

共同作業管理のための文脈利用方式

高田考平¹ 櫻井義尚¹ 鶴田節夫¹ Avelino J. Gonzalez² Johann Nguyen²

1. 東京電機大学 情報環境学部 2. University of Central Florida, School of EECS

専門家達の対話がウェブベースの協調ツールに制限される複雑なプロジェクトにおいて、製品品質の確保、予算超過や遅延の防止等、プロジェクトの目的達成を保証するためには、プロジェクト管理が不可欠である。通常これはプロジェクト管理者(PM:Project Manager)の仕事である。本論文では、状況の認識とそれに対応する知識を記号表現した文脈知識をベースとする推論である文脈ベース推論(CxBR: Context Based Reasoning)を用いて、プロジェクトマネージャの知能をPMエージェントとして実現する方式を提案する。また、ロケット設計プロジェクトをモデルとした実験システムにより、その有効性を評価する。

Context Exploiting Method to Manage Collaborative Work

In complex projects that require contributions from various experts whose interaction may be limited to a web-based collaborative tool, maintaining control of a project is difficult but it is essential to ensuring that the project objectives are met. This can help avoid cost overruns, shipment delays, but most importantly, product performance. This is typically the job of a project manager(PM). In this paper, a method is proposed to construct a PM agent, which exploits Context-based Reasoning (CxBR) as a tool of choice for implementing control measures typically used by competent PMs. Effects of the proposed method are evaluated, using an experimental system modeling a rocket development project.

1. はじめに

コンピュータによる共同作業支援は、プロジェクトの各構成メンバーの知的共同作業を効果的に管理するためのツールや手法を提供しようとする研究分野である。この研究分野は、グライブラ[1]により CSCW (Computer Supported Cooperative Work)と名付けられ、様々な研究が行われている。それらの中には、プロジェクト・マネージャ(PM)を補助する PM 支援システムの提案がいくつかある[2][3][4][5][6][7]。

このようなシステムは、PM が重要な役割を担うような複雑なプロジェクトにおいて、複数の知識労働者間の共同作業を支援するのに有効である。例えば、軍事作戦、チームで行うスポーツ、および設計プロジェクトなどがこれにあてはまる。本論文は、このうち設計プロジェクト関連、中でも、対面的な意思疎通が出来ないため、PM にとって状況認識が難しいウェブベースの開発プロジェクトや共同作業の支援を対象とする。

さて、調査した上記の従来研究では、PM の状況認識について、直接には言及されていない。状況認識(situation awareness)は Endsley[8]が提唱したもので、現在の状況とその意味、即ちそれが及ぼす影響を認識し対処する人間やエージェント(ソフトウェア)の能力を表す。状況認識によって、PM はプロジェクトの問題を解決したり、予測したりできる。従って、システムがこの能力を提供すれば PM を支援できる。

本論文では、非対面でもあり状況認識が難しいウェブベースの共同作業の支援のため、有能なプロジェクトマネージャの状況認識の知能を PM エージェント(PMA)として実現する方式を提案する。この PMA は PM の替わりをするものではない。有能な PM の管理能力の基本的なものを PMA に組込むことにより、プロジェクトの状況悪化を予測し、人間の PM に警告したり、対処法を提案したりするなどの支援を行う。この PMA の実現のため、具体的には、[9]で述べられている目的(使命、目標)、すなわち、スコop(要件:製品仕様あるいは顧客の要求範囲など)や費用や納期を満たすための開発プロセスにおける PM の状況認識に注目する。

即ち、ソフトウェアエージェントである PMA に、状況とそれに対応する知識や状況間の関係を記号表現した知識である文脈を組み込み・活用することにより、状況認識の知能が容易に実現できると考える。本研究対象であるウェブベースの CSCW では、対面意思疎通の不足のため状況認識が難しいので文脈を利用すること

は特に重要である。そこで、文脈ベース推論(CxBR) [10]を利用して、PMA を実現する。CxBR を用いて文脈を組み込み利用することにより、PMA が状況認識を容易にできるからである。実際、CxBR は、適切な状況認識が生死を決めるほど重要な戦術立案エージェントのモデル化に成功してきている[10]。

