

2208

体系的設計プロセスモデルに基づいたオープンな協調設計環境の研究* (日本機械学会指定テンプレートファイル利用について)

中沢 俊彦*¹

A Study of open source design environment based on a systematic design process model (About the Use of the JSME Specification Template File)

Toshihiko NAKAZAWA*¹

*¹ Modeling Technologies Co. Ltd.
Shirokane 3-23-6, Minato-ku, Tokyo, 108-0072 Japan

As a form of design activities to create innovative products, collaborative design by unspecified number of designers and engineers in public work place (open source design) has attracted attention. In order to execute an open source design, services and tools to support communications and collaborations among participating designers and engineers are indispensable. In this paper, requirements of services and tools to support open source design activities are clarified, and a concept of service is proposed based on the identified requirements.

Key Words : Collaborative design, Open source design, the systemetic rdc model,

1. 結 言

これまで、また今日行われている設計活動のほとんどは企業内の設計室においてその企業に雇用された設計者(本稿では、デザイナーと設計技術者を区別なく設計者と呼ぶ)、契約設計者、部品供給企業などから派遣された設計者など限定されたメンバーにより実施されている。最近では、ハッカソンと呼ばれる不特定のソフトウェア開発者が一時的に一箇所に集まりソフトウェアの開発を競うイベントや、スタンフォード大学の D.School に代表されるようなワーキングスペースに異なるスキルやバックグラウンドを持つ設計者が集まり設計活動を行うオープンな協調設計の形態が注目されている。エリック・レイモンドはその著書「伽藍とバザール」の中で前者のような組織内での閉じた設計活動の形態を「大聖堂」開発モデルと呼び、後者のような不特定多数の設計者がプロジェクト単位で集まり設計活動を行う設計形態を「バザール」方式と呼んだ⁽¹⁾。バザール方式でのソフトウェア開発で顕著な成果を残した例として、Linux カーネル、Firefox web ブラウザー、Apache OpenOffice などのオープンソースソフトウェアの開発が有名である⁽²⁾。ソフトウェア以外にも、ここ数年、大容量のイメージファイルやテキストファイルをインターネット上で共有することのできるソーシャルネットワークサービス(以下、SNS)やファイルの共有が可能なクラウドストレージサービスなどの登場によりインターネットを介した情報交換コストが飛躍的に減少しており、異なる組織に所属しかつ離れた場所に居住する複数の設計者がこれらのサービスを利用して、協調設計活動を試みる例が散見されるようになってきている。本稿ではこのような複数の設計者が、所属する組織の壁を越え、公開された場で協力して設計活動を行う設計形態をオープンデザインと呼ぶことにする。

オープンデザインでは地理的に離れた設計者間での情報共有や情報交換を実現する仕組みが欠かせない。しかしながら現在、オープンデザインの試みで利用されている e メール、チャット、ファイル交換や SNS といった仕組みは設計活動を支援する目的で提供されているものではないため、これらの仕組みを使って協調して設計活動を行うことには難しさも存在する。そこで本稿では、インターネットを介して不特定多数の設計者によるオープ

*¹ 一般、モデリングテクノロジー (株) (〒108-0072 東京都港区白金 3-23-6)
E-mail: nakazawa@designearth.com

ンデザイン環境の要件を検討し、オープンデザインを支援するための新しいサービスのコンセプトを提案する。

2. 協調設計環境の要件

2.1 ソフトウェア協調開発環境の機能要件

ソフトウェアの協調開発の支援環境としてよく知られる GitHub はソフトウェアの実装作業を複数のソフトウェア開発者に分散し並行的に作業を行うための環境を提供している。ソフトウェア開発者は GitHub 上に公開されている開発中のマスターコードを自身のブランチ開発環境にコピー（Fork と呼ばれる）し、機能の追加や修正などを行う。作業が終了すると作業を行った開発者は作成または編集したコードのマスターコードへの取り込み依頼（Pull Request と呼ばれる）を行い、別の開発者のレビューを経てマスターコードと統合される⁽³⁾。このように GitHub ではソフトウェアの協調開発をマスターフローとブランチフローの2つの作業の流れに分けることにより効果的に実現しており、多くのソフトウェア開発者に支持されている。

