



卒業研究報告書

平成26年度

研究題目

要求仕様書のレビューにおける読み方の教示によるレビュー効率の変化

指導教員 上野秀剛 助教

氏名 葛井健文

平成27年1月8日 提出

奈良工業高等専門学校 情報工学科

要求仕様書のレビューにおける 読み方の教示によるレビュー効率の変化

上野研究室 葛井 健文

本研究の目的は、要求仕様書のレビューを行う際に要求仕様書の特性についての教示を行うことの有用性を明らかにすることである。要求仕様書の内容について、曖昧な記述や矛盾があればバグの原因となることを教示することで、レビューにおける効率が向上すると考えられる。本研究では被験者を1)教示無しグループと、2)簡易教示グループ、3)厳密教示グループの3つに分け、教示なしグループには教示を行わず、簡易教示グループと厳密教示グループにはそれぞれ簡易な教示と厳密な教示を行い、各グループのレビュー効率の差を計測する。実験では要求仕様書レビューを行う。実験の結果、教示を行ったグループは教示を行わなかったグループに比べ、バグの指摘精度は低下したものの、バグ発見率は向上した。このことから、要求仕様書の特性の教示には要求仕様書のレビュー効率の向上に有用であるといえる。

目次

1	はじめに	2
2	関連研究	3
2.1	レビューにおける読み方の教示	3
2.2	要求仕様書のレビュー	3
3	教示	4
3.1	要求仕様書の特性	4
3.2	簡易な教示	5
3.3	厳密な教示	5
4	実験	7
4.1	レビュー対象	7
4.1.1	非あいまい性に関連するバグ	8
4.1.2	完全性に関連するバグ	8
4.1.3	無矛盾性に関連するバグ	9
4.2	実験環境	10
4.3	実験手順	11
5	結果と考察	12
5.1	バグ発見率	12
5.2	発見したバグの傾向	14
5.3	指摘数と指摘精度	15
6	おわりに	16
	謝辞	17
	参考文献	18

1 はじめに

ソフトウェアレビューとは成果物に混入したバグを発見するために、開発者がソースコードや要求仕様書を精読することである。要求仕様書とは、ソフトウェアの機能や性能を定義する文書である。要求仕様書はソフトウェア開発の初期段階で作成されるため、要求仕様書のバグが開発の後工程で発見された場合、修正に多大なコストがかかる [1]。そのため、要求仕様書のレビューは開発コストの削減という面において重要である。

ソフトウェアレビューにおいては、レビュー前に読み方を教示することでレビュー効率が向上することが明らかになっている [2]。一方で、先行研究で用いられた教示内容はソースコードレビューを対象に作成されており、要求仕様書のレビューにおいても同様にレビュー効率が向上するか明らかになっていない。ソースコードレビューの際に用いる設計書は要求仕様書レビューの段階で完成しておらず、またソースコードと要求仕様書ではバグとなる記述の特徴が異なるために、ソースコードレビューのための教示はそのまま適用できない。そこで、本研究では要求仕様書を対象としたレビューの効率を向上させるための教示について、有効性を被験者実験により調査する。

ソースコードレビューを対象とした先行研究では、プログラムの全体像を把握することや、設計書に書かれたメソッドの設計とソースコードに書かれたメソッドの動作が等しいか確認することでレビュー効率が向上するとして教示内容が作成されている。本研究では、レビュー作業者に要求仕様書の特性を理解させることでレビューの際にバグを発見する助けになり、要求仕様書レビュー効率の向上に繋がると考え、要求仕様書の特性について記述した教示内容を作成した。この研究から要求仕様書のレビュー効率の向上につながれば、ソフトウェア開発時の手戻りが減少し、ソフトウェア開発のコスト削減につながる。

以下、2章では関連研究について述べ、3章で教示内容について説明し、4章で実験環境や実験手順などを示し、5章で実験結果を述べ、考察を行う。6章ではまとめと今後の課題について述べる。

2 関連研究

2.1 レビューにおける読み方の教示

應治はソースコードのレビュー時に教示を行うことでレビュー効率が向上するか分析している [2]. 應治の研究における教示は1)設計書をよく読む, 2)コード全体にざっと目を通す, 3)設計書に書かれているフィールドやメソッドが正しくソースコードに実装されているか確認する, の3つだった. これは, 作業者がプログラムの内容を把握し, プログラムへの要求を理解すること, プログラムの全体像を把握すること, 設計書のメソッドの設計とソースコードのメソッドの動作が等しいかを確認することがコードレビューの効率の向上に繋がるという考えに基づいて作成されている.

被験者実験において教示有りグループと教示無しグループのバグ発見率を比較した結果, 教示有りグループのバグ発見率が6.67%向上した. 一方で, 要求仕様書のレビューにおいても教示を行うことでレビュー効率が向上するかは明らかになっていない. 本研究では要求仕様書を対象としたレビューの効率を向上させるために, 要求仕様書の特性について教示を行うことでレビュー効率が向上するか調査する.

2.2 要求仕様書のレビュー

これまでに要求仕様書を対象にしたレビューは複数行われている. [3][4][5] 野中らは要求仕様書の特性に着目した個人レビュー手法を提案, 評価している [6]. 実験では, レビュー担当者の知識や経験に基づいてレビューを行う Ad Hoc Reading(AHR), 過去の経験などに基づいたチェックリストを用いてレビューを行う Checklist-Based Reading(CBR), レビュー実施時のシナリオを作成し, それに基づいてレビューを行う Perspective-Based Reading(PBR)の3つのレビュー手法について比較している. 実験の結果, PBRによるレビューではAHRに比べて要求仕様書の特性の1つである完全性に関連するバグの発見率が6%向上している.

本研究では, CBRやPBRではなく, 教示によって要求仕様書のレビュー効率が向上するか調査する. 野中らの研究においてCBRで用いたチェックリストやPBRで用いたシナリオは, いずれも要求仕様書内にバグとなる記述は含まれていないか, という書き方をしている. 本研究では, 記述のあいまいさに関する「非あいまい性」, 記述の十分さに関する「完全性」といった要求仕様書の特性について教示することで, それら特性についてのバグを発見しやすくなると考えた. また, 特性についてどこまで教示するかでレビュー効率が異なると考え, 特性に対して異なる教示をしたときのレビュー効率を被験者実験で計測する.

