

電子社会の仕様作成と社会規則の動的変更

岩井 淳

群馬大学 社会情報学部

概要 高度な安全性をもつ社会基盤情報システム構築のため、電子社会の仕様とモデルを先行して作成するという安心電子社会の研究指針がある。ここで仕様化の対象は情報システムとして実現可能な社会規則に限られるとされるが、システム実現上の難度の階層等の検討はまだ今後待つところが多い。本稿では、規則変更の規則の有無に着目し、安心電子社会研究を、仕様の動的変更手続きを含むクラスと含まないクラスに二分することを提案した。ソフトウェア開発過程の形式化技法であるプロセスプログラミングの理論を用いて、前者が技術的により難度の高い問題を含むことを示した。

Specification Description for an Electronic Society and Dynamic Changes in Social Rules

Atsushi IWAI

Faculty of Social & Information Studies, Gunma University

Abstract: An approach to establish a very safe social information infrastructure for a future electronic society is to describe preceding well-defined specifications and formal models for it. The principal approach is called safe e-society, in which the descriptive objects are basically expected to be the social rules that programmers can implement as information systems. However, it is not clear enough which type of rule is adequate to model and to be implemented. This paper proposes a framework in which a group of social rules that contains factors to modify other factors is divided from the other groups that do not contain such factors.

1. はじめに

近年、社会は複雑化と大規模化を続ける情報システムへの依存を増しつつある。正当性やアカウントビリティ確保の意味において、より安心な社会基盤情報システムの構築が期待されている。

安心な電子社会の実現のため、電子社会の仕様とモデルを先行して作成するという研究指針がある⁵⁾。(以下、安心電子社会研究と参照)。ここで電子社会とは、社会システムのうち情報システムで実現される部分であり、仕様化の対象は情報システムとして実現可能な社会規則に限る。しかし、対象としての社会規則の分析はまだ今後待つところが多い。システム実現上の難度の階層の検討等は少ない。

本稿は、規則変更の規則の有無に着目し、安心電子社会研究を仕様の動的変更手続きを含むクラスと含まないクラスに二分することを提案する。形式的仕様はモデルが満たすべき条件を与え

るので、形式化対象を選択する上での仕様の動的変更への着目は作成モデルにおける動的変更への着目と同等である(第2節参照)。

以下、第2節では安心電子社会研究の仕様化対象の分類基準として動的変更規則の有無を選択する着想について説明する。第3節では、仕様の動的変更のために要求される形式化技法が通常の場合と比較し異なる理由をプロセス形式化技法に関する既存の枠組みを応用して説明する。

2. 電子社会の仕様と動的変更の問題

2.1 電子社会の仕様作成

安心電子社会研究の構成とこの中での電子社会の仕様作成の位置づけをまず整理する。目的とする安心概念には複数の側面があるが、例えば、電子社会のサービスや機能に以下の特徴があることを重視する。1)誤りがないこと(正当性)、

2) 質問に対する説明可能性 (アカウンタビリティ), 3) 内包するデータへの不正アクセスや漏洩がないこと (セキュリティ), 4) 社会変化に応じ電子社会を進化させられること (進化性), 5) 故障高信頼情報基盤に支えられること (ディペンダブル基盤). 特に検証性と進化可能性への着目に特徴づけられる.

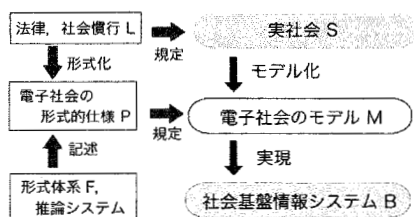


図 1. 仕様作成の位置 (文献 5) p.518. 一部省略.)

形式的仕様とモデルを先行して重視するのは、正当性の高いシステム開発に形式的手法が、また進化性を要求されるシステムではモデル駆動開発方法論が、有利とされることによる。ここで電子社会として実現すべきものは、法律や社会慣行 L により規定されていると考える。その内容が形式仕様 P として形式化される。P は電子社会仕様記述のための形式論理体系 F によって記述される。電子社会のモデル M は実社会 S を反映して構築され、それは形式仕様 P を満たすものでなくてはならない。モデル M は最終的に社会基盤情報システム B として実現される (図 1)。

2.2 仕様の動的変更の問題

法システム(L)が立法のための法を内包するように、社会システムの進化自体も社会システムの機能を用いた結果である。定められた作業の背景にある同作業の決定プロセスも考えに含めるならば、一般に社会は、「作業の決定」と「作業の実行」の両プロセスの循環と考えられる (図 2)。

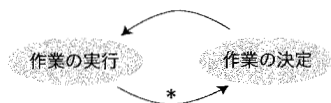


図 2. 作業の決定と実行の循環

作業の決定への回帰(*)は、地方自治体の行政サービスが天災により変更を余儀なくされるプロセス等として見られる。また、災害のない場合

も定期的に行政サービスは修正され続ける。これらの修正プロセスも支援できればより望ましい。

以上が、仕様化対象の分類基準として動的変更規則の有無に着目する理由である。メタレベルの規則の記述は、それ自体の修正等さらにメタレベルの記述を要しうるため (階層無限)、形式体系 (F) の準備が困難であると思われる。しかし、必ずしも不可能でない。以下、仕様の動的変更のために要求される形式化技法の論点を整理する。

