

# idea in: プロジェクト管理を効率化する Web アプリケーション その 3 - タスクチャート手法によるプロジェクトの目的別可視化

浜田真成<sup>†1</sup> 安藤大地<sup>†1</sup> 笠原信一<sup>†1</sup>

本研究は、専門知識や難しい操作を必要とせず扱えるユーザーフレンドリーなプロジェクト管理支援 Web アプリケーションの開発と、そのシステムの実現を目指すものである。前報では、そのための手法としてバージョン管理と表形式のインターフェースによってプロジェクトを管理する「タスクチャート手法」を提案した。今期研究では、上述のタスクチャート手法に、日時・進行度・期限・優先順位・制作者等をキーとし、目的別の切り口から全体を閲覧できるビジュアライゼーション機能を導入することによって、プロジェクトの過程を残しながら、それぞれの役割に応じて効率的にプロジェクトを管理する手法を提示する。

## idea in: A Web Application for Project Management Part. 3 – Project Visualization by Task-Chart method

MASANARI HAMADA<sup>†1</sup> DAICHI ANDO<sup>†1</sup> SHINICHI KASAHARA<sup>†1</sup>

This study proposes user-friendly project management system without a specialized knowledge and a difficult operation, researches the way of efficiency of collective work in the process of project management, and aims to build this system by developing the web application software. In the previous paper, the authors proposed the “Task-Chart” that manages projects by using a version control and a tabular interface.

In this study, the authors added a visualization function to the “Task-Chart” in order to overlook the whole project from different viewpoints based on different purposes by using a key of date, progress, term, priority, person, and so on. Finally, the authors propose effective project management method to keep history and manage through each different work style.

### 1. 概要

本研究は、主に物を制作するプロジェクトにおいて、専門知識や難しい操作を必要とせず利用できるユーザーフレンドリーな Web アプリケーションの開発と、そのシステムの実現を目指すものである。

これまでの研究では、デザイナーやプログラマーといった異なる役職間の連携に関する問題点や現状を踏まえ、プロジェクトを円滑に進めるための改善手法を提案、その手法に基づく Web アプリケーションの構築し、フィードバックを行ってきた。

今期研究では、プロジェクトにおけるタスクをバージョン管理と表形式のインターフェースによって管理する「タスクチャート手法」に、日時・進行度・期限・優先順位・制作者等をキーとし、目的別の切り口から全体を閲覧できるビジュアライゼーション機能を導入することによって、プロジェクトの過程を残しながら、様々な役割に応じて効率的にプロジェクトを管理する手法を提示する。

### 2. 背景

#### 2.1 異なるプロジェクトタイプ・役職間の連携の問題

異なる分野間の人々が連携するプロジェクトでは、分業によって情報共有が困難となり易く、プロジェクトの全体像が見えにく

くなる問題がある。

また、ソフトウェア開発や作品制作など、各専門分野でのプロジェクトの管理方法を見てみると、下記のように一つのプロジェクトでも、分野ごとに異なった専門手法が導入され、別々に運用されている現状がある。

#### (1) ソフトウェア開発手法

分散バージョン管理の手法を用いて、変更をソースコードの修正差分として管理し、開発を進める。

#### (2) プロジェクトマネジメント手法

マイルストーンを設定し進行状況を管理し計画的に進行を行う。また、アーンドバリュー評価や、バーンダウンチャートによってプロジェクトの進捗状況を数値によって可視化し、計画を常に見直ししながら最終目標に到達させる。

#### (3) 作品制作における、制作者の手法

主に作品を納期までに仕上げるために、担当ごとにスケジュール管理をしながら行う方法が一般的である。制作では、途中経過などをやりとりしながら、その都度調整を行う。

一方で、同じプロジェクト内でも、デザイナーとソフトウェア開発者との連携など、管理手法が異なる分野間の連携をしなければならない状況がある。こうした異分野間の連携を行うプロジェクトの場合、全体の進捗状況等を他分野から俯瞰的に把握することは、いっそう困難となり易い。

