Length: 12
+	Checksum: 0x5a0a [validation disabled]
+	Data (4 bytes)
	Data: 756e646f
	[Length: 4]
0000	48 d2 24 44 a0 75 b4 82 fe 6d 03 d1 08 00 45 00 H.\$D.u.. .m....E.
0010	00 20 08 fc 00 00 80 11 9a 69 c0 a8 0b 0f c0 a8i.....
0020	0b 08 1a 43 1a 43 00 0c 5a 0a 75 6e 64 6f ...C.C.. Z.undo

図 4-12 : IPtalk からの取消のパケット

4.5 IPtalk との連携機能の実装

ここまでで、IPtalk のプロトコルが判明したので設計方針に従い実装方法を述べる。

4.5.1 IPtalk 入力者との接続

前節で確認した内部動作に対して、IPtalk との連続に必要なのは以下の 3 つである。

- (1) 6722/UDP で、IPtalk のメンバ探索リクエスト(ブロードキャスト)を待ち受け
- (2) 6718/UDP で、リクエストを待ち受けた後、確認の返事を送る
- (3) 6718/UDP で、IPtalk の確認を受信する

これらに対応するため、「CaPPTioner」は .NET Framework のクラス・ライブラリ内の System.Net 名前空間及び System.Net.Sockets を利用する。

それぞれの命名空間内で使われるクラスと説明は表 4-4 に示す。

表 4-4 : IPtalk との連携機能に関するクラス

System.Net 名前空間	
UdpClient	ユーザーデータグラムプロトコルのネットワークサービスを提供
IPEndPoint	IP アドレスとポート番号でネットワークエンドポイントを表す
System.Net.Sockets 名前空間	
BeginReceive	接続されている Socket からの非同期のデータ受信を開始する
AsyncCallback	対応する非同期操作が完了したときに呼び出されるメソッドを参照
EndReceive	保留中の非同期読み込みを終了
NetworkStream	ネットワーク アクセスの基になるデータストリームを提供

UDP 通信 [11]の実装のフロー図と説明は図 4-13 のように示す。

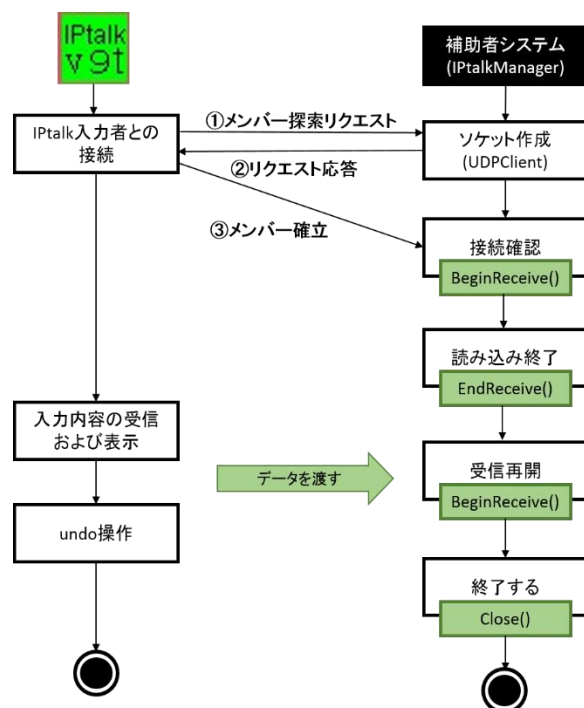


図 4-13 : UDP 実装のフロー図

4.5.2 IPtalk 入力の受信および表示

前節で確認した内部処理に対して、IPtalk からの入力を受信および表示するため、6711/UDP で待ち受ける。受け取った内容を、字幕管理するクラスである CaptionManager の Append メソッドに渡すことで字幕が表示され、スクリーンに表示する。

また、後述する undo 機能の為、図 4-14 のように、CaptionManager クラスから IPtalk 字幕の ID の返り値をとって undo 機能の為、別保存しておく。

```
id = CaPPTioner.captionManager.Append(captionOrder.text, captionOrder.kind);
iptalkManager.CaptionPush(id);
```

図 4-14 : IPtalk 字幕 ID の保存のソースコード

4.5.3 IPtalk からの取消

前節で確認した内部処理に対して、IPtalk からの undo 命令を受信するため、6723/UDP で待ち受ける。受け取った後、スタックを利用して保存された最新の IPtalk 字幕 ID を取り出し、字幕管理するクラスである CaptionManager の

captionManager.Remove メソッドを呼ぶ。そうすると、最後に送り出した IPtalk 字幕はスクリーンから削除できる。これらの操作が、5 回までできる。

図 4-15 は、スタックを利用して、IPtalk 字幕 ID を管理するソースコードである。

<pre>/// <summary> /// スタックにIDをPushする /// スタックのサイズが5より大きい場合、1番目を削除する /// </summary> /// <param name="id">PushするID</param> public void CaptionPush(int id) { if (count < 5) { items.Add(count++, id); } else { for (var i = 1; i < 5; i++) { items[i - 1] = items[i]; } items[4] = id; } }</pre>	<pre>/// <summary> /// スタックからIDをPOPする /// </summary> /// <returns></returns> public int CaptionPop() { int id; if (count > 0) { if (items.ContainsKey(--count) && items.Count > 0) { id = items[count]; items.Remove(count); return id; } } return -1; }</pre>
字幕追加する時IDを保存	undoする時IDを取り出す

図 4-15 : IPtalk 字幕 ID 管理のソースコード

4.6 まとめ

機能テスト (第 6 章) で、筆者担当した IPtalk との連携機能に関するテスト項目が 16 項目である。そのうち、1 件のバグを発生した。その原因は、各機能間の影響範囲を十分に考慮せず、不十分なプログラム設計を行ったためである。それらのバグに対して、再設計を行って実装をやり直した。

成果物として、「CaPPTioner」と IPtalk の連携を実現した。実現するために、「Wireshark」を利用して IPtalk の通信プロトコルを確認した。実装した機能により、IPtalk からの字幕入力内容を表示することを可能とした。また、最後送り出した文を削除できる undo 機能も実現した。これらの機能によって、「CaPPTioner」と IPtalk を連携でき、質疑応答への対応を実現した。

5 音声合成との連携機能

聴覚障害者は、その聴力の程度や失聴の時期によって、また、音声言語の獲得環境によって、発話の困難を抱えている場合がある。また、重複障害等で発声することが困難な講演者もいる。

そのため、研究発表会で発話が困難な講演者のサポートを可能とするために、音声合成機能が必要と考えた。音声合成は、画面に入力したテキストから現実の人間の音声に似ている言語化表現を出力するものである。

3.3 節において定義された要件のうち、合成音声との連携機能は機能要件として挙げられていないが、より多くの講演者を対象としてできるように、追加実装した。設計・実装にあたっては非機能要件「h2：インストールが不要である」を重視して行った。

5.1 要件の詳細化

h2 要件について、音声合成との連携機能を実現するために以下の要素が必要である。

- (1) 表示する文を音声合成で発声できる
- (2) 発声終了が講演者に知覚できる

各要素に関して、以下に解説する。

- (1) 表示する文を音声合成で発声できる

音声エンジンと連携して、表示する文字を音声合成で発声できるようにする。音声合成エンジンは Windows 内蔵のものを利用可能とし、他にもインストールされていれば、それらも使えることが望ましい。

- (2) 発声終了が講演者に知覚できる

聴覚障害者の中には、完全に聞こえない人がいる。聴覚障害者が音声合成機能を利用して講演する時、次の文に進めるためには発声のタイミングが分かり、合成されている音声の終わりを知覚できることが必要である。そのため、表示を工夫し、発声終了が講演者に知覚できるようにする。

5.2 音声合成との連携機能の設計

前節で述べた要求に対する設計方針に関して述べる。

「CaPPTioner」音声合成機能は、セルフモードで文単位の字幕表示をしつつ講演する時だけ利用可能とする。理由としては、研究発表会で補助者がいる場合、発話が困難な聴覚障害者が講演者として手話や字幕の形式で講演して、補助者が通訳として発話する事もできるためである。

音声合成機能は補助者がいない場合を想定しており、発話が困難な聴覚障害者が合成音声機能を利用して発話するためである。また、合成音声機能を利用して講演するので、段落やページ単位では聴講者が聞き取りにくいと考え、文単位での表示を行う場合のみの使用とする。