表 1: 要求仕様書の特性

特性	概要
妥当性	要求仕様書の記述が，ソフトウェアが満たすべき要求であること
非あいまい性	要求仕様書の記述が，一意にしか解釈できないこと
完全性	要求仕様書の記述が，ソフトウェアの挙動に関して十分に書かれていること
無矛盾性	要求仕様書の記述が，矛盾していないこと
順位付け	要求に対して重要度が設定されていること
検証可能性	要求仕様書の記述が要求を満たしているかをチェックできること
修正可能性	要求が変更されても修正が可能であること
追跡可能性	中間生産物がどの要求に関わっているのかがわかること

3 教示

本研究では要求仕様書レビューにおいて作業者に教示を行うことでレビュー効率が向上するか調査する。一般に，要求仕様書の作成にかけることのできる時間は短いため，効率的に品質を確認・改善することが求められる。要求仕様書をすべての観点から評価することは困難であるが，要求仕様書の8つの特性が満たされていることの確認は比較的短時間で行うことができ，また，要求仕様書の品質が十分であるかの確認として有用とされている[7]。そこで，要求仕様書の特性に着目したレビューをすることで要求仕様書の品質を高められると考えた。本研究では，要求仕様書の特性について教示を行うことで，要求仕様書のバグの源となる部分に注目して読むことの助けになり，作業効率が向上すると考えた。

3.1 要求仕様書の特性

要求仕様書の作成において，バグの原因とならないよう要求仕様書が満たしているべき8つの特性が存在する[8]。それぞれの特性について表1に示す。これらの特性はいずれもニーズを正しく仕様化できているか確認するための項目が中心となっている。要求仕様書の記述がこれらの特性を満たしていれば，ニーズに正しく則ったソフトウェア開発が可能になる。一方で，要求仕様書の記述がこれらの特性を満たしていないと，その要求仕様書に基づいて作成されるソフトウェアの品質に悪影響を与えるため，そのような記述はバグと判断される。

本研究では，要求仕様書の記述から上記の特性を満たしていない箇所をレビューにおいて指摘できるような教示をすることで，効率よく要求仕様書のバグが発見できると仮定した。先行研究[2]においてもレビュー効率を高めるためにレビューにおける読み方を教示しているが，ソースコードレビューを対象として設計書と比較するような教示であり，要求仕様書の特性に着目した教示とは異なる。

本研究では、8つの特性のうち、a)非あいまい性、b)完全性、c)無矛盾性の3つの特性に着目して教示を行う。ニーズ記述書と要求仕様書のみを見て行うレビューでは他の特性についての判断は難しいと考えた。また、本研究では簡易な教示と厳密な教示の2種類の教示内容を作成し、それぞれの教示内容の効果を被験者実験により調査する。簡易な教示では、要求仕様書の特性についてのみ教示する。厳密な教示では、要求仕様書の特性に加え、どのようなときにその特性が満たされなくなるか教示する。次節以降で簡易な教示と厳密な教示の詳細を説明する。

3.2 簡易な教示

簡易な教示として、以下の3点を被験者に教示する。

- 記述内容があいまいで一意に解釈できず、設計の際に困ることがあります。
- すべての要求が仕様書に記述されていなかったり、必要でない情報が記述されていることがあります。
- 要求仕様書内で記述が矛盾していることがあります。

簡易な教示では、要求仕様書の特性である非あいまい性、完全性、無矛盾性について、各特性が満たされていない状態があることを教示する。この教示により、作業者は要求仕様書のレビューの際に要求仕様書のバグと判断される記述についての予測を立てることができ、レビューの際にバグを発見する助けになると考えた。

3.3 厳密な教示

厳密な教示として、以下の3点を被験者に教示する。

- 仕様書の中で用語、指示範囲などがあいまいに記述されて一意に解釈できず、設計の際に問題が発生することがあります。例えば、同じ単語が複数の意味を持っていたり、きちんと範囲を示すべき箇所で代名詞が使われていて具体的な指示範囲がわからなかったり、正しく意味を理解できない用語があると、読み手によって解釈が異なり、バグの源となります。
- 仕様書に記述されるべき内容が記述されていない（記述漏れ）と、設計の際に要求を満たせなくなることがあります。例えば、ある機能の中で記述すべき事柄が一部記述されていなかったり、必要な機能が丸ごと抜けていることもあります。このような記述漏れがあると必要な機能や処理が作成されず、バグの源となります。
- 仕様書に記述されている内容が別の場所の記述と一致してなかったり、ある記述と別の記述を両方正しいものとして設計できない場合があります。こ

のような矛盾を含む記述があると、要求を満たすシステムが作成できず、バグの源となります。

厳密な教示は要求仕様書の特性である非あいまい性、完全性、無矛盾性について、各特性が満たされていない状態があることに加え、なぜ満たされないのか、どのような記述がバグの原因となるか例を交えて教示する。この教示により、作業者は簡易な教示による効果に加え、特に注意すべき箇所や、ある記述がバグかどうかの判断が容易になり、簡易な教示よりもさらに要求仕様書のレビュー効率の向上に繋がると考えた。

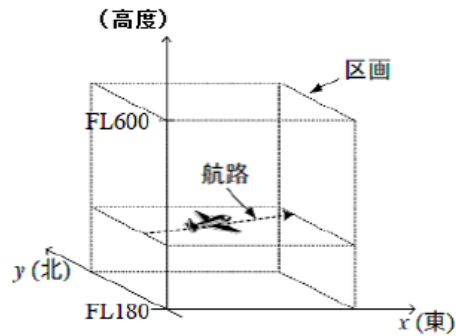


図 1: 三次元領域のイメージ

表 2: 実験で使した文書

文書	内容	章数	行数
ニーズ記述書	飛行計画の管理システムについてのニーズが記述されている	3	90
要求仕様書	飛行計画の管理システムについての仕様がニーズ記述書をもとに記述されている, システムの概要と機能, データ定義, インターフェースについて記述されている	6	216

4 実験

ニーズ記述書をもとに, 要求仕様書のバグを見つけてもらう実験を行う. 被験者を1) 教示無しグループと, 2) 簡易教示グループ, 3) 厳密教示グループの3つに分け, 教示の効果を調査する. 被験者は情報工学科5学年の学生16名で, 要求仕様書レビューの経験がなく, 要求仕様書レビューに対しての理解度はほぼ一律であると考えられる.