次節では、ロケット設計プロジェクトをケーススタディとして、ウェブベース CSCW 向け PMA 実現のための CxBR 適用型文脈利用方式の具体的な提案を行う。3 節では、本提案方式の評価実験と評価結果を述べ、4 節に結言を述べる。

2. PMA 実現のための文脈利用方式

プロジェクトに影響を与える出来事(イベント)は毎日発生する。PM はプロジェクトを成功させるためにイベントを適切に管理・解釈して必要な対応や行動を取る。PM はすべての出来事に完全に把握しなければならない。それらは例えば、工程遅延や設計変更・不良など重大事件の発生や上司によるプロジェクトの目標変更の把握、プロジェクトに対する資金の削減などである。PM は、すべてのこのような出来事(例えば、マイルストーンの完了、上級管理者によるプロジェクトの目標変更、プロジェクトに対する資金削減などを完全に認識し把握しなければならない。この認識・把握が状況認識である。PM はそのような出来事を反映する何らかの一般的な情報、つまり事実ベース(FB)を持っていると考えられる。PM はプロジェクトの進捗状況を把握するために絶えず(プロジェクトの)内容を監視できなければならない。我々は CSCW、とくに Web などを使うため対面的な意思疎通ができないことにより、状況認識が困難となるプロジェクトのコンピュータ支援に焦点を合わせる。この様な困難な状況認識の支援のためには、既に述べた通り、有能な PM をモデル化することが重要な技術課題である。

多くの最新アプリケーションでは最先端技術であるマルチエージェントシステムに焦点を合わせている[7]。我々も PMA とそのアシスタントを造るのにエージェントを使用する。しかし、本論文のアイデアは上記の状況認識を PMA に持たせることである。PMA はプロジェクトの状況を監視し、まさしくその瞬間のプロジェクトの文脈(つまり状況とその対処法、関連状況)を認識する。

具体的に PMA は局面の悪化を緩和するため、識別した各文脈に関連する 1 組の組込み手続きを活性化させる。これらの組込み関数の記述・組織化を助けるのに、Contextual Graphs(CxGs)[11]と呼ぶ有向非巡回グラフを使用する。

文脈ベース推論(CxBR)[10]は、文脈を使用して予想をたてることにより、PM による上記の状況認識を容易にする。従って、この CxBR の文脈利用型予想駆動的な特徴は CSCW システム、特にウェブを利用した CSCW システムにおける PMA の状況認識機能の実現を容易にすると考えられる。以下に、ロケット開発プロジェクトをケーススタディとして、本提案の CxBR を用いた文脈利用方式による PMA の実現方法を具体的に述べる。

2.1 ロケット開発を事例としたプロトタイプシステム

本提案を具体化しその有効性を評価するため、観測用ロケット開発プロジェクトを事例、つまりケーススタディとしたプロトタイプを作成した。

ロケットが製品要求仕様、スケジュールおよび予算、即ちミッション通りに開発されるのを保証することが PM の義務である。納期や予算は Mission Context に反映される。設計どおりでロケットの打ち上げが成功するかをチェックするために NASA[12]によって提供されたシミュレータを使用する。

ロケット開発は、以下の項目の決定を要求する。

- 1) エンジンとロケット胴体の材料と質
- 2) 多段ロケットのステージの数と垂直安定板の厚さを含むロケットの構造
- 3) 制御システム

この要求仕様に対する決定、つまり設計は、設計エージェント(DA)が行う。DA はソフトウェアでも良いが、本プロトタイプでは人間である。

本提案の PMA は文脈ベースのエージェントである。このエージェントは様々な共同作業員又はそのエージェント(ソフトウェア)によって事実ベース(FB)に「ポスティング」、つまりプロジェクト活動に関して入力されたすべてのイベントにアクセスする手段を持っている。PMA は、ポスティングされたデータを解釈して、プロジェクト全体が現在どの文脈、つまりどの状況の中にあるかを決定する。

この決定に専門的技術が必要なら、PMA は補助エージェント(AA : Auxiliary Agent)を利用する。AA は、人間もしくはソフトウェアである。本プロトタイプでは、提案方式の評価を容易にするため、これらはソフトウェアエージェントであると仮定する。