<sup>†1</sup> 首都大学東京システムデザイン学部  
Tokyo Metropolitan University

## 2.2 導入障壁の問題

プロジェクトの進行を補助するためのツールとして、多くのプロジェクト管理ソフトウェアの既往事例がある。

既往事例は大きく分けて、専門的ソフトウェアと、簡易的ソフトウェアに分類することができる。

一般のユーザーは、多くが簡易的なソフトウェアを導入しているが、こうした管理ソフトウェアの多くは日程管理の機能に限られており、プロジェクトを効率的に管理できる専門手法が導入されていない場合が多い。

一方で、専門手法を利用する管理ソフトウェアは利便性がある反面、専門手法に独自の操作方法や事前知識が必須となってしまうことが導入障壁となり、一般のユーザー同士が管理ソフトウェアを導入する際には利用されていない傾向にある。

## 2.3 タスクチャート手法の導入

こうした現状の問題点を解決するため、本研究では、

- i) 各分野の人々が利用でき、プロジェクトの進行状況をひと目で俯瞰的に把握できる。
- ii) 各分野で利用されている様々な専門手法を簡素化して実装するアプローチを行い、一般ユーザーでも専門手法が直感的に利用できる。

上記2点を重視し、プロジェクトにおける課題(=タスク)を表形式に管理するインターフェースと、主にソフトウェア開発で用いられるバージョン管理の手法でファイルを管理できる、「タスクチャート手法」を提案した。また、本システムを利用した Web アプリケーション(idea in)の構築・運用し、そのフィードバックを行った。(図1)

タスクチャート手法では、横軸にチームごとのカテゴリーを、表示されているボックスがプロジェクトにおけるタスクを表し、それらを簡易的な時系列で表形式に整理することによって、プロジェクトの全体像を表現する。

多くの管理ソフトウェアで用いられるタスク管理や、期限による工程管理を、表形式のインターフェースで管理できるようにしたことで、今どのチームで何が行われているのか、どこまで進捗しているのかといったプロジェクトの全体像を、ひと目で俯瞰的に見渡すことを可能とした。

また、タスクを開いた画面の右パネルではファイルとその簡易的な差分管理として閲覧することができ、いつ誰がなんのファイルの更新を行ってきたか、その過程を閲覧することができる。反対の左パネルでは、そのタスクに対して掲示板形式で簡易的なコミュニケーションをとることができる。(図2)

こうした簡易的にビジュアル化されたインターフェースにより、差分管理や工程管理といった専門的手法の概念に抵抗のあった利用者層も専門知識を覚えることなく利用でき、また今までこうした専門手法を意識してこなかったユーザーグループにおいても、本システムを利用することで過程を残しながらプロジェクトの管理を行うことが可能となる。

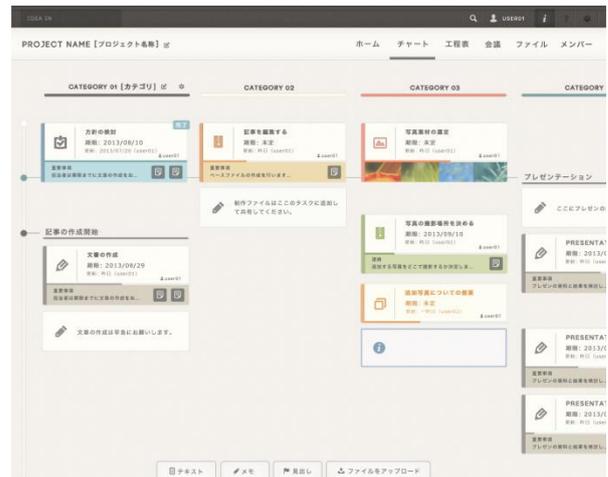


図 1. idea in 全体像



図 2. 差分管理とコミュニケーション

## 3. 目的別可視化機能

### 3.1 現行システムの改善案の検討

今期研究では、本手法を利用した Web アプリケーションの運用の結果、閲覧できる情報が平面的になる課題が挙げられた。

2.1の背景で指摘したとおり、プロジェクトは分野ごとによって進行方法が異なるため、各自の役割の違いにより閲覧すべき情報が異なる。例えば、プロジェクトの統括者は、全体の進捗状況を把握しながら、各役割に指示を与えてプロジェクトを進捗させる。一方で、制作担当者は、自らのタスクを優先的に進めるためにタスクの重要度を重視するとともに、関係する担当間の進捗を優先して把握しなければならない。また、経理担当者場合は、コストを優先して把握しなければならない。