4.1 レビュー対象

レビュー対象は野中らの研究 [6] で使用された要求仕様書にバグを埋め込んだものである. 本実験で使用したニーズ記述書と要求仕様書について表2に示す. ニーズ記述書と要求仕様書は飛行計画管理システムについての文書である. このシステムは図1に示す三次元領域内を飛行する飛行機を管理するプログラムであり, 飛行機ID, パイロットの名前, 搭載燃料, 航路の情報からなる飛行計画を入力することで区画内を管理する.

要求仕様書のバグは要求仕様書の特性によって分類される. 本研究ではバグの作成と分類が容易で, ニーズ記述書と要求仕様書の2箇所から検出しやすいこと

から、非あいまい性、完全性、無矛盾性の3種類の特性に対応するバグを埋め込んだ。野中らの研究と同様に計21個(3種類の特性に対応するバグを7つずつ)のバグを埋め込んだ。

4.1.1 非あいまい性に関連するバグ

非あいまい性に関連するバグを次に示す。要求仕様書の記述が用語の意味や指示範囲について一意に解釈できないような書き方であった時、非あいまい性に関連するバグと判断される。ここでの指示範囲とは、「その」や「この」などの指示代名詞が指す範囲、「～し」、「あるいは」などの接続詞が指す範囲、「～のとき」や「～でないとき」などの条件節の範囲のことである。

- (1) 区画データという言葉が「その区画における飛行機IDのリスト」という意味で使用されているが、他の箇所でも「システムが管理する範囲」という意味でも使用されている。
- (2) 処理の場合分けについて「そうでない場合」という記述があり、具体的な条件が記述されていない。
- (3) 「飛行機の数」の正しさを確認するような記述があるが、何機ならば正しいのか明確に記述されていない。
- (4) 「チェックの結果によって、承認メッセージを出力し、当該飛行計画の高度を変更するか、承認できない理由を記述したメッセージを出力し、当該飛行計画の高度変更を拒否する」という記述があるが、チェックの結果が具体的にどうなればどちらの出力に分岐するかが明確に記述されていない。
- (5) 意味が説明されていない略語(TJFR)が使用されている。
- (6) 距離および針路の値が小数点を含むときに必要な桁数の変換について、どのように変換するか不明確。
- (7) 飛行計画リストのフォーマット例が一部省略されており、記述の意味が曖昧になっている。

4.1.2 完全性に関連するバグ

完全性に関連するバグを次に示す。要求仕様書の記述に、ニーズにある情報が記述されていなかったり、ニーズにない情報が記述されている場合、完全性に関連するバグと判断される。

- (1) メニュー欄に高度を変更する機能が表示されていない。

- (2) ニーズ記述書に記載のある「区画内を同時刻に同一高度で飛行する飛行機は水平方向に少なくとも0.5海里離れているか」判定する処理について記述が無い。
- (3) 飛行データを登録する際に承認メッセージを表示するようニーズ記述書には記載されているが、仕様書に記述がない。
- (4) 飛行計画を削除する機能について記述が無い。
- (5) データのフォーマット例について、ニーズ記述書に記載のある「終点」の項目が無い。
- (6) 飛行計画の情報について、ニーズ記述書に記載のない「速度」が追加されている。
- (7) 初期化された場合に残った飛行計画がどうなるかについて記述されておらず、必要な情報が不足している。

4.1.3 無矛盾性に関連するバグ

無矛盾性に関連するバグを次に示す。要求仕様書の記述に、別の記述と内容が一致していないものがあったり、ある記述を実装すると別の記述を正しいものとして設計できないような記述がある場合、無矛盾性に関連するバグと判断される。

- (1) 時刻表記のフォーマット指定がHHMMであるのに対して「14:10」とする記述がある。
- (2) 飛行計画について「最大で80件まで保持」する記述と「1日分の飛行計画を管理する」という記述がある。
- (3) 搭載燃料の表示について小数点第1位まで表示するという記述と整数で表示するという記述がある。
- (4) システムが管理する高度の範囲について、FL200以上FL600以下という記述とFL180以上600以下という記述がある。
- (5) 「x方向は北を、y方向は東を表す」という記述と「x方向が東、y方向北を表す」という記述がある。
- (6) 飛行機IDの定義について、整数5桁とする記述と6桁とする記述がある。
- (7) 飛行計画の表示機能は3章9節で説明するという旨の記述があるが、実際には3章8節で説明されている。