5.2.1 表示する文を音声合成で発声できる

「CaPPTioner」の音声合成機能では、Windows に搭載された音声合成機能を利用する。

Windows アプリケーションで音声認識や音声合成を使うためにマイクロソフトが開発した API である、Speech Application Programming Interface (Speech API、SAPI) [12] [13]を使用する。SAPI は自由に再配布可能なコンポーネントであり、SAPI を使用するアプリケーションに同梱することが可能である。SAPI は、.NET Framework3.0 がインストールされている Microsoft Windows XP 以降の Windows OS で利用可能である。

SAPI は言わば、アプリケーションと音声合成エンジンの間のインターフェースあるいはミドルウェアである。アプリケーションは直接エンジンとやり取りできるが、エンジンのメソッドを直接呼び出す代わりに単純化された高レベルのオブジェクトを使うこともできる。音声合成機能に関するクラスが表 5-1 である。

表 5-1 : 音声合成機能に関するクラス

System.Speech の System.Speech.Synthesis 名前空間	
SpeechSynthesizer	インストールされている音声複合エンジンの機能へのアクセスを提供するクラス
VoiceInfo	インストールされた音声合成エンジンを表すクラス
SpeakAsync	オブジェクトから非同期で出力される音声を生成するメソッド

5.2.2 発声終了が講演者に知覚できる

発声終了が講演者に知覚できるようにするには、様々な方法が考えられる。

(1) 字幕スクリーン上に表示させる

話している音声が終わる時点で、字幕スクリーン上である画面部品が点滅する。この方法は、専用の画面部品を用意する必要がある。

(2) 発声終了時点で字幕に特殊マークを表示する

文の表示と同時に発声し、発声終了時に、特殊な記号を表示する方法を考えた。しかし、文単位で字幕を表示する時、単位毎にマークを付けることが、字幕を見にくくなる可能性が高い。

(3) 字幕の表示方法を変える

表示の最小単位は文であるが、字幕の表示方法を変更して、音声の終わりを表示できるようにする。例えば、「明日は、いい天気です。」と発声する時、まず「明日は、いい天気です」のみを表示して、発声終了時に句点「。」を表示する方法が考えられる。

しかし、ノートには、句点がついていない文があると考え、句点の表示により音声の終わりを判断させる方法は確実ではない。従って、字幕の表示方法を変えることで発声終了を知覚するなら、より明確な表示方法を工夫しなければならない。

以上の考えを踏まえて調査したところ、SAPIには `SpeechSynthesizer.SpeakProgress` イベントがあり、音声合成メソッド `Speak()` あるいは `SpeakAsync()` で渡された文字列が、今どの部分の文字列を発声しているのかを通知することができる。

イベントで渡される `SpeakProgress.EventArgs.Text` を受信し、4.3 節で説明した `captionManager.Append()` メソッドを使用すれば、部分文字列単位で表示することで、発話と同期して文字が表示できる。そうすると、発話終了が文章の表示終了によって知覚できる。図 5-1 では、時間遷移と共に、字幕を送り出すイメージである。

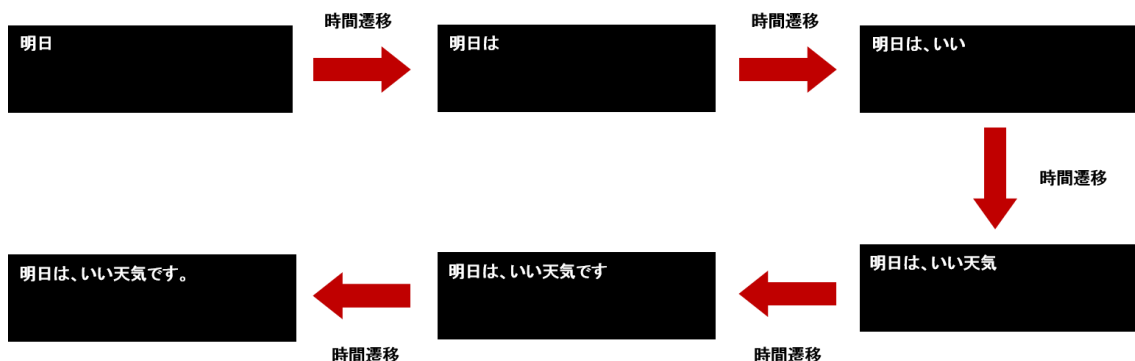


図 5-1：部分文字列で字幕を出すイメージ図

テストプログラムで試したところ、発声しない部分文字列は `SpeakProgress()` では通知されないことが分かった。そのため、「明日は、いい天気です。」を発声させ、`SpeakAsync()` で渡される部分文字列をつなぎ合わせても、「明日はいい天気です」にしかない。これについては、実装上の対策が別途必要である。

5.3 音声合成との連携機能の実装

本節では、設計方針に対する実装を述べる。

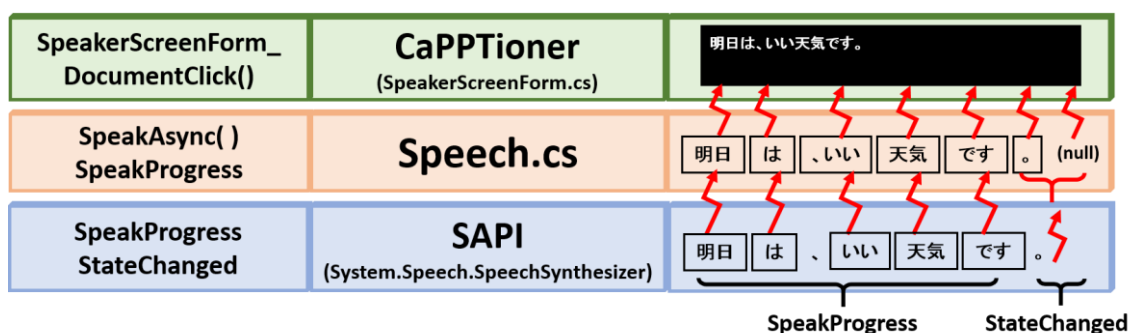


図 5-2：実装説明図

図 5-2 の図を用いて実装方法を説明する。

Speech.cs クラスで、音声合成エンジンの連携と文字列処理を行う。句点が消えると次の文と繋がってしまうので、これを補うため、System.Speech.SpeechSynthesizer のラッパークラス Speech.cs を実装した。Speech.cs の役割が表 5-2 で示す。

表 5-2：Speech.cs の役割

クラス名	役割
Speech.cs	(1) 表示する文を音声合成で発声できる。また、音声エンジンのインストールされていない場合の発声動作を抑制する
	(2) 発声終了が講演者に知覚できる - 発生が起こらない部分文字列を復元する - イベントを SpeakProgress に一元化する。発声終了 StateChanged を受けて、SpeakProgressEventArgs.Text が null である SpeakProgressEvent を発生させる

ここから、Speech.cs の役割に対して、具体的に述べる。

(1) 表示する文を音声合成で発声できる

Speech クラスに SpeakAsync メソッドを実装する。発声対象の文字列を内部に保存し、System.Speech.SpeechSynthesizer. SpeakAsync を呼ぶ。

(2) 発声終了が講演者に知覚できる

```
void tts_SpeakProgress(object sender, SpeakProgressEventArgs e)
{
    int i = text.IndexOf(e.Text) + e.Text.Length;
    string s = text.Substring(0, i);
    text = text.Substring(i);
    OnProgress(new ProgressEventArgs(s));
}
```

図 5-3 : SpeakProgress のソースコード

Speech クラスは SpeakAsync メソッドをあり、System.Speech.SpeechSynthesizer. SpeakProgress イベント (図 5-3) および System.Speech.SpeechSynthesizer. StateChanged イベントのリスナになる。System.Speech.SpeechSynthesizer. SpeakProgress イベントを受けて、Speech. SpeakProgress イベントを発生させる。この際に、StateChanged イベント(図 5-4)を受けて、Speech. SpeakProgress イベントを発生させ、テキストの残りを上位レイヤに渡す。さらに、発声終了を通知するため、Speech. SpeakProgress イベントで、null を渡す。