こうした認識事項の違いは、1.2で挙げた制作手法の違いのように、それぞれの役割の違いが連携を行う際の齟齬につながる。

また、一方でフィードバックから、アプリケーションの操作においても、タスクとなるアイテムを手動で時系列のみの表形式で

整理する機能は閲覧したい情報が分散してしまう UI 上の問題が挙げられた。時系列のみでタスクを表示した場合、過去のタスクが画面下に押しやられてしまうため、効率的にプロジェクトを閲覧できないケースがあった。

そこで、簡易的な時系列のみによる整理ではなく新しい指標を加えることで、これら問題点の改善を図った。

### 3.2 多面的切り口からのプロジェクトの可視化

前述の問題点を解決するため、様々な立場の人間が異なる指標でプロジェクトを閲覧できるように、各自が重視する事項で優先して抽出して可視化できる機能(目的別可視化機能)を提案する。

本機能では、これまで簡易的な時系列のみで整理されていたタスクの管理を、選択された基準をもとに動的にソートして一覧表示でき、プロジェクトを様々な切り口から多面的に捉えることができる。(図 3)

基準は日時・進行度・期限・優先順位・制作者・コスト等を設定し、閲覧者はそれぞれをキーにして、タスクチャートの画面を適宜切り替えることができる。これにより、プロジェクトを時系列のみの一面的な情報からだけでなく、それぞれの立場上で必要な事項を優先して閲覧することが可能となる。

例えば、プロジェクトの統括者においては「進行度」をキーにしてタスクを表示する。これにより、タスクの進捗状況を基準にプロジェクトを俯瞰でき、遅延しているタスクを把握しサポートにつなげることができる。また、制作担当者は、「重要度」や「日時」をキーにして表示することで、重要なタスクを中心におきつつ、関連するタスクの進捗や、制作されているファイルを把握することが可能となる。

### 3.3 利用の想定

また、こうしたプロジェクトを様々な側面から捉える機能は、現時点の進捗状況だけでなく、プロジェクトをあとから振り返る場合にも効率的に利用できる。時系列だけではなく、様々な切り口からプロジェクトが進行してきた過程を追うことができるため、プロジェクトの反省や改善を行う際に、情報の把握が行い易い。

また、本機能が効果的に利用できる分野として、アートプロジェクトの分野がある。アートプロジェクトは一度きりのイベントとなる場合が多いため、後発のプロジェクトが以前に開催されたイベントの手順を参考にするための資料が失われていることが多く、過程を残すことが重要とされてきている現状があった。[4][5]

本システムはプロジェクトの進行と、軌跡を残すことを目的としており、こうした事例に対して有効に活用できる可能性がある。目的別にプロジェクトを俯瞰できるようになったことで、過去のプロジェクト状況を多面的に閲覧することができるようになり、後発のアートプロジェクトのための軌跡の体系化に応用できることが期待できる。

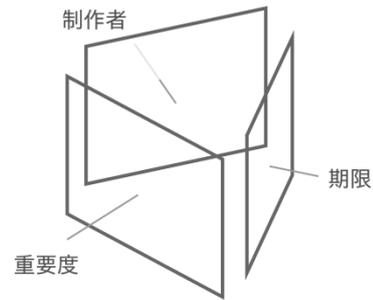


図 3. 多面的切り口からのプロジェクトの可視化する

## 4. 機能概要

3で述べた目的別可視化機能についての概要を以下に示す。

- i) タスクを表すボックスに、付加情報を持たせる。(図 4)
- ii) 付加情報は、ユーザーによるタスクの編集、移動、優先度の変更、時間の推移などによって自動的に更新される。
- iii) ユーザーは、日時・進行度・期限・優先順位・制作者・コストなど、付加情報から参考にした閲覧基準を選択し、これらをキーとして保存する。
- iv) 更新された情報をもとに、ユーザーが選択しているキーを縦軸の指標にしてタスクをソートし、ビューの更新を行う(タスクの再配置)。この工程はビューの更新のみを行い、付加情報の変更は行わない。(図 5)