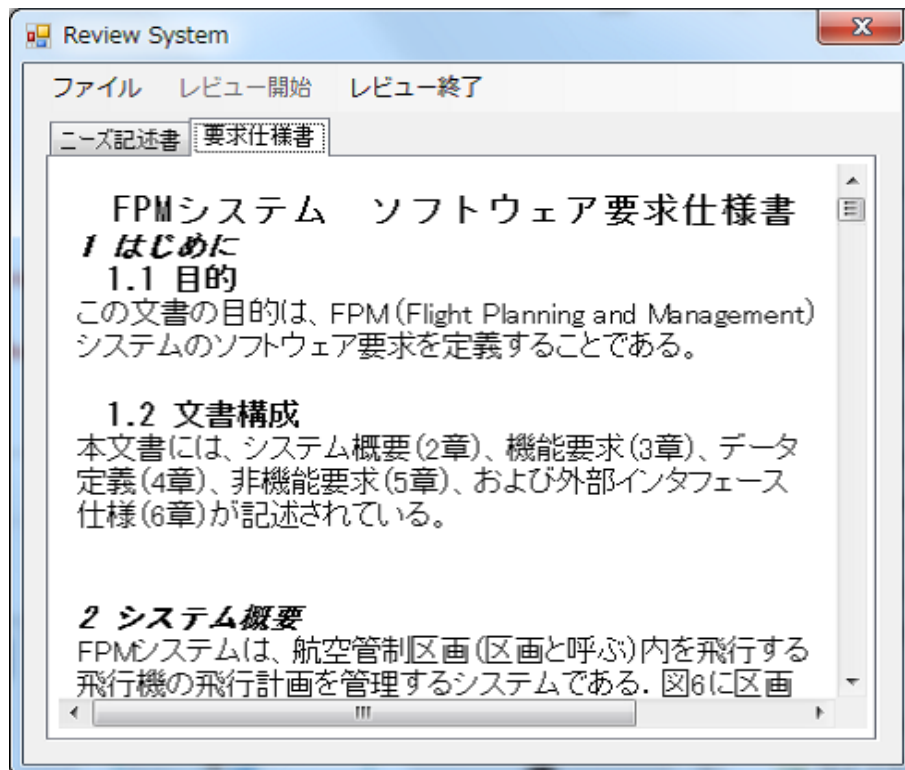


図 2: 実験用ツールの実行画面

4.2 実験環境

本研究では、ニーズ記述書と要求仕様書を複数タブで表示する実験用ツールを作成した。ツールの実行画面を図2に示す。ソフトウェアはC#言語によって作成されたGUIアプリケーションで、複数のタブにニーズ記述書と要求仕様書を表示し、画面上部のタブをクリックすることで表示する文書を切り替えることができる。作業者は要求仕様書をレビューし、バグがある行を見つけたらダブルクリックすることでバグ報告用のテキストボックスが表示される。作業者はテキストボックスにバグの内容を記述することで、発見したバグを報告する。

また、本ツールはレビュー開始と同時に時間をカウントし、報告の箇所と内容、報告された時間を記録し、CSVファイルとして出力する。出力されるCSVファイルの例を図3に示す。図3のtimeは指摘が行われた時間(秒)を、placeは指摘された箇所を、descriptionは指摘の内容を表している。最終行はレビュー終了のメッセージである。

実験に使用したパソコンは全てOSがWindows Vista、CPUがIntel(R) Core(TM)2 6300@1.86GHzである。実験をした部屋は本学情報工学科棟の情報処理実習室であり、被験者1人ずつに十分なスペースと、他人の画面を見ることが出来ない間隔がある。

time	place	description
408.861,	3.6.2,	ニーズに存在しない仕様である
533.981,	3.7.4,	最大飛行機数が異なる
575.101,	3.8.2,	終点座標が無い
1714.653,	review,	finish

図 3: CSV ファイルの出力例

4.3 実験手順

まず, 被験者の能力や実験データの偏りを避けるために, 無作為に5人, 5人, 6人のグループに分割し, それぞれを”教示なし”と”簡易な教示”と”厳密な教示”とする. 次に全てのグループに対してテキストファイルで注意事項を表示し, 教示を行うグループに対しては同時にテキストファイルで教示内容を示す. その後, 被験者を実験環境に慣れさせるための練習タスクを実施してもらう. 練習タスクはニーズ記述書と要求仕様書に対するレビュー作業で, 実験用ソフトウェアでタブが切り替えられること, 任意の行にダブルクリックで指摘が書き込めることを確認してもらう. その後, 要求仕様書レビューを行う.

練習タスク終了後に, 本番のレビューではバグを発見したらその箇所にバグの理由を指摘してほしい旨と出来るだけ多くバグを発見してほしい旨を伝える. レビュー時間は30分で, 途中終了は認めず30分を超えたら強制的に終了させる.

表 3: 各グループのバグ発見率

グループ	非あいまい性	完全性	無矛盾性	全体
教示なし	11.4%	14.3%	28.6%	18.1%
簡易教示	14.3%	20.0%	28.6%	21.0%
厳密教示	14.3%	19.0%	21.4%	18.3%

5 結果と考察

5.1 バグ発見率

バグ発見率 $detection$ は各グループの被験者数を n , i 番目の被験者 ($i = 1, 2, \dots, n$) のバグ発見数を X_i とし, 以下の式で求める.

$$detection = \left(\sum_{i=1}^n X_i \right) / n \quad (5.1.1)$$

グループごとの平均バグ発見率を表3に示す. 表3より, 全体のバグ発見率は教示なしのグループが18.1%であったことに対し, 簡易教示グループは21.0%と約3%の差が見られた. 簡易な教示を行ったことで要求仕様書を読む際に着目すべき特性が分かり, 効率よくレビューをすることができたためと考えられる.

一方で, 厳密教示グループは18.3%と, 教示なしグループとほぼ同じで, 簡易教示グループよりも低下した. 厳密な教示を行ったことにより様々な箇所にバグとなる記述が存在する可能性に気付き, 全ての箇所の記述について時間をかけて確認したためにレビュー時間内に確認しきれなかったことが原因と考えられる. 各グループのバグ発見率の差に有意差があるか確かめるために2グループ間におけるバグ発見率(全体)の差についてt検定を行ったが, 差が最も大きい教示なしグループと簡易教示グループでのP値が0.65と, いずれの結果にも有意な差は見られなかった. 次に, バグの種類ごとの発見率に着目すると, 教示を行った2つのグループは教示なしのグループに比べて完全性に関するバグ発見率が5%から6%向上している. 完全性は要求仕様書の記述に過不足がある場合に満たされなくなる. レビュー作業者が完全性について注意していない場合, 記述の過不足に気付いてもバグとして理解できず, 仕様化する際に付け加えられたり削られた記述だと考える恐れがある. しかし, 完全性を満たさない記述がバグであることを理解し, 注意をしながらレビューしたことで完全性に関連するバグを指摘できたと考えられる.

非あいまい性について着目すると, 教示を行った2つのグループは教示なしのグループに比べて完全性に関するバグ発見率が約3%向上している. 特に, 指示範囲が曖昧なことによるバグについては教示なしのグループよりも発見数が大きく増加しており, 教示によって曖昧な指示範囲に関する記述を検出しやすくなったと考えられる.