設計活動は一般的には企画、要件定義、設計、試作、テストといった一連の過程により構成されているが、これらの過程の中で、GitHub は主に試作とテスト過程での作業成果物（この場合、ソフトウェアのコード）の共有と交換を支援していると言える。GitHub や類似のホスティングサービスやツールで提供されている機能と利用方法を観察することにより、協調設計環境に必要と考えられる以下のような機能要件を見出すことができる。

- ① 課題（Issue）管理
- ② タスク（必要な作業）管理
- ③ レビュー前の作業成果物の一時保管
- ④ 成果物の統合
- ⑤ 設計成果物の競合管理
- ⑥ 成果物の履歴管理
- ⑦ 成果物の共有

2.2 ソフトウェア以外でのオープンデザイン実現の要件

ソフトウェアの協調開発を実行する上では、GitHub が注力していない目標管理や要求管理ならびにそれらを定義するための議論や情報共有も重要であるが、これらの作業はEメール、インスタントメッセージ、最近では Slack などのタイムライン機能を持った情報共有サービスを利用して行われている。また、進捗管理にはタスク管理サービスである Trello、バグ管理の仕組みとして様々なバグトラッキングシステムなどが知られている。

ソフトウェア以外の事物を対象にした開発ではGitHubのような中心となるツールやサービスが存在しないことから、Slack や Trello に加え、データ共有に Google drive などのクラウドストレージサービスを利用するなど、必要な機能を獲得するために複数のサービスやツールを組み合わせたオープンデザイン環境の模索が続いている。

それらソフトウェア以外の事物のオープンデザインの試み（例えば、rloop⁽⁴⁾）を観察することにより、前節で述べた既存のソフトウェア開発環境が満たしている機能要件に加えて、以下のような要件が存在することが理解できる。

- ⑧ 設計意図や要求の共有
- ⑨ 合意形成のためのコミュニケーション環境
- ⑩ 設計レビューやテストなどの検証項目と検証結果の共有
- ⑪ 多様なファイル形式のデータの共有

2.3 設計プロセスを動的に共有するための要件

オープンデザインを実現するための環境構築では設計プロセスそのものと生成される情報特有の性質にも配慮する必要がある。かねてより設計プロセスは試行錯誤を特徴とする循環型（Iterative）なプロセスであると言われてきた。この特徴がフローチャートなどに代表される逐次進行型のプロセスを前提とした記述手法によって設計プロセスを記録することを困難にしている。また、設計作業の進行に伴って生成される設計情報も修正と上書き

が繰り返されながら徐々に情報量が増え完成度が増していく。そこで、オープンデザインを支援するサービスやツールはこれらの設計プロセスの特性に由来する以下の要件への適合性も備えていなければならない。

- ⑫ 試行錯誤的なプロセスへの適合
- ⑬ インクリメンタルな（少しずつ増加する）情報の共有と管理

3. オープンデザイン支援サービスの実現手段

設計プロセスの特徴である試行錯誤的な過程から生成される設計情報を予め決められたステップに従って画面が遷移するソフトウェアインターフェース（以下、UI）を介して記録することは困難である。なぜなら、利用者は創発的な思考過程から生成される情報をソフトウェアが要求する入力順序に再構成しなければならなくなり、設計作業そのものに対する思考の集中が大きく阻害されてしまうからである。どのような情報がどのような順序で創発されても記録ができる柔軟な UI が求められるが、その実現手段として情報を生成順に保存するタイムライン形式の UI はそのような人間の気まぐれな思考過程の記録に適していると考えられる。生成される全ての設計情報をタイムライン上に保存時刻を付与しながら時系列に記録してしまえば、インクリメンタルな情報についてもデータの上書きをすることなく記録することができ、その新旧も明らかになる。また、タイムライン形式の UI は自由書式のコメント記録機能との親和性も良く、情報を共有している設計者間のコミュニケーションの場としての性格を併存させることも容易である。