字幕の表示を行う SpeakerScreenForm は Speech. SpeakProgress イベントのリスナになる。イベントで渡される SpeakProgressEventArgs.Text を受信し、null でなければ、captionManager.Append メソッドに渡すことで逐次表示が実現できる。

```
void tts_StateChanged(object sender, StateChangedEventArgs e)
{
    if (e.State == SynthesizerState.Ready)
    {
        OnProgress(new ProgressEventArgs(text));
        text = null;
        OnProgress(new ProgressEventArgs(text));
    }
}
```

図 5-4 : StateChanged のソースコード

図 5-2 の右側に示した通り、句点がない文字列「明日はいい天気です」を Speech.cs で処理した後、「明日は、いい天気です。」という文字列を「CaPPTioner」で表示できる。

5.4 まとめ

音声合成との連携機能に関する機能テスト (第 6 章) は、2 回目の機能テストにおいて行った。この機能に関するテスト項目を実施することで、顧客と合意した機能要件に沿っ

て機能を実現できていることを確認した。全てのテストケースを通過したことから、機能要件を満たしていると判断し、音声合成との連携機能の開発を完了した。

本機能の実現によって、「CaPPTioner」と音声合成エンジンを連携することで、合成音声で講演を行うことが可能になった。音声合成との連携機能によって、発話が困難な障害者も講演を行いやすくなる。このことにより、「CaPPTioner」の目的である、より広く情報保障を実施することの達成に貢献することができた。

6 品質管理

本プロジェクトにおいて筆者は品質管理を担当した。本章では筆者が実施した品質管理の計画、実施内容、及び結果と、そこから得られた知見やプロジェクトへの貢献について述べる。

6.1 品質とは

ISO/IEC 25000 [14]によると品質とは「指定された特定の条件で利用する場合の、明示的または暗示的なニーズを満たすソフトウェア製品の能力」と定義されている。

品質は 4 つの種類に分類される。図 6-1 (JIS X 0129-1 から引用) を用いて説明する。

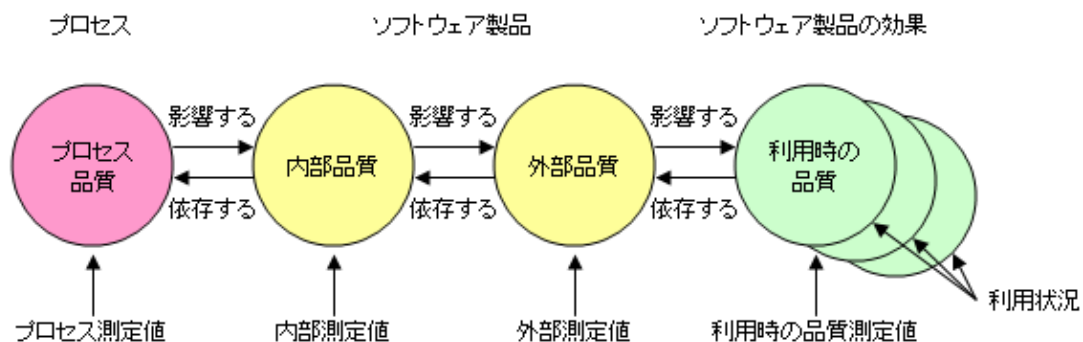


図 6-1：品質の 4 つ種類

図 6-1 は示す、右側のソフトウェア製品の効果から説明する。

ソフトウェアを実際にユーザが利用する段階の品質のことを「利用時の品質」と呼び、利用者の必要性に合致するかどうかを表す。「利用時の品質」に直接影響するものが「外部品質」である。ソフトウェアが実行されときの品質のことであり、テストデータを用いたテストの結果（実行時のコードの振る舞い）などが該当する。「外部品質」に直接影響するものが「内部品質」 [15]である。ソフトウェアの内部的な特徴で、ソースコードのみならず、仕様書なども測定対象になる。静的な測定によって得られるものがほとんどで、例えばモジュール性（ソフトウェアが適切に構造化され、変更、修正などが局所的なもので済むようになっている性質）や追跡可能性（要求から実現されたものへの関連、および実現されたものから要求への関連を追いかけることができる性質）などが該当する。「内部品質」に直接影響するものが「プロセス品質」である。プロセスとは設計／開発のやり方や手順のことである。この「やり方」の品質が結果的にすべての品質に影響を及ぼす、ということになる。

ソフトウェア品質を測定するための世界標準として定められた ISO/IEC25000 に基づき、本プロジェクトの品質マネジメントで重視する品質特性を決定した。

ISO/IEC9126 [16]では、ソフトウェア品質を示す特性に「品質特性」とそれをさらにブレイクダウンした「品質副特性」を定めて、表 6-1 に示す。

表 6-1：ソフトウェア品質特性・副特性(ISO/IEC25000)

品質特性	品質副特性	主な内容
機能性	合目的性	目的から求められる必要な機能の実装の度合い
	正確性	
	相互運用性	
	標準適合性	
	セキュリティ	
信頼性	成熟性	機能が正常動作し続ける度合い
	障害許容性	
	回復性	
使用性	理解性	分かりやすさ、使いやすさの度合い
	習得性	
	運用性	
効率性	時間効率性	目的達成のために使用する資源の度合い
	資源効率性	
保守性	解析性	保守（改訂）作業に必要な努力の度合い
	変更性	
	安定性	
	試験性	
移植性	環境適用性	別環境へ移した際そのまま動作する度合い
	設置性	
	規格適合性	
	置換性	

6.2 本プロジェクトの品質管理

本プロジェクトにおける管理の方針，及び考慮すべき点を以下に述べる。

6.2.1 基本方針

本プロジェクトの基本方針として表 6-1 で示した品質の 6 特性のうち、本プロジェクトでは以下の特性を重視することにした。

●機能性

機能性とは、「ソフトウェアがある目的を持って求められる、必要な機能を実装している度合い」と言える。

本プロジェクトは顧客から受託開発によって作成されたソフトウェアである。従って、顧客に必要な機能が実装されているかを担保する機能性を重要視する。

●使用性

使用性（以後、ユーザビリティと呼ぶ）とは、ソフトウェアシステムの「使いやすさ」「(使うことにかかる) 労力」「(使うことによって) 得られる結果」の良し悪しを評価するものである。

本システムを利用するのは一般人である。したがって、「ユーザにとって理解しやすく、使いやすいかどうか」を担保するためにユーザビリティを重視する。

6.2.2 実施内容

これらの品質特性を重視するために、本プロジェクトでは以下の活動を実施した。

■ 顧客との積極的な関わり

本プロジェクトでは、顧客との認識の齟齬を軽減するために、週 2 回ミーティングを行い、顧客と関わる機会を設けた。顧客と話し合いを重ねる事によって、顧客から潜在的な要求を抽出し、機能性の副特性である合目的性の向上を目指した。

■ 機能テスト

機能テストとは、システム開発の際に行われるテストのうち、顧客側から要求された機能をシステムが満足しているかどうかを検証するテストのことである。機能テストは、要求仕様を満たしているかどうかを確認することが目的のため、機能性の副特性である合目的性と正確性の向上を目指した。

■ ユーザビリティ評価実験

評価者が実際にシステムを利用することで、評価で行い問題点を発見する。これらのユーザビリティ評価を確認することで、使用性の副特性である理解性、習得性、運用性の向上を目指した。

6.3 機能テスト

6.3.1 機能テスト方針

本プロジェクトの機能テストは、要求事項仕様書に基づいて作成したテスト項目に沿って実施し、機能間およびシステム全体が仕様・要求の通りであるかどうかを確認するためのテストである。機能テスト工程における「機能テスト計画」、「機能テスト設計」、「機能テスト実施」の全てを専任する担当者は用意していない。機能テスト設計・実施の作業は、開発メンバ全員で分担して実施した。

機能テストは全て機能の実装を完了後に実施した。終了条件として、チームレビューしたテスト項目が全て合格となるか、または、不具合に対する対処方法が決定していることである。計画として、1回実施する予定である。機能テストについての手順が図 6-2 のように設定した。