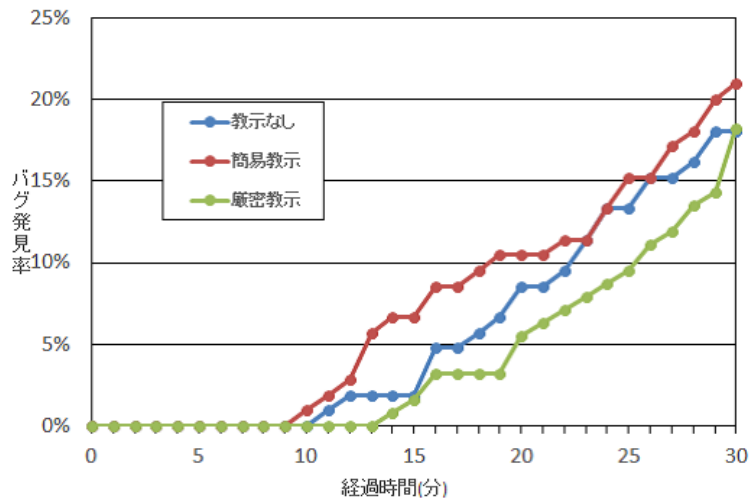


図4: バグ発見率の推移

無矛盾性について着目すると、厳密教示グループのバグ発見率が他の2グループに比べて低い。非あいまい性と完全性に関するバグはニーズ記述書と要求仕様書を比べることで発見できるのに対し、無矛盾性のバグを発見するためには要求仕様書内の記述を比較する必要がある。厳密教示グループの被験者はニーズ記述書と要求仕様書の比較に集中した結果として要求仕様書内の比較がおろそかになったと考えられる。一方で、無矛盾性に関連するバグのうち1つについては厳密教示グループの発見数が最も多かった。このバグはニーズ記述書の記述と要求仕様書の記述の間にも矛盾がみられることから、厳密教示グループはニーズ記述書と要求仕様書の記述の差については深く注意していたと考えられる。

各グループのそれぞれの特性に関連するバグ発見率に対してt検定を行った結果、差が最も大きい教示なしグループと簡易教示グループの完全性に関連するバグでのP値が0.53と、いずれの結果にも有意な差は見られなかった。

グループごとのレビュー時間とバグ発見率の推移を図4に示す。図4において、横軸はレビュー開始からの経過時間(秒)、縦軸はバグ発見率を表している。図から、いずれのグループもバグ発見率がレビュー終了直前まで増加しており、レビューを続けていけばバグ発見率がさらに上昇したと考えられる。特に厳密教示グループについてはレビュー後半でのバグ発見率の増加率が高く、レビューを続けることで多数のバグの発見が期待できると考えられる。

レビュー中の全時間を通じて厳密教示グループは簡易教示グループよりもバグ発見率が低い。これは、厳密教示グループは教示によって要求仕様書の特性を意識しながら読んだため、1つ目の指摘までにやや時間がかかったと考えられる。図から、簡易教示グループはレビュー開始8分頃からバグを検出しているのに対して、厳密教示グループは12分頃からバグ検出割合が増加を始めており、4分の

表 4: 各グループのバグ発見率

バグ	教示なし	簡易教示	厳密教示	全体
非あいまい性(1)	20.0%	0.0%	33.3%	18.8%
非あいまい性(2)	0.0%	20.0%	0.0%	6.3%
非あいまい性(3)	20.0%	60.0%	33.3%	37.5%
非あいまい性(4)	0.0%	0.0%	0.0%	0.0%
非あいまい性(5)	20.0%	0.0%	16.7%	12.5%
非あいまい性(6)	0.0%	20.0%	0.0%	6.3%
非あいまい性(7)	20.0%	40.0%	33.3%	31.3%
完全性(1)	0.0%	20.0%	0.0%	6.25%
完全性(2)	20.0%	0.0%	16.7%	12.5%
完全性(3)	0.0%	0.0%	0.0%	0.0%
完全性(4)	20.0%	20.0%	50.0%	31.3%
完全性(5)	20.0%	0.0%	33.3%	18.8%
完全性(6)	60.0%	20.0%	16.7%	31.3%
完全性(7)	0.0%	20.0%	0.0%	6.3%
無矛盾性(1)	0.0%	20.0%	0.0%	6.3%
無矛盾性(2)	40.0%	20.0%	16.7%	25.0%
無矛盾性(3)	20.0%	40.0%	33.3%	31.3%
無矛盾性(4)	80.0%	60.0%	50.0%	62.5%
無矛盾性(5)	40.0%	40.0%	50.0%	43.8%
無矛盾性(6)	0.0%	20.0%	0.0%	6.3%
無矛盾性(7)	0.0%	20.0%	0.0%	6.3%

差が見られる。厳密教示グループはニーズ記述書と要求仕様書の記述について深く読み進めた結果、1つ目の指摘までに時間がかかったと考えられる。一方、簡易教示グループはバグとなる記述についての注意のみをしながら読み進めた結果として、3つのグループの中で1つ目の指摘をするまでの時間が最も早くなったと考えられる。

5.2 発見したバグの傾向

個々のバグに対して各グループのバグ発見率を表4に示す。各グループが発見したバグの種類に着目する。教示なしグループは3グループの中で最もバグ発見率が低かったが、無矛盾性(4)“システムが管理する高度の範囲について、2種類の矛盾する記述がある”バグの発見数のみ3グループの中で最も多い。システムが管理する高度の範囲についてはニーズ記述書で文章と図の両方で示され頭に残りやすかったこと、教示を行ったグループは非あいまい性や完全性に関連するバグを意識してレビューを進めた結果、単純な数字の違いを見落としてしまったことなどが考えられる。

3グループの中で最もバグ発見率が高かった簡易教示グループは、このグループでのみ発見しているバグが6種類存在する。3種類の特性に関連するバグがそれぞれ2種類ずつ発見されており、簡易な教示を行ったことでバグのある記述を予測し、疑われる箇所によくの指摘をしたことが考えられる。

厳密教示グループは教示なしグループとバグ発見率がほぼ同じであったが、非あいまい性、完全性に関連するバグの発見数は増加している。非あいまい性に関連するバグでは、非あいまい性(1)“区画データという言葉が2種類の異なる意味で使用されている”が3グループの中で最も発見数が多い。教示によって、語句そのものはあいまいに見えなくても、同じ単語が別の意味で使われている、非あいまい性に関連するバグであると判断できたためと考えられる。

完全性に関連するバグでは、完全性(4)“飛行計画を削除する機能について記述が無い”バグと、完全性(5)“データのフォーマット例について、ニーズ記述書に記載のある「終点」の項目が無い”の発見数が多い。2つのバグはどちらも必要な情報が記述されていないことによりバグと判断される記述であり、厳密な教示によって記述漏れに注意しながらレビューを行った結果であると考えられる。

どのグループでも発見できなかったバグは非あいまい性(4)“条件に応じた処理の分岐について、条件が明確に記述されていない”と完全性(3)“飛行データを登録する際に承認メッセージを表示するようニーズ記述書には記載されているが、仕様書に記述がない”の2つである。どちらもバグとなる記述が非常に短く、発見難易度が高いため見落とししてしまったと考えられる。

