

システム仕様書テンプレート_ver1.0

参考資料

IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specification

システムインストルメンツ仕様書

旧 Labwiki の先生のページ

1. 目次

2. はじめに

2.1. 文書の目的

この文書はコンテンツ制作支援システムの仕様書である。

2.2. システムについての簡単な説明

BPMN（ビジネスプロセス記述言語）を発展させたタスクユニット記述法を新たに開発して、知識やスキルなどのコンテンツ制作に必要な情報のモデル化・記述が可能な web ベースの複数人制作支援システムを開発する。タスクユニットの達成履歴の視覚化により、成長プロセスをユーザにフィードバックして、能力特性の自己認識を容易にする。

2.2.1. システムの名称

候補考える

~~・Concreat（コンクリート）（Contents Creation Support）~~

- ・コスモツリー（ユグドラシル が意味的にはいいと思うが、名前が一般的すぎる。コスモ（宇宙）＋ツリー（木）でユグドラシルと同じ意味）

2.2.2. システムができること

知識やスキルなどのコンテンツ制作に必要な情報のモデル化・記述が可能な web ベースの複数人制作支援システム。タスクユニットの達成履歴の視覚化により、成長プロセスをユーザにフィードバックして、能力特性の自己認識を容易にする。

2.2.3. 予想させる利点・目標

- ・アマチュアでも制作プロセスを理解して効率的に分担や作業を行うことを可能にする

- ・能力特性の自己認識により自発的な学習を促す

2.3. 用語定義

3. 開発目的・コンセプト

3.1. 開発背景

3.1.1. 問題設定

コンテンツ制作は、多様な分野の知識やスキルが必要となることに加えて、制作プロセスが不透明であるため、アマチュアでは制作管理や学習が容易ではないことが問題として挙げられるが、従来研究では制作プロセスの分析や管理を行う基本手法の検討が行われて来なかった。

3.1.2. システムの目的

本研究では制作プロセスの構造化と視覚化により、アマチュアでも制作プロセスを理解して効率的に分担や作業を行うことを可能にするとともに、能力特性の自己認識により自発的な学習を促進する基本システムを提案する。

「アマチュアが一般公開可能なコンテンツを制作するまでのプロセスの支援」を目的とする。

3.1.3. 他のシステムとの差異・比較

作業計画やワークフローを記述するための記法として、UML[]と BPMN[]が良く利用されている。しかし、これらの記述法では細かな作業の記述をする際に可読性が下がる問題があったり、制作者のスキルなどのコンテンツ制作の計画の際に必要な情報の記述ができなかったりと、コンテンツ制作のためのワークフロー記述法として適切ではない。

また、作業管理をシステムが行うソフトも多数存在している。Redmine[]は Web ブラウザ上でタスク管理が可能で、進捗状況やタスクをチケットで管理することで、作業管理を行っている。作業支援に関する研究としては、管理システムによる自動化を用いてユーザの負担軽減を目的としたアプローチがある。大向らは、人間関係ネットワークに基づく情報フィルタリングを用いることで、複数プロジェクトに関わる人間のスケジュールをプライバシーに考慮して公開する、スケジュール管理システムを提案した[]。複数プロジェクトに係わる場合の時間管理をより効率的に行うことで、ユーザの支援を行っている。ワークフローの動的再計画による研究では、垂水らはマルチエージェントシステムを用いることでシステムが動的に時間管理や再計画を行うシステムを提案している[]。しかし、これらの既存の手法では作業中の作業管理はサポートしているが、作業前の計画段階での支援や、作業後のフィードバックが十分になされていない問題がある。