以下は、テスト項目の作成、テストの実施とテストの結果について述べる。

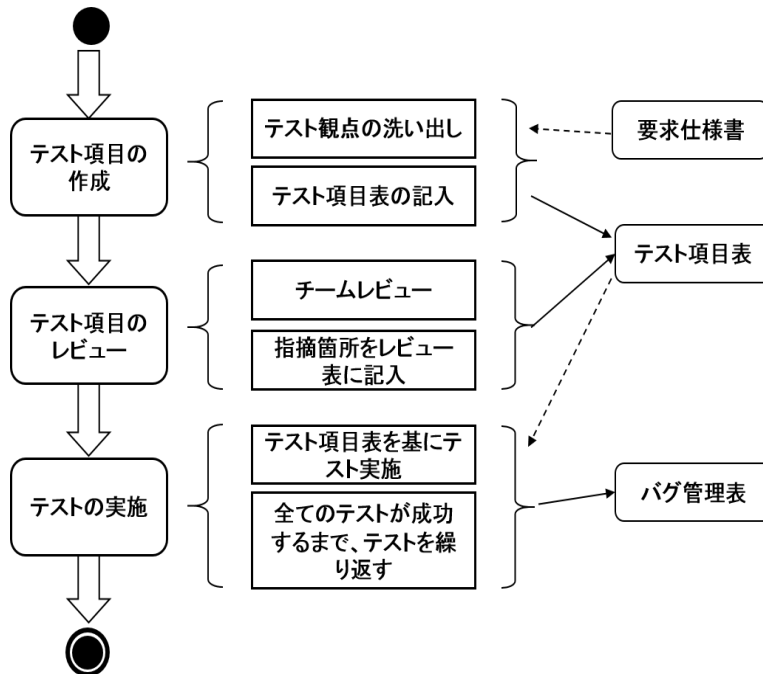


図 6-2 : 機能テストについての手順

6.3.2 機能テスト項目の作成

本節では、テスト項目の洗い出し方法と管理する方法について述べる。

■ CFD 技法 [17]でテスト項目の網羅性を確保

テスト項目の網羅性を確保することが重要と考えられる。しかし、テスト項目に網羅性を確保する事は難しい。その理由の 1 つとして、テスト条件の組み合わせの数が非常に多くなることが挙げられる。

文字列を分析するソフトウェアという簡単な場合を考えてみる。8 文字の英数字ファイル名のチェックを考えた場合、このファイル名の組み合わせは英数字の種類 36 の 8 乗＝約 3 兆になり 1 秒に 10,000 の組み合わせをテストしても、全ての組み合わせをテストするには 9.5 年かかる。このような簡単なソフトウェアでさえも、考えられるすべての組み合わせをテストすることはできない。従って、テスト項目を削減する方法について考える必要があり、同時に削減法は合理的なものである必要がある。

本プロジェクトでは、合理的なテスト項目を作成するため、CFD(Case Flow Diagram)技法を用いた。

・ CFD 技法

CFD 技法は、原因結果グラフの発展型とも言うべき技法で、同値分割、デジションテーブルの手法を組み合わせたものである。

・ CFD 技法でのテスト項目削減原理

CFD でのテスト項目削減は、「同値分割」「有効系の優先」「原因間の相互依存の除」「階層化」によって実現される。

「同値分割」により、原因結果グラフと同様に類似のテスト項目を排除する。「有効系の優先」とは、より重要である有効系に対しては優先的にテスト項目を抽出し、無効系に対しては優先度を落とすことで、テスト項目を削減することである。「原因間の相互依存の排除」では、依存関係の無い原因間のテスト項目を排除することにより削減する。そして「階層化」では、機能を階層で分割し、上位階層で結合する際に有効系の組み合わせを優先し、無効系同士の組み合わせを排除することにより、テスト項目を削減する。

・ CFD 技法でのテスト項目抽出方法

まず、CFD 技法を用いてテスト項目を抽出するまでの基本的な流れを説明する。

- ① 仕様から原因を抽出し、同値分割する。
- ② 仕様から結果を抽出し、有効系と無効系に分類する。ここで、有効系とは仕様に定められた処理を行う結果を指し、無効系とはそれ以外のエラー処理等で定められた処理が行われない結果を指す。
- ③ 各原因と結果の関係を有効系と無効系とを明確に分け流れ図で表現する。これが CFD である。

- ④ CFD をデジジョンテーブルに展開し、テスト項目を抽出する。

CFD 技法でテスト項目表を抽出する例は図 6-3 に示す。

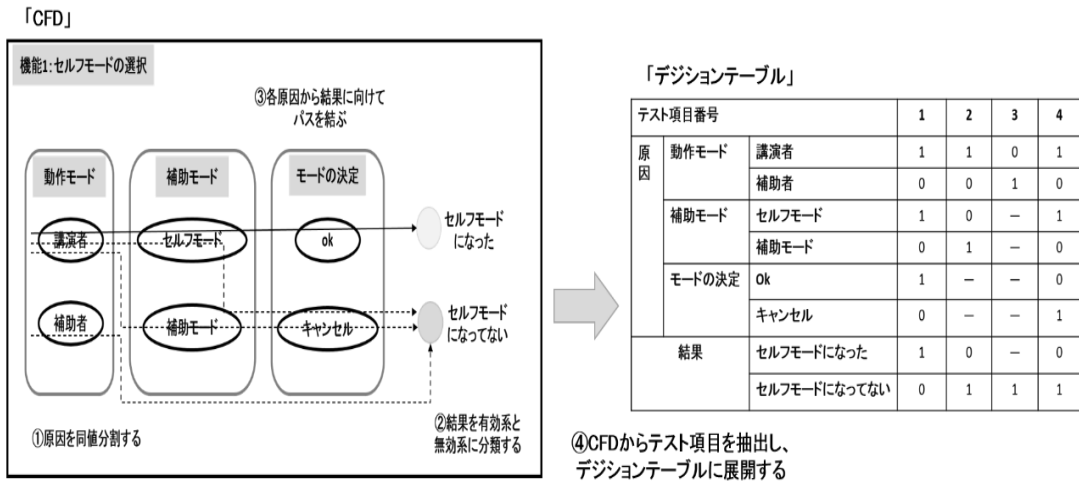


図 6-3 : CFD 技法で抽出したテスト項目表例

■ Redmine を利用

作成したテスト項目表の一部抜粋を図 6-4 に示すように、Redmine のプラグイン Impasse を用いて仕様の作成、テストの割り当て、実行、バグチケットの発行を行った。

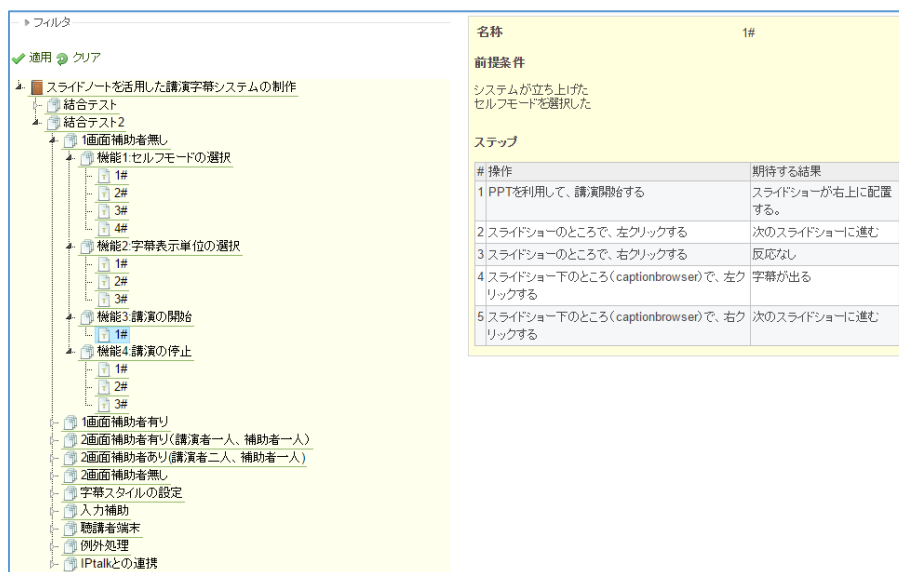


図 6-4 : Redmine のテスト項目図例

6.3.3 機能テストの実施

機能テストは計 2 回行った。1 回目で発覚したバグを修正し、漏れていた項目を追加して、再度テストを行った。1 回目の結果を図 6-5 に示す。

Test suite	total cases	executed cases	bugs	(open)	progress rate	bugs rate
結合テスト 1	177	175	11	0	98.87%	6.21%
結合テスト	177	175	11	0	98.87%	6.21%
1画面補助者無し	4	4	0	0	100.00%	0.00%
1画面補助者有り	19	19	3	0	100.00%	15.79%
2画面補助者無し	7	7	0	0	100.00%	0.00%
2画面補助者有り	19	19	1	0	100.00%	5.26%
入力補助	6	6	6	0	100.00%	100.00%
聴講者端末	69	69	0	0	100.00%	0.00%
字幕スタイルの設定	45	43	1	0	95.56%	2.22%
IPtalk	7	7	0	0	100.00%	0.00%
(Test case)	1	1	0	0	100.00%	0.00%

