

平成21年度

筑波大学第三学群情報学類

卒業研究論文

題目

ソフトウェア開発プロジェクトの  
活動状況の視覚的な分析のためのツールの構築

主専攻 情報科学主専攻

著者 矢崎 聖也

指導教員 三末和男、志築文太郎、高橋伸、田中二郎

## 要 旨

ソフトウェア開発プロジェクトの過去から現在に渡る活動状況の把握は、プロジェクトの関係者が現状の把握や新事実の発見、そして意志決定を行うために有用なものである。プロジェクトの活動状況を把握、分析することを支援するための研究はすでに存在するが、それらは「予めどのような傾向を見いだしたいのかを知っていることが要求される」「時間と属性値およびイベントの3つ全てを一カ所で概観し、さらに対比することができていない」といった理由から、大量の活動状況データについて必ずしも満足行くものではなかった。

本研究では、ソフトウェア開発プロジェクトの活動情報における情報探索のために、視覚的表現手法と操作体系を構築した。視覚的表現手法と操作体系は、プロジェクト管理ツールの有するチケットと呼ばれるデータの概観と傾向の発見、対比に適したものにした。そして、それらを使ったツール「ColorWave」を開発した。ColorWaveは、数千件(かそれ以上)の大量のチケットデータの時間と属性値、イベントを一画面で視覚的に表現し、どの側面から一望するかをユーザの操作によってインタラクティブに切り替えることで対比できるツールである。活動情報を一望した上で、それを様々な側面に対比することを可能にする。

ケーススタディを行った結果、ColorWaveを用いることで中・大規模のソフトウェア開発プロジェクトの活動状況について様々な発見を得ることや、多角的に知見を深めることで発見を検証したり深化することが可能であると示された。

本研究の貢献は、活動の傾向についての事前知識無しでも時間と属性値およびイベントの3要素の全てを一カ所で概観できるようにするための視覚的表現を構築したことと、その視覚的表現を通して様々な面から対比するための操作体系を構築したことである。活動に関する情報を持つ大規模なプロジェクトは実世界に多く存在し、その分析に対する必要性も高いことから、本研究の成果はそれら活動の分析の支援に役立つことが期待される。

# 目次

<b>第1章 序論</b>	<b>1</b>
1.1 プロジェクト管理ツールとチケット	1
1.2 活動分析	1
1.2.1 活動分析の要件	2
大量の活動情報の俯瞰	2
多角的な対比	2
1.2.2 活動分析の現状	2
データの羅列を用いた活動分析	2
事前知識を必要とする活動分析	3
1.3 本研究の目的	3
1.4 本研究の貢献	3
<b>第2章 関連研究</b>	<b>5</b>
2.1 ソフトウェア開発プロジェクトの活動情報の可視化・分析	5
2.2 大量の時間変化するデータの可視化	5
<b>第3章 対象データ</b>	<b>7</b>
3.1 一般的なチケットデータ	7
3.1.1 プロジェクト管理ツールとチケット	7
3.1.2 チケットの例	7
3.1.3 属性	8
3.1.4 イベント	8
3.2 チケットの定式化	8
<b>第4章 現状分析</b>	<b>9</b>
4.1 活動分析の現状	9
4.1.1 チケットの検索を用いた分析の現状	9
4.1.2 チケットの集計を用いた分析の現状	11
4.2 着目点	13
4.2.1 事前知識を要求しないようにする	13
4.2.2 様々な面からの俯瞰を可能にする	13
4.2.3 容易に対比することを可能にする	13

<b>第 5 章</b>	<b>要件定義</b>	<b>14</b>
5.1	活動分析において得たい情報 . . . . .	14
5.2	支援のための要件 . . . . .	14
5.3	ユーザにとって何が出来るべきなのか . . . . .	15
<b>第 6 章</b>	<b>圧縮可能な俯瞰表現</b>	<b>16</b>
6.1	課題 . . . . .	16
6.2	個々のチケットの持つ情報の視覚的表現 . . . . .	16
6.3	チケット群の表現 . . . . .	17
6.4	大量の情報の圧縮 . . . . .	18
<b>第 7 章</b>	<b>対比するための操作体系</b>	<b>20</b>
7.1	視覚的表現の操作可能な部分 . . . . .	20
7.2	対比のための操作体系 . . . . .	20
<b>第 8 章</b>	<b>チケットデータ分析支援ツール「ColorWave」</b>	<b>22</b>
8.1	ColorWave の設計方針 . . . . .	22
8.2	ColorWave の概観 . . . . .	23
8.3	ColorWave の機能 . . . . .	23
8.3.1	チケットデータの視覚的表現 . . . . .	23
8.3.2	属性と色の対応の変更操作 . . . . .	23
8.3.3	色の濃さの変更操作 . . . . .	24
8.3.4	チケットの並び順の変更操作 . . . . .	24
	長さによる並び (Sort by length) . . . . .	24
	時間位置による並び (Sort by start, Sort by end) . . . . .	24
	色のクラスタリングによる並び (Sort by color) . . . . .	26
	属性値による並び (Sort by attribute) . . . . .	27
8.3.5	時間軸の変更操作 . . . . .	27
	グローバルな時間軸 . . . . .	28
	チケット内のローカルな時間軸 . . . . .	28
8.3.6	埋もれている値のハイライト操作 . . . . .	28
8.4	ツールの実装 . . . . .	29
8.4.1	システム構成 . . . . .	29
8.4.2	実装言語と入力データ形式 . . . . .	29
<b>第 9 章</b>	<b>ケーススタディ</b>	<b>31</b>
9.1	Mono project (7452 チケット) . . . . .	31
9.2	Redmine (2253 チケット) . . . . .	42
9.3	考察 . . . . .	48
9.4	まとめ . . . . .	49

第 10 章 結論	50
謝辭	51
参考文献	52

# 図目次

4.1	検索画面の例: <a href="https://bugzilla.novell.com/query.cgi?format=advanced">https://bugzilla.novell.com/query.cgi?format=advanced</a> のスクリーンショット	9
4.2	検索結果の例: <a href="https://bugzilla.novell.com/query.cgi?format=advanced">https://bugzilla.novell.com/query.cgi?format=advanced</a> から、Status:ASSIGNED であるチケットの検索結果のスクリーンショットの一部。さらに下へと円グラフが大量に並んでいる。	10
4.3	視覚的な集計画面の例: <a href="https://bugzilla.novell.com/query.cgi?format=report-graph">https://bugzilla.novell.com/query.cgi?format=report-graph</a> のスクリーンショット	11
4.4	視覚的な集計結果の例: ある時点における、Status 属性値を Classification ごとに集計した円グラフの並びの上方(ごく一部分)のスクリーンショット	12
6.1	チケット一つの描画例(横軸が時間に対応)	16
6.2	大量のチケットの表現(縦軸が時間に対応)。千ピクセル程度の幅で7452チケットを表現した。	17
6.3	視覚的表現の一部を拡大した図(縦軸が時間に対応)	18
6.4	色の合成の例	19
8.1	ColorWave の概観	22
8.2	色に対応させる属性をドロップダウンによって選択する UI: ドロップダウンを展開している図である	23
8.3	色の濃さの調整 UI	24
8.4	ソート種別の選択 UI	24
8.5	長さで並べた図	25
8.6	始端位置で並べた図	25
8.7	終端位置で並べた図	26
8.8	色でクラスタリングした図	27
8.9	ソートに用いる属性名と、それらの優先度を決定するための UI	27
8.10	ローカルな時間軸で表示した図	28
8.11	アニメーション過程(1)	29
8.12	アニメーション過程(2)	29
8.13	アニメーション過程(3)	29
8.14	システム構成	30

9.1	Mono: Classification 属性、長さ順	32
9.2	Mono: Classification 属性、始端順	33
9.3	Mono: Classification 属性、終端順	34
9.4	Mono: Product 属性、始端順	35
9.5	Mono: Product 属性、終端順	36
9.6	Mono: Product 属性、色によるクラスタリング	36
9.7	Mono: Priority 属性、長さ順	37
9.8	Mono: Priority 属性、始端順	37
9.9	Mono: Priority 属性、終端順	38
9.10	Mono: Priority 属性、色によるクラスタリング	38
9.11	Mono: Priority 属性、始端順 (埋もれている色を強調)	39
9.12	Mono: Priority 属性、Priority の属性値順	39
9.13	Mono: Priority 属性、Priority の属性値順 (埋もれている色を強調)	40
9.14	Mono: Priority 属性、色によるクラスタリング	40
9.15	Mono: Priority 属性、色によるクラスタリング (埋もれている色を強調)	41
9.16	Redmine: ステータス属性、長さ順	42
9.17	Redmine: ステータス属性、始端順	43
9.18	Redmine: ステータス属性、終端順	44
9.19	Redmine: ステータス属性、始端順 (埋もれている色を強調)	45
9.20	Redmine: ステータス属性、色によるクラスタリング	46
9.21	Redmine: 担当者属性、始端順	46
9.22	Redmine: ステータス属性、担当者、次に開始日時でソート	47
9.23	Redmine: ステータス属性、担当者、次に開始日時でソート (埋もれている色を強調)	47

# 第1章 序論

## 1.1 プロジェクト管理ツールとチケット

インターネットや大人数によるソフトウェア開発の発展により、ソフトウェア開発プロジェクトにおける活動が電子化され、電子化された情報を介して活動状況を共有されるようになった。現在、プロジェクト管理システムによって、多くのソフトウェア開発プロジェクトにおける活動情報が蓄積されている。プロジェクト管理システムが有する活動情報には、バグや要望の内容や進捗に関する情報があり、それらは「チケット」と呼ばれる形で蓄積されている。バグや要望が出た時に、プロジェクトのメンバーないしは第三者が、対応するチケットを作成し、さらに進捗があった時にチケットを改変する。そのとき、バグや要望の概要、詳細、進捗状況などの内容がチケットの属性値として記録され、さらに属性値が進捗に応じて改変される。メンバーも第三者も、既存のチケットを見ることで現状を把握でき、さらにチケットの改変履歴を見ることでこれまでの進展を把握することもできる。

プロジェクトを管理あるいは観察するためには、そのプロジェクトで行われている活動を知る必要がある。特に、いわゆる管理者などの、強い影響力と責務を有するメンバーにとっては、現状やこれまでの進展を把握することは意志決定のために重要である。更に、それらを把握できることは、プロジェクトの観察者にとっても、プロジェクトについての評価を下したり今後の関わり方を考える上での助けになる。よって、バグや要望についての活動はソフトウェアの開発プロジェクトについて知るための興味の対象となるものである。かつ、プロジェクトにおけるバグや要望の内容が属性として、進捗が属性に対する改変としてチケットに記録されているので、チケットの属性値や改変の傾向がバグや要望の内容や処理の進捗の傾向に対応している。そのため、チケットの属性値や改変の傾向を知ることによって、ソフトウェア開発プロジェクトについての知見を深めることができ、意志決定や価値判断などの助けになる。なお本論文では、活動情報を通してプロジェクトにおける活動についての知見を深めることを「活動分析」と呼ぶ。また、活動情報のうち特にチケットデータを主な対象とする。

## 1.2 活動分析

本研究では、プロジェクトのメンバーや観察者が、意志決定のための材料を得るために、対象プロジェクトにおいてどのような活動が行われてきたのかを調べるのが活動分析であると捉えている。活動分析によって分かったプロジェクトの過去や現在の活動の傾向を元に、プロジェクトのメンバーは今後の自らの行動やさらなる調査について意志決定できる。特にプ

プロジェクトの管理者や強い影響力を持つメンバーにとっては、現在の傾向や、過去の行動の結果を知ることが、意志決定の上で重要である。また、活動分析を行うことで、そのプロジェクトの成果物と今後の展望に対する裏付けや推測の材料を発見したり、それらを補強することができる。このように、活動分析によって、プロジェクトの実際の活動状況を踏まえた判断や調査が可能になる。

### 1.2.1 活動分析の要件

本研究では、活動分析は以下が特に必要とされていると考えている。

#### 大量の活動情報の俯瞰

活動分析の支援は特に大規模なプロジェクトにおいて必要とされている。その場合、活動分析は大量の活動情報の俯瞰を必要とすると考えられる。大量の活動情報を俯瞰することで、属性値やイベントおよび時間の傾向を知ることができ、それによって大規模なプロジェクトについての分析が可能になると考えられるためである。なお、本論文では、このように属性値やイベントおよび時間の傾向を知ることが、「概観する」と表現する。また、大規模でないならば、活動情報の生データを眺めるだけでも傾向を知ることが可能である。しかしながらも、大量の活動情報について傾向を現実的な時間と労力で見いだすためには、大量の情報を何らかの形で一括して認知する必要がある。

#### 多角的な対比

活動分析では、知見をより深化させたり、さらなる発見したりするため、見つけた傾向について様々な側面から多角的に比較検討できる必要がある。また、他の側面からの概観がないと気づけない傾向が存在しうることから、様々な側面の対比の必要性が言える。

### 1.2.2 活動分析の現状

活動分析を行うにあたって、現状では、データの羅列を見るか、事前知識を必要とされる分析支援のどちらかを用いることになる。

#### データの羅列を用いた活動分析

多くのプロジェクト管理ツールでは、検索条件を入力すると、該当するチケットを列挙する機能が提供されている。ここで、検索と検索結果の閲覧を通して活動分析することが考えられる。しかし、活動情報の量が多くなるにつれて必要な労力が大きくなる。例えば、検索結果が数万件にわたる場合、検索・表示処理に長い時間が掛かる上、数万行に渡るデータに目を通すために時間と労力を多く必要とされる。他にも、検索結果の件数に着目することも

考えられる。しかし、どの検索条件が件数にどのように関連するのかを知るために多数の試行錯誤が必要である。

### 事前知識を必要とする活動分析

いくつかのプロジェクト管理ツールでは、集計条件や描画方法を指定することで棒グラフや円グラフを生成する機能が提供されている。しかし、どのデータをどのように集計しどのようなグラフを描画すべきかを事前に知らない場合、多くの数のオプションや描画形式から適切な選択肢を見いだす必要がある。さらに、集計によるグラフでは、チケットデータの持つ属性値、時間情報、イベント情報のうち一部のみが表現されているため、限定的な概観となる。加えて、異なる集計方法でグラフを生成して対比する場合、別個の図同士を対比できる認識力が必要となる。このように、既存のプロジェクト管理ツールの機能を用いる場合、どのような傾向があるのかについての事前知識が必要とされるか、高い認識力が必要とされる。

また、活動情報を概観したり対比したりすることが可能な先行研究 [3] は、プロジェクトの健全性評価に着目した上で指標などの前提知識を持ち込んでいる。しかし、特定の側面や指標を前提とすることは、未知の傾向を発見したり様々な側面から知ることと相反する。

## 1.3 本研究の目的

本研究では、大規模なプロジェクトの活動状況の傾向の発見と多角的な対比を支援するための手法とツールを開発する。すなわち、大量のチケットのデータについて、いつどのような属性値や変更が存在したのかを概観でき、さらに多角的な対比を行えるツールを開発する。それによって、大量の活動がいつどのような傾向だったのか、を事前知識なしで探索することを可能にする。

## 1.4 本研究の貢献

属性、時間位置、イベント、を持つ大量の情報について、単一のビューにおける色と点の分布としてそれら全ての傾向を表現する手法を構築することで、チケットの持つ情報をまとめて俯瞰できるようにし、さらにその表現を用いた多角的な対比のための操作手法も構築した。これまでの情報可視化の研究では、属性や時間位置、変更のどれかを表現することは多く研究されてきたが、それらを全て兼ね備えることは考えられていなかった。また、これまでの活動分析についての研究では、特定の事前知識を必要としたりすることなく、汎用的に活動分析をすることも考慮されてこなかった。

活動の分析を可能とすることで、今まで行えなかった、活動の時間特性といった傾向の発見と多角的な分析が可能となった。例えば、プロジェクトの活動の活発さや、プロダクトの安定度、進捗度合いなどの変化を捉えて分析することが期待できる。棒グラフや円グラフといった集計を伴う表現手法に比べて、情報をできるだけ生のままで表現する可視化手法は今

まであまり利用されてこなかったが、将来的にはプロジェクトに関する意志決定のためなどの調査の際に、本研究で開発した手法が積極的に利用されるようになると考えられる。

## 第2章 関連研究

本研究は、ソフトウェア開発プロジェクトの活動情報の可視化・分析と、大量の時間変化するデータの可視化に関連している。この章では、それらの領域における先行研究について議論する。

### 2.1 ソフトウェア開発プロジェクトの活動情報の可視化・分析

ソフトウェアの開発プロジェクトの活動に関する情報を可視化し、そこから知見を得ることを支援するための研究は多く行われてきている。

例えば、Storey らの調査 [8] によれば、レポジトリから得られるソースコードのファイルや行、およびそれらに対する変更履歴に着目した研究が数多くある。しかし、本研究では、ソフトウェア開発プロジェクトのプロジェクト管理ツールについて着目している。