2.2 概念的な構造

ここでは本プロトタイプの概念的な構造について説明し、正式なソフトウェア・アーキテクチャは後で述べる。

図 1 は本プロトタイプにおける、エージェントの関係を示す。

事実ベース(FB: Fact Base)はすべてのコミュニケーションが行われるデータ構造であり、プロジェクトに関連する全ての設計と出来事が揭示される。PMA は、新しい出来事がプロジェクトに影響しているかどうかを見るために事実ベース(FB)を監視しなければならない。

幾つかの点において、FB は黒板に類似している。すなわち、CxBR と黒板アーキテクチャ(BBA)[4]の間には、幾つかの類似性がある。しかしながら、CxBR は制御が中央管理エージェントにあるというより、主文脈(詳細は後述)の間に分散されているという点が BBA と異なっている。主文脈は中央エージェントより良く状況を把握しているので、CxBR の分散の方が望ましい。分散していなければ、管理エージェントはすべての文脈で見つかるすべての知識を含む必要があり、知識の拡張性や局所化が難しくなるからである。

さらに、CxBR は、作業において違ったレベルの粒度を定義するが、BBA はそれを個々のモジュールに任せている。これは、本論文で提案するように、CxBR が CxGs と結合するときにより効果的に定義される。垂直と水平の両方で定義された主文脈内外のインタラクションが CxBR を BBA と著しく異なったものにする。

本プロトタイプでは 3 つの設計エージェントと 5 つの補助エージェントを定義する。

設計エージェントは以下を行う。エンジンとロケットボディの選択(DA-1)、ロケットの構造の設計(DA-2)、適切な制御システムの選択(DA-3)である。エージェントは FB を介して通信するが、PMA は補助エージェントと直接に通信できる。

PMA は定期的に FB を調べる。本プロトタイプのためには、このサンプリング間隔を 1 日に設定した。但し、デモンストレーションでは、15 秒に一度(1 日が 15 秒に)なるように加速する。FB からの関連情報を検索すると、PMA は状況の判断をしてどの主文脈(Major Context)がアクティブになるべきかを定める。この主文脈が現在アクティブな主文脈と異なるなら、この新たに選択された主文脈がアクティブになり、活性化される。主文脈には PMA が実行するあらゆる行動を定義する。この実行は、次のサンプリングサイクルまで続く。

CxBR の文脈は、それらが主文脈かそうでないかにかかわらず制御文脈を記述する総称的な言葉である。すべての文脈は以下の 3 つの基本要素からなる：

- 1) エージェントが文脈の範囲内でカプセル化された振舞いを行うのに必要な行動知識つまり動作知識(Action knowledge)。特定の状況下でのエージェントの知能を表す。文脈(状況)に対応するための制御知識(Control Function)と、下位文脈(Minor Contexts)への遷移のための行動規則(Action rules)から成る。行動規則は CFI の一部を形成する関数内の局所変数か、事実ベース(FB)上の事実(fact)を使用できる。
- 2) 主文脈間の遷移条件を示す遷移知識(Transitional knowledge)。主文脈(Major Contexts)間の遷移規則を表現する。
- 3) 文脈の意味を説明する宣言的知識(Declarative knowledge)。

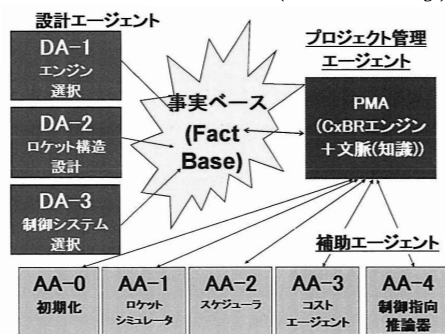


図 1. 本プロトタイプの概念図

2.3 設計エージェント

設計エージェントは多数の設計パラメータを決める。設計パラメータはそれぞれ以下の選択肢を持っている。

エンジン設計者に対応するエージェント DA-1 はロケットエンジンの選択とボディに使用される材料の決定を担当する。エンジンと材料選定の組み合わせに関しては工学的見地から完全に直感的というわけではないが、ロケットモデラーシステムによりそれらを統合・検証する。エンジン設計ではプロジェクトの要件を満たし、かつすぐに入手できるエンジンや材料の選択をするだけである。

ロケット構造設計者 DA-2は他の選択をする。例えばペイロード(有効搭載量)と高度を与えられると、ロケットが1段か、2段か、3段のどれが良いかを決定する。この判断にはエンジン設計者との調整が必要である。