自由書式のコメントと時系列のファイル保存機能を備えたタイムライン形式の情報共有サービスやツールは既に複数存在しているものの、それらはオープンデザイン活動の支援を目的とするものではない。今のところ、それら既存サービスやツールと他の手段を併用したオープンデザイン環境の模索が続いていることは既に述べた通りであるが、合理的な意思決定の積み重ねによって実現可能な設計情報を完成させることを目指す設計過程を自由書式のタイムライン型 UI 上で成立させることは難しい。それ故、上で述べた様々な要件を満たした設計活動に特化したオープンデザイン支援サービスやツールが求められている。

設計活動の支援を行うためのサービスやツールに対する機能要件は目標設定、要求定義、設計、検証といった設計過程特有の過程とそれらの過程から生成される設計情報の性質に沿った設計情報の記録機能の必要性を示唆している。そこで、筆者は設計過程の記録手法である体系的 RDC モデル⁽⁵⁾の設計作業要素によって分類される設計情報の記録機能をタイムライン形式の UI 上に構築することで、柔軟性と設計情報特有の形式の双方に対応したオープンデザイン支援サービスを実現することにした。

体系的 RDC モデルは設計過程を概念設計と詳細設計に分解し、それぞれにおいて要求定義と定義された要求の検証、設計定義と定義された設計情報の検証作業

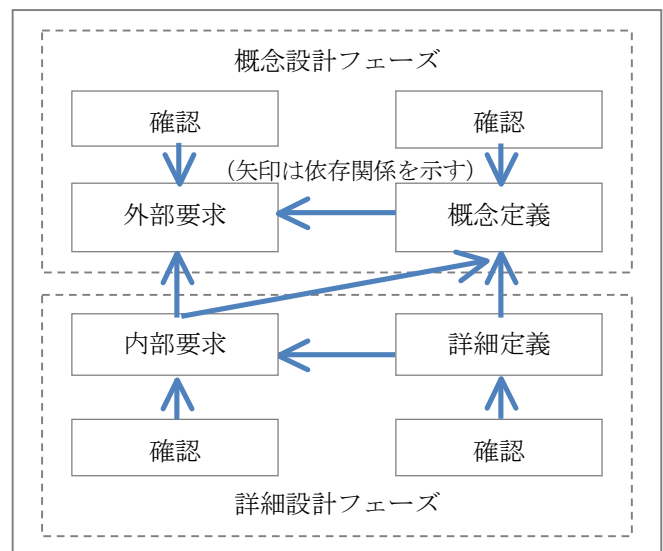


Figure 1: The systematic RDC model

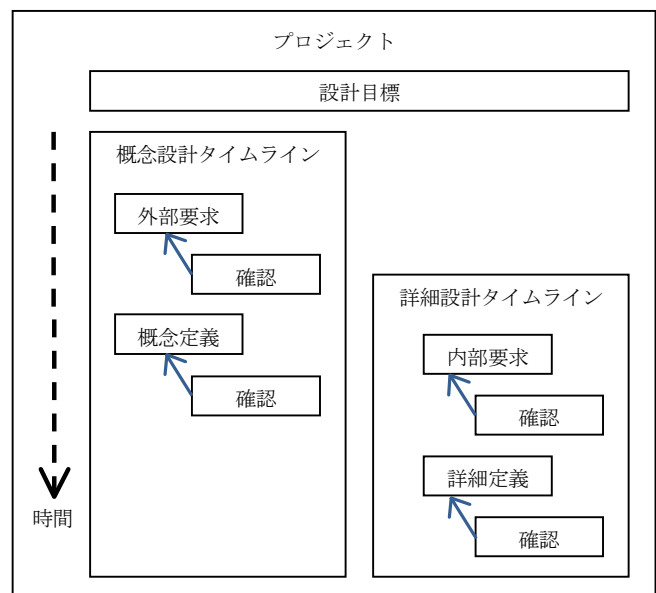


Figure 2: RDC model elements on the time line

を分析し、さらにそれらの関連を記述する静的な設計過程分析手法である（図1）。本手法は、過去に実施された設計過程を記述する分析手法として提案され、過去の設計活動の中で行われた複数の意思決定作業とそれら意思決定作業間の依存関係を記録することに主眼が置かれていた。

筆者は本手法を発展させ、不特定多数の設計者により現在進行している設計プロジェクトにおいて、生成される設計情報を併行する2つのタイムライン（概念設計タイムラインと詳細設計タイムライン）上に動的に記録し共有するオープンデザイン支援サービスのコンセプトを作成した（図2）。