MDSViews[4] は、CVS(レポジトリ)と Bugzilla(プロジェクト管理ツール)から得られる情報に多次元尺度構成法と呼ばれる手法を適用することで、プロジェクト内の要素同士の関連性をグラフとして表現するものである。これに対して、本研究では、プロジェクト全体の活動を、個々の活動の情報の集合として視覚的に表現することに重きを置いている。

Social Health Overview (SHO) [3] は、点と色として表現された Bugzilla から得られた個々のチケットとその属性を見ることで、プロジェクトの活動状態の健全性を評価するためのツールである。SHO は、プロジェクト管理ツールから得られる個々のチケットの情報を視覚的に表現することで、全体の活動の様子を視覚的に表現する点と、表現に対して操作を行うことで多角的に分析する点で本研究と類似している。しかし、SHO はいくつかの時点での状態それぞれの視覚的表現を並べることで時間的な推移を表現しており、一つの視覚的表現の中でかつ連続的に時間的な推移を表現している本研究とは異なる。また、SHO は～3500 件程度のチケットを対象としており、～数万件やそれ以上のチケットを対象としている本研究とは、規模の面でも異なっている。

### 2.2 大量の時間変化するデータの可視化

時間変化する何らかのデータ(時系列データ)を大量に可視化することで、大量のデータに対する理解を助ける研究も多く行われてきており、Aigner らの調査 [1] においてそれらの先行研究が示されている。また、Extreme Visualization[7] には、単なる表でない方法でより視覚的に大量のデータを表現することについての先行研究が示されている。

複数の属性値を持つ大量のデータについての傾向を示す研究としては、上記以外にも [2] や [6] が挙げられる。ガン検査の結果についての大量のデータをクラスタリングして並べて表示している [2] は、色の合成を用いながら大量のデータを並べて傾向を示す点が本研究と類似しているが、データの時間位置や長さというものが存在しない。クラスタ上でやりとりされた大量の MPI<sup>1</sup> メッセージの傾向を可視化している [6] は、大量のデータの属性値のみならず時間位置や長さも表現し、データの並びを変えることで傾向を見いだす点も似ているが、イベントを表現していない点、データが重なって隠れてしまいうる点、切り替えなどによる対比を行っていない点、が異なる。

このように、時間変化するデータの可視化についての多くの研究と異なり、本研究はデータの時間位置、属性値、イベントの全てを同時に表現し、かつ異なる側面から対比することを可能にしている点が特徴的である。

---

<sup>1</sup>Message Passing Interface

## 第3章 対象データ

この章では、プロジェクト管理ツールにおけるチケットデータについて説明する。

### 3.1 一般的なチケットデータ

#### 3.1.1 プロジェクト管理ツールとチケット

プロジェクト管理ツールとは、ソフトウェア開発プロジェクトへのバグ報告や要望などをチケットとして受け付け、「バグを解決した」といったチケットに関する進捗の記録もするシステムである。チケットは、バグや要望一つについての、詳細と今までの進捗の情報を持っている。プロジェクトの参加者や観察者は、プロジェクト管理ツールの保持しているチケットを見ることで、個々のバグや要望などについての現状とこれまでの進捗を知ることが出来る。

なお、Bugzilla ではチケット相当のものが「バグレポート」と呼ばれるが、このバグレポートは、バグに限らず要望などの記録にも使われる。本論文ではチケットという表現に統一している。

#### 3.1.2 チケットの例

ソフトウェア開発プロジェクトにおけるチケットの実例を一つ示す<sup>1</sup>。このチケットは、表 3.1 に示す属性値を初期状態に持っており、表 3.2 に示すイベントを持っている。イベントそれぞれは、いくつかの属性値を変化させた。この例では、バグが担当者に割り当てられてから、解決と再開を繰り返しており、その過程でいくつかの属性値が変更されている。

表 3.1: 属性の初期値の例

属性名	属性値
Assignee	mono-bugs@lists.ximian.com
Status	NEW
Resolution	(なし)
Severity	Critical
Priority	P5 - None
Product	Mono: Runtime
Found in Version	2.0.x

<sup>1</sup>例は [https://bugzilla.novell.com/show\\_bug.cgi?id=489019](https://bugzilla.novell.com/show_bug.cgi?id=489019) の一部を引用

表 3.2: イベントと属性値の変更履歴の例 (部分)

イベント日時	属性名	新しい属性値
2009/03/26 06:14:04	Assignee	gonzalo@novell.com
	Severity	Normal
2009/03/27 20:08:07	Status	RESOLVED
	Resolution	FIXED
2009/03/30 08:09:39	Status	REOPENED
	Resolution	(なし)
08:10:20	Status	NEEDINFO
2009/03/30 17:08:53	Status	RESOLVED
	Resolution	FIXED
2009/04/01 10:05:53	Status	REOPENED
	Resolution	FIXED
10:06:43	Status	NEEDINFO
15:15:05	Status	REOPENED
2009/04/01 15:15:38	Priority	P2 - High
	Severity	Major

### 3.1.3 属性

プロジェクト管理ツールは、いくつかの種類属性と、その属性の取り得る属性値<sup>2</sup>を提供する。属性の種類は、個々のプロジェクトの事情に応じてカスタマイズされていることもあるが、一つのプロジェクトの中では一定である。

### 3.1.4 イベント

チケットの持つ属性値は、何らかの進捗がある度に改変されてゆく。本稿では、チケットに記録された進捗を「イベント」と呼ぶ<sup>3</sup>。イベントについて、チケットにはそのイベントの発生時刻と、イベントによって改変された属性の値が記録されている。

## 3.2 チケットの定式化

チケットは以下の構造を持つデータと見なすようにした。

$$\begin{aligned}
 \textit{Ticket} &= (\textit{State}, \textit{Modifications}) \\
 \textit{State} &= \{(\textit{Attribute}, \textit{Value}), \dots\} \\
 \textit{Modifications} &= \{\textit{Modification}, \dots\} \\
 \textit{Modification} &= (\textit{ModifiedAt}, \textit{StateDiff}) \\
 \textit{StateDiff} &= \{(\textit{Attribute}, \textit{Value}), \dots\}
 \end{aligned}$$

<sup>2</sup>属性によっては、任意の文字列を入力できるものもある

<sup>3</sup>Bugzilla などでは個々のイベントを指して Activity (活動) という単語が用いられているが、本論文では「活動」と紛らわしくなることを防ぐために、あえてその呼び名を用いていない。

# 第4章 現状分析

この章では、ソフトウェア開発プロジェクトの活動分析における現状について述べる。

## 4.1 活動分析の現状

既存のプロジェクト管理ツールを用いて活動分析しようとする場合、以下に示すように、チケットの検索機能を用いた分析か、チケットの集計機能を用いた分析を行うことになる。

### 4.1.1 チケットの検索を用いた分析の現状

一般的なプロジェクト管理ツールが提供する、チケットの検索機能を用いて活動分析を行う場合、以下の流れになる。

ユーザは、図4.1のような、検索フォームに何らかの条件を入力して、チケットを検索する。

The screenshot shows the 'Search for Bugs' advanced search interface. It features several sections for defining search criteria:

- Summary and Comments:** Includes a 'Summary' field with a search button and a 'Comment Matches' field.
- Product Information:** A table with columns for Classification, Product, Component, Found in Version, and Fixed in Milestone. Values include 'Missing Product Information', 'CASA', 'Driver-Sharepoint', and various version numbers.
- Bug Details:** Fields for 'The URL', 'Whiteboard', and 'Keywords' with search buttons.
- Impact:** A grid of dropdown menus for Status (e.g., UNCONFIRMED, NEW), Resolution (e.g., FIXED, INVALID), Severity (e.g., Blocker, Critical), Priority (e.g., P0 - Critical), Hardware (e.g., All, ASP), and OS (e.g., NetWare 5.1).
- Who and What:** Checkboxes for 'ANY of' (bug assignee, reporter, etc.) and 'AND ANY of' (bug assignee, reporter, etc.).
- Query Options:** 'Sort results by' (Importance) and 'Remember these as my default search options'.

図 4.1: 検索画面の例: <https://bugzilla.novell.com/query.cgi?format=advanced> のスクリーンショット

ID	Sev	Pri	OS	Assignee	Status	Res	Summary
545887	Cri	P1	Other	security-team	ASSI		Adobe Reader 9.5
552712	Cri	P1	Other	pmishelac	ASSI		Need a script to create right hid.lst for OOO release
554449	Cri	P1	openSUSE 11.1	mccarrion	ASSI		MoonAsABridge doesn't display the proper accessibles when more than one plugin is loaded
559524	Cri	P1	Other	mrvueckert	ASSI		Installing lighttpd patch crashes webstart
566296	Cri	P1	Other	shvabec	ASSI		privat-wireless-update forces gnome-keyring usage
568012	Cri	P1	Other	jkim	ASSI		Users in the APAC area having issues with edit in place options never updating files
528388	Haj	P1	openSUSE 11.0	tscho	ASSI		Intel ICH9R raid controller not supported on 10.3 since beta3, all dmraid arrays broken after boot (ibata problem)
569897	Haj	P1	Windows Vista	phurley	ASSI		Teaming add entry "Teaming" inside from word"
483735	Nor	P1	Other	rothbr	ASSI		Import schema from Netes fails
412308	Nor	P1	All	rothbr	ASSI		Duplicate records are getting set when importing records
416111	Nor	P1	openSUSE 10.3	mono-bugs	ASSI		[PWC1] [DataGrid] System.IndexOutOfRangeException: list position
486259	Nor	P1	Other	hag	ASSI		non-merged-helios issues
488928	Nor	P1	Other	jezootton	ASSI		Resized WIKI edit window is not sticky
505464	Nor	P1	Other	misoddard	ASSI		Client install breaks Apache on SLES11
505462	Nor	P1	Other	misoddard	ASSI		Need to provide synopsis for core files
513720	Nor	P1	OES - Linux	jezootton	ASSI		The field to enter the tag name stops following the cursor after 12 characters
514565	Nor	P1	Other	misoddard	ASSI		Desktop/Application share keeps dropping during its use . . .
522501	Nor	P1	SLES 10	turguhart	ASSI		Novel Version and Copyrights symbol are not displayed correctly into the toolbar.
523209	Nor	P1	Other	jezootton	ASSI		The foreignName field is not being set correctly when a user logs into Teaming
525463	Nor	P1	Other	phurley	ASSI		State Change not reported or Triggered when using entry value
538639	Nor	P1	Windows	dgilles	ASSI		Displaylist from "Invite new participant" stays opened
532274	Nor	P1	Other	dogies	ASSI		Product wont install & run on 64-bit Linux Operating Systems
534691	Nor	P1	Other	misoddard	ASSI		GWIM conferencing integration not working on Linux
545634	Nor	P1	Other	mkuenger	ASSI		Create method goes wrong in situation
546881	Nor	P1	Other	jezootton	ASSI		There is no UI for disabling individual user account
562011	Nor	P1	Windows 7	jkim	ASSI		Edit in-place feature on Windows 7
565482	Nor	P1	Other	turguhart	ASSI		Include FCS EUJA
566411	Nor	P1	Other	phurley	ASSI		Remove the "beta" watermark before the release
568997	Nor	P1	All	stapan	ASSI		Tasks in OVI Calendar not updated in Teaming
569863	Nor	P1	Other	lchrisat	ASSI		Need to update Splash Screen for Novell Teaming
569864	Nor	P1	Other	lchrisat	ASSI		Need to update Splash Screen for KabiLink Teaming
570012	Nor	P1	Other	ngiberson	ASSI		Create SUSE Studio Welcome Image for Dell
572024	Nor	P1	Other	jkim	ASSI		Upgrade 3rd party libraries used for Teaming's web services implementation
572146	Nor	P1	Other	jezootton	ASSI		user group getting deleted and then re-created when a user group object is moved
573115	Nor	P1	Other	nglibert	ASSI		Trash has been translated wrong
573117	Nor	P1	Other	nglibert	ASSI		Creating a new teaming system with german language uses the wrong name for workspace root
511603	Min	P1	Windows Vista	nglibert	ASSI		Teaming2.0_LIN : PL Translation of "Add Guest Book Signature" should be "Dodaj wpis w księdzce gości"
512939	Min	P1	SLES 11	lchrisat	ASSI		Italic style is used for Japanese.
512945	Min	P1	SLES 11	lchrisat	ASSI		Italic style is used for Japanese.
512979	Min	P1	SLES 11	lchrisat	ASSI		Italic style is used for Japanese.
512980	Min	P1	SLES 11	lchrisat	ASSI		Italic style is used for Japanese.
512989	Min	P1	SLES 11	lchrisat	ASSI		Italic style is used for Japanese.
512991	Min	P1	SLES 11	lchrisat	ASSI		Italic style is used for Japanese.
512994	Min	P1	SLES 11	lchrisat	ASSI		Italic style is used for Japanese.
513812	Min	P1	SLES 11	lchrisat	ASSI		Italic style is used for Japanese.
520282	Min	P1	Windows Server 2008	lchrisat	ASSI		Tooltip incorrectly indicates Mini business card view is available in Profile tab
520423	Min	P1	Windows XP	jezootton	ASSI		Misplaced help button.
536135	Min	P1	Other	lchrisat	ASSI		A different font is shown
546437	Min	P1	Other	lchrisat	ASSI		Regex "Go to An Entry..." widgets on various folder navigation bars.
550263	Min	P1	Windows XP	jezootton	ASSI		ISBN-Null in error message is not translated if password is null
554044	Min	P1	Windows Server 2008	jezootton	ASSI		"incoming and" should be deleted from "enable or disable incoming and outgoing e-mail"
556612	Min	P1	SLES 11	phurley	ASSI		Error is displayed: SQL Error: Data truncation: Data too long, using more than 255 char as Blog Entry name
559669	Min	P1	All	jezootton	ASSI		Page Scrolling stops when adding Item to Landing Page drag/drop
568850	Min	P1	OES - Linux	phurley	ASSI		User name is maintained from Add a User box
561809	Min	P1	Windows Server 2008	diffraiser	ASSI		ISBN - Windows Vista only - Local chars are displayed wrong in error message
561561	Min	P1	Other	jezootton	ASSI		Dialog for checking folder ownership covers the list of names from which to select
569889	Min	P1	Windows XP	jezootton	ASSI		It display garbage chars if Name and Template title contain "<>" when modifying folder template
571987	Min	P1	Other	phurley	ASSI		Mobility / Recent Places / Lacks context
505741	Evh	P1	All	arnesi	ASSI		We need clear and complete UI guidelines
118072	Evh	P1	All	mjasmine	ASSI		Implement undo/redo for new schema tree control
118348	Evh	P1	All	mjasmine	ASSI		Need direct print support for policies
118426	Evh	P1	All	arnesi	ASSI		Provide a mechanism to close related issues

図 4.2: 検索結果の例: <https://bugzilla.novell.com/query.cgi?format=advanced> から、Status:ASSIGNED であるチケットの検索結果のスクリーンショットの一部。さらに下へと円グラフが大量に並んでいる。

検索結果(図4.2)の件数を見ることで、どれぐらいのチケットがその条件に該当したのかを知ることが出来、さらに条件を変えて再検索することで件数を対比することが可能である。しかし、チケットの検索には多数の項目の入力をしなければならない。また、検索を利用した対比を行う場合、闇雲な試行錯誤が要求される。なぜなら、どの検索条件が件数にどのように影響したのかは明示されず、少ない情報からを予測することになるためである。また、対比するためには、検索条件入力画面と検索結果画面を往復する必要がある。加えて、適切な検索条件ないしはそれに近い条件を知らなかった場合、多数の検索条件をどう指定すべきか知るためには、多数の試行錯誤が必要である。さらに、検索結果にあるそれぞれのデータが、時間的にどのような分布をしているのかは表示されない。よって、時間について絞り込むことで分布を明らかにするためには、何回もの試行をして結果を比較することになる。

なお、検索結果(図4.2)の内容を見ることで、検索結果にあるチケットの傾向を知ることとも考えられる。ただし、大規模なプロジェクトではチケット数が数千や数万、さらには数十万に達する。そのため、長い時間が必要とされる<sup>1</sup>。

#### 4.1.2 チケットの集計を用いた分析の現状

いくつかのプロジェクト管理ツールが備える、チケットの集計機能を用いて活動分析を行う場合、以下の流れになる。

ユーザは、図4.3のようなフォームにて、集計手法や描画方法を指定する。

図4.3: 視覚的な集計画面の例: <https://bugzilla.novell.com/query.cgi?format=report-graph> のスクリーンショット

そして、描画された折れ線グラフや円グラフ(図4.4)などを見ることで、条件を満たすチケット数の推移や比率を見ることが出来る。しかし、グラフを描画させるために、多数の項目を入力しなければならない。また、これらのグラフは2軸であるので、チケット(数)と属