ビジネスプロセスなどの一般的な場面ではなく、特定の状況での使用に焦点を当ててプ

プロセスを記述する方法を開発する研究も行われている。医療分野では、下野らは通常のビジネスプロセスと異なる特殊性を持っていることについて言及している[1]。やり直しがきかない、患者ごとにプロセスを変化させる必要があるなどが言及されている。医療分野における特殊性を考慮した医療プロセス記述法として、Avrumin らからは Little-JIL という手法が提案されており、モデル化・解析が行われている[1]。標準的なプロセス実行だけでなく、様々な例外やメッセージにリアクティブに対応する処理の記述が可能なが大きな特徴である。このように特定の用途に応じて必要となる要素を分析し、適切なプロセス記述法を提案することは重要である。

3) システム開発法の研究（開発プロセス、ドキュメンテーション、要件定義・システムの記述法）

4) コンテンツ制作支援の研究（ないかもしれない、事例ベースの何か）

5) グループウェアの研究（できれば制作支援に近いところ、開発におけるコミュニケーションの研究も含む）

3.2. 対象ユーザ

大学生などのアマチュアによるコンテンツ制作の場面を想定する。教科書どおりの制作ならなんとかできるレベル。

4. システム概要

4.1. システム要件

4.1.1. ワークフロー記述法の構造化

- ・コンテンツ制作プロセスに必要な情報が記述できる
 - ・制作に必要な制作対象モデルなど
 - ・コンテンツ制作では一般的な業務プロセス記述法だけでは不十分であるため、それを補えるように必要な情報を記述できるようにする。詳細については後述。
- ・視認性を損なわない
 - ・UML や BPMN ではタスクの種類が増えたり、作業が増えると視認性が下がる問題がある。タスクユニットによる粒度の違いによる表示をすることで魚眼レンズのような詳細表示と俯瞰表示ができるようにする
- ・制作者が複数人いても対応できる

コンテンツ制作はしばしば複数人で行われる。このような場合でも作業管理ができる必要がある。

- ・ワークフローに流動性がある

他人が作ったワークフローや自分が作ったワークフローでも頻繁に変更をしたい場合がある。その時、全体の流れをごっそり変える場合でも、一部だけ変えたい場合でも破綻しないようにする。