3. 仕様の動的変更手続きの形式化

仕様の動的変更のために要求される形式化技法が通常の場合と比較し異なる点を既存のプロセス形式化技法の枠組みに依拠して説明する。

3.1 形式化技法としてのプロセスプログラミング

依拠するプロセス形式化技法としてプロセスプログラミングを採用する。

プロセスプログラミングはソフトウェア開発過程の形式化手法の 1 つであり、プログラム形式で記述することによる形式化を行う。プログラム形式の記述を可能にすることで、開発過程の (1) 明確な表現、(2) プログラムとしての保存と再利用、(3) 実行時における機械的な支援の 3 点を実現し、大規模で高品質なソフトウェアの効率的な開発を図る 6)。このアプローチは安心電子社会の目標 1) 正当性や 2) アカウンタビリティの確保等の点で整合性が高い。ソフトウェア開発を主な対象とする点では本稿と異なるが、この相違に基づく論点については最後に改めて整理する。

3.2 動的変更を含まないプロセス

最初に動的変更を含まない通常プロセスの形式化を整理する。プロセスプログラミングの枠組みのうち特に HFSP⁴⁾ に依拠する。これは、HFSP が動的変更の形式化を議論に含むことと、そのシンプルな木構造の履歴データが以下の説明をより明快にするのに資するためである。ただし以下は本質的に他のプロセスプログラミングの枠組みにも当てはまるため、ここでは HFSP 自体の詳細は省略し以下に関わる項目のみを整理する。

HFSP はソフトウェアの開発工程を記述するための計算モデルであり、属性文法を基礎とする。人間の関わるプロセスをアクティビティとよぶ操作単位の階層構造としてとらえる。

1 つのアクティビティは、通常、入力値である

文書から出力値の文書を得る人間の作業を表す。複雑なアクティビティはより単純なアクティビティに分解し、この段階の詳細化をアクティビティ分解と呼ぶ（図3）。

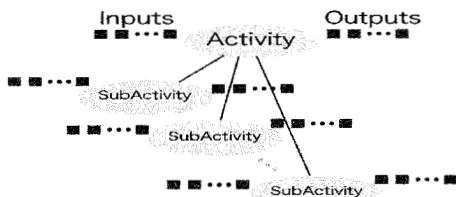


図3. アクティビティと分解

プロセスプログラム（以下必要に応じて「PP」と略す）はアクティビティを列挙した内容をもつ。属性文法の性質から、その実行に伴い、実行系では木構造をもつ実行履歴が生成される（図4）。

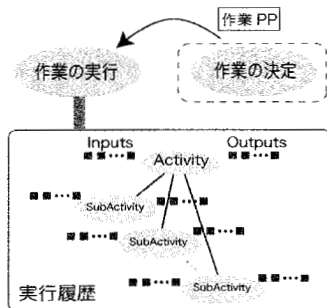


図4. 作業の実行と履歴の構成

作業の決定、すなわち作業PPの修正過程の形式化が動的変更過程の形式化の問題である。

3.3 動的変更を含むプロセス

動的変更プロセスでは途中停止した作業PPの実行履歴の扱いがまず論点となる。この履歴は通常消去できない。例えば、公共住宅の建設途中で計画縮小が生じて、既に完成した住宅は廃棄されず新計画で利用が引き継がれる。このように、動的変更プロセスのPPの実行では、履歴データの継承と再成長の機構が期待される（図5）。

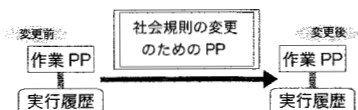


図5. 社会規則変更のためのプロセス

HFSPでは履歴データがシンプルな木構造をとるため、その保存や再成長は木構造のデータの接木処理として実現できる。

以下、HFSPの各論のうち動的変更に関するHFSP/T³⁾の用語を用いてこの機構を説明する。

HFSP/Tは、Work, Try, Museの3つのアクティビティをもつPP(F)の実行で、図2の循環を実現する。Workでは作業計画の実行であるTryと作業計画の修正であるMuseが反復実行される。初期目標から最終出力を得るための通常作業プロセスのスクリプトPP(T)と、何らかの理由でPP(T)の実行が中断された場合の実行状態変更プロセスのスクリプトPP(M)が最初からあり、TryではPP(T)が、MuseではPP(M)を実行する。

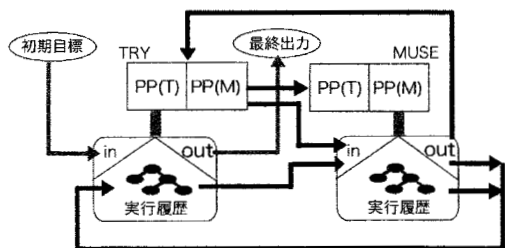


図6. PP(F)実行時の属性値の流れ

両履歴とPP(T), PP(M)は、図6のように循環する。PP(T)とPP(M)の双方がPP(M)の実行における入力属性値であり、ここで両記述の変更を実現する。PP(T)の実行履歴もPP(M)の実行の入力属性値となるため、履歴の再利用が図られる。