<sup>1</sup>また、システムによっては、限られた件数のチケットのみが表示されるように運用上の制約が課されていることもある

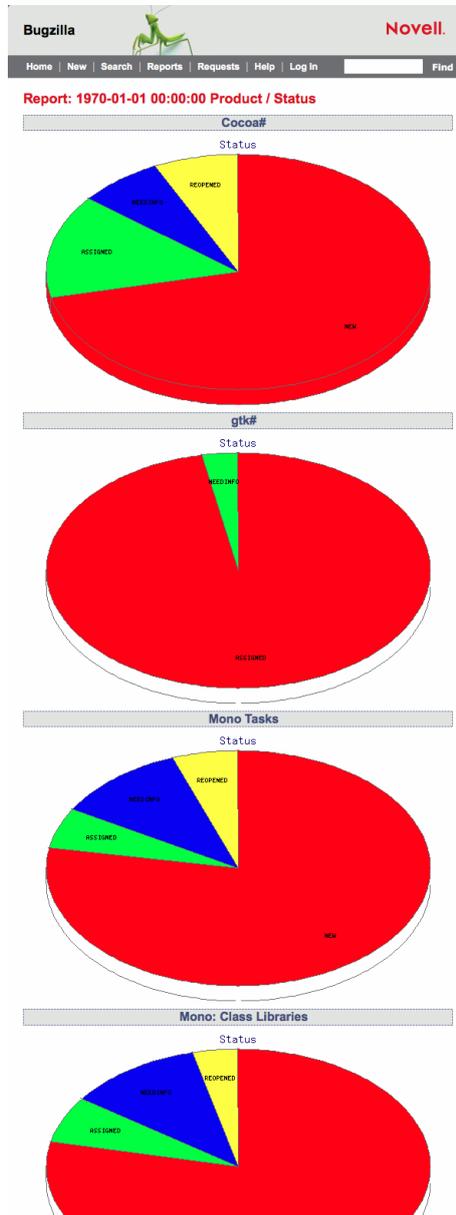


図 4.4: 視覚的な集計結果の例: ある時点における、Status 属性値を Classification ごとに集計した円グラフの並びの上方(ごく一部分)のスクリーンショット

性値と時間の3つを一度に表現するといったことはできず、活動情報のごく限られた側面だけが表現されている。また、あくまで数の集計であるので、イベントといった概念は表現されない。さらに、対比するためには、入力画面と結果画面を往復しながら、個別の図の相関性を見いだす必要がある。

## 4.2 着目点

本研究では、活動分析のために以下の改善が可能であることに着目した。

### 4.2.1 事前知識を要求しないようにする

多くの属性を持ち、さらに時間情報やイベントも持つチケットを、検索や集計しようとする場合、どう検索や集計すべきかを事前に知っていることが求められる。または、適切な検索条件や集計方法を見いだすまでに多数の試行錯誤を行うことになる。また、一部のツールや既存研究のように特定の指標や統計手段を前提とすることは、特定の事前知識を前提とすることと同義である。そのため、どのような傾向があるのかを模索するためには適さない。そこで、活動分析に取りかかりやすくするために、わずかな事前知識でも分析を可能にすることを考える。

### 4.2.2 様々な面からの俯瞰を可能にする

多くの属性を持ち、さらに時間情報やイベントも持つチケットを、表やあるいは折れ線グラフや円グラフなどの手法で表現することで、チケットの持つ情報のうち一部についての俯瞰が得られる。そのため、一部の情報を元に、活動状況を概観したり、概観するために試行錯誤することとなる。また、チケットの持つ時間情報や属性およびイベントを、大量の図表や文字列の羅列で表現する方法は、数千件やそれ以上といった大規模なデータに対しては、俯瞰とならない。そこで、活動分析の過程での自由度を確保するために、属性や時間情報、イベントの情報をまとめて俯瞰できるようにすることを考える。

### 4.2.3 容易に対比することを可能にする

クエリやパラメーターを与え、チケットデータの持つ情報の一部を表示するモデルでは、さまざまなクエリやパラメータを与え、出力された多数の別々の表現から相関性を見いだすことで対比することになる。傾向を見いだすためにこのような試行錯誤をするためには、クエリやパラメーターの再入力と結果の表示を何回も反復しつつも、同時に多数の結果それぞれの間にある相関性を見いだす能力が必要とされる。そこで、多角的な分析や対比のために、対比を容易に行えるようにすることを考える。

## 第5章 要件定義

この章では、開発する手法やツールがどのような要件を満たすべきであることを述べる。

### 5.1 活動分析において得たい情報

チケットそれぞれについて見えるべき情報は以下であると考ええる。

1. チケットの開始や終了の時間位置(日時)
2. あるチケットについて、任意の時点における属性値
3. あるチケットについて、イベントの存在とイベントのあった日時

また、上記について、時間、それにチケット間での傾向が見えるべきであると考ええる。

### 5.2 支援のための要件

活動状況の把握や発見を支援するためには、以下の要件があると考えている。

#### 大量のチケット群を俯瞰できること

本研究は、大規模なプロジェクトの活動状況を概観することを目的としている。従って、何らかの形で、大規模なプロジェクトに存在する大量のチケット全体について、上記の情報をまとめて把握できる必要がある。

#### 時間やイベントが見えること

本研究は、チケットのデータを通じて、プロジェクトの活動を知ることが目的としている。活動は時間的な経過を伴うものであるため、特に時間的な位置(日時)は常に表現されているべきである。

### **多角的な概観が可能であること**

活動が単一の側面(属性)のみを持つことは希であるので、複数の側面を対比するといった多角的な分析が可能であるべきである。

### **何を知りたいのかを事前に知っている必要がないこと**

本研究は、ユーザが明確に理解していないか、そもそも存在に気づいていない特徴について、把握や発見を助けることを目的としている。従って、どのような側面を概観したいのかについて、事前に具体的な知識を要求する造りになるべきではない。

### **試行錯誤が容易であること**

ユーザが事前に理解していない事柄について調べていくことを想定しているため、試行錯誤によって発見することを支援し、かつ発見について多角的に対比していくことが可能とすべきである。

## **5.3 ユーザにとって何が出来るべきなのか**

上記の要件を満たすために、ユーザがツールを使うことで以下の結果を得られるようにツールを設計すべきと考えた。

1. 前知識なしで、表示して一望できること。
2. 一見して、時間と属性値およびイベントの傾向を読み取れること。
3. 多角的な対比をその場でできること。

## 第6章 圧縮可能な俯瞰表現

本章では、前章で述べた要件を基に、本研究で開発した視覚的表現手法について述べる。この手法は、チケットそれぞれの視覚的表現をチケットの数だけ並べることで、チケット群の傾向を視覚的に表現するものである。

### 6.1 課題

要件を踏まえてチケットデータを表現するためには、以下の3点を同時に満たすという課題が特に考えられる。

1. チケットそれぞれを時間と共に表現すること。
2. 一般的な画面の横や縦の画素数よりも多いデータ数(数千や数万)全てを一画面にて表現すること。
3. 属性値やイベント、およびそれらの時間位置の全てについての傾向を一見して読み取れるように表現すること。

この課題に対処すべく、以下に述べる視覚的表現を構築した。

### 6.2 個々のチケットの持つ情報の視覚的表現

まず、個々のチケットの属性値、時間位置、イベントをごく小さい幅の中で、図6.1のようにして表現する。

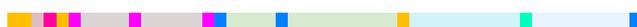


図 6.1: チケット一つの描画例 (横軸が時間に対応)

指定された属性の値が色、イベントが点に対応する。また、チケットの生存期間<sup>1</sup>やイベントの時間位置が、線分や点の座標に対応する。

このようにすることで、限られた幅の中でチケットの持つ属性やイベントおよび時間を表現する。限られた幅の中で表現することは、大量のチケットを並べられるようにするためである。

<sup>1</sup>チケットの始端から終端までの区間

### 6.3 チケット群の表現

大量のチケットの表現を、上記の表現手法を用いたチケットを並べることで表現する。例えば、ケーススタディ(9.1 節)で用いているチケットの実データ 7452 件を表示すると、図 6.2 のようになる。なお、この図の一部を拡大すると、図 6.3 のようになっている。

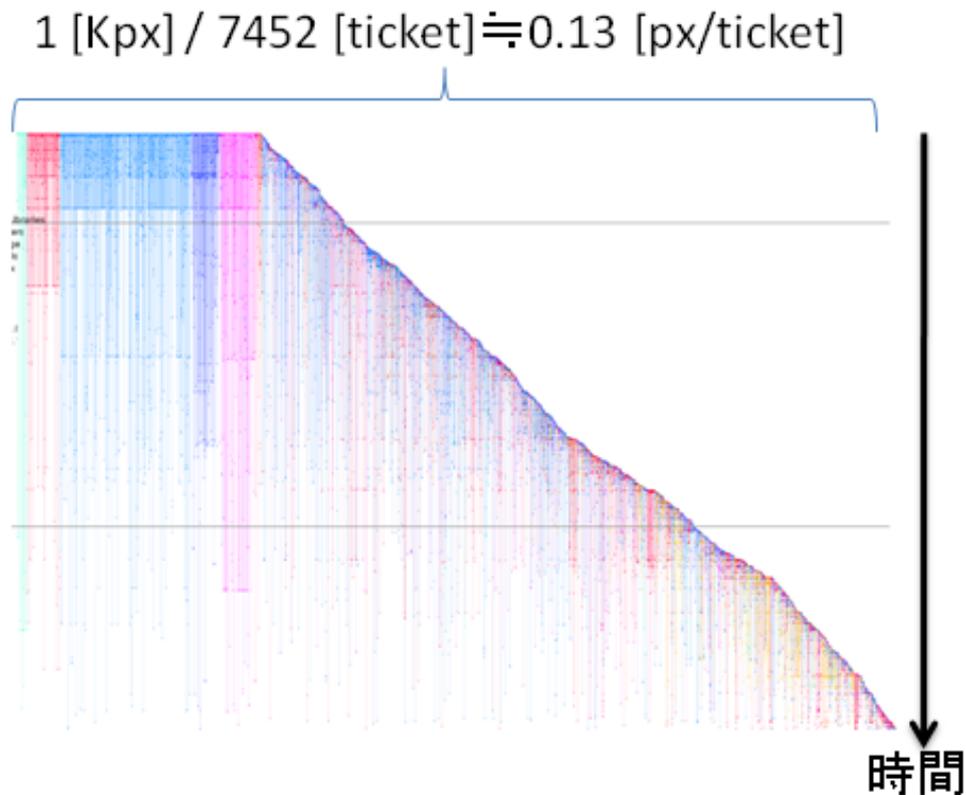


図 6.2: 大量のチケットの表現 (縦軸が時間に対応)。千ピクセル程度の幅で 7452 チケットを表現した。

ユーザーには、この表現によって、色の傾向や点の集まりが見える。色や点は属性値やイベントに対応しているので、ユーザーは属性値やイベントの傾向が読み取れる。また、ユーザーは位置を見ることで、それらの時間位置も理解できる。このように、時間情報を損なわないようにしつつも一つのチケットの持つ情報を狭い幅で表現し、それらの集合としてチケット群を表現することで、ユーザーに傾向を見せている。

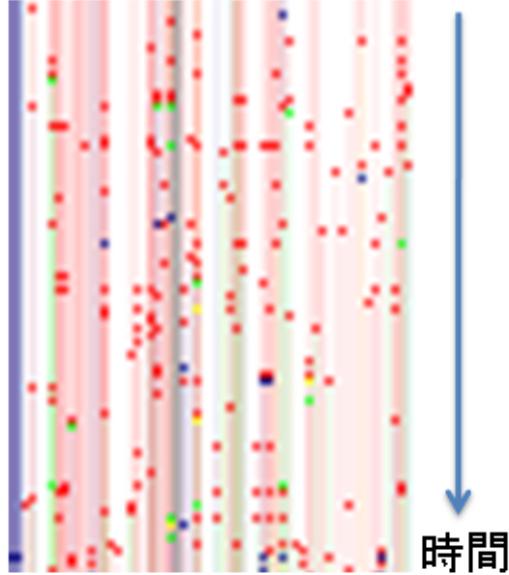


図 6.3: 視覚的表現の一部を拡大した図 (縦軸が時間に対応)

## 6.4 大量の情報の圧縮

図 6.2 で示されているように、多くのチケットを一望出来るようにする必要がある。また、チケットの数が画面のピクセル数を上回る場合、チケット 1 つの幅が 1 ピクセル未満になる。そこで、チケットが画面上で占めるべき領域が、ピクセルの境界と揃わない場合、描画時に色を混合するようにする。具体的には、ピクセル  $(x, y)$  の色  $PixelColor(x, y) = (Red, Green, Blue)^T$  は以下のように計算される。

$Left(T)$  = チケット  $T$  の左端の座標値

$Right(T)$  = チケット  $T$  の右端の座標値

$Top(T)$  = チケット  $T$  の上端

$Bottom(T)$  = チケット  $T$  の下端

$$CoverageX(T, x) = \begin{cases} \min(x + 1, Right(T)) - \max(x, Left(T)) & (Left(T) \leq x \cap x + 1 \leq Right(T)) \\ 0 & Otherwise \end{cases}$$

$$Coverage(T, x, y) = \begin{cases} CoverageX(T, x) & (Bottom(T) \leq y \cap y \leq Top(T)) \\ 0 & Otherwise \end{cases}$$

$Color(T, y)$  = (チケット  $T$  の  $y$  における色の Red 成分, Green 成分, Blue 成分) $^T$

$$PixelColor(x, y) = \sum_{t=Ticket} Coverage(T, x, y)Color(T, y)$$

例えば2つのチケットが、あるピクセルの列に0.5 px ずつ重なるなら、図6.4のように色が合成され、合成結果の色がそのピクセルの描画時の色として用いられる。これによって、1

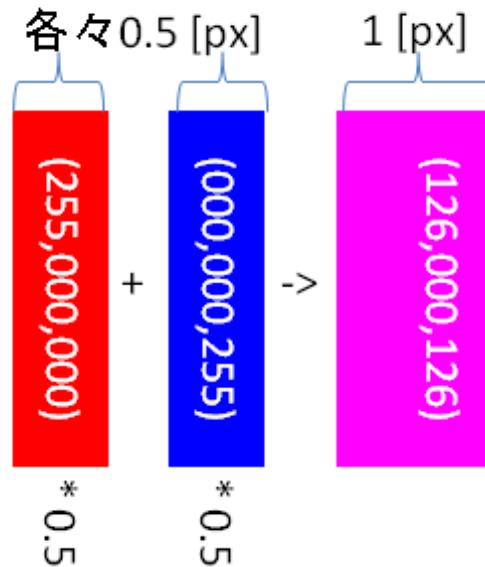


図 6.4: 色の合成の例

ピクセル未満になる部分の色も描画に反映される。また、仮にチケット1つの幅が数ピクセルであったとしても人間の目でそれを判別することは困難を伴うが、数ピクセル程度の太さのチケットについても、表示機器や人間の目の特性によって自ずから色が混合されることを利用する。

そうすることで、表示機器の解像度やユーザの目の分解能の限界まで細部の色や点を保持しつつも、傾向を表現することが出来る。これは、情報の表現に色と色の合成演算を用いることによって可能になっていることである。

## 第7章 対比するための操作体系

本章では、先に述べた要件と視覚的表現を踏まえ、多角的に対比するためにどのような操作体系にするのかを述べる。この操作体系は、視覚的表現を通して、インタラクティブに多角的な分析や対比をすることを可能にするためのものである。

### 7.1 視覚的表現の操作可能な部分

この節では、視覚的表現についてどの部分がどのように制御可能であるかを述べる。

#### 属性と色の対応

先述の視覚的表現を構築するために、どの属性の値を色として表現するのかを与える。どの属性を色として表現するのかを選択することで、どの側面について一望するのかを制御できる。

#### チケットの並び順

先述の視覚的表現を構築するためには、チケットの並び順を与える必要がある。チケットの順序を決める規則やパラメータを制御することで、チケット同士のどのような相関性について一望するのかを制御できる。

#### 属性値とイベントの色の濃さ

属性値とイベントをどれほどの濃度で描画するのも制御できる。これらを調節することで、属性値とイベントのどちらかを強調したり、あるいは双方をバランス良く見ることができる。また、色が重要な役割を果たす表現を使うにあたって、ユーザの表示機器や色彩感覚に合わせた調整できるということでもある。

### 7.2 対比のための操作体系

多角的な分析や対比を可能とするために、以下のように上記の操作を行えるようにすることを考えた。

## **見ながら操作**

視覚的表現の表示と操作の両方を並列して行えるようにする。これによって、操作の結果を即座に認識したり、視覚的表現を見ることで思いついた操作を即座に実践することが可能になる。

## **インタラクティブに操作**

その場でインタラクティブに制御するようにする。これによって、図を少しずつ変えながら見る事が出来るので、特別な労力や認識力を使わずに、対比することができる。

## 第8章 チケットデータ分析支援ツール 「ColorWave」

先述の表現・操作体系を実装したチケットデータ分析支援ツール「ColorWave」を開発した。図 8.1 にツールの全体図を示す。本章では、ツールの使い方を機能説明と共に述べ、また実装上のポイントも述べる。