図 6-5 : 1 回目のテスト記録

修正した後、再度テストを行った。2 回目に行ったテストの結果を図 6-6 に示す。

Test suite	total cases	executed cases	bugs	(open)	progress rate	bugs rate
結合テスト 2	233	179	16	0	76.82%	6.87%
結合テスト 2	233	179	16	0	76.82%	6.87%
1画面補助者無し	16	16	0	0	100.00%	0.00%
1画面補助者有り	21	21	6	0	100.00%	28.57%
2画面補助者無し	8	6	0	0	75.00%	0.00%
2画面補助者有り(講演者一人、補助者一人)	20	20	6	0	100.00%	30.00%
入力補助	6	6	0	0	100.00%	0.00%
聴講者端末	61	22	4	0	36.07%	6.56%
字幕スタイルの設定	64	64	0	0	100.00%	0.00%
IPtalkとの連携	16	14	0	0	87.50%	0.00%
2画面補助者あり講演者二人、補助者一人)	11	1	0	0	9.09%	0.00%
例外処理	9	8	0	0	88.89%	0.00%
(Test case)	1	1	0	0	100.00%	0.00%

図 6-6 : 2 回目のテスト記録

機能テストの結果、計画したテスト項目を実施することで、システム全体が仕様・要求の通りであることを確認した。

6.3.4 機能テストの振り返り

反省点としては、1 度目のテストでテスト項目の網羅性を確保できなかったことが挙げられる。最初のテスト項目は、各メンバが自分の担当部分のテスト項目を要求事項仕様書に沿って作成した。

初回のテストで利用した項目は、各メンバが作成してチームレビュー後に実施した。しかし、そのテスト項目では十分に網羅していなかったため、テスト実施時に、テスト項目に漏れや重複が発生してしまった。十分に網羅していなかった理由として、テスト条件の組み合わせの数が多くて、開発者の経験や勘による属人的方法でテスト項目を作成したため、十分に網羅することができなかったのではないかと考えた。そのため、筆者が CFD 技法を導入し 2 回目のテスト項目を作成した。

6.4 ユーザビリティ評価実験

本システムのユーザビリティを向上するため、ユーザビリティ評価実験を行う。

ユーザビリティ評価実験を行い、ユーザビリティの観点からシステムの品質を判断し、その問題点を発見する。発見した問題点を解決することによって、ユーザビリティの向上を目指す。

6.4.1 代表的な評価方法

代表的な評価法として、ユーザビリティテスト・ヒューリスティック評価法・チェックリスト評価法・認知的ウォークスルーの4つの方法を取り上げ、概要を説明する。

ユーザビリティ・テストは、システムやプロトタイプを実際のユーザに使用してもらい、その観察記録から問題点を指摘する評価手法。ユーザがシステム利用時に行う主要なタスクをいくつか設定し、それらの実行過程におけるユーザの行動や発言を記録し、開発者が見落としがちなテスト対象の問題点を効率的に洗い出すことができる。

ヒューリスティック評価法は、評価のためのガイドラインを事前に用意しユーザに関しての知識を習得した専門家が自らの知識と経験からガイドラインに従いテスト対象の評価を行う方法であり、専門的な知見に照らし合わせた問題点を洗い出すことができる。

チェックリスト評価法は、事前に設定したチェックリストに基づきテスト対象がその確認内容を満たしているかどうかを評価する方法であり、チェックリストの質に依存するもののテスト対象の問題点を広く洗い出すことができる。

認知的ウォークスルーは、システムを利用するユーザにとってどのような目標があり、それを実現するためにどのような手続きが必要で、そのために何が手がかりとなるのか、そしてその結果はどのようにフィードバックされるのか、という流れで評価する方法である。具体的には、評価者が想定されるタスクに沿って実際にテスト対象を利用しながらユーザの思考過程や操作を推測して問題点を洗い出す方法である。

本プロジェクトでは、想定されるユーザが全て一般人であり、一般人がシステムを利用して評価をいただくことが一番よいと考える。今回の開発メンバーが、開発経験が少ない学生である。ユーザの行動や発言を記録し、開発者が見落としがちなテスト対象の問題点を効率的に洗い出すことができる。また、ユーザが利用する時の様子を見ることで、開発者が客観的に判断できる。以上の理由で、ユーザビリティ・テスト評価方法を取りあげ実験する。

6.4.2 評価実験の計画

ユーザ評価は、講演者、補助者、聴講者の3つの視点から行う。

聴講者向けの実験は、筑波技術大学に在学する 12 名の聴覚障害がある学生に協力していただいた。聴講者として字幕の見やすさや端末システム操作の分かりやすさについて評価を行った。

1 回目の講演者・補助者向けの実験は、筑波大学高度 IT コース修士 2 年の 13 名の学生が協力していただいた。講演者・補助者向けの実験では講演者・補助者システムの操作の分かりやすさについて評価を行った。

2 回目の講演者・補助者向けの実験は、筑波大学高度 IT コース修士 1 年の 10 名学生に協力していただいた。筑波大学高度 IT コース修士 1 年の学生が、毎週定例的な発表会がある。発表会が 5 分間のプレゼンテーションと 10 分間の質疑応答である。本システムが、講演本番に近い環境で対応できるかどうかを確認することが目的である。

■ 期間

12 月 8 日～12 月 15 日

■ 評価方法

- GQM [18]という手法で、評価に必要な指標値を定義
- アンケートおよび思考発話法から、定義した指標に対するデータを抽出して評価

■ GQM 手法

概要

プロセス改善活動などの実施効果を分析する観点を系統的に抽出するための思考方法である。図 6-7 に示すように、組織の当該活動の実施目的（Goal）が抽象度の高い形で先ず定義され、次に Goal の達成要件が質問文 Q の集合へと分解され、最後に各質問に対応するデータ収集用の定量化尺度すなわちメトリクス M が定義される。

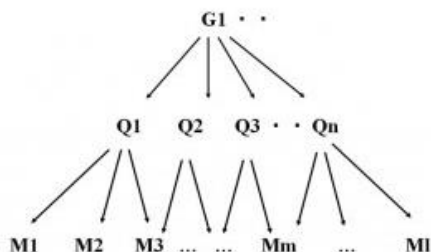


図 6-7 : GQM 手法

GQM 手法によるアンケート質問の抽出

図 6-8 に、GQM 手法によって抽出した質問を示す。3 つのゴールとして「字幕の見やすさを評価する」、「操作の分かりやすさを評価する」、「表示や構成の分かりやすさを評価する」を定義する。

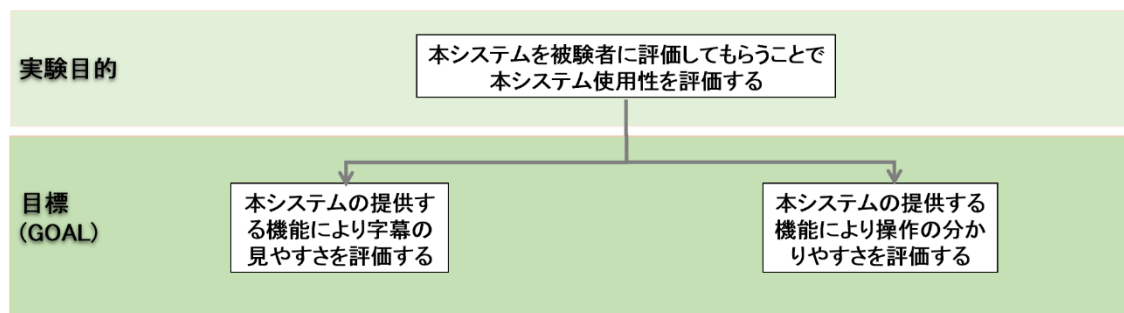


図 6-8 : GQM 手法によって抽出したアンケート質問

端末システムが聴覚障害者以外の人でも利用すると考えて、同時に評価を行う。図 6-9 は、目標「本システムの提供する機能により字幕の見やすさを評価する」に対する質問とメトリクスである。