4.1.2. 制作管理システム

- ・制作者の負荷をできるだけなくす

導入や運営が面倒だと意味がない。仕様書を穴埋め方式にする、適用限界問題に対して作業データはできるだけ自動的に取得する、入力するデータを少なくするなどが必要。

- ・進捗管理ができる

進行中のタスクや締め切りまでの状況が一目で分かるようにする。このとき、情報可視化はワークフローを用いることで、複雑な管理にならないようにする。

- ・達成履歴などの学習に必要なデータを視覚的にフィードバックができる

作業履歴をただ提示するだけでは意味が無い。理解しやすいような分かりやすいフィードバックとして、グラフィカルなフィードバックが有効と考えられる。

4.2. システム機能一覧

4.2.1. ワークフロー記述法の構造化

- ・BPMN ベースなので、BPMN ができることは基本的に全て
- ・タスクユニットによるタスクの階層構造化
- ・制作に必要な情報のモデル化

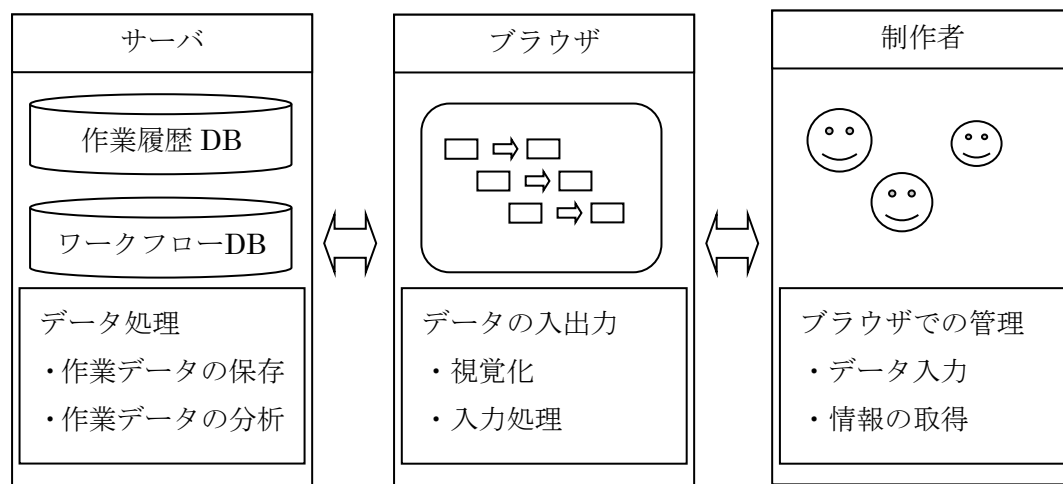
4.2.2. 制作管理システム

- ・システム設計書や仕様書などの制作者間のコミュニケーションに必要なテンプレの用意
- ・典型的なワークフローの推薦による作業計画補助機能
- ・ワークフロー作成機能
- ・制作中のワークフロー管理機能
- ・進捗管理
- ・作業状況の記録
- ・達成履歴や能力特性などのデータの視覚化

4.3. システム構成

本システムはサーバとの通信を頻繁に行うことや PC での使用を想定し, 気軽に使いやすい Web ブラウザ上で動作する Web アプリとして実装する.

4.3.1. 全体の構成図



4.3.1.1. 制作者

制作者はブラウザを介して制作管理を行う. 作業データの入力やワークフローの作成はブラウザ上で行う.

4.3.1.2. ブラウザ

制作者の入力処理やデータの視覚化を行う. ワークフローを作成する際などにブラウザ上だけで完結する. 視覚化は **JavaScript** でグラフ表示やガントチャート表示を用いてグラフィカルに分かりやすく表示する.

4.3.1.3. サーバ

サーバはデータ処理を行う. 入力された作業データの保存を行い, そのデータを元に進捗状況や制作プロセスに関する分析を行い, データベースに格納する. ブラウザには分析済みのデータを送信する.