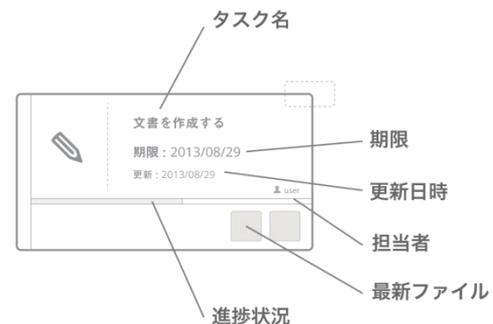


図 4. タスクに情報を付加



図 5. 選択されたキーをもとに再配置

## 5. システムの実装

システムはこれまでと同様、Node.js を用いて Web アプリケーションの形態で実装を行う。

今回の目的別可視化機能では、クライアントサイドにて動的にビューの変更と構築を行う必要がある。また、更新情報は多岐にわたるため、それに伴うビューの更新は非常に頻繁に行われることとなる。こうしたことから、仮想 Dom[8][9]を用いたライブラリによるビューの更新が設計に効果的に利用できる可能性があり、導入の検討を行っている。

また、モジュール管理ライブラリを利用することで、フィルタリングや移動のアルゴリズムに、サーバー側とクライアント側で同一のコードを再利用することが可能となり、機能の重複が多い本アプリケーションでは開発を効率的に進めることが可能となった。

一方で、プロジェクトを円滑に進めるための管理ソフトウェアとしては、多人数の同時操作でも情報がシームレスに同期され、いつどの端末で見ても最新の情報が閲覧できることが重要視される。前回の報告では、更新処理に定期的にポーリングを行う Ajax を用いた処理から WebSocket[a]通信を利用した処理に変更することによって、サーバー負荷の少ない更新処理を実現することが可能となったが、今回の実装においても、上記処理の利点は大きいと同様の更新処理を採用した。

## 6. 今後の展望

以上の機能により、タスクチャート手法を用いた、プロジェクトを目的別の切り口から可視化する機能について述べた。今後も引き続き、具体的なプロジェクトでのテスト運用によってシステムの妥当性について検討を図り、早期の実現を目指してフィードバックを踏まえた改善を行っていく予定である。

## 参考文献

- 1) 西村克己(2000, 7). 『よくわかるプロジェクトマネジメント: 入門マネジメント & ストラテジー』, 日本実業出版社
- 2) 内館町子(2008, 4). 『ひと目でわかる Microsoft Office Project 2007』, 日経 BP ソフトプレス
- 3) Project Management Institute, inc.(2004). A Guide to the Project Management Body of Knowledge (PMBOK Guide)
- 4) 帆足亜紀(2013, 3). 『アートプロジェクト運営ガイドライン—運用版』, 公益財団法人東京都歴史文化財団 東京文化発信プロジェクト室
- 5) アート・アーカイブ・キット | Tokyo Art Research Lab : [http://www.tarl.jp/cat\\_output/cat\\_output\\_record/10411.html](http://www.tarl.jp/cat_output/cat_output_record/10411.html)
- 6) Stoyan Stefanov(2010, 9) “JavaScript Patterns Build Better Applications with Coding and Design Patterns” O'Reilly Media
- 7) WebSockets | MDN : <https://developer.mozilla.org/ja/docs/WebSockets>
- 8) Psaila, G. (2008, 9). Virtual DOM: An Efficient Virtual Memory Representation for Large XML Documents. In Database and Expert

---

a WebSocket は、ブラウザとサーバー間で双方向通信を可能にする通信プロトコル技術の一つで、応答のためにサーバーへリクエストを送ることなく、イベントドリブンなレスポンスを受信できる

Systems Application, 2008. DEXA'08. 19th International Workshop on (pp. 233-237). IEEE.

9) Matt-Esch/virtual-dom : <https://github.com/Matt-Esch/virtual-dom>

10) Microsoft Project :

<http://www.microsoft.com/japan/project/default.aspx>

11) dotProject : <http://www.dotproject.net/>

12) 5pm : <http://www.5pmweb.com/>

13) Backlog : <http://www.backlog.jp/>

14) Brabio! : <http://brabio.jp/>

15) Zoho Project : <http://www.zoho.jp/projects/>

16) Google Groups : <https://groups.google.com/>

17) ChatWork : <http://www.chatwork.com/>

18) Fluent : <http://fluent.io/>

19) サイボウズ Live! : <https://live.cybozu.co.jp/>