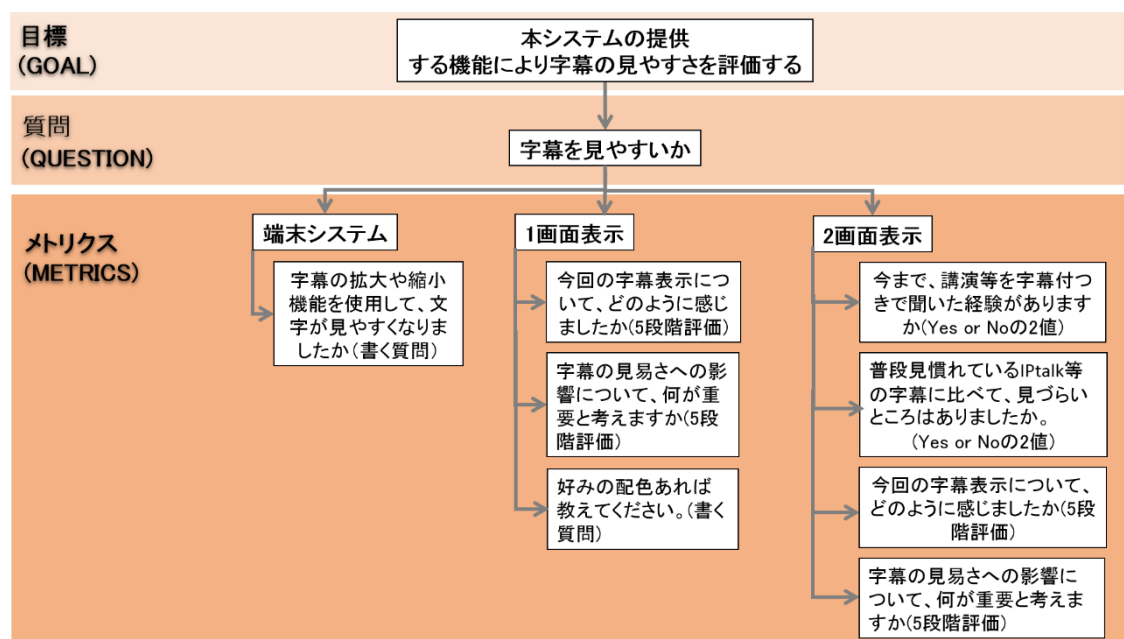


図 6-9 : GQM によって抽出した質問とメトリクスの例

■ 思考発話法

思考発話法とは、被験者にタスク（課題）を提示し、その実行過程において考えていることを話しながら操作してもらう手法である。被験者の行動と発話から、インターフェース上のどの部分に問題があるのか、なぜその問題が起きたのかを詳細に把握できる。

本実験は、被験者が実行過程における失敗や混乱、さらに発話内容から開発者が見落としがちなテスト対象の問題点を効率的に洗い出すことによって、発話を記録・観察し、分析の材料とする。

6.4.3 評価実験の実施

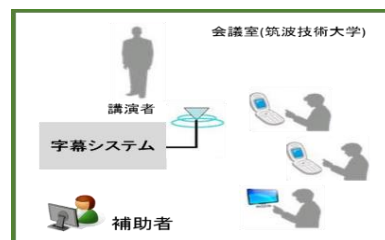
■ 聴講者向け

- ・ 実験対象の選定

端末システム、1画面での表示、2画面での表示

- ・ 実験イメージ図

本実験では、チームメンバが講演者と補助者を担当して、本システムについて説明を行う。被験者の手元にある、携帯電話などの端末を利用して、字幕がスクリーンだけではなく手元の端末で見られるようにする。



- ・ 実験の構成と内容

本実験では、被験者が表 6-2 に示す 3 つのタスクを課した。また、実験の様子をビデオカメラに収録し、被験者の発話データを記録した。次に、各タスクの終了後に、表 6-2 に示す各タスク内の細目について評価アンケートを実施した。

表 6-2：聴講者向けの実験のタスク

番号	タスク名	タスク内容
①	2画面での閲覧	開発メンバが2画面で講演する。被験者が画面表示を見る。講演終わった後、インタビューとアンケートの形で評価する。
②	1画面での閲覧	開発メンバが1画面で講演する。被験者が画面表示を見る。講演終わった後、インタビューとアンケートの形で評価する。
③	端末での閲覧	被験者が自分用の端末で字幕システムにアクセスする。端末で字幕を見て、インタビューとアンケートの形で評価する。

・ アンケートの質問と結果

① 2画面での閲覧

質問内容		選択肢			回答(人数)	
質問1：今まで、講演等を字幕付つきで聞いた経験がありますか。		経験あり			12	
		経験なし			0	
質問2：普段見慣れている IPtalk 等の字幕に比べて、見づらいところはありましたか。		IPtalk と同じ			11	
		IPtalk と違い			1	
質問内容	評価項目	評価 1 (人数)	評価 2 (人数)	評価 3 (人数)	評価 4 (人数)	評価 5 (人数)
質問3：今回の字幕表示について、どのように感じましたか。	字幕の大きさ	0	1	10	1	0
		小さすぎる<----->大きすぎる				
	行間サイズ	0	0	12	0	0
		狭すぎる<----->広すぎる				
	スクロールの速度	0	4	6	2	0
		遅すぎる<----->速すぎる				
	配色	0	3	6	2	1
		見にくい<----->見やすい				
質問4：字幕の見易さのためには、何がどの程度重要だと思いますか。重要と考えるものから、大きい値を選んでください。(同じ値を繰り返し選んでも構いません)	字幕の大きさ	0	0	1	6	5
		小さすぎる<----->大きすぎる				
	行間サイズ	0	0	5	5	2
		狭すぎる<----->広すぎる				
	スクロールの速度	0	0	3	5	4
		遅すぎる<----->速すぎる				
	配色	0	0	0	3	9
		見にくい<----->見やすい				
質問5：補助者がいない時にも、講演者だけで字幕つき発表を行うモードについて教えてください。	1.即興発言や質疑応答について字幕が出ないので、講演本編だけでは意味がない。	0	2	1	3	6
	2.講演本編だけでも字幕がつけば、福祉以外	0	1	5	4	2

(5段階評価:1 そう思わない、3 どちらでもない、5 そう思う)	の分野の発表も聞けてよいと思う。					
	3.補助者なしで字幕つき発表とする場合、レーザーポインタで字幕を追ってもらうことは重要だと思う。	0	3	2	6	1

② 1画面での閲覧

質問内容		選択肢			回答(人数)	
質問1: 1画面と2画面の、それぞれよい悪いと感じた点を教えてください。総合的にはどちらが好みですか。		1画面がすき			4	
		2画面がすき			4	
質問2: 2画面が準備できない会場で、1画面で字幕を表示する機能が有用だと思いますか?理由も教えてください。		有効			10	
		有効ではない			2	
質問内容	評価項目	評価 1 (人数)	評価 2 (人数)	評価 3 (人数)	評価 4 (人数)	評価 5 (人数)
質問3: 今回の字幕表示について、どのように感じましたか。	字幕の大きさ	1	7	3	1	0
		小さすぎる<----->大きすぎる				
	行間サイズ	0	2	10	0	0
		狭すぎる<----->広すぎる				
	スクロールの速度	0	0	7	5	0
		遅すぎる<----->速すぎる				
	配色	0	0	6	4	2
		見にくい<----->見やすい				
質問4: 字幕の見易さへの影響について、何が重要と考えますか。	字幕の大きさ	0	1	2	5	4
		小さすぎる<----->大きすぎる				
	行間サイズ	0	2	6	3	2
		狭すぎる<----->広すぎる				
	スクロールの速度	1	1	2	4	4
		遅すぎる<----->速すぎる				
	配色	0	1	4	3	4

		見にくい<----->見やすい
--	--	-----------------

③ 端末での閲覧

質問内容	選択肢	回答(人数)
質問 1: アクセスできるまでの手順が分かりやすいですか? 分からないところがあったら、教えてください。	分かりにくかった	11
	分かりやすかった	1
質問 2: 字幕の拡大や縮小機能を使用して、文字が見やすくなりましたか? もし、見にくいところがあれば、教えてください。	見やすくなった	11
	見にくかった	1
質問 3: 自動更新の機能では、リアルタイムに字幕が表示されましたか?	表示された	12
	表示されなかった	0
質問 4: 画面のデザインが分かりやすいですか? もし、分かりにくいところあったら教えてください。	分かりやすかった	10
	分かりにくかった	2
質問 5: 端末の字幕配信機能があれば、スクリーンに字幕を出さなくても大丈夫だと思いますか? 理由を教えてください。	スクリーンだけ必要	0
	端末だけ必要	0
	両方必要	12
質問 6: 今後、字幕の配信があるなら、端末で閲覧したいですか? 理由を教えてください。	閲覧したい	11
	閲覧したくない	1