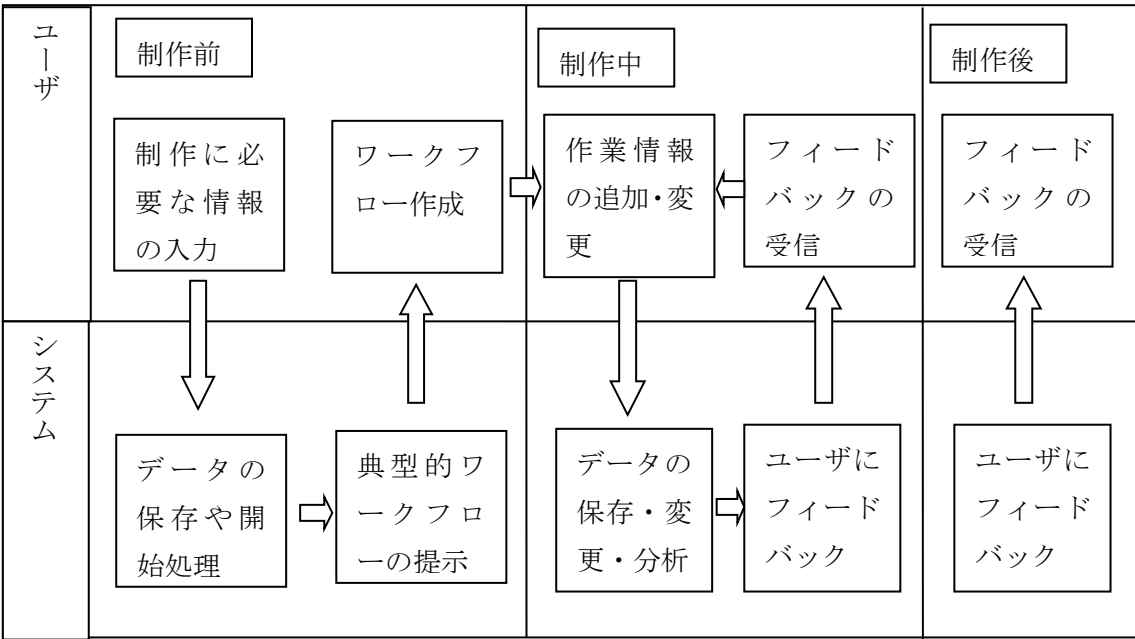
4.3.2. ユーザインターフェース

ユーザとシステムとのインターフェースについて記述する. ユーザとシステムが行うやり取りについて概略を書く. 図を推奨.

4.3.3. ソフトウェアインターフェース

ソフトウェア内部での構成について概略を記述する。図を推奨。

4.4. 利用シナリオ



4.4.1. 制作前

ユーザは制作に必要な制作者モデルや制作対象モデルなどの制作に必要な情報を入力する。入力を受け取ったシステムは、データを保存し制作の開始処理を行う。開始処理後、入力データから近い典型的なワークフローをユーザに提示することで、ユーザのワークフロー作成の支援をする。ユーザは提示結果を参考にしてワークフローを作成する。

4.4.2. 制作中

制作中はユーザが適宜仕様書やタスク処理状況などのデータをシステムに入力をする。システムは受け取ったデータを保存し、進捗状況や制作プロセスを分析する。デッドラインを超えそうなタスクや制作に問題を発見した場合はユーザにフィードバックを行う。（ワークフローのタスクの色を変えるなど）それを見たユーザはフィードバック結果を元に適宜作業の変更や期日の変更を行う。

4.4.3. 制作後

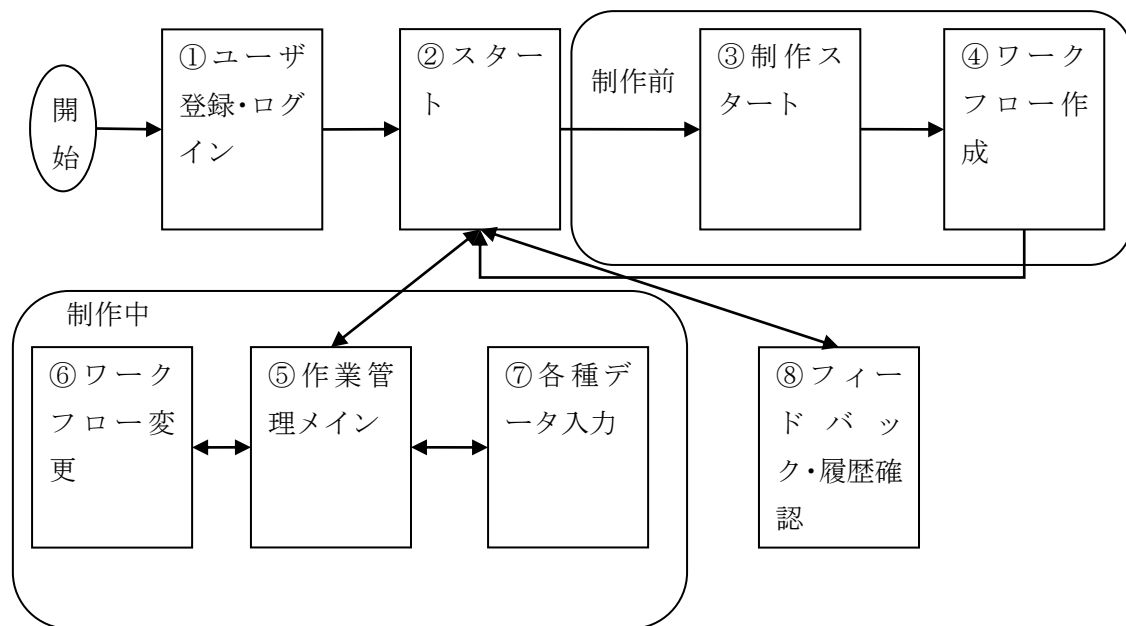
システムは制作中や制作前に集めた情報を元に制作プロセスや成長プロセスを視覚的にフィードバックする。その結果を元にユーザは反省や学習を行う。