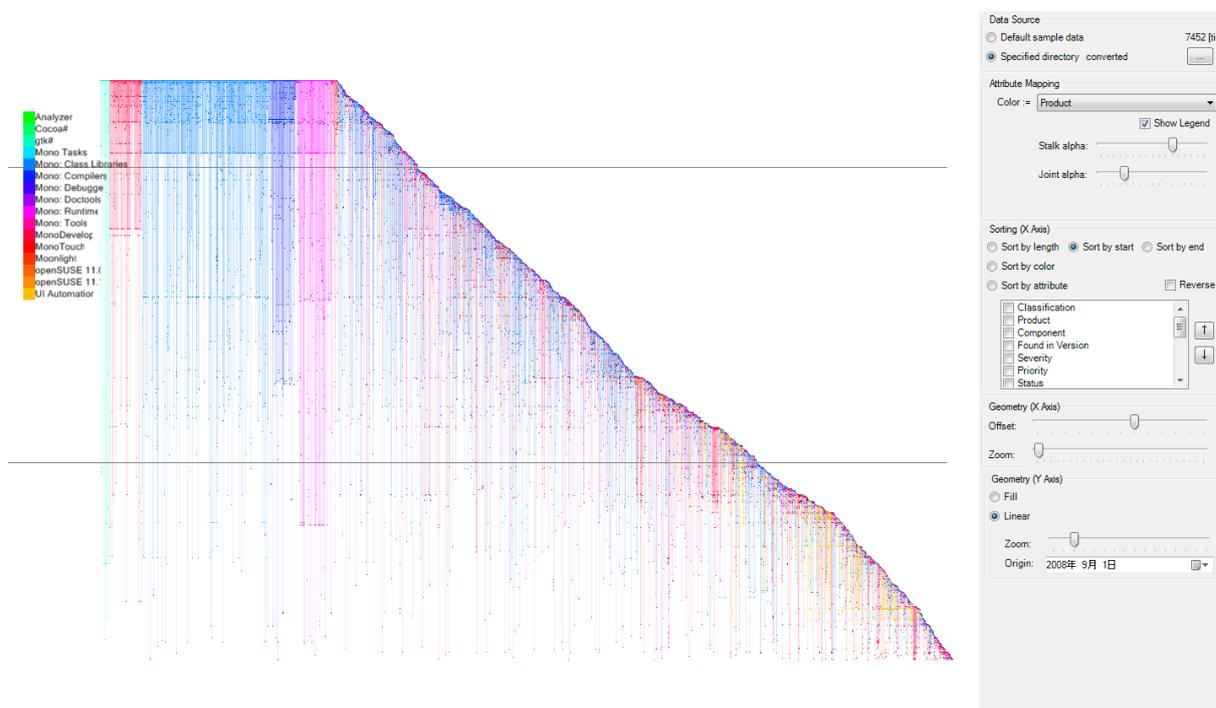


図 8.1: ColorWave の概観

### 8.1 ColorWave の設計方針

ColorWave では、読み込んだチケットデータを先述の視覚的表現を用いて提示する。また、先述の操作体系における操作それぞれを行うための GUI も提供する。属性値を色とマッピングする操作のように、チケットが持つ属性や属性値のバリエーションに対応した GUI が使わ

れる場合、読み込んだチケットデータに応じて、操作のための GUI を提供するようにする。

## 8.2 ColorWave の概観

ツールは画面右側の操作ペインと、それ以外を占める表示ペインからなる (図 8.1)。表示ペインでは、チケットのデータが視覚的に表現されており、また色がどの属性値に対応するのにかについての凡例も表示されている。操作ペインには、視覚的表現に対する操作を行うために必要な GUI が配置されている。

## 8.3 ColorWave の機能

### 8.3.1 チケットデータの視覚的表現

読み込まれたチケットは、先述の視覚的表現によって表示ペインに表示される。これによって、読み込まれたチケットを俯瞰することができ、チケットの時間位置や属性値とイベントの傾向の概観を読み取ることもできる。また、表示ペインには色の凡例も表示される。

### 8.3.2 属性と色の対応の変更操作

操作ペインにあるドロップダウンリストから属性名を選択することで、どの属性を色として表現するのかを選択することができる (図 8.2)。

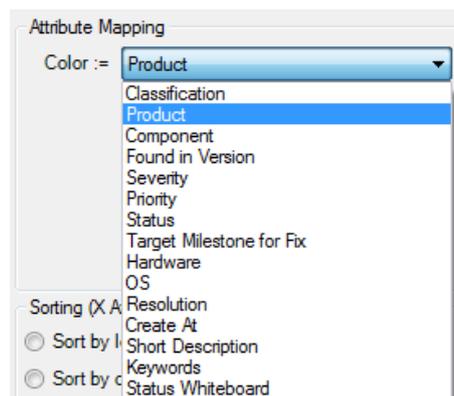


図 8.2: 色に対応させる属性をドロップダウンによって選択する UI: ドロップダウンを展開している図である

これによって、表示ペインにおける色の傾向がどの属性の傾向なのかを切り替えることが出来る。これを用いることで、複数の属性の値の傾向の対比が出来る。なお、属性と色の対応を変更しても、チケットの位置は変化しない。

### 8.3.3 色の濃さの変更操作

操作ペインにあるスライダを用いて、属性値に対応する色の透明度やイベントに対応する点の透明度を変更することができる(図 8.3)。

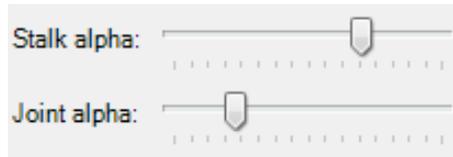


図 8.3: 色の濃さの調整 UI

これによって、属性値に対応する色やイベントに対応する点の視認性を調節することが出来、属性値とイベントのどちらかを強調したり、あるいは両方をバランス良く概観することが出来る。

### 8.3.4 チケットの並び順の変更操作

操作ペインにおいて、ラジオボタンで示された 5 種類のチケットの並び順(図 8.4)からいずれかを選ぶことが出来る。これによって、チケットの様々な側面について概観することができる。なお、チケットの並び順を変更しても、それぞれのチケットの色と、時間位置は変化しない。

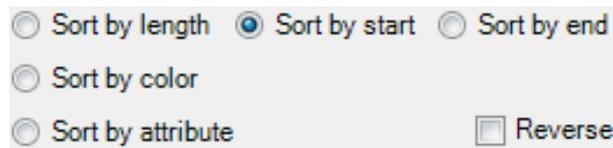


図 8.4: ソート種別の選択 UI

#### 長さによる並び (Sort by length)

始端と終端の間の時間長によってチケットを並び替えることができる(図 8.5)。

これによって、より長引いた、あるいは早く終わったチケットはどのようなものなのか、という側面についての概観が可能になる。

#### 時間位置による並び (Sort by start, Sort by end)

始端ないしは終端の時間位置によってチケットを並び替えることが出来る(図 8.6, 8.7)。

これによって、より昔の、あるいはより最近のチケットはどのような傾向があるのか、という側面について概観できる。

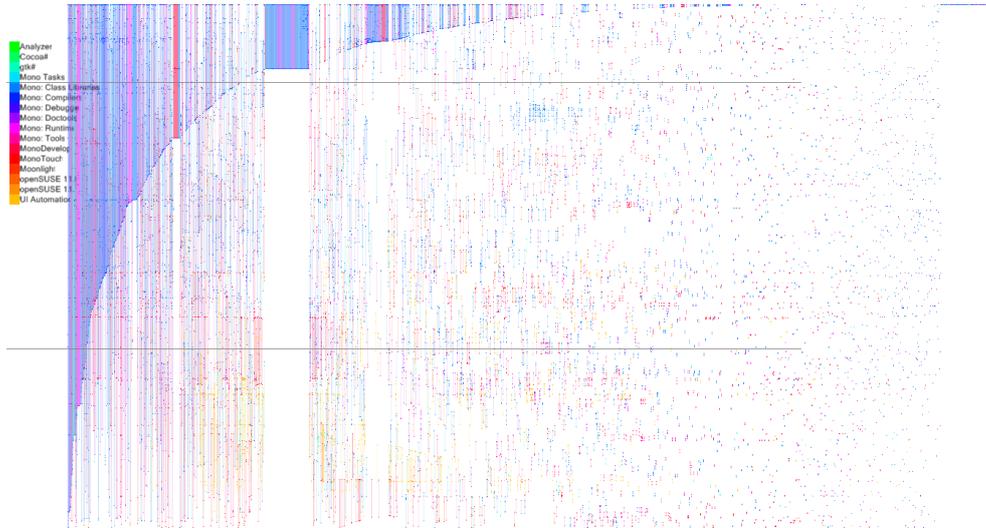


図 8.5: 長さで並べた図

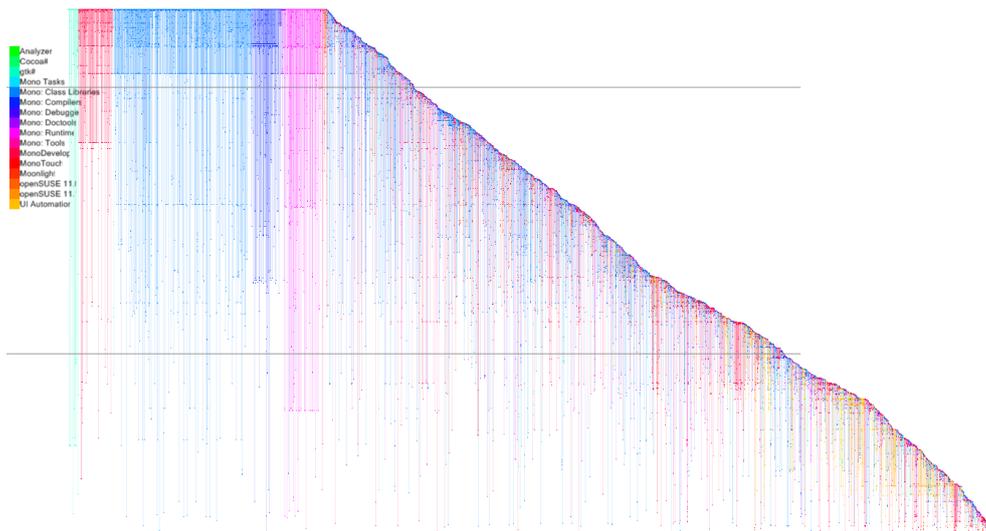


図 8.6: 始端位置で並べた図

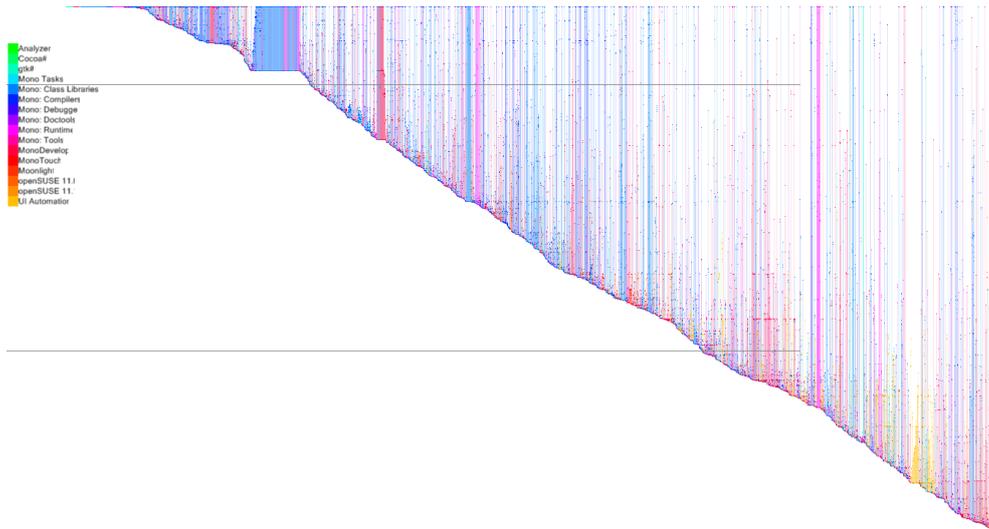


図 8.7: 終端位置で並べた図

### 色のクラスタリングによる並び (Sort by color)

画面上で隣接するピクセルの色がなるべく近くなるようにチケットを並び替えることが出来る (図 8.8)。

このために、以下のようにチケット 2 つ ( $T_1$  と  $T_2$ ) の間の距離  $D(T_1, T_2)$  を定義している。

$Y_{max}$  = 縦軸の最大値

$Y_{min}$  = 縦軸の最小値

$Y_{min}(T)$  = チケット  $T$  の始端

$Y_{max}(T)$  = チケット  $T$  の終端

$$H(T, y) = \begin{cases} \text{チケット } T \text{ の縦軸の位置 } y \text{ における色相} & (Y_{min}(T) \leq y \leq Y_{max}(T)) \\ \text{表示ペインの背景色の色相} & (y \leq Y_{min}(T) \cap Y_{max}(T) \leq y) \end{cases}$$

$$D(T_1, T_2) = \sum_{y=Y_{min}}^{Y_{max}} |H(T_1, y) - H(T_2, y)|$$

その上で、チケットに最も近いチケットを右隣に並べる操作を繰り返すことで、色によるクラスタリングを行えるようにした。これによって、似た外見のチケットを近づけたときの全体を概観できる。

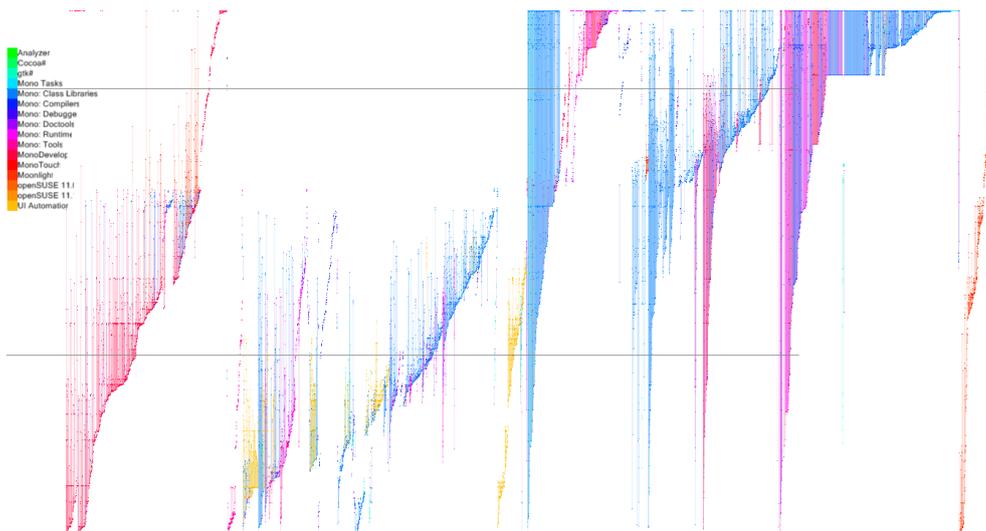


図 8.8: 色でクラスタリングした図

### 属性値による並び (Sort by attribute)

1つ以上の属性名をリスト(図 8.9)にてチェックすることで、チェックした属性の値によってチケットを並び替えることが出来る。

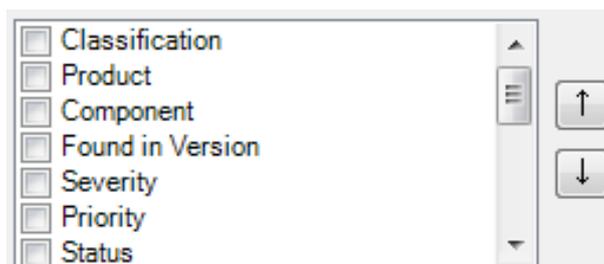


図 8.9: ソートに用いる属性名と、それらの優先度を決定するための UI

複数の属性名をチェックした場合、チェックされている属性名のうち最も下の属性から最も上の属性へ、という順序で属性値による安定ソートを行う。これによって、ある属性についての属性値が等しい場合にのみより下の属性の値で比較されるようになる。また、リストの横の上下ボタンによって、属性名の順序を変更することができる。