■ 講演者・補助者向け 1

・ 実験対象の選定

端末システム、セルフモード、補助者ありモード

・ 実験の構成と内容

本実験が、表 6-3 に示す 3 つのタスクを被験者に課した。同時に、被験者の発話データを記録した。次に、各タスクの終了後に、表 6-3 に示す各タスク内の細目についての 5 段階評価(1 そう思わない、3 どちらでもない、5 そう思う)によるアンケートを実施した。(端末での閲覧について、1 人が評価してないので、合計 12 人)

表 6-3 : 講演者・補助者向けの実験のタスク

番号	タスク名	タスク内容
①	端末での閲覧	開発メンバがシステムの概要について講演する。被験者が自分用の端末で字幕システムにアクセスする。端

		末で字幕を見て、インタビューとアンケートの形で評価する。
②	セルフモードでの講演	被験者 1 人ずつセルフモードを行う。実験終わった後、インタビューとアンケートの形で評価する。
③	補助者ありでの閲覧	被験者 2 人ペアで講演者と補助者ありのモードで講演する。講演終わった後、インタビューとアンケートの形で評価する。

・ アンケートの質問と結果

① 端末での閲覧

質問内容	評価 1 (人数)	評価 2 (人数)	評価 3 (人数)	評価 4 (人数)	評価 5 (人数)	平均 評価 (点数)
質問 1 : アクセスできるまでの手順が分かりやすかった	0	2	6	1	3	3.4
質問 2 : 字幕の拡大や縮小機能を使用して、文字が見やすくなった	0	1	3	3	5	4
質問 3 : 自動更新の機能を使用したら、リアルタイムに字幕が表示された	0	2	1	4	5	4
質問 4 : 画面のデザインが分かりやすかった	1	1	1	6	3	3.8
質問 5 : 簡単に使えました	0	0	1	6	5	4.3

② セルフモードでの講演

質問内容	評価 1 (人数)	評価 2 (人数)	評価 3 (人数)	評価 4 (人数)	評価 5 (人数)	平均 評価 (点数)
質問 1 : 操作方法がわかりやすかった	0	3	5	3	2	3.3
質問 2 : 操作手順が単純だった	0	1	2	5	5	4.1
質問 3 : 普段通り講演できた（ノート意識してしまって、話しにくいことはなかった）	0	3	2	4	4	3.7
質問 4 : セルフモードで、講演しながら補助することは容易だった	0	2	3	2	6	3.9

質問 5 : セルフモードでの講演は、補助者有のときと比べて講演しやすかった	2	4	2	2	3	3
質問 6 : 聴覚障害者が実際に来場するのであれば、ノートを準備してもよい	0	0	3	1	9	4.5
質問 7 : 聴覚障害者が実際に来場するのであれば、本システムを使って講演したい	0	0	3	2	8	4.4

③ 補助者ありでの講演

質問内容	評価 1 (人数)	評価 2 (人数)	評価 3 (人数)	評価 4 (人数)	評価 5 (人数)	平均 評価 (点数)
質問 1 : 操作方法がわかりやすかった	0	3	4	5	1	3.3
質問 2 : 操作手順が単純だった	0	2	4	5	2	3.5
質問 3 : 自分が参加する学会等において、聴講者に聴覚障害者がいる場合、本システムを使った補助に協力したい	0	0	2	7	4	4.2

■ 講演者・補助者向け 2

- ・ 実験対象の選定
補助者ありモード

- ・ 実験の構成と内容

本実験は、講演本番環境で講演する。講演終わった後、5段階評価(1 そう思わない、3 どちらでもない、5 そう思う)によるアンケートを実施し、発話形式で評価を行う。

- ・ アンケートの質問と結果

質問内容	評価 1 (人数)	評価 2 (人数)	評価 3 (人数)	評価 4 (人数)	評価 5 (人数)	平均 評価 (点数)
質問 1 : 操作方法がわかりやすかった	0	0	1	1	3	4.4
質問 2 : 操作手順が単純だった	0	1	1	1	2	3.8
質問 3 : 予測変換簡単に使えた	3	1	1	0	0	1.6
質問 4 : 予測変換を使って補助速度を向上できた	3	1	1	0	0	1.6

質問 5：自分が参加する学会等において、聴講者に聴覚障害者がいる場合、本システムを使った補助に協力したい	1	0	1	1	2	3.6
--	---	---	---	---	---	-----

6.4.4 評価実験の考察

本節では、評価実験で集計した結果を分析して考察を述べる。

■ 聴講者向けの実験

「2 画面での閲覧」のアンケートにおける質問項目 2 の評価結果から、普段見慣れている情報保障と大体同じであり、見づらいところが特にないことを分かった。同アンケートの質問項目 5 の評価項目 1 の評価結果から、聴覚障害者が字幕の表示をちゃんと見えることを分かった。質問項目 5 の評価項目 2 に対して、発話で意見を聞く時に「自動補助する時、文字の大きさが途中で変わるとどこを読んでいるか分からなくなることがある、そのせいで目を疲れる」と意見があった。その理由は、スライドノートの記述量が多い場合、全ての字幕をスクリーン上に表示しようとして字幕を縮小していくため見にくくなるのである。しかし、その問題に対して、ユーザビリティが低いことが設計段階で予想していたが、「補助者がいないとき、自動補助表現が有効か」を尋ねると、有効であると感じている人が多数であったので、実現することを決めた。この問題への対応方法は、レーザーポイントで字幕を追って発表することである。質問項目 5 の評価項目 3 から、この方法が有効であることが分かった。

1 画面での閲覧する時、質問項目 1 と質問項目 2 から、1 画面が有効だが、見にくいことが分かった。また、被験者からの意見より、1 画面での閲覧が字幕の表示幅が小さいと字幕文字が小さいということが分かった。

1 画面と 2 画面の両方のアンケートにおいて、字幕の大きさや行間といった表示形式に関する質問をした。アンケート結果や被験者から頂いた意見より、個人の好みによって見やすい表示形式が違ってくるのが分かった。本システムでは文字の大きさや配色などを、ある程度自由に調整できる。従って、講演者がそれらを調整することによって、1 画面での閲覧のアンケートと 2 画面での閲覧のアンケートにおける質問項目 1 と 2 の問題を解決できると判断する。

端末での閲覧する時、質問項目 6 と発話の意見から字幕の配信機能がとても有効という意見が多数を占めた。そして、質問項目 2「字幕の拡大縮小機能」、質問項目 3「更新・自動更新機能」を利用して、字幕を見やすくできることが分かった。質問項目 1 と質問項目 4 から、少数の聴講者に対して画面デザインが分かりにくいことと多数の聴講者に対してアクセス手順が分かりにくいことが分かった。特に、「フラッシュが大きい」、「自動スクロールがない」という開発中に気が付くことができなかった問題点を発見できた。

■ 講演者・補助者向けの実験

端末での閲覧する時、聴講者向けの実験から得られた結果と大体同じであった。

セルフモードでの講演アンケートで、ノートの前準備とセルフモードの操作状況について質問した。本システムにおける既存の講演字幕手法と異なる点は、講演者にノートの前準備が必要な点である。従って、この点に着目し質問した。質問項目 6 と質問項目 7 から、聴覚障害者が来る場合、ノートを準備してもよいという回答が得た。また、質問項目 1 から質問項目 5 までの質問から、セルフモードでの講演が情報保障として有効であるが使いにくいということが分かった。しかし、被験者から直接意見を聞いたところで意見を聞く時、慣れたら使いやすい回答が多数あった。

補助ありでの講演で、質問項目 3 から、補助者に対してシステムを使える結果が得られた。しかし、質問項目 1 と質問項目 2 の平均評価が普通であったので、システムを使えるけどつかいやすくないと判断した。発話で意見を聞く時、「編集・削除操作が間に合わない」などの意見を得た。また、予測変換の現場対応について、「予測変換が簡単に使えた」と「予測変換を使って補助速度を向上できた」について質問したが、平均評価が 2 以下であった。