5. システム詳細

5.1. 開発環境

レンタルサーバを借りてクライアント側を JavaScript と html, サーバ側を PHP と mySQL での実装を予定している。(マッテオさんは node.js をおすすめしているので、そちらも見てみる)

5.2. 画面遷移図



① ユーザ登録・ログイン画面

システムを使用する前にユーザ登録を行う。すでに登録済みの場合はログインのみ。初回ログイン時にはプロフィールなどの情報を入力させる。

② スタート画面

様々な画面に遷移するための画面。遷移するためのボタンが設置してある。

③ 制作スタート画面

制作を始める際は、この画面で必要事項の入力をする。

④ ワークフロー作成画面

制作開始前にワークフローの作成を行う。このとき、③で入力された制作対象モデル情報に基づき、システムが類似する典型的なワークフローの提示や、ユーザが他のワークフローの参照を行うことができるようにすることで、ワークフロー作成の手助けを行う。

⑤ 作業管理メイン画面

制作中に参照するメインとなる画面。現在のワークフローの状況や各種データ参照、ガントチャート表示などを行う。

⑥ ワークフロー変更画面

制作中にワークフローに変更を行う際に遷移する画面。基本的な構成は④の画面と同じ。

⑦ 各種データ入力画面

制作中に新しくデータ入力や変更を行う際に移行する画面。仕様書のアップロードや制作対象情報などのワークフロー以外の情報変更関係を扱う。

⑧ フィードバック・履歴確認画面

制作プロセスに対するフィードバックを行う画面。作業プロセスのフィードバックや成長プロセスのフィードバックなどをグラフ等を用いてグラフィカルに行う。

5.3. システムのデザイン

まずは必要最低限の機能のみを実装し、デザインについてはその後考えたい。

5.4. システム構成詳細

5.4.1. システム全体の構成詳細

5.4.1.1. ワークフロー記述法の詳細

- BPMN ベースなので、BPMN ができることは全て
- タスクユニットによるタスクの階層構造化

階層化構造でタスクを表現することで

- ・視認性の向上
- ・流動性の向上

が見込まれる。

- 制作に必要な情報のモデル化

- ・制作者モデルの記述（スキル情報、過去の制作事例）
- ・制作対象モデルの記述（システム設計書）
- ・利用環境モデルの記述（いつ、どこで、誰に使うか）
- ・タスクモデルの記述（必要スキル、必要ツール、類似するタスク）
- ・制作者間でのコミュニケーションモデルの記述（仕様書、やり取りするデータをワークフロー上で記述、テンプレを用意）

5.4.1.2. 制作管理システムの詳細

- システム設計書や仕様書などの制作者間のコミュニケーションに必要なテンプレの用意

コミュニケーションの際に必要なであろうデータ集のテンプレを用意しておく。ユーザは穴埋めをするだけで円滑にユーザ同士のコミュニケーションが可能になる。穴埋め方式にして提示することで作業負荷の軽減を図る。

- 典型的なワークフローの推薦による作業計画補助機能

入力された制作対象モデルに似ている典型的なワークフローをデータベースから検索して提示する。ユーザは何もない状態よりも参考にできるので、より効率よく計画が立てやすくなる。他の人が制作したワークフローなども提示できるようにする。

推薦機能は類似度を計算して提示する。制作物に対して直接比較は難しいので、VR、ネットワークなどの属性を制作物に付加する。付加された属性（使用予定のツール、デバイス、人数、環境など）を元に類似するワークフローを提示する。