### 8.3.5 時間軸の変更操作

操作ペインにおいて、時間軸を以下の2種類から選べるようにした。これによって、複数の時間軸についての概観を可能にした。

## グローバルな時間軸

絶対的な時間(日時)が画面上の位置と比例するように描画できる。これによって、日時について概観し傾向を知ることができる。

## チケット内のローカルな時間軸

個々のチケットの始端が画面の上端、終端が画面の下端に対応するように描画することができる(図 8.10)。

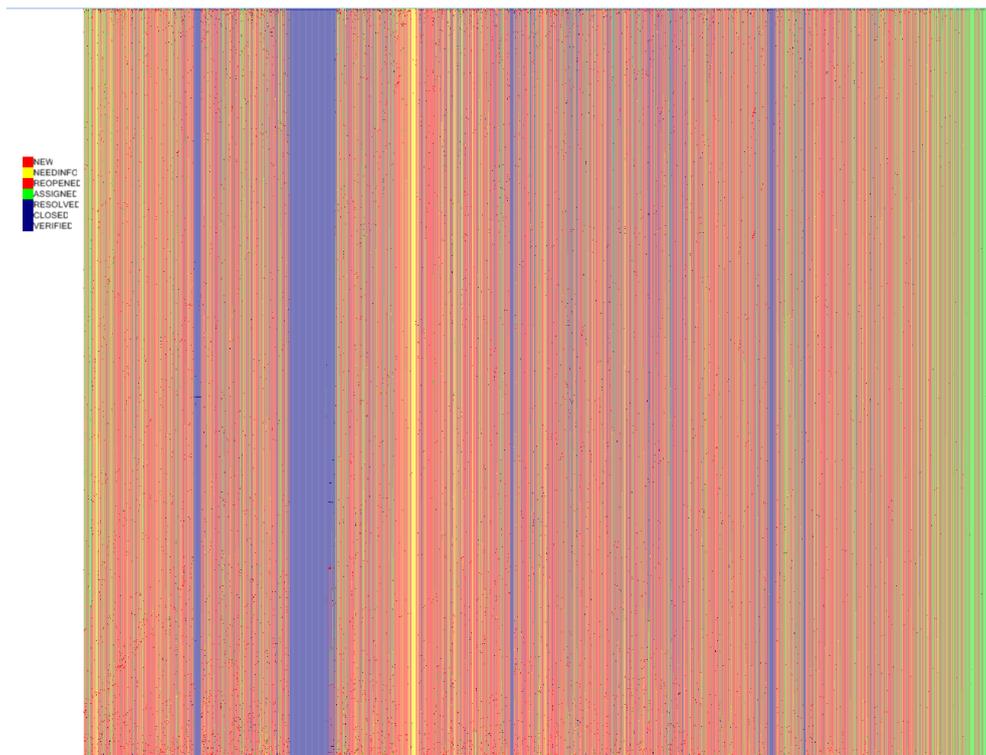


図 8.10: ローカルな時間軸で表示した図

これによって、絶対的な時間位置の代わりに、チケットの始まりの方や終わりの方に近づくほどどのような傾向があるのか、という点で概観できる。

### 8.3.6 埋もれている値のハイライト操作

先述の視覚的表現の特性上、1ピクセルに多くのチケットの色が合成されて描画されうる。そのため、隣接するチケット間で値が大きく異なっても、その分散の大きさが読み取れないという問題がある。

そこで、埋もれている値の存在をアニメーションによって強調することを可能にした(図 8.11, 8.12, 8.13)。あるピクセルについて、ユーザによって指定された閾値以上、そのピクセルの色から離れた色相となる色が、ピクセルの色の合成元に含まれていた場合、これを埋もれている色であると見なす。この機能では、埋もれている色を、左右に広げて描画する。また、広げる幅は、本来の幅の 1 倍から、任意の幅の間でスムーズにアニメーションする。

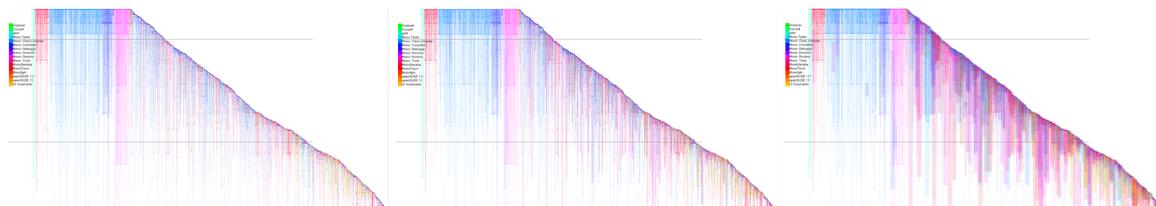


図 8.11: アニメーション過程(1) 図 8.12: アニメーション過程(2) 図 8.13: アニメーション過程(3)

これによって、色が埋もれている場所とそうでない場所を見分けることや、埋もれている色を見て取ることができる。また、アニメーションすることによって、本機能で埋もれている色を強調しなかった場合の概観と、強調した場合の概観を連続的に対比できる。

## 8.4 ツールの実装

### 8.4.1 システム構成

図 8.14 は、ColorWave とその周辺のシステム構成である。赤が一般に存在するもの、緑が本研究で構築した部分、青が実験用のデータである。プログラムによる作用は灰色かつ中空の矢印として表されており、ユーザとの相互作用は黒い実線の矢印で表されている。システムは、チケットデータ収集部と、ColorWave 本体に大別される。チケットデータ収集部では、ダウンローダーが bugzilla や redmine などのプロジェクト管理ツールからチケットの生データを HTML や XML 形式で収集する。そして、post process することで、それらを解析、統合し、特定のプロジェクト管理ツールに依存せずかつ扱いやすい独自形式のチケットデータへと整理する。ColorWave は、チケットデータ収集部によって得られたデータを読み込み、ユーザに提示し、ユーザによる操作を受け取り、インタラクティブに動作する。ユーザへの提示は提案した視覚的表現を用いる。また、提案した操作が行われた時、視覚的表現の設定を変化させるようにすることで、インタラクティブな動作を実現する。

### 8.4.2 実装言語と入力データ形式

チケットデータ収集部は、Linux や Mac OS X 上で動作する、ダウンローダー(シェルスクリプト)とポストプロセッサ(C# で記述されたコマンドラインアプリケーション)で実装されている。ColorWave 本体は、C# によって記述され、.NET Framework 3.5 と XNA Framework

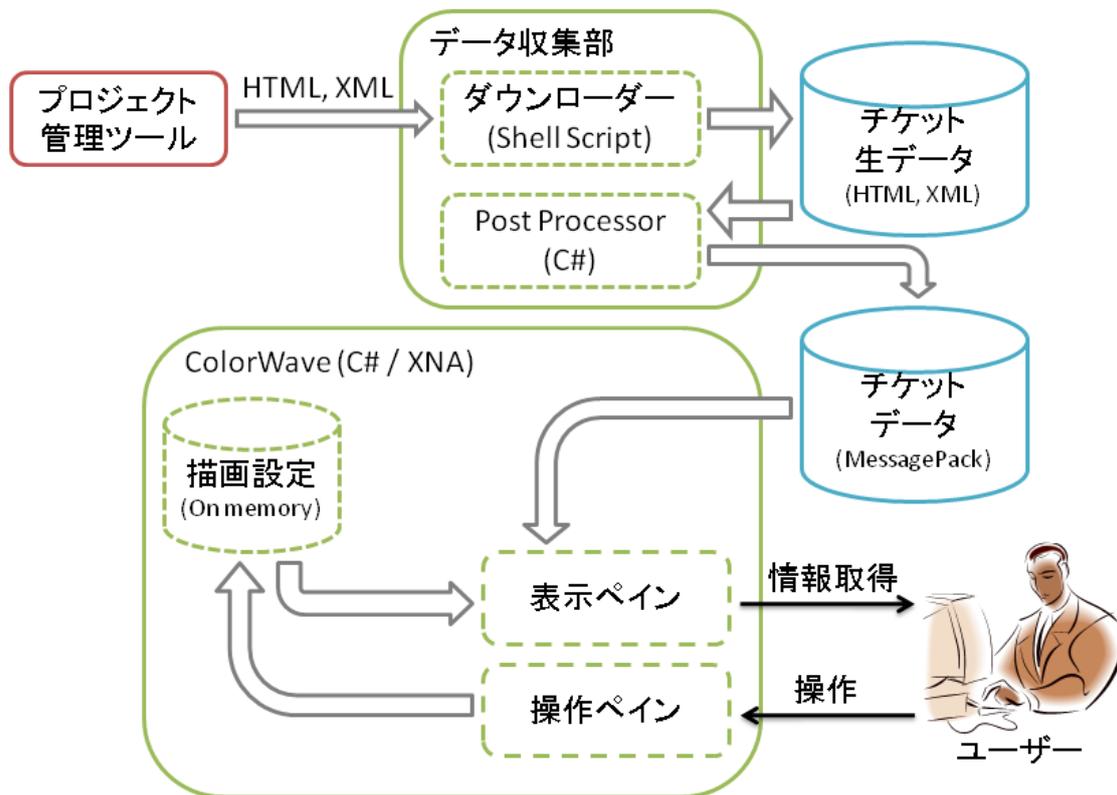


図 8.14: システム構成

3.1 を用いたプログラムである。XNA を通して DirectX を利用することによりビデオカードの処理性能を利用し、大量の描画要素を高速に描画することで、インタラクティブな操作を可能とした。チケットデータのフォーマットには、MessagePack<sup>1</sup> と呼ばれるバイナリフォーマットを用いている。これによって、GB 単位に渡る大量のデータを、XML などのテキストベースのフォーマットよりも軽量で高速に読み書きできる形で取り扱っている。

<sup>1</sup><http://msgpack.sourceforge.jp/>

## 第9章 ケーススタディ

本章では、オープンソースソフトウェアについての実在するチケットデータについてケーススタディを行い、ColorWave がどのように活用できるものなのかについて実例を示す。まず Mono project についてのケーススタディで一例を、それ以降のケーススタディでは、共通点や差異に着目しながら述べる。

なお、本章にある図は、特に断りがない場合は表示ペイン全体のスクリーンショットである。また、キャプションにはケーススタディのソースの名前、属性と色の対応、そしてチケットの並び順が書かれている。

### 9.1 Mono project (7452 チケット)

Mono project<sup>1</sup> のチケットデータ 7452 件を、bugzilla<sup>2</sup> から取得し、これについてケーススタディを行った。

まず、収集したデータを ColorWave にロードさせた。すると、図 9.1 のようになる。図 9.1 を見ると、上端から始まっているチケットの特徴的な集まりの存在と、長さの分布が見て取れる。特に、上端から始まっているチケットの終端が二次関数のようなカーブを描いていることから、現在の Mono の bugzilla が始まった時点から存在するチケットの終端が二次関数のような分布を示していることが分かる。また、点が横に並んでいる部分が複数ある、すなわちイベントが同時に多発した部分があることから、まとめて作業したことがあったことが分かる。さらに、図 9.1 全体や左にある凡例を見ることで、Mono project では Classification 属性は必ず "Mono" になっており、それ以外の値を持たないことも見て取れる。

ここで、他の並び順にするとどうなるだろうかと考え、並び順をチケットの長さ順から、チケットの始端順にする。すると、図 9.2 のようになる。図 9.2 を見ると、Mono project の現在の bugzilla の開始時からあるチケットが相当数あり、それ以降のチケットの始端がほぼ線形に並んでいることが分かる。また、チケット群の上方ほど密であり、下方ほど疎らになっていることが、濃淡として読み取れる。このことから、長さの短いチケットが多数あり、長いチケットは少ないことが分かる。また、上方から下方に掛けての濃淡の変化の仕方が、横位置によってある程度異なっている。この差異から、下方近くになっても色が濃い横位置にあるチケットは全体的に長引いていることや、上方近くで色が薄くなっている横位置にあるチケットは早々に終わっていることが分かる。

---

<sup>1</sup>[http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)

<sup>2</sup><http://www.mono-project.com/Bugs>

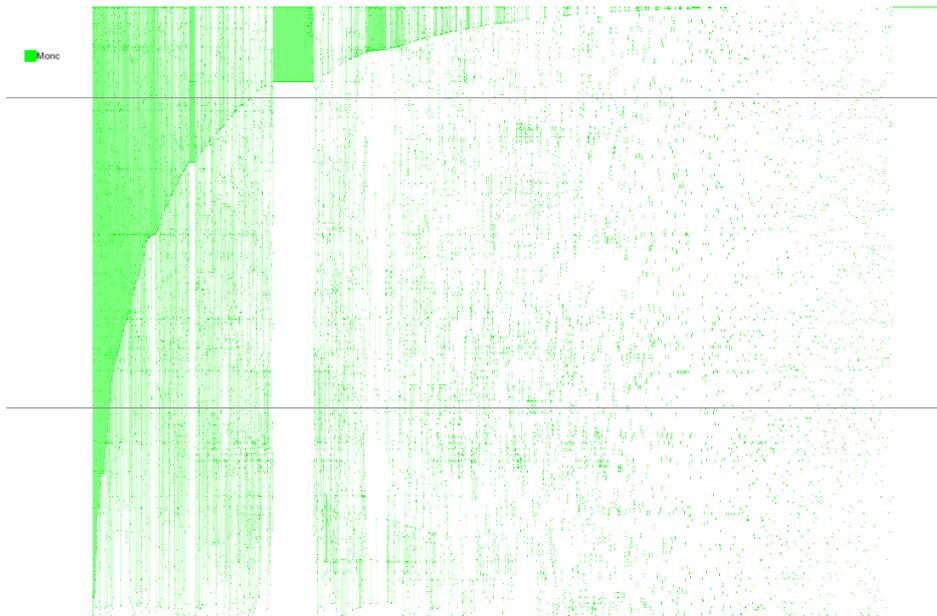


図 9.1: Mono: Classification 属性、長さ順

さらに、水平な点線のように見える、すなわち点が密に横並びしている部分が複数あることから、特定の時点で大量のイベントが発生したことがあったと分かり、特定の時点でまとめて作業したことがあったのだと知ることができる。かつ、点の横並びが図 9.1 よりも濃くなっていることから、まとまった作業の対象は、同じ頃に作られたチケットに対して行われる傾向があることも読み取れる。

ここで、始端順でなく、終端順にする(図 9.3)。図 9.3 を見ると、チケットの終端もほぼ線形であることが分かる。図 9.2 でチケットの始端が線形であったことも踏まえると、Mono project では順調にチケットが増加し、そして順調に処理されていることが伺える。また、図 9.3 では、複数の線が密集することで太い線のように線の固まりが見えている部分が多い。このことから、古いチケットをまとめて終わらせていることが何回もあることが分かる。このような線の固まりは 図 9.2 には少ないことから、「同じ頃に始まったチケット複数まとめて処理されることがある」という傾向と「昔からあったチケット複数まとめて処理されることがある」という傾向が読み取れる。一見するとどちらも自然に思えるが、前者はあまり現実と一致せず、後者は現実に沿っていたことが判明した。

より多くの情報を得るために、Classification 属性より小さい分類を値に持つであろう Product 属性で見る。例えば 図 9.2 の状態から、色として表現する属性を Product へと切り替えると、図 9.4 のようになる。すると、図 9.2 などの Classification 属性の時と異なり、色とりどりの図になる。また、色の偏りが随所に見て取れることから、特定の Product についてのチケットが同一時点でまとめて作られることがままあったことが分かる。さらに、初期は青

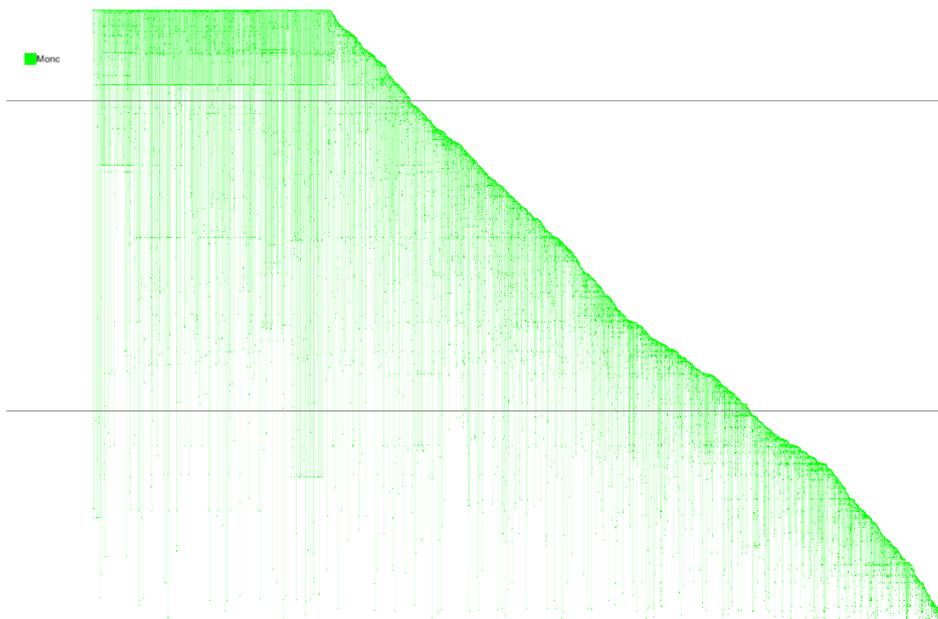


図 9.2: Mono: Classification 属性、始端順

(Compilers や Debuggers) が多く、次に赤 (Tools や Runtime) が多かったが、最近になると赤が増え、黄色 (UI Automator) が出現している。このことから、作られるチケットの傾向、すなわち Mono project における活動の対象のシフトが分かる。かつ、点がほぼ水平に並んでいる部分について、それらの点の色は時期にかかわらず同じであるという傾向が読み取れる。そのため、Product を跨がずに単一の Product 内で、長引いているチケットがまとめて処理されていることが伺える。

ここで、図 9.4 から並び順を終端順にするか、あるいは図 9.3 から Product 属性への切り替えを行うと、図 9.5 のようになる。図 9.3 で述べた線の固まりを図 9.5 で見ると、線を構成するチケットの色がほぼ均一であることから、それらのチケット群は同じ Product のものであることが分かる。また、チケットの終端付近の青 (Compilers や Debuggers) や赤 (Tools や Runtime) の点が、多数固まっていることから、これらの Product については、ある時点で昔からあったチケットについて活発な活動が行われ、結果として同時期に多数のチケットが終わることがあったのだと分かる。