5.3 指摘数と指摘精度

各グループの平均指摘数と平均正答数を表5に示す。教示を行った2グループは教示なしグループに比べて指摘数が増加しており、教示によって要求仕様書の様々な記述についてバグの可能性を疑った結果、指摘数の増加に繋がったと考えられる。一方で、教示を行った2グループとも正答率は教示なしグループに比べて低下している。教示を行ったことで、バグの可能性のある記述に対して多くの指摘を行ったことが原因と考えられる。これは、教示内容に具体例を増やすなどして、バグの含まれている記述と含まれていない記述を判断できるようになれば改善できると考えられる。

簡易教示グループは指摘数と正答数が教示なしグループに比べ増加したが、正答率が約12%低下している。バグの含まれている記述に対して予測を立て、30分のレビュー時間中に多数の指摘をしたことで多数のバグを検出できたと考えられる。教示なしグループの指摘数が少なく、正答率が高いことについて、教示を行わなかったことで疑わしい記述に対しての指摘をせずにレビュー中に作業者が確実にバグであると考えた箇所についてのみ指摘が行われたためと考えられる。

表 5: 各グループの指摘数と正答数

グループ	指摘数	正答数	正答率
教示なし	6.00	3.80	56.2%
簡易教示	10.60	4.40	44.8%
厳密教示	7.67	3.83	49.8%

6 おわりに

本研究では要求仕様書のレビューにおいて作業者に要求仕様書の特性について教示することでレビュー効率が上昇するか検証した。被験者を1)教示無しグループと、2)簡易教示グループ、3)厳密教示グループの3つに分け、バグ発見率と指摘精度に差があるか検証した。実験の結果、教示を行った2グループとも教示なしグループに比べてバグ発見率が向上した。これは要求仕様書の特性について教示をすることで要求仕様書を読む際にバグとなる記述についての予測を立てることができ、効率よくレビューをすることが出来たためと考えられる。

実験の結果から、要求仕様書のレビューにおいてはレビュー開始前に作業者に簡易な教示を行うことで教示をしないよりも効率よくレビューできると考えられる。また、厳密な教示は作業者のレビュー速度を下げたため、結果としてレビュー効率に寄与しないと考えられる。要求仕様書のレビュー開始時に要求仕様書の特性について簡易な教示をすることで教示をしないよりも同じ作業時間内でレビュー効率が向上し、ソフトウェア開発において有用であると考えられる。今後の課題として、最終的なレビュー効率を確かめるために実験時間を伸ばして実験を行うこと、被験者数を増やして統計的な分析を行うことが挙げられる。また、教示内容に具体例を増やすなど、バグのある記述についてははっきりと判断できるようにすることで正答率を高めることが出来ると考えられる。

謝辞

本研究を進めるに当たり、様々なご指導を頂きました指導教員の上野秀剛助教に深謝いたします。また、実験や輪講において多くのご指摘を下さいました上野研究室の先輩・同期の皆様には感謝いたします。そして、実験の際に被験者を快く引き受けてくださった皆様には感謝いたします。

参考文献

- [1] ロバート・L・グラス”ソフトウェア開発55の真実と10のウソ”(山浦恒央訳), 日経BP pp. 115-117(2004).
- [2] 應治沙織“ソフトウェアレビューにおける読み方の教示によるレビュー効率の変化”, 奈良工業高等専門学校情報工学科2013年度卒業論文(2014).
- [3] 松川文一, Giedre Sabaliauskaite, 楠本真二, 井上克郎:“UMLで記述された設計仕様書を対象としたレビュー手法CBRとPBRの比較評価実験”, オブジェクト指向最前線2002(2002).
- [4] A. A. Porter, L. Votta:“Comparing Detection Methods for Software Requirements Inspection: A replication using professional subjects, ” Empirical Software Engineering: An International Journal, Vol.3, No.4, (1988).
- [5] F. Lanubile, G. Visaggio:“Evaluating Defect Detection Techniques for Software Requirements Inspections, ” ISERN Technical Report-00-08(2000).
- [6] 岡本博幸, 内山敬太, 鈴木彩子, 高橋一仁, 西山茂, 野中誠“要求仕様書の特性に着目した個人レビュー手法の実験的評価”, ソフトウェア品質管理研究会2004年度分科会成果報告 第6分科会(2004).
- [7] アラン・M・デービス”成功する要求仕様失敗する要求仕様”(高嶋優子訳), 日経BP pp. 154-163(2006).
- [8] ビジネスコミュニケーション <http://www.bcm.co.jp/site/2004/2004Dec/04-youkyuu-kougaku-12/04-youkyuu-kougaku-12.htm>

実験で使用したニーズ記述書と要求仕様書を示す。

FPMシステム ニーズ記述書

FPM(Flight Planning and Management)システムは、航空管制区画(区画と呼ぶ)内を飛行する飛行機の飛行計画を管理するシステムである。FPMシステムのユーザは航空管制官である。

問題領域の概要

(1) ひとつの区画は、次の特徴をもった三次元領域で構成される。

- a) 垂直方向は、FL_LからFL600の範囲で区切られる。なお、飛行機は境界上を飛行することができる。
- b) 50海里₂以上150海里以下の矩形で表される。

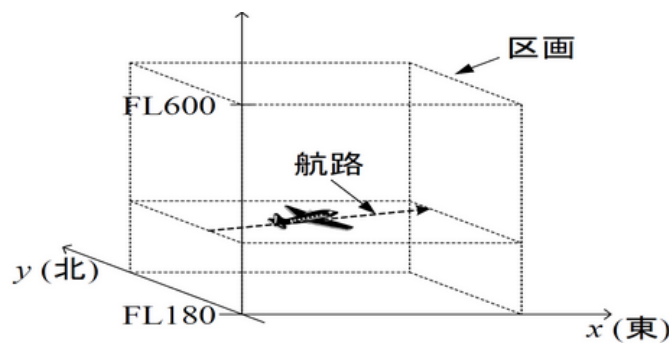


図5 区画のイメージ

(2) 同じ高度を同時に飛行できる飛行機は、最大で4機までであり、互いに少なくとも0.5海里離れていなければならない。

(3) 異なる高度を飛行する飛行機は、少なくともFL20離れていなければならない。

(4) 飛行機は、区画内を飛行するための飛行計画を持つ。飛行計画は次の情報を含む。

- a) 飛行機ID
- b) パイロットの名前
- c) 離陸時の搭載燃料(飛行時間で表記, 飛行機の種類と速度に基づいて算出される)
 - 出発点座標と出発時刻
 - 高度 (FLで表現, 区画内を飛行している間は一定)