■ 考察

以上得られた問題や質問に対して、それぞれ機能の担当者に報告した。そして、チームレビューで各問題に対する対策を議論し、優先順位を付けた。例えば、優先順位高い問題として、端末での閲覧する時、「フラッシュが大きい」と「自動スクロールがない」などが挙げられる。これらの問題は、端末での字幕閲覧時に、字幕の見やすさに大きな影響を与えたので、まず解決すべきと判断した。そして、実装担当者が技術調査を行い、解決した。

これらのユーザ評価により、問題発見・解決したことでユーザビリティ向上の目標を達成した。

6.4.5 今後の展望

ユーザ評価より、本システムは有効であると感じた方が多かったものの、同時に、システムの UI や操作方法については使いにくいと感じた方も多かった。これらの点に対する改善対策を行えば、システムのユーザビリティをさらに高められるものと期待される。

また、本システムを運用するため、講演者が予め原稿をスライドノートに書く必要がある。この労力を軽減する事ができれば、利用者の増加が見込めるものと思われる。具体的な方法としては、音声認識を利用したノートの自動作成機能が考えられる。

7 終わりに

本報告書は、「スライドノートを活用した講演字幕システムの実現」のプロジェクト開発に関する報告書である。

本プロジェクトは、研究発表会において発表内容がある程度決まっているという特徴を利用し、講演者は事前に講演内容を PowerPoint のノート機能に記述し、字幕として活用する講演字幕システム「CaPPTioner」を実現した。これにより、人的コスト・特殊な機材の制約の問題を解決でき、より広く実施できる情報保障を実現する。

「CaPPTioner」は、スライドノートの抽出や字幕の表示といった基本機能とより多くの環境で利用するための拡張機能がある。筆者は、拡張機能の 1 つである、外部システムとの連携機能を担当した。外部システムとの連携機能として 2 つを実現した。1 つは、講演大会の質疑応答時に字幕入力が増えすぎて問題を解決するため、聴覚障害者への情報保障である IPtalk との連携である。もう 1 つは、発話が困難な障害者が講演を行いやすいため、字幕を読み上げる音声合成エンジンとの連携である。また、成果物の品質を確保するため、必要な機能が満たされるかどうかの確認のための機能テストとユーザ視点から問題点を発見するためのユーザビリティ評価実験を担当した。

結果として、聴覚障害者が実際に来場するのであれば、本システムを利用したいという評価が約 9 割の情報保障の実施者から得られて、字幕の見やすさが従来手法とそれほど変わらないという評価が過半数以上の聴覚障害者から得られた。以上の評価で、情報保障の実施のための運用コストを抑える目的を達成できることを判断した。ただし、操作について不便なところがあるので、将来的により使いやすいシステムの実現を目指している。

謝辞

本報告書を執筆するまでに、多くの人にご指導、ご協力をいただきました。

本プロジェクトの課題担当教員である秋川友宏准教授には、日頃から多くのご指導をいただきました。プロジェクトを進めていく上で困っている際には相談にも乗っていただき、多くのことを学ばせていただきました。心より感謝申し上げます。報告書の執筆や発表などにおいて大変丁寧なご指導とご助言をいただきました。ここで厚く御礼申し上げます。

筆者の指導教官である田中二郎教授には、筆者が本学へ入学する以前から本コースの内容や受験に関しての助言や応援のメールを頂き、大変お気遣い頂きました。入学後も事あるごとにご指導を頂き、筆者の学業や進路において大きな励みになりました。

被験者実験の際にも多くの方にご協力をいただきました。特に筑波技術大学の西岡知之先生には被験者実験を実施するため聴覚に障害のある学生を対象とした評価実験の機会を設定いただき、心より感謝を申し上げます。

本プロジェクトのチームメンバである顧毅捷氏、落合遥堂氏、後藤慎也氏、横山快氏にはプロジェクトを遂行するにあたり、様々な面でお世話になりました。ここに感謝の意を表します。

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 7 期生のみなさまには、プロジェクト活動から日常生活まで数多くのご意見やご指摘をいただき、充実した学生生活を送ることが出来ました。ここまで来ることが出来たのは、みなさまから常に刺激を受けることが出来たからだと思っています。この場を借りて心より御礼申し上げます。

最後に、これまで私を支えてくれた家族や友人、大学院生活でお世話になったすべての方々に、心より感謝いたします。本当にありがとうございました。

参考文献

- [1] “要約筆記者養成等調査検討事業報告書,” [オンライン]. Available:
http://www.mhlw.go.jp/bunya/shougaihoken/cyousajigyoku/jiritsushien_project/seika/research_09/dl/result/11-07a.pdf.
- [2] “要約筆記 wiki ページ,” [オンライン]. Available:
<http://ja.wikipedia.org/wiki/%E8%A6%81%E7%B4%84%E7%AD%86%E8%A8%98>.
- [3] 中村研二, “プロの道具、ステノタイプ【ものづくりの視点】,” [オンライン]. Available: http://www.diva.co.jp/archive/mail_magazine/2012/146b.html.
- [4] 田中芳則, “アメリカでの聴覚障害学生の就学支援,” 2005.
- [5] “IPtalk ホームページ,” [オンライン]. Available:
http://www.geocities.jp/shigeaki_kurita/.
- [6] 三好茂樹, “聴覚障害学生支援セミナー分科会 2. 「音声認識を使った情報保障」,” 筑波技術大学・障害者高等教育研究支援センター.
- [7] “ミーティング・会議支援のアプリ UD トーク,” [オンライン]. Available:
<http://plusvoice.jp/UDtalk/>.
- [8] 竹下恵, パケットキャプチャ入門第3版(LAN アナライザ Wireshark 活用術), リックテレコム, 2014.
- [9] B. Usha, V. Ashutosh, S. Mukul, “Evaluation of the Capabilities of WireShark as a tool for Intrusion Detection,” International Journal of Computer Applications, Volume 6- No.7, pp.358-363, September 2010.
- [10] Wireshark を使った解析. [オンライン]. Available: http://beginners-network.com/wireshark_analyze.html.
- [11] C. YOO, H.-W. JIN, S.-C. KWON, “Asynchronous UDP,” IEICE transactions on communications, VOL.E84-B, No.12, NO.12 DECEMBER 2001.
- [12] 桃崎浩平, 原義幸, 正井康之, 松浦博, 新田恒雄, “音声合成/認識 API と応用ソフト開発環境の動向 97(16),” 情報処理学会研究報告. SLP, 21-26, 1997-02-07.
- [13] 住吉貴志, 李晃伸, 河原達也, “音声認識エンジン Julius/Julian の API 実装,” 情報処理学会研究報告. SLP, 音声言語情報処理 2001(68), 91-96, 2001-07-13.
- [14] “ISO/IEC 25000,” [オンライン]. Available:
<https://www.iso.org/obp/ui/#iso:std:iso-iec:25000:ed-1:v1:en>.

- [15] 三宅武司, 富士仁, 西山茂, “設計レビュー項目の十分性に関する検討,” 全国大会講演論文集 第 47 回平成 5 年後期(5),275-276, 1993-09-27.
- [16] “ソフトウェア品質特性(ISO/IEC9126)の解説,” [オンライン]. Available: <http://www15.ocn.ne.jp/~squality/software-quality/iso9126.htm#%82V%81D%81u%8Eg%97p%90%AB%81v>.
- [17] “CFD 技法の評価と事例に基づくガイドの作成,” [オンライン]. Available: http://www.juse.or.jp/software/pdf/20_spc/8/7_d_report.pdf.
- [18] “CMMI 導入の為に GQM 手法による測定データの研究,” [オンライン]. Available: http://www.juse.or.jp/software/pdf/20_spc/1/1_b_report.pdf.
- [19] 菅田直美, ソフトウェア品質会計 NEC の高品質ソフトウェア開発を支える品質保証技術, 日科技連, 2010.
- [20] “平成 18 年身体障害児・者実態調査結果,” 厚生労働省社会, 2008-3-24.
- [21] 榛葉悟, 室溝誠治, 内山裕文, “アジャイル開発製品のテストに対する品質工学の適用品質工学 19(5),” 品質工学会, pp.584-591, 2011-10.
- [22] 足達直, 平野健一, 山田敦, 田中恵美, “アジャイル開発における品質管理の取り組みと評価,” プロジェクトマネジメント学会誌 13(2),pp.24-29, 2011.