ここで、Product 属性について、色によるクラスタリングを行うと、図 9.6 のようになる。青 (Compilers や Debuggers) や赤 (Tools や Runtime) の占める幅が多いことから、それらが多いのだという先述の所見が裏付けられる。また、黄 (UI Automator) のみならず、鮮赤 (Tools) も最近になって始まったものであることが分かる。さらに、青 (Compilers や Debuggers)、黄 (UI Automator)、鮮赤 (Tools) のどれについても、線形にチケットが発生し、そこから過去へと尾を引くという形状が共通していることが伺える。かつ、濃青 (Debugger) の並びは、時間

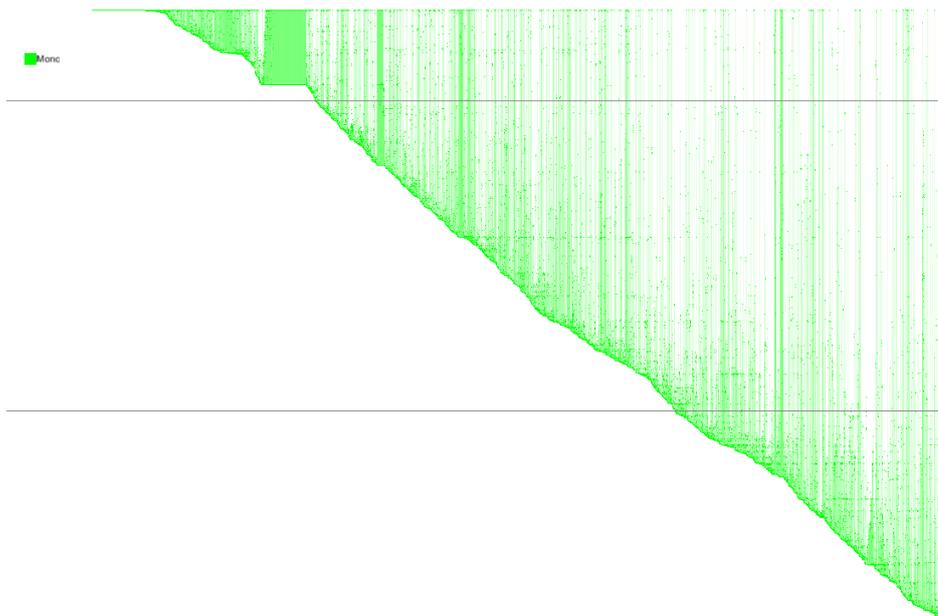


図 9.3: Mono: Classification 属性、終端順

長がごく短いチケットが並んでいることから、それらについてのチケットは長引くことが希であることも分かる。

ここで、Priority 属性を色として表示すると、図 9.7、図 9.8、図 9.9、図 9.10 のようになる。図 9.7 を見ると、上端から始まり、終端が二次関数のようなカーブを描く特徴的な集まりがあることが分かる。また、その集まりの Priority がシアン (P3 - Medium) であり、それ以外は青 (P5 - None) であることも分かる。このことについては、始端順に並べた図 9.8 を見ると、Priority がシアン (P3 - Medium) であった集まりは Mono project の bugzilla に最初からあったチケットであり、青 (P5 - None) はそれ以降のチケットが None であることが、より明確に分かる。そのため、基本的に Medium にする運用ポリシーから None にするポリシーへの、運用ポリシーの変化があったのだらうと知ることができる。

また、それらの図から、大半のチケットは Medium ないし None の Priority であるとも分かるが、図 9.8 を見ると黄色 (P1 - Critical)<sup>3</sup>のチケットも比較的最近には少数あったことが、黄色のチケットの集まりの存在から見て取れる。ここで、Medium や None それに Critical 以外の Priority のチケットが無いか気になる。そこで、埋もれている色を強調させると、図 9.11 のようになる。すると、確かに一部にはある程度以上の長さを持った赤 (P0 - Critical) チケットがあったことが分かる。加えて、短い Critical なチケットの存在を確かめるために、ローカルな時間軸で表示させると、図 9.12 や図 9.13、図 9.14 や図 9.15 のようになる。

まず、Priority 属性値順で見る (図 9.12) と、Medium や None 以外のチケットの固まりは見

<sup>3</sup>凡例では橙色だが、色の合成によって彩度が下がっているのewithゆる黄色に見える

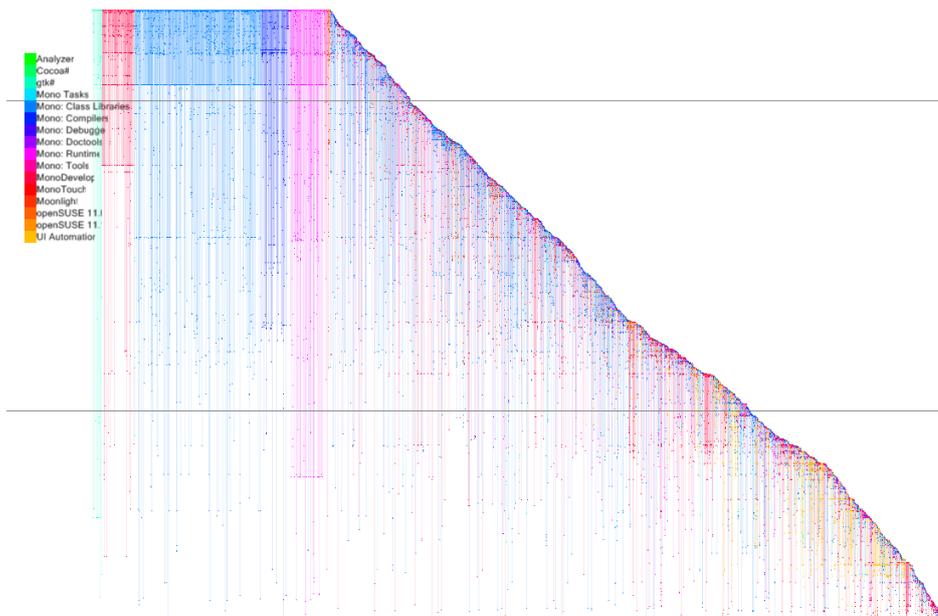


図 9.4: Mono: Product 属性、始端順

あたらない。なぜなら、属性値によるソートは原理的にチケットのある瞬間の属性値<sup>4</sup>でソートすることになる。そのため、途中で属性値の変わるチケットがあった場合に、必ずしも同じ属性値を持ったチケットが固まらないため、Medium や None 以外の固まりが見あたらないのである。このことは、埋もれている色を強調させる (図 9.13) と、青いチケットたちの中に青色以外のチケットが埋もれていることが分かることから裏付けられる。そこで、属性値による並び替えの代わりに、色によるクラスタリングを行う (図 9.14) と、青や水色以外のチケットがある程度存在することが、図の右端付近から分かる。加えて、属性値によるソートでは埋もれていたがクラスタリングすることでチケットの集まりが見えたことから、青や水色以外のチケットは、チケットの属性が途中から変更されているものであるという推測が裏付けられる。また、色によるクラスタリングを行っても赤いチケットは殆ど見えないことと、埋もれた色を目立たせて初めて赤色が見える (図 9.14 の右側など) ことから、Critical なチケットはごくわずかであることが分かる。