属性が新たに増える可能性も十分に考えられるため、階層的クラスタ分析によるクラスタリング処理を行う。

計算処理はそれなりにかかるが、そこは問題でないので、Ward 法を用いてクラスタリング処理を行う。分類感度がよい手法として有名。

また、推薦機能とは別に検索機能も実装する必要があると考えられる。

- ワークフロー作成機能

ブラウザ上でワークフローの作成操作が完結するように UI を設定する。

- 制作中のワークフロー管理機能

制作中にワークフローを見るだけで管理ができるようにする。各タスクの詳細情報や進捗状況、期日などがワークフローから分かるように色で分けたり、タスクユニットをクリックすることで詳細が見られたりするようにする。

俯瞰表示と詳細表示をタスクユニットを用いて表現することで、視認性を確保する。

- 進捗管理

基本はワークフロー上で進捗状況が分かるようにする。タスクユニット内の最小粒度のタスク（ミニマムタスク）の達成か未達成かの1か0で進捗を管理する。そうすることで制作者が進捗を細かに記述することを容易にする。

- 作業状況の記録

全ての作業状況はサーバで保存する。ワークフローのデータは図表表現のため、XML 形式で保存する。他の作業履歴などのデータは mySQL を用いてデータベースに保存する。

- 達成履歴や能力特性などのデータの視覚化

保存されたデータを分析し、視覚的に分かりやすいように提示する。タスクの達成履歴

や制作プロセスからユーザの成長プロセスや制作の問題点が分かるようにグラフなどを利用して提示する。

5.4.2. ユーザインターフェース

5.4.3. ソフトウェアインターフェース

5.5. ワークフローデータの扱い

ワークフローを図情報のまま扱うことは難しい。そのためシステム内部では XML を用いてワークフローの記述を行う。

5.6. 制作手順

想定している作業手順について記述する。制作手順の記述にはワークフローを用いる。

5.7. 動作に必要な仮定及び依存性

全ブラウザで動けばいいが、とりあえずは Google Chrome で完全に動くものを制作する。
(IE6 などとはとりあえず無視)

スマートフォン用の表示やタブレットでの表示も考慮した方がいいが、まずは動作優先で PC 版のブラウザでの対応を第一に考える。

5.8. 必要素材一覧

サーバ

6. 予備実験

6.1. 実験の目的

本実験前に試験的に使用し、システムの問題点や改善点を洗い出す。

6.2. 実施予定日

2月頃

6.3. 実施場所

研究室

6.4. 参加者

新 B4

6.5. 実験手法

リハビリゲームを制作してもらう。本システムを使用するグループと使用しないグループを比較して検証する。

6.5.1. 実験詳細

タスクユニットの達成履歴から、段階的な制作スキル水準、実行時間を計測するとともに、制作者間のコミュニケーションの頻度・内容を計測する。コンテンツの質を対象ユーザに評価して貰い、これらの関連性を明らかにする。

6.5.2. 分析手法

6.6. 実験準備スケジュール

1月末にプロトタイプが完成

7. 本実験

7.1. 実験の目的

アマチュアでも制作プロセスを理解して効率的に分担や作業を行うことが可能になるとともに、制作スキルの成長プロセスを明らかにすることができるかどうかを検証する。

7.2. 実施予定日

4月頃～

7.3. 実施場所

研究室

7.4. 参加者

EC 演習受講者

7.5. 実験手法

4～5月：こちらで用意した内容を使用する・使用しないグループで制作してもらい、基本的な制作スキルの獲得に有効かどうかを検証する。その後、自由課題を行うことで、制作スキルの成長プロセスを評価する。

7.5.1. 実験詳細

タスクユニットの達成履歴から、段階的な制作スキル水準、実行時間を計測するとともに、制作者間のコミュニケーションの頻度・内容を計測する。コンテンツの質を対象ユー

ザに評価して貰い，これらの関連性を明らかにする．

7.5.2. 分析手法

7.6. 実験準備スケジュール

3月までに完成版ができる．

8. 付録

適宜必要なら何かしら書く