3・1 設計目標の宣言と概念設計タイムラインの生成

任意の設計過程は何らかの課題もしくは設計対象物に対する設計意図を持って開始されるが、オープンデザイン活動に参加する設計者間でこの課題認識や設計意図が明確に共有されないと、直接の会話などで認識の整合を取り難いコミュニケーション環境において円滑な協調設計活動を行うことは望めない。そこで、本オープンデザイン支援サービスでは、オープンデザインプロジェクトを開始する際の必要条件として、設計意図を明示するとともに、設計対象の現在と実現したい未来の姿の明確化をプロジェクト作成者に求める入力画面を用意することにした（図3）。

ここに記述された内容を概念設計タイムラインと詳細設計タイムラインのヘッダに表示することによって、プロジェクト作成者の設計目標が共有され、この目標に対する設計解が得られる方向へ議論が収束し、的の絞られた協調設計活動が展開されることを期待した。

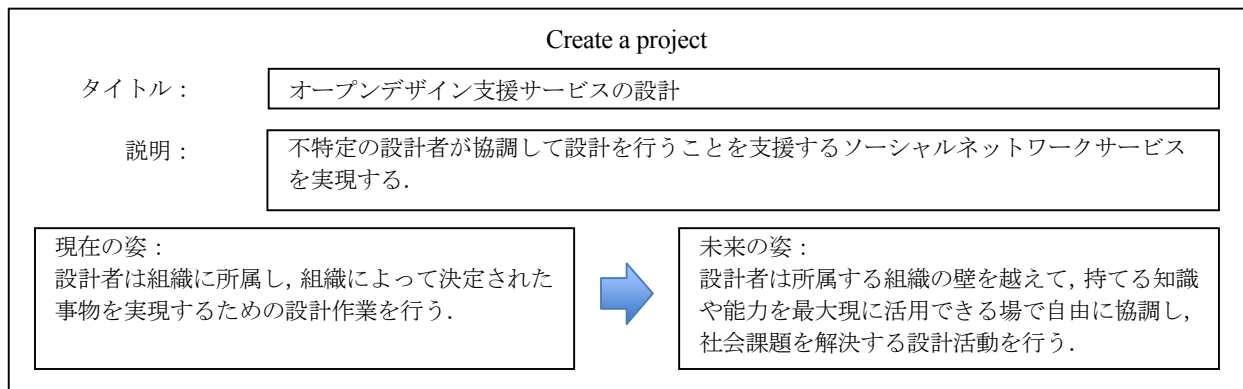


Figure 3: Description of a design project when creating the concept design time line

3・2 外部要求情報の記録と検証

設計目標が宣言されると概念設計タイムラインが生成され、協調設計者はこのタイムライン上において設計情報の記録や交換ができるようになる。

概念設計タイムライン上で設計情報の記録や交換が行われる中で、協調設計者の誰かによって設計対象事物が適合しなければならない（例えば、ユーザーのニーズや法規、利用環境など設計対象事物の外に存在する）要求や条件が認識されると、その協調設計者は認識された情報を外部要求情報として概念設計タイムラインへ投稿することができる（図4）。投稿はタイムライン上にテキスト形式で打ち込むこともできるし、任意のフォーマットのデータファイルをアップロードすることでも行うことができる。この結果、タイムライン上に体系的RDCモデルが定める外部要求情報の属性を持つ設計情報が保存され公開される。

投稿された外部要求情報は他の協調設計者によるピアレビューを受けることができ、その確認結果は対象となった外部要求情報と関連付けられる。また投稿された外部要求情報に対して他の協調設計者が賛同を表すこともでき、情報の確からしさを暗示する一つと指標にすることができるようにした。

設計上は全ての外部要求の洗い出しが設計過程の上流で行われることが望ましいが、実際は設計が進む中で徐々に明らかになってくることもある。タイムライン方式による記録方法を採用したことによって、外部要求情報は設計活動のどの段階でも認識された時点で投稿することができる。