<sup>4</sup>ColorWave では、属性の初期値

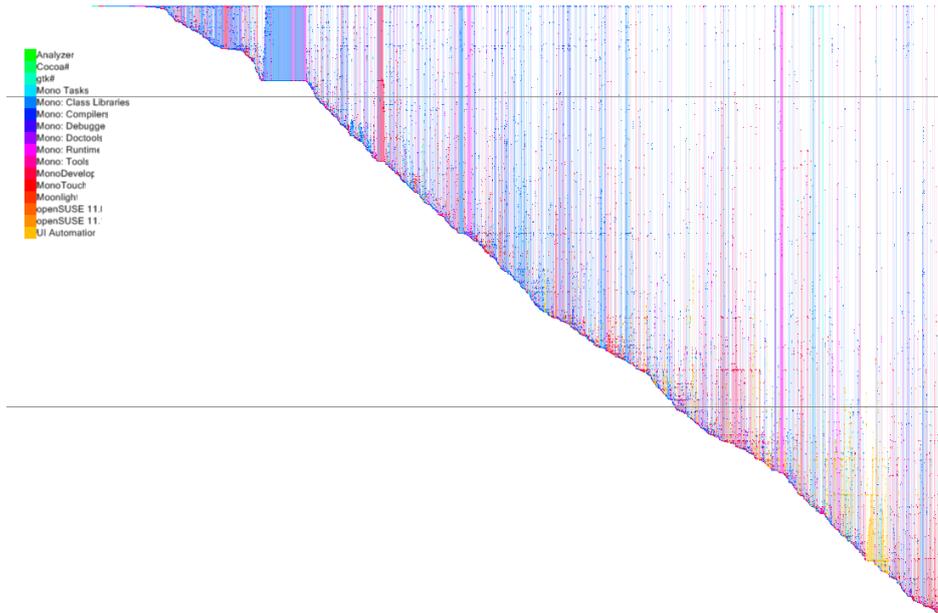


図 9.5: Mono: Product 属性、終端順

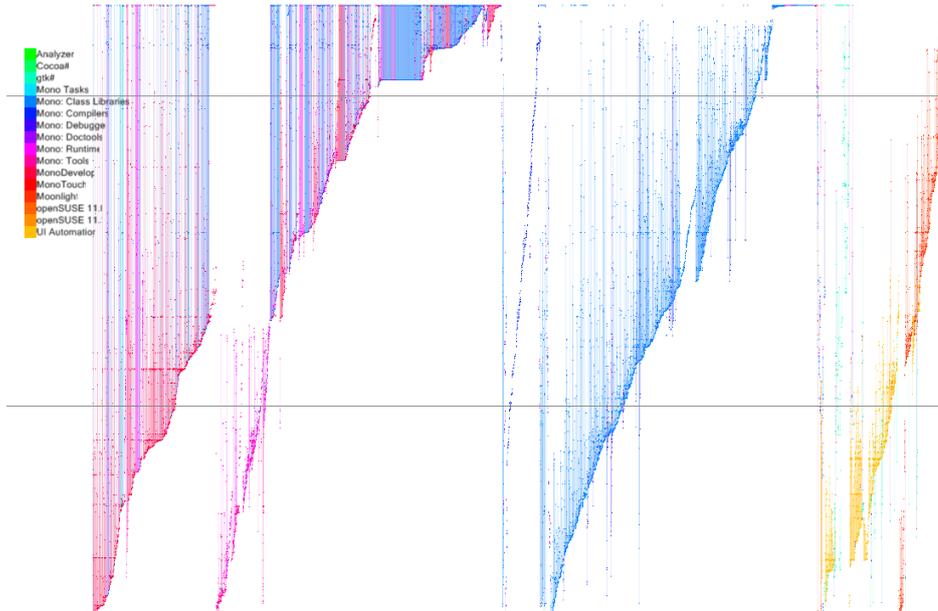


図 9.6: Mono: Product 属性、色によるクラスタリング

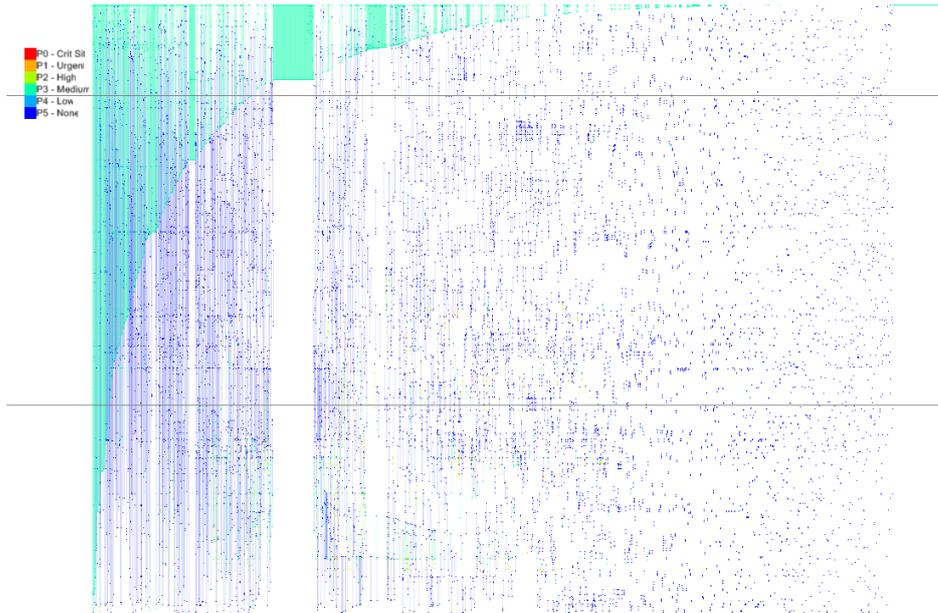


図 9.7: Mono: Priority 属性、長さ順

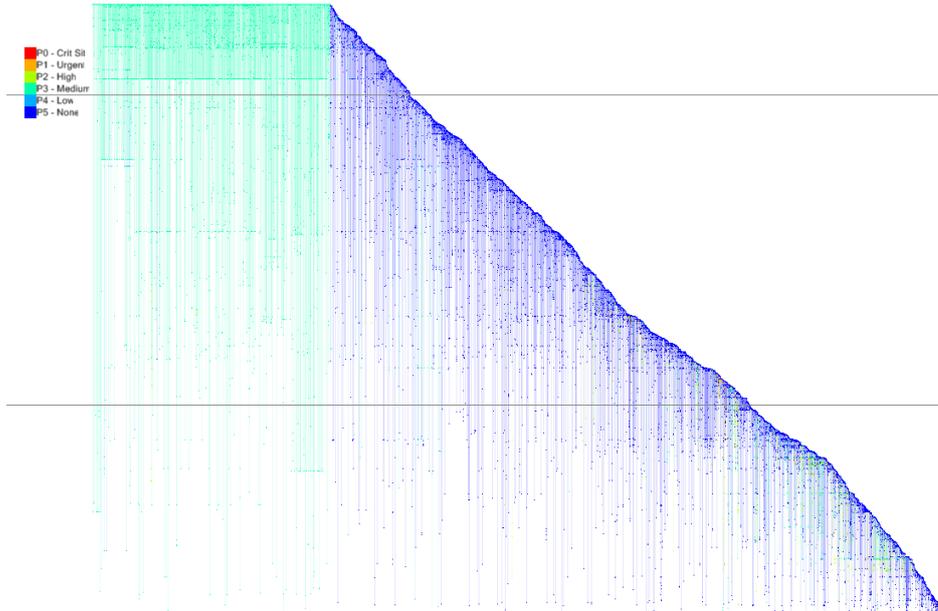


図 9.8: Mono: Priority 属性、始端順

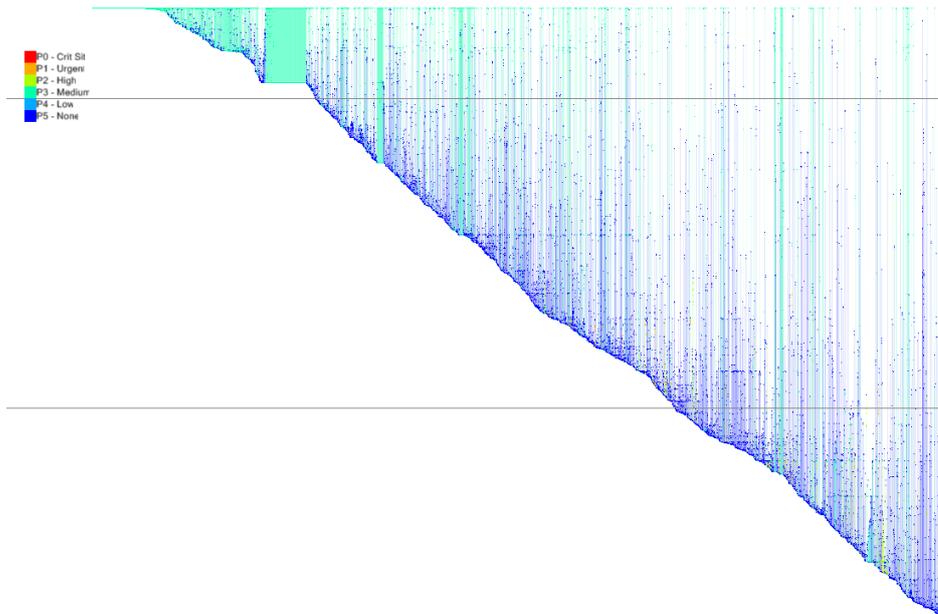


図 9.9: Mono: Priority 属性、終端順

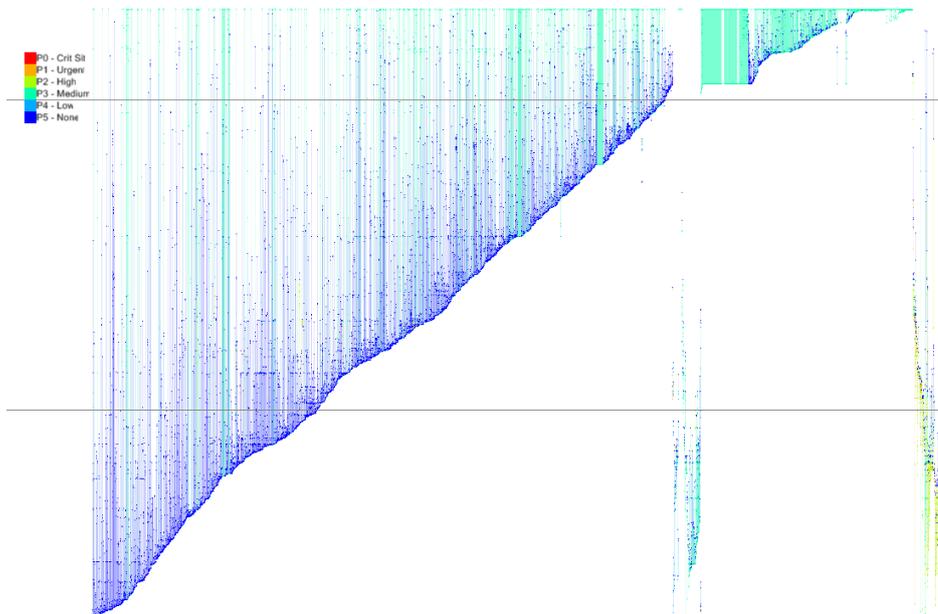


図 9.10: Mono: Priority 属性、色によるクラスタリング

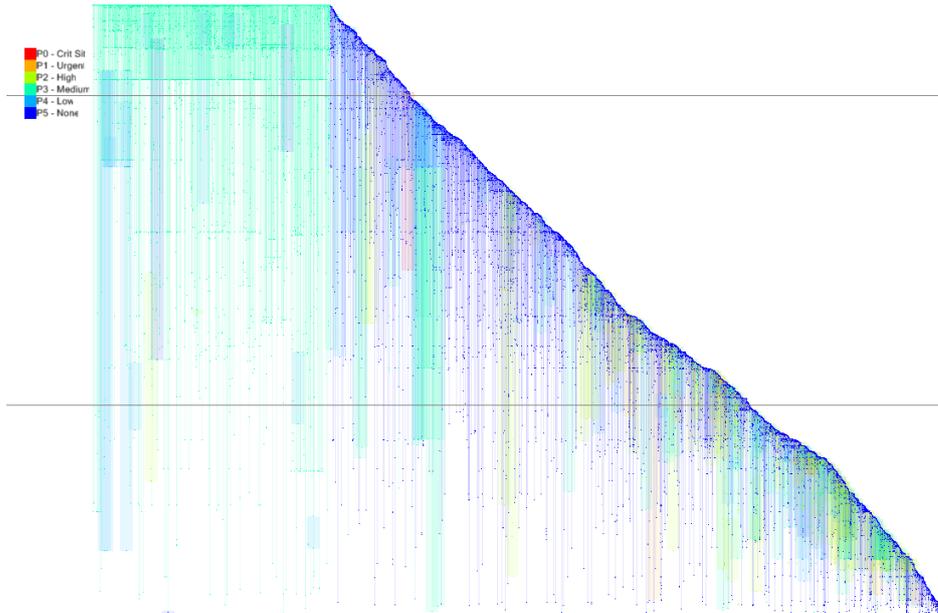


図 9.11: Mono: Priority 属性、始端順 (埋もれている色を強調)

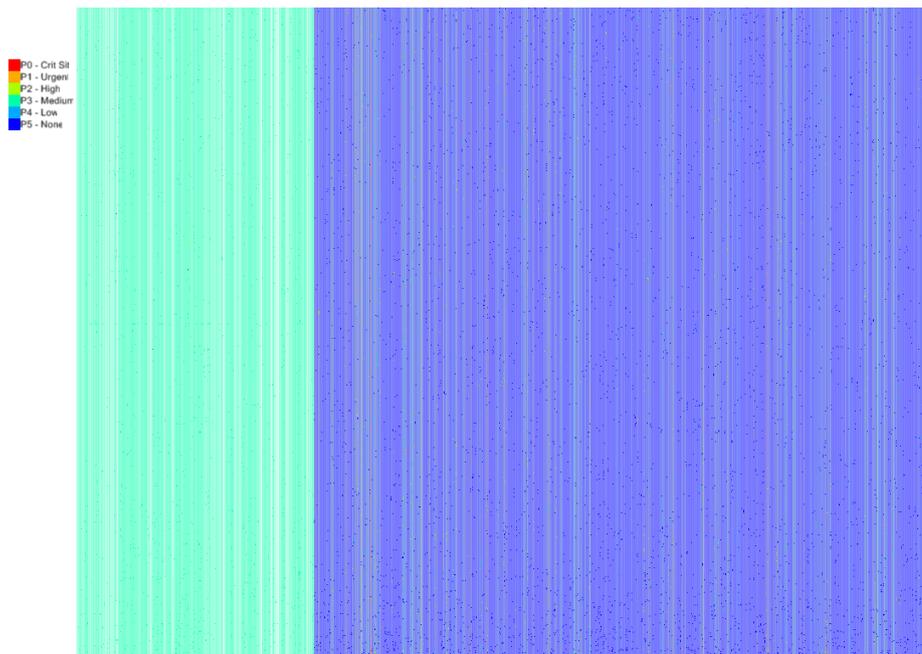


図 9.12: Mono: Priority 属性、Priority の属性値順

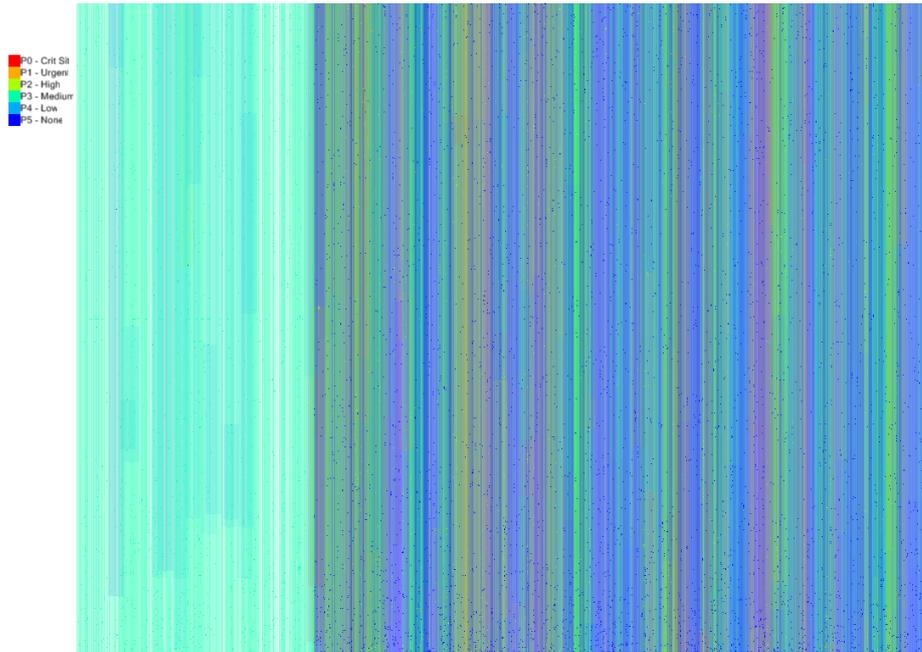


図 9.13: Mono: Priority 属性、Priority の属性値順 (埋もれている色を強調)

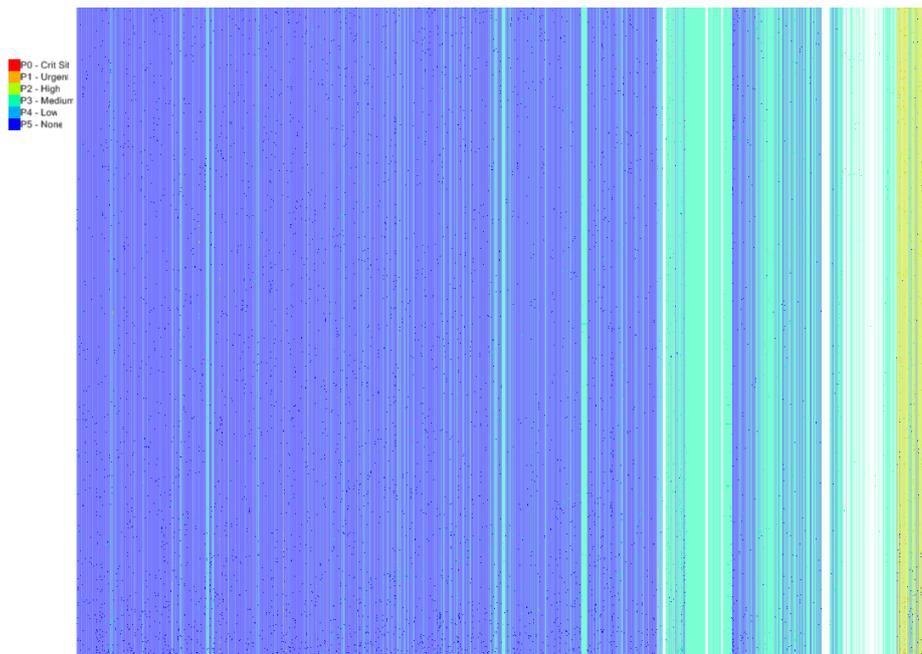


図 9.14: Mono: Priority 属性、色によるクラスタリング

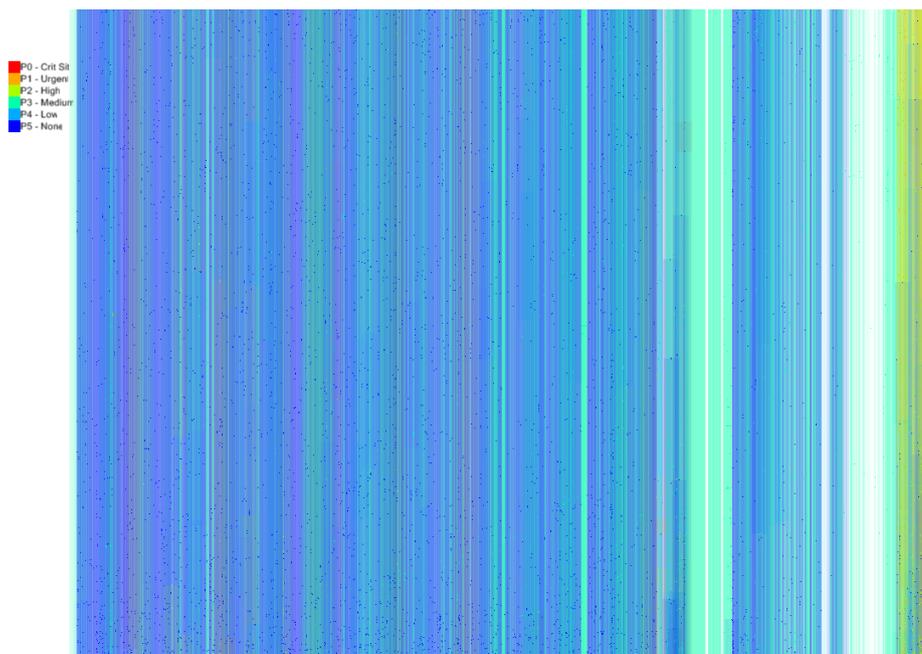


図 9.15: Mono: Priority 属性、色によるクラスタリング (埋もれている色を強調)

## 9.2 Redmine (2253 チケット)

プロジェクト管理ツールである Redmine<sup>5</sup>それ自体の開発についてのチケットデータ<sup>6</sup> 2253 個を取得し、これについてケーススタディを行った。このデータは、ケーススタディの中では比較的小規模である。

まずデータを読み込むと、図 9.16 のようになる。チケットの時間分布に着目すると、二

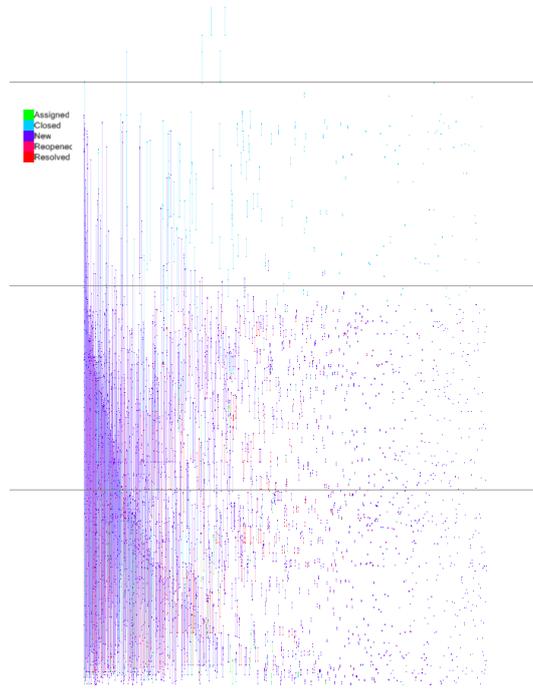


図 9.16: Redmine: ステータス属性、長さ順

次関数的な曲線が見いだせることは Mono project と類似しているが、Mono project では上端から始まり、下端が曲線を成していたのに対して、Redmine では上端が曲線を成し、下端付近で終わっているという、形状が上下逆である点が異なっている。ここで、図 9.17 のようにすると、チケットの増加速度が、紫(ステータスが NEW)が支配的になった時点を境に大きくなっていることと、最近にまで伸びているチケットの多くは紫が支配的になってからのものであることが読み取れる。以上から、長引いたチケットが時間に対して二次関数的になるという共通性と、Mono project のようにチケットの終了が二時間数的になるパターンと、Redmine のようにチケットの開始が二時間数的になるパターンの両方があり得るというバリエーションが分かる。

また、図 9.17 からは、チケットの増加速度が速くなった時点から紫が支配的な傾向があると分かるが、この傾向は 図 9.18 でより明白になる。直感的には、似た背景のチケットは同

<sup>5</sup><http://www.redmine.org/>

<sup>6</sup><http://www.redmine.org/projects/redmine/issues>

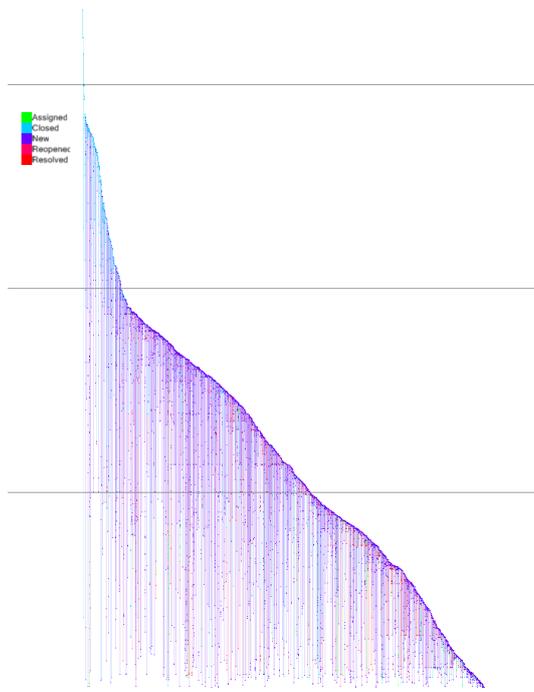


図 9.17: Redmine: ステータス属性、始端順

じ時期に作られるとも考えられるのだが、実際には、Mono project でも、Redmine でも、同じ時期に終わるチケットの方がより傾向が明白になることは興味深い。ただし、点の横並びについては、Mono project においても Redmine においても、チケットの始端でソートした方がよりはっきりと見えている。このことから、内容に共通点の多いチケットは開始日よりも終了日に強い相関があるが、メンバーがまとめて作業する場合には開始日に着目して作業しているという、ささやかだが興味深いギャップが見いだせる。

ここまでは図の形状からの発見を主に述べたが、図の色からの発見もある。例えば、図 9.17 や 図 9.18 を見ると、古いチケットは最初から最後までシアン (ステータスが Closed) であるという、いささか不思議な傾向が読み取れる。実はこれは、Redmine のチケットが開発中の Redmine そのものによって管理されており、初期の Redmine はステータス属性が存在せず、後からステータス属性の情報が補完されたためである。バグや要望と言ったインシデントを管理するという目的からすると、チケットの現状を記録する機能を最初から付けなかったのは不思議にも思えるが、どうやら Redmine の作者は、ステートの管理よりも、コメントを付けることでドッグフードを食べる<sup>7</sup>ことが可能な状態にすることを優先したようである。

また、ここまでの図では、シアン (Closed) か紫 (NEW) が殆どであったが、他のステータスは使われていないのかが気になるので、図 9.19 や 図 9.20 を見る。図 9.19 を見ると、一部のチケットで赤 (Reopened) や緑 (Assigned) が使われていることや、チケットの先頭でも末尾