コントロールシステム設計者 DA-3はすぐ入手できる(在庫のある)制御システムを選択する。1段、2段、3段、および多段ロケット用の4つの制御システムを選択することが可能である。例えば、4番目の制御システムはいかなるタイプのロケットにも適用可能である。一方、より高価であり、届くのに時間がかかる。

2.4 補助エージェント

本プロトタイプは、合計で5つの補助エージェントAAを持つ。主な3つのものは、NASAのロケットモデルを用いたロケットシミュレータ(AA-1)、スケジューリングエージェント(AA-2)、コストエージェント(AA-3)である。AA-1は、使命の達成に関してロケット設計に何か問題があるか、シミュレーション計算により決定できる。AA-2はスケジューラであり、遅延を解決したり、解決できない重大なものがあるかを検出したりする。AA-3は設計判断や遅延のプロジェクト予算への影響を計算し基本的な解決を図る。

上記の3つのAAに加えさらに、AA-0はスケジュールと費用を維持・管理する。AA-0は承認された最終納品日と最大費用に対する変更を許容できるエージェントである。PMはAA-0を通してこれらを変更する。

最後に、AA-4は、専用の文脈知識の適用で新たに生じた問題に対し、どの修正行動を開始したらよいかを決めるより汎用的な問題解決エージェントである。PMAは様々な制約間の競合や矛盾を認識した時点で、すべての規制を満たせる解を求めなければならない。これは人間でさえ難しい。AA-4は、プロジェクトの現在の状況下でのすべての要件と規制を満たす設計・費用・配送スケジュールの解決策を見つけようとする。

(AA-4における)成功とは、問題の制約が強すぎず、許容解が存在したことを意味する。失敗は、問題の制約が過剰であり、制約に違反しない解が見つからないことを示す。後者の場合、PMAは後述の行詰まり主文脈(Impasse Major Context)に移行する。AA-4は制約指向プログラミングからいくつかの概念を使用する一種の制約指向推論器であるが、それだけで問題解決を行なう実現・拡張の容易でない一般的なものではなく、専用の文脈(知識)を補助するものである。

AA-4は以下を必要とする。1)ロケット設計パラメータ、2)配送スケジュール用に設計された要素プログラム、3)コスト計算用に設計された要素プログラム、4)費用とスケジュールに影響する外部イベント(外部の出来事)。AA-4は設計の実行可能性、費用、およびスケジュールなどの制約が満たされるならRocket設計パラメータを返す。制約過剰(解が存在しない)なら0を返す。

さて、実際には極めて複雑となるこれらのエージェントのコーディングを簡単にし、本方式の評価を実現可能にするための前提を説明する:

- すべての配送は時間的に並行である。最長時間かかるものだけが、最終納品日を制御する。
- 配送時間・費用は当初、即ち設計変更されなければ製造時間・費用に含まれていると仮定する。配送先は発射場とする。
- 外部イベントはロケットのスケジュールや費用にのみ影響を与える。設計には全く影響を与えない。
- 実験開始日(時刻ゼロ)の後にされた設計変更には、設計変更をした日から教えられる新納入日が必ず設定される。
- 設計変更は費用と配送の両方に影響しうる。
- 「動く」つまりロケットがミッションあるいは顧客要求通りの飛び方をするとAA-1がみなすときのみPMAは設計変更を受け入れる。

- 「動かない」つまりロケットが顧客要求通りの飛び方をしないときAA-1がみなすときPMAは設計変更を受け入れない。
- PMAはスケジュールやコストが問題になりそうだからといって、「動く」新設計を拒絶できない。
- 設計に無関係な外部イベントから生じる許容できない遅れやコスト過剰を解消するために設計を変えることができる。競合的な文脈つまり競合状況にあるときにこれを解決する以下の2つのオペレータ(対応知識、動作知識)を文脈知識の要素として定義する。当然、それらの動作範囲つまり動作する文脈(状況)は限られていて、本プロトタイプに特化している。

1)修正措置オペレータ#1(ca1):

コストエージェントAA-3が使用する。

省エネ励行や清掃費など間接費削減により総コストを削減する。これは、二度までできるだけである:

- 1回目、総コストの最大10%の減少
- 2回目、総コストの最大5%の減少

2)修正動作オペレータ#2(ca2):

スケジュールエージェントAA-2が使用する。

超過勤務手当を作業者に支払うことによって、部品納期を短縮する。例えば5万円/日の納期減少が設定できる。但し、本プロトでは最大30日減少しかできないものとする。

一方、AA-4は制約指向エージェントで、汎用的な制約指向論理を用い、最初に、設計変更、時間外手当、間接費削減のいずれも必要としない解決策を見つけようとする。これに失敗した場合は、まず、予算上限を超えない限り、時間外手当を支払うことによって納期遅延短縮を図る。予算超過だけでなく、超過人員削減などによるコスト削減を試みる。

これらがいずれも不十分なら、予算やスケジュールが制約範囲内になるような設計変更パラメータを探索する。

問題解決のすべての試みが失敗するなら、PMAが行詰まり(Impasse)を宣言し、後述の行詰まり主文脈がアクティブになる。

2.5 PMA

PMAはプロジェクトを管理する。これは、本提案の文脈駆動方式を中心とするソフトウェアエージェントである。CSCWの知識の情報処理における本研究の貢献はこのエージェントを文脈ベースにして、その実現容易性・有用性を評価することである。

さて、この文脈ベース化のためには、PM、それゆえPMAが最初知っておくべき事柄は何かを明らかにすることである。すなわち、ロケットの性能に対する要求は何か、ロケット製作に使用できる予算はどれだけか、仕事完了の時間枠は何かである。本提案のPMAはこの情報を使命文脈(Mission Context)として記憶する。既に述べたように、使命文脈は処理を制御するだけでなく、使命の情報を記述するだけである

1)PMAのための使命文脈: 本ケーススタディでは、使命(目的)は、要求された仕事を実行するロケットを時間内にかつ限られた予算で完成することである。表1に示すのはMission#1で、コストの安い場合の使命文脈の例を示す。

使命文脈の重要部は主文脈(Major Context: 次節で説明する)に関する実行計画である。戦略行動つまり、エージェントが計画に従い、しかもいつでも出来事に対応できる必要があるために、これが重要である。この実行計画とは入念に設計された遷移基準を持つ主文脈(Major Contexts)の系列である。

物事が(全て)計画通り進み、PMはプロジェクトが予定通りに進み予算の範囲内にあることを確認する以外に何もする必要がない場合は、本ケーススタディには適用できない。この様な時はプロジェクトは常に後述の正常主文脈に留まると考えられ、PMAは全くどんな行動を取る必要もない。

使命文脈は(使命の)目標(mission goal)、(課された)制約、使命実行のための上記計画などを属性として含む。

使命文脈は目標(goal)属性としてロケットシミュレータ内における目的地(到達高度など)の記述、および目的地に到達したかを示す論理型変数を持つ。制約としてシミュレータ内の変数に対し、定数値あるいは有効範囲をPMAに伝える。

表 1. 使命文脈の内容

属性	Mission
予算上限	10 MYen
最終納期	6 months
到達距離	100 m
搭載積量	490 g
重量	< 2,000 g
優先度	Low cost

2) 主文脈(MC, Major Context):

多くのことが開発プロセスに影響する。例えば、設計過程でいくつかの異なる状況が起こりうる。各MC(主文脈)はPMAが直面し解決しなければならない問題状況を反映する。MCはPMAがプロジェクトを正常な状態に戻すための知識・ノウハウを含むべきである。本ケーススタディのプロトタイプに組み込み使用するMCは、正常文脈(Normal MC)と、設計変更主文脈(DesignChange MC)と、外部事象主文脈(ExternalEven MC)と、行詰まり主文脈(Impasse MC)である。

これらの文脈は問題状況でどう行動するかに関する手順だけでなく期待も含む。それぞれのMCは以下の通りである:

7) 正常主文脈(Normal MC): すべてが正常であり、プロジェクトが計画通り進んでいるなら、文脈は正常(Normal)といわれる。この主文脈(MC)では、PMAは、nシミュレーションサイクル毎に(1日に一度ならn=1)状況をモニターし続ける。正常文脈は、初期文脈かつデフォルトの文脈である。PMAは、アクティブな副文脈として、エージェントAA-0を、この