3・3 概念定義情報の記録と検証

概念定義は概念設計段階において起案される設計コンセプトである。概念定義情報をタイムライン上に投稿する際には、概念設計タイムラインに投稿されている外部要求情報をリスト化して表示する機能を実装した。投稿しようとしている概念定義情報を起案した際に考慮した外部要求情報をこのリストの中から選択することにより、概念定義と外部要求情報の間の参照関係が記録できるようにした。

設計には既に存在する人工事物に改良を施す流用設計と、存在しない人工事物を一から創造する新規設計がある。通常、我々が実施している設計活動は何らかの既存設計を基にしている流用設計であることが多いが、そのような流用設計の場合には外部要求が暗黙知化されていたり、標準化された構造やパラメータに隠蔽されているなど表面的に認識されていないことがある。そのような場合、設計目標から直感的、経験的に直接、概念定義が導かれることがあるが、タイムライン方式を採用したことにより、このような暗黙的な知識から直感的に生成される概念定義情報も思い付いた時点で記録することができる。

概念定義情報が記録された後に外部要求情報が認識され投稿された場合、投稿済みの概念定義情報に後に投稿された外部要求情報を関連付けることもできるようにした。

外部要求情報と同様に投稿された概念定義情報は他の設計者によるピアレビューを受けることができ、その確認結果が付与される。さらに他の協調設計者からの賛同を記録する機能は投稿された概念定義情報の確からしさの指標となる。

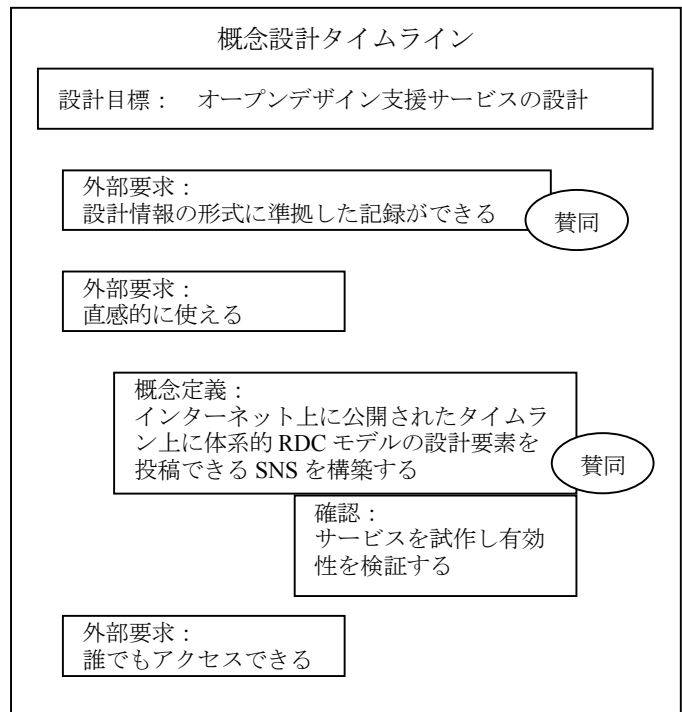


Figure 4: The concept design time line

3・4 詳細定義タイムラインの生成と内部要求情報の記録と検証

任意の協調設計者は、概念設計タイムライン上に投稿された概念定義情報が詳細設計を行うに値すると判断した場合、その概念定義の詳細設計情報を記録し交換するための詳細設計タイムラインを生成することができる(図1)。詳細設計タイムラインは概念設計タイムラインと基本的な機能は同じであるが、詳細設計タイムラインには体系的 RDC モデルが定義する設計者自身が外部要求を考慮しながら設定する設計対象事物の機能と非機能目標である内部要求情報(一般的には仕様と呼ばれる)を投稿することができる。内部要求は概念設計タイムラインに記録されている外部要求情報と概念定義情報の関係から推論することができるため、その論理展開を補助する UI は有用である(図5)。投稿された内部要求情報は他の協調設計者からのピアレビューを受けることができる。