- 針路（角度で表現，真北を0とする）
- 終点座標と終点時刻

(5)FPMシステムに入力された飛行機は，上記(2)と(3)の制約を満たした時に承認される。

(6)FPMシステムは，同一の飛行機に対して複数の飛行計画を保持できる。ただし，それらは区画内を同一時刻に飛行する飛行計画であってはならない。

¹ 1FL(Flight Level)=100 フィート

² 1海里 = 約1.8km

機能ニーズ

(1) 区画の初期化

(2) 飛行計画の入力

- 入力された飛行計画をチェックし、問題が発見されなかった場合は、承認メッセージを出力し、その飛行計画をFPMシステムに登録する。
問題が発見された場合は、承認できない理由を記述したメッセージを出力する。
- 飛行計画の入力には、対話式による個別入力と、ファイルによる一括入力がある。

(3) 飛行計画のチェック

- 飛行計画の正確性および登録済み飛行計画に対する整合性をチェックする。

(4) 飛行計画の削除

- 指定した飛行計画を削除する。

(5) 飛行計画の表示

- 指定した飛行機の飛行計画を出力する。

(6) 飛行機の検索

- 時刻を指定すると、その時刻に区画内を飛行する飛行機のリストを出力する。
- 時刻と高度を指定すると、その時刻・高度に区画内を飛行する飛行機のリストを出力する。

(7) 高度の変更

- 指定した飛行計画の高度を変更する。

(8) 飛行機間距離の出力

- 指定した2機の飛行機について、異なる高度の場合は飛行機間の高度差を出力する。同じ高度の場合は、飛行機間の最小距離とその時刻を出力する。

(9) 指定距離以下となる飛行機の組合せの検索

- 距離を入力すると、高度が同じ飛行機について、飛行機間の最短距離が指定距離以下となる飛行計画を持つ飛行機の組合せと、その時刻を出力する。

機能二一ズ

(1) 時刻は、HHMM表記とする。HHは時刻の24時間表記、MMは分である。例えば、午後3時3分は1503と表す。

(2) 搭載燃料は自然数で表す。

(3) 高度 (FL) は自然数で表す。

(4) 水平方向の距離（海里）は小数第2位まで表す。

(5) 速度 (mph) は小数第1位まで表す。

(6) 針路（度）は小数第1位まで表す。

(7) FPMシステムは、1日分の飛行計画が管理できればよい。

(8) FPMシステムは、一つの区画を維持管理できればよい。

(9) FPMシステムは、応答時間に関する制約はない。

(10) 飛行計画が承認されていない飛行機が、区画内に進入することはない。

FPMシステム ソフトウェア要求仕様書

1 はじめに

1.1 目的

この文書の目的は、FPM (Flight Planning and Management) システムのソフトウェア要求を定義することである。

1.2 文書構成

本文書には、システム概要 (2章)、機能要求 (3章)、データ定義 (4章)、非機能要求 (5章)、および外部インタフェース仕様 (6章) が記述されている。

2 システム概要

FPMシステムは、航空管制区画 (区画と呼ぶ) 内を飛行する飛行機の飛行計画を管理するシステムである。図6に区画のイメージを示す。

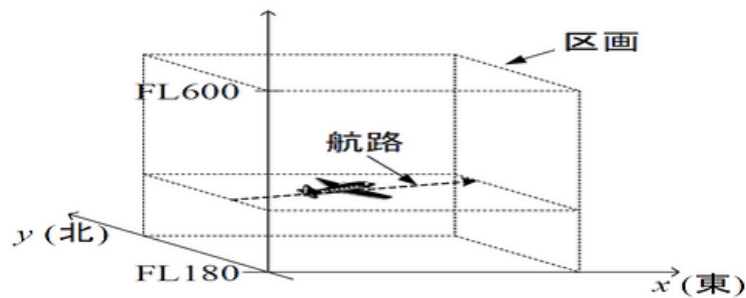


図6 区画のイメージ

FPMシステムは、一つの区画を維持管理し、区画に進入する飛行機の飛行計画を保持する。

FPMシステムに登録可能な飛行計画は、飛行計画に関する各種制約を満たしていなければならない。制約を満たしていない飛行計画は、FPMシステムへの登録が拒否される。

FPMシステムのユーザは航空管制官のみである。FPMシステムは、他のシステムとは相互作用しない。FPMシステムの入力インタフェースには、ユーザインタフェースと、ファイル入力インタフェースの二つがある。出力インタフェースは、ユーザインタフェースのみである。

3 機能要求

3.3 区画の初期化

FPMシステムを起動した際に、ユーザから区画データの入力を受け付ける。入力された値に基づいて区画を初期化する。区画に関する制約を満たしてい

ないデータが入力された場合は、そのデータの入力を拒否する。

3.4 飛行計画の入力

ユーザから飛行計画の入力を受け付ける。飛行計画データの正確性および整合性をチェック（チェック機能は3.6節および3.7節）し、承認メッセージを出力し、飛行計画を登録する。そうでない場合、承認できない理由を記述したメッセージを出力し、飛行計画の登録を拒否する。

3.5 飛行計画リストの入力

テキストファイルに記述された飛行計画のリストを読み込み、その並び順に従って、飛行計画を逐次的に入力する。各飛行計画について、データの正確性および整合性をチェックする（チェック機能は3.6節および3.7節）。データに問題が無い場合、飛行計画を登録する。

問題が発見された場合、承認できない理由を記述したメッセージを出力し、飛行計画の登録を拒否する。

なお、データファイルのフォーマットの正しさは保証されているものとし、フォーマットのチェックは行わない（データフォーマットは6.18節）。

3.6 飛行計画の正確性チェック

入力された飛行計画の正確性をチェックする。チェックは、各データ項目のフォーマットチェックを行う。また、次のチェックを行う。

- 出発点座標と終点座標は、区画の水平方向の境界面上にあるか。
- 搭載燃料は適切な値か。
- 速度は、出発点座標と終点座標の距離を、出発時刻と終点時刻の差で除算した値
- 針路は、航路に対して正しい値か。正しさの判定は図7に基づいて行う。