<sup>7</sup>自らの作っているものを自らで使うことである [5]。

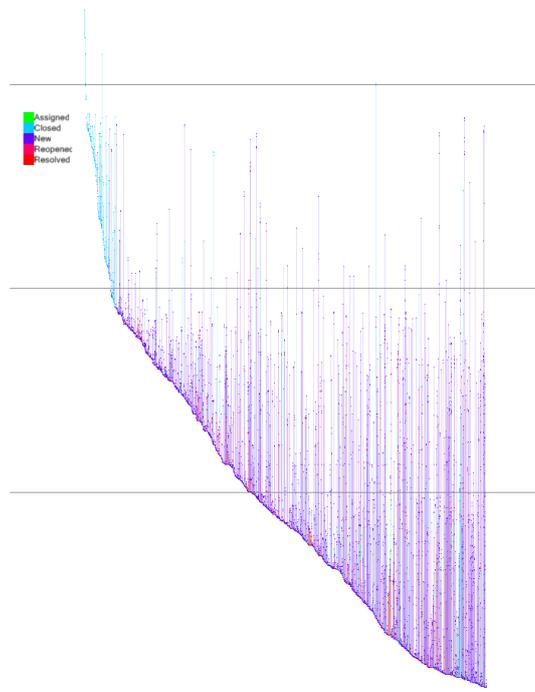


図 9.18: Redmine: ステータス属性、終端順

でもなく途中で赤や緑になることが一般的である様子が伺える。また、図 9.20 を見ることで、赤や緑はかなり少数であることも分かる。

ここまで言及してきたように、Redmine は比較的小規模なプロジェクトであるので、チケットの担当者毎に見てみると図 9.21 のようになる。また、傾きが急に変わる場所(支配的なステータス属性が紫から緑へ変わったポイントでもある)以前の担当者は - (無記入) であったのが、それ以降は複数人になっていることが読み取れ、これらのことから Redmine の開発プロジェクトが急激に複数人開発へとシフトしたことが伺える。

ここで、図 9.22 のようにチケットを担当者、次に開始日時でソートすると、これもまた興味深い。図の中央を占めているのは担当者が - (無記入) のチケットであり、左に緑 (Azamat) やその他のチケット、右にシアン (Chaoqun) や濃青 (Jean) のチケットの固まりが見受けられる。まず、中央の担当者が無記入である固まりに着目すると、左側の散らばっている部分と、それより右側の線形になっている部分に大別される。開始日でソートされているにもかかわらず左側が散らばって見えるのは、これらのチケットが Redmine の開発の初期のものであるために、開始日などの情報も入っていないためである。初期の Redmine の開発では、ステータスのみならずタイムスタンプも実装されていなかったことが分かる。また、複数の開発者が活動し初めてからも担当者が無記入のチケットが多数、コンスタントに作られてきていることから、今も昔も、担当者属性はある程度限られたチケットについてのみ設定されていることが明らかになった。ここで、中央の緑 (無記入) の固まりと、右にあるシアンや濃青の固

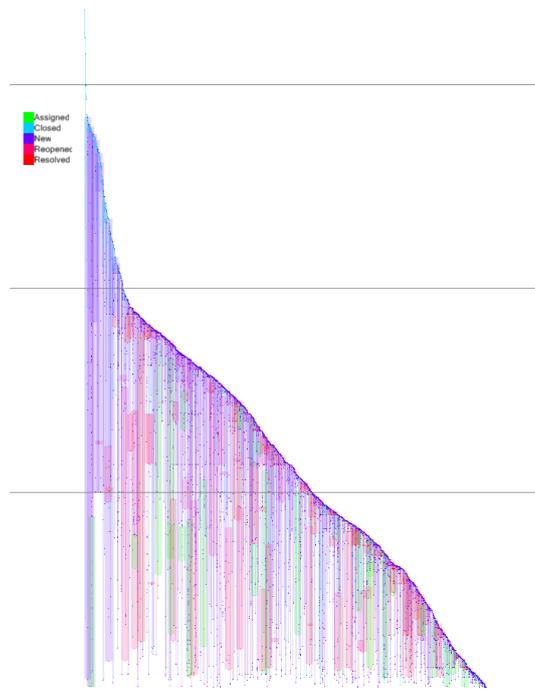


図 9.19: Redmine: ステータス属性、始端順 (埋もれている色を強調)

まりを見比べると、無記入のチケットはごく最近まで長引くことが多いが、シアンや濃青のように予め担当者へ割り当てられているチケットは長引くことが希であることが分かる。このことから、担当者に明確に割り振られているものは、そうでない物よりも長引きにくいことが伺える。左端の緑 (Azmat) の固まりについては、緑以外の色がちらほらと混ざっているもので、図 9.22 と対比する。すると、チケットが始まってからある程度してから、緑以外の色に変化していることが多いことや、緑以外の色のチケットは大抵長いことが分かる。このことから、緑 (Azmat) に割り当ててあるチケットは、その後に緑 (Azmat) から他の人間へタスクがディスパッチされることがあり、かつディスパッチされたタスクが得てして長引いていることが伺える。

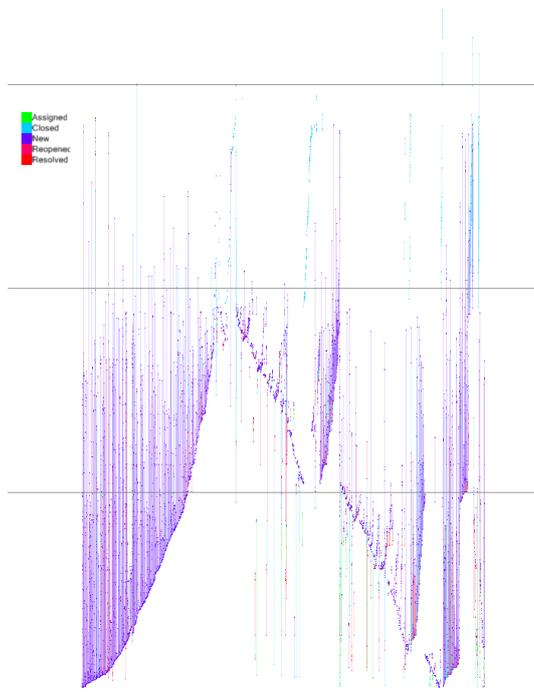


図 9.20: Redmine: ステータス属性、色によるクラスタリング

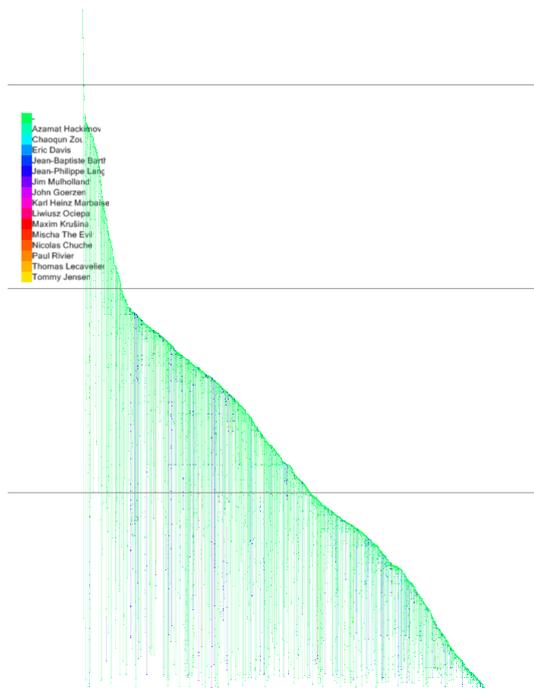


図 9.21: Redmine: 担当者属性、始端順

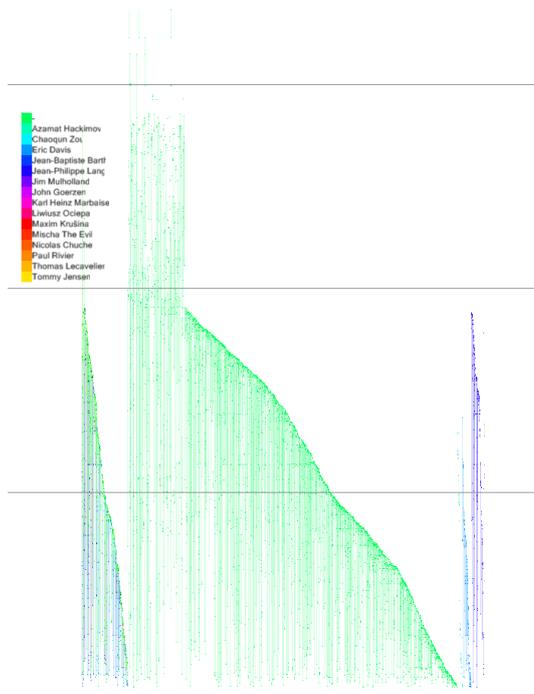


図 9.22: Redmine: ステータス属性、担当者、次に開始日時でソート

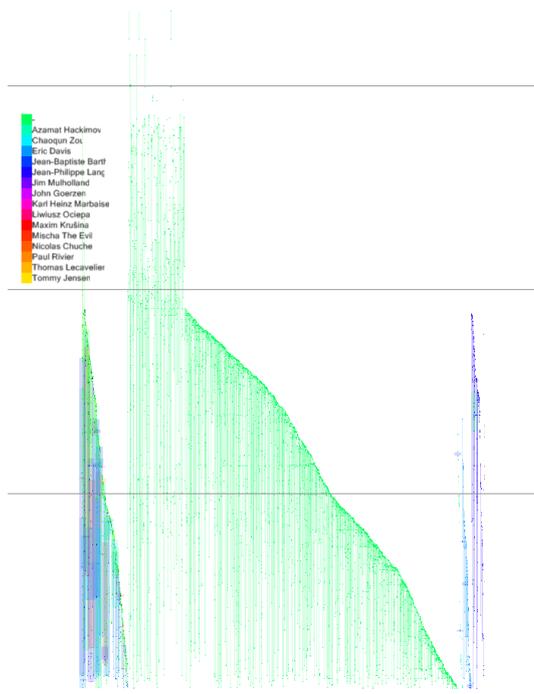


図 9.23: Redmine: ステータス属性、担当者、次に開始日時でソート (埋もれている色を強調)

### 9.3 考察

ケーススタディで挙げたプロジェクトのメンバーではなく内情に詳しくもない筆者でも、低いハードルで作業を開始できていた。このことから、ColorWave では前提知識が要求されることなく、分析に取りかかることができると考えられる。

チケット群のシルエットの形状や点の分布から、時間的な位置を読み取ることが出来ていた。また、色の濃淡から、それぞれの時点におけるチケットの数量を見て取ることも出来ていた。このことから、視覚的表現によって、チケットやイベントがどの時間にあるのか、を表現できていたと考えられる。

色を見ることで、それぞれ属性値の存在率や、あるチケットの固まりの属性を見て取ることも出来ていた。また、同じ色の線や点の並びに気づくことも多く、それによって特定時点でのチケットやイベントの属性値の偏りを一見して把握できていた。このことから、視覚的表現では、属性値を色によって表せていたと考えられる。

操作をインタラクティブに行えたことで、前の発見を踏まえたさらなる発見をすることが出来ていた。特に、チケットの並び順の変更操作と、属性と色の対応の変更操作は、発見を得たり仮説を検証するに当たって、よく用いられていた。チケットの時間位置による並びによって、チケットの時期に着目した分析が行えていた。それに対して、長さによる並びには、時期に依存せずに、支配的な要素を見いだすことに貢献していた。色のクラスタリングによって、属性値が近いチケットのクラスタの中の傾向や、属性値の異なるチケット同士の傾向の異なりを分析できていた。また、色のクラスタリングでは、クラスタ同士の数の違いを見て取ることもできており、多くの情報を読み取れていた。属性値による並びは、特に色によるクラスタリングと対比することで役に立っていた。色によるクラスタリングで得られたクラスタが、どのようなクラスタなのかを検証する際に、属性値によってソートした結果と比べることで対比できていた。

時間軸については、多くの知見はグローバルな時間軸において得られたものであった。ケーススタディにおいては、グローバルな時間軸上で各種操作を行っていく過程でチケットのローカルな時間軸における傾向も見えていることが多かったので、ローカルな時間軸の必要性は低いようである。ただし、チケットの長さを無視して、チケットの存在だけを見る用途では、ローカルな時間軸が有用であった。

埋もれている色のハイライトは、直接的には、支配的な色の中に埋もれている色の発見に役立っていた。しかし、ケーススタディにおいては、チケットの生涯における限られた時点でしか使われていない属性値が、チケットのどの時間位置を占めているのかを発見したり、クラスタの内の下の方では何色が多くなっているといったクラスタの内部構造を発見するといった、他の用途にも役立っていた。

このことから、操作体系の多くの操作は、次々と発見を得たり、仮説を検証することの役に立っていたと考えられる。特に、埋もれている色のハイライトは、視覚的表現をそれほど大きく変えないものの、様々な場面で役に立っていた。また、属性値による並びやローカルな時間軸は、それ自体が有用な局面は少ないが、他の操作との対比によって仮説を検証する場合や、特定の条件下では役立っていたと言える。

## 9.4 まとめ

属性や並び順を切り替えたり、時間軸のグローバルとローカルでの切り替えや埋もれている色の強調を用いることで、チケットデータからさまざまな知見を読み取ったり、仮説を見だし、それを確かめることができた。Mono project については、Product の開発時期や規模などについて、筆者が漠然と抱いていた印象が実情とは異なることが分かるなどの知見を得られた。また、開発がコンスタントに行われていることを、具体的なデータから知ることが出来た。Redmine については、単独による開発から、複数人開発へと急激にシフトしたことによる様変わりを、チケットの数や担当者の割り当て方などからまざまざと感じ取れた。加えて、タスクの分担がうまく機能する状況とそうでない状況の違いも知ることが出来た。

なお、本章で述べた操作は本論文に書かれている順序通りに行う必要は無く、様々な切り替えを思いつくままに縦横に行うことが可能である。

## 第10章 結論

ソフトウェア開発プロジェクトの活動状況を概観・分析するために、大量のチケットデータについての視覚的表現と操作体系を開発し、ツールを制作し、ケーススタディを行った。まず、チケットの属性値やイベントおよび時間位置の3つ全てを、限られた面積における色や点の分布として表現することで、チケット群から傾向を読み取れるようにする視覚的表現を開発した。また、時間やチケットの並びや属性値を切り替えることで、多角的な分析を可能とする操作体系も開発した。そして、上記の視覚的表現と操作体系を備えたツール「ColorWave」を開発した。

さらに、実在するプロジェクトのチケットの実データを用いたケーススタディによって、知見や仮説の発見や検証を行った。

チケットデータの視覚的な表現と多角的な分析のための操作体系により、今まで難しかった、大量のチケットの概観と分析を時間情報も含めて多角的に行うことが可能になった。

## 謝辞

研究および本論文の執筆にあたり、指導教員である三末和男先生、田中二郎先生をはじめ、志築文太郎先生、高橋伸先生には、日頃のゼミやミーティングなどを通じ、研究への取り組み方や研究内容、論文執筆など多岐に渡って、深いご指導やアドバイスを頂きました。心より感謝しております。また、筑波大学システム情報工学研究科インタラクティブプログラミング研究室 (IPLAB) のメンバーの方々からも、ゼミや日頃の研究活動を通して、多くの有意義なフィードバックや刺激を頂いた上、毎日の研究生活についてもアドバイスや助けを頂き、深く感謝しております。さらに、研究に従事することを快諾していただき、また日頃より支えと先達としてのアドバイスをしてくださった株式会社ワイズアンドテクノロジーの皆様に感謝申し上げます。最後に、様々な面で力になっていただいた家族や友人、大学生活でお世話になった全ての方に心から感謝致します。

## 参考文献

- [1] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60, 2008.
- [2] Jin Chen, Alan M. MacEachren, and Donna J. Peuquet. Constructing overview + detail dendrogram-matrix views. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):889–896, 2009.
- [3] Jason B. Ellis, Shahtab Wahid, Catalina Danis, and Wendy A. Kellogg. Task and social visualization in software development: evaluation of a prototype. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 577–586, New York, NY, USA, 2007. ACM.
- [4] Michael Fischer and Harald Gall. Mds-views: Visualizing problem report data of large scale software using multidimensional scaling. In *Proceedings of the International Workshop on Evolution of Large-scale Industrial Software Applications (ELISA)*, September 2003.
- [5] Jerome Kelli. *Inside Out: Microsoft—in Our Own Words*. Diane Pub Co, 2005.
- [6] Chris Muelder, Francois Gygi, and Kwan-Liu Ma. Visual analysis of inter-process communication for large-scale parallel computing. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1129–1136, 2009.
- [7] Ben Shneiderman. Extreme visualization: squeezing a billion records into a million pixels. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 3–12, New York, NY, USA, 2008. ACM.
- [8] Margaret-Anne D. Storey, Davor Čubranić, and Daniel M. German. On the use of visualization to support awareness of human activities in software development: a survey and a framework. In *SoftVis '05: Proceedings of the 2005 ACM symposium on Software visualization*, pages 193–202, New York, NY, USA, 2005. ACM.