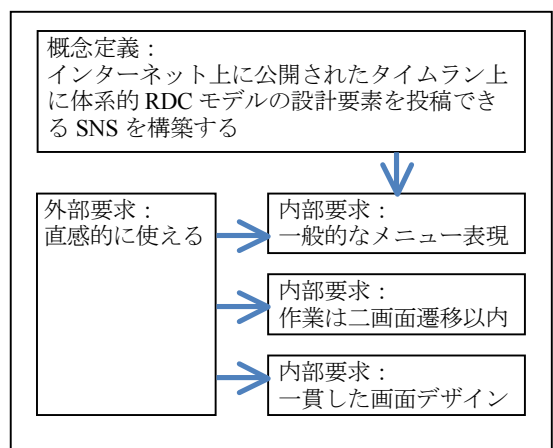


Figure 5: Internal requirement finding tool

3・5 詳細定義情報の記録と検証

詳細定義情報は設計パラメータの集まりであり、その多くは製造準備過程に渡すための設計情報である。流用設計では詳細定義の内容が標準化されていたり、経験的に決まっていることがある。そのような場合には、往往にして内部要求は詳細定義の内容に内包されていて設計過程の中で明示的に検討されないため、詳細定義タイムラインは概念定義情報から詳細定義情報が直接発想されたときにも生成することができるようにした。

一つ概念定義は複数の詳細定義に分解され検討されることもあるため、詳細設計タイムラインにはある概念定義情報に関連付けられた複数の詳細定義情報を投稿することができる。また、概念定義情報と外部要求情報の関連付け UI と同様に、詳細定義情報を作成する際には既にタイムライン上に投稿されている内部要求情報との関連付けができるようにした。投稿された詳細定義情報は他の協調設計者からのピアレビューやテストを受けることができ、その結果は確認情報として詳細設計タイムライン上に記録することができる。

3・6 体系的 RDC モデルに基づいた外部要求と概念定義情報間関係の記録

以上の機能により、タイムラインという時系列の記録 UI 上に、体系的 RDC モデルの設計作業要素に沿って入力される設計情報と情報間の関連を記録する仕組みを構築することができる。

タイムラインには自由書式のコメントや質問も投稿することができるようにすれば、設計者間の自由なコミュニケーションが体系的 RDC モデルの形式的な文章の行間を埋めることができ、協調設計者間の意思疎通に役立つであろう。逆に、協調設計者間で自由書式による頻繁なやりとりが行われた場合には、雑然とした情報の中に有効な設計情報が埋もれてしまうことが懸念されるが、体系的 RDC モデルの属性がついた投稿のみを区別することにより、有用な設計情報のみを抽出することができる。

3・7 詳細定義間の関連

あるプロジェクト内で生成された詳細定義情報は同じプロジェクト内の他の詳細定義情報と依存関係を持っている。この依存関係をデザインストラクチャーマトリックス (DSM) 上に整理する機能を持たせることによって、詳細定義情報をネットワーク化した統合設計情報を持つことができる。このネットワークを辿ることにより個々の詳細定義情報間の適合性を検証することも可能になるであろう。

4. 結 語

現在、本稿で述べたサービス機能の実装が進められており、次の段階として複数の協調設計者のオープンデザイン活動での試用による有効性の検証に移っていく。

GitHub ではソフトウェアの協調開発を実現しているが、GitHub の特徴ともなっている開発したコードをメインコードへ統合する仕組み (Fork と Pull request) が必要かどうかは、今後の検討課題である。また、本サービスでは強制のない協調設計環境を目指したため、強制を伴うタスク管理や日程管理などのプロジェクト管理機能は検討していないが、本サービスを利用し、納期の伴う設計作業を行うような局面では必要な機能になろう。

類似プロジェクトを検索し、類似プロジェクトがあるのであればそちらへの参加を促し、同様のプロジェクトが複数作られることによるリソースの分散を防ぐ思想 (Not invented here approach) の具現化はオープンデザイン支援環境として必要不可欠であろう。これらの機能をどのように実現していくかは本サービスの今後の重要な設計課題である。

文 献

- (1) エリック・レイモンド, “伽藍とバザール”, USP 研究所, (2010)
- (2) https://en.wikipedia.org/wiki/Open-source_software_development
- (3) GitHub Flow : <http://scottchacon.com/2011/08/31/github-flow.html>
- (4) <http://rloop.org>
- (5) 中沢俊彦, 増田宏, “設計品質管理のための設計プロセス計画と検証及び要求開発のための一手法に関する研究”, Journal of the Japanese Society for Quality Control vol. 41, No. 2 (2011), pp. 81-93