3.7 飛行計画の整合性チェック

入力された飛行計画の整合性をチェックする。チェック項目は次の通り。

- 区画内を同時刻に同一高度で飛行する飛行機の数は正しいか。
- 区画内を同時刻に飛行する飛行機は、少なくともFL20離れているか。

なお、上記の制約に従うと、区画内には同時刻に最大86機の飛行機が飛行でき、システムによって管理される。

3.8 飛行計画の表示

ユーザが指定した飛行機IDの飛行計画を表示する。

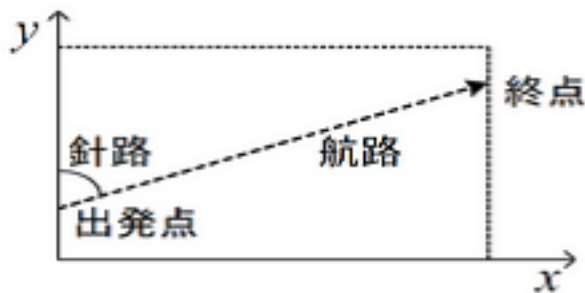


図 7 針路と航路

飛行計画は、次のフォーマットで画面に出力される。

```

飛行機ID: DAL111
パイロット名: J. Doolittle
搭載燃料: 60.0
出発点: 東10.00海里 北40.00海里
出発時刻: 1410
高度: FL220
速度: 300.0mph
針路: 39.8度
終点時刻: 1425

```

3.9 飛行機の検索（時刻指定）

ユーザが指定した時刻に区画内を飛行する飛行機について、飛行機IDのリストを表示する。ユーザが指定した時刻と、出発時刻または終点時刻が同一の飛行計画を持つ飛行機も表示される。指定時刻が無効な値の場合、エラーメッセージを表示し、検索を行わない。

3.10 飛行機の検索（時刻・高度指定）

ユーザが指定した時刻と高度に区画内を飛行する飛行機について、区画データを表示する。ユーザが指定した時刻と、出発時刻または終点時刻が同一の飛行計画を持つ飛行機も表示される。指定時刻および高度が無効な値の場合、エラーメッセージを表示し、検索を行わない。

3.11 高度の変更

「飛行計画の表示」機能(3.9節)によりユーザが特定した飛行計画に対して、ユーザから新しい高度の数値の入力を受け付ける。高度を変更した飛行計画データの正確性および整合性をチェックする(チェック機能は3.6節および3.7節)。チェックの結果によって、承認メッセージを出力し、当該飛行計画の高度を変更するか、承認できない理由を記述したメッセージを出力し、当該飛行計画の高度変更を拒否する。

3.12 飛行機間距離の出力

ユーザが指定した2機の飛行機IDについて、それらの飛行機が区画内を同時に異なる高度で飛行する場合は、飛行機間の高度差を出力する。それらの飛行機が区画内を同時に同一高度で飛行する場合は、飛行機間の最小距離とその時刻を出力する。

3.13 指定距離以下となる飛行機の組合せの検索

区画内を同時に同一高度で飛行する飛行機のすべての組合せについて、飛行機間の距離が、ユーザが指定した距離以下となる飛行機IDの組を出力する。

4 データ定義

4.15 区画

区画は、次の特徴をもった三次元領域で構成される。

- 垂直方向は、FL200以上FL600以下の範囲で区切られる。飛行機は垂直方向の境界上を飛行することができる。自然数で表す。
- 水平方向は、50海里以上150海里以下の矩形で表される。矩形上の点は(x, y)平面上の点で表され、x方向は北を、y方向は東を表す。小数第2位まで有効な固定小数点数で表す。

4.16 飛行計画

飛行計画は次の情報からなる。

- 飛行機ID。6桁の文字列。
- パイロットの名前。文字列。
- 離陸時の搭載燃料。自然数。
- 航路。次の情報を含む。
 - 飛行機ID
 - パイロットの名前
 - 離陸時の搭載燃料（飛行時間で表記、飛行機の種類と速度に基づいて算出される）
 - 航路。次の情報を含む。

飛行計画は最大で80件まで保持する。

5 非機能要求

- 時刻は、HHMM表記とする。HHは時刻の24時間表記、MMは分である。例えば、午後3時3分は1503と表す。

- FPMシステムは、1日分（時刻0000から2359まで）の飛行計画を管理する。
- FPMシステムは、応答時間に関する制約はない。
- 距離および針路に関する計算によって小数点数が得られた場合は、それぞれの数値に求められている小数点部桁数で表した値とする。
- 距離および針路に関する計算の正確性は、それぞれの数値に求められている小数点部桁数まで一致していれば正確であると判定する。
- FPMシステムは、TJFRについて考慮しない。
- 飛行計画が承認されていない飛行機が、区画内に進入することはない。

6 外部インタフェース仕様

6.17 ユーザインタフェース

FPMシステムのユーザインタフェースは、テキストベースのメニュー型とする。初期画面は次の通り。

```
*****
* Flight Planning and Management System v1.0
* Sat Oct 16 21:24:58 1999
*****
区画の座標を指定します。
区画の原点（南西）のX座標を入力して下さい：
```

区画に関する情報がユーザから入力されたのちに、次のメニュー画面が表示される。

```
*****
* MENU
*****
1. 飛行計画の入力
2. 飛行計画リストの入力
3. 飛行計画の削除
4. 飛行計画の表示
5. 飛行機の検索（時刻指定）
6. 飛行機の検索（時刻・高度指定）
7. 飛行機間距離の出力
8. 指定距離以下となる飛行機の組合せの検索
9. ヘルプ
99. 終了
->
```

各メニューの仕様は省略する。

6.18 飛行計画リストファイル

ファイルは次のフォーマットに従っている。区切り記号にはカンマを使用する。

6.19 エラーメッセージ

省略

行番号	内容
1	飛行機ID,パイロットの名前
2	離陸時の搭載燃料
3	出発点座標(x,y), 出発時刻, 高度, 速度, 針路, 終点座標(x,y), 終点時刻

飛行計画リストの例を次に示す.

行番号	内容
1	DAL111, J. Doolittle
2	60
3	10.00, 40.00, 14:10, 220, 300.0, 39.85, 60.00, 100.00, 14:25
4	AF237, H.Arnold
5	90
6	80.00, 20.00, 14:10, 220, 400.0, 36.32, 20.00, 100.00, 14:26
7	...