PP(F)の実行は、階層的に2種の実行履歴を構成する。1つはPP(F)自体の実行履歴、もう1つはPP(F)の実行の内側で構成される、PP(T)とPP(M)の実行履歴を表す履歴である。後者は接木で1つの履歴として扱える（図7）。

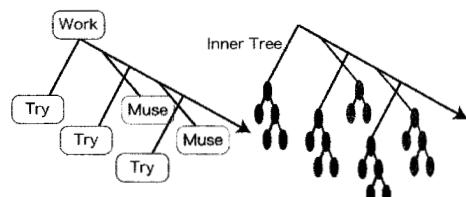


図7. PP(F)実行時の履歴データの蓄積

この枠組みでは、TryとMuseの間でPP(T)とPP(M)および内側の履歴を渡しあう構造をもつ

ことにより、実行状態の動的変更を PP(M)のプロセスの実行として実現することができる。すなわち、社会規則の変更のための規則は PP(M)の中のコマンドとして記述できることになる。

PP(M)では、PP(M)とその実行履歴自体も再利用の対象と位置づけられるため、PP(M)の変更手続きの形式化も可能である。論理的には無限の階層性をもつメタレベルの修正プロセスであるが、これを Try と Muse の2つのモードの反復として形式化している点に特徴がある。ただし、他の形式化技法の場合も、履歴の再利用の機構をもつ通常プロセスの実行系を3系連動させることで同様の動的変更手続きの支援が原理的には可能となる。従って、仕様の動的変更手続きの形式化が難度の高い問題であるのは、履歴の再利用の機構と実行系数の意味においてであると整理できる。

3.4 特徴的な課題

プロセスプログラミングの枠組みに依拠した上記内容はソフトウェア開発を主な対象とする点で本稿の目的と異なる。この相違に基づく論点のうち重要と思われる点を補足する。

第一に、社会規則の動的変更の場合、各アクティビティの終了が集団意思決定に関連する点に注意が必要である。現実世界の意思決定は多くの場合単純多数決に基づくが、電子社会ではその妥当性が問われうる。単記投票方式としての多数決が数学的に問題を含むことは従来から指摘されており²⁾、これが許容されてきたのはボルダ方式やコーブランド方式等の代替技法のコストが非現実的に大きかったことによる。しかし、電子社会においては、計算機環境を前提とするため、民主的意思決定の手段として単純多数決を採用することの根拠が失われかねない。

社会規則の動的変更を形式化対象に含めることは、従って、「正当性：誤りのない」の意味を拡大させる。第一義では技術的ミスがないことであつたが、提供されたサービスが民主的な意思決定を経て提供されたものであるかどうかを問う意味が加わりうる。この問題は Arrow¹⁾以来の社会的選択理論にも関わる。

第二に、社会規則の場合、意思決定を行う組織が階層的で、本会議とは別に委員会等でも決定と作業の循環がある。ソフトウェア開発では原理的に意思決定は最上層部の問題であつた。組織に関するこの相違をどのように実現するのかは上記

の議論にまだ含まれていない。

以上は、技術的な困難という意味での課題とは異なるが、設計上の大きな分岐点であり、注意して機能を追加する必要があると思われる。

4. おわりに

本研究では、規則変更の規則の有無に着目し、安心電子社会研究を、仕様の動的変更手続きを含むクラスと含まないクラスに二分することを提案した。ソフトウェア開発過程の形式化技法であるプロセスプログラミングの議論を応用し、前者が後者に比べ技術的に難度の高い問題を含むことを示した。

本稿は、規則変更手続きを含めた電子社会の実現を検討する概念的枠組みを与えるものであり、具体的に動的変更手続きの支援を実現していくための基礎研究である。

今後の課題としては、特徴的な課題として整理した、民主的な決定手続きの問題や階層的な組織構成を考慮した形式化技法の開発と、これを用いた具体的な記述実験がまず重要であると考えられる。

謝辞

本研究の一部は21世紀COEプログラム「検証進化可能電子社会」の助成による。

参考文献

- 1) Arrow, K. J.: *Social Choice and Individual Values*. New York. Wiley. 2nd. (1963).
- 2) Condorcet, Marquis de. *Essai sur l'Application de l'Analyse, à la Probabilité des Décisions Rendues à la Pluralité des Voix*. Paris. (1785).
- 3) 岩井 淳, 片山卓也: プロセスモデル HFSP における実行状態の動的制御機構, コンピュータソフトウェア, Vol.13, No.1, pp71-84 (1996).
- 4) Katayama, T.: A Hierarchical and Functional Software Process Description and its Enaction, *Proceedings of the 11th International Conference on Software Engineering*, pp343-352 (1989).
- 5) 片山卓也: 検証進化可能電子社会, 情報処理, Vol.46, No.5, pp515-521 (2005).
- 6) Osterweil, L.: Software Processes are Software Too, *Proceedings of the Ninth International Conference on the Software Engineering*, Monterey, California, pp2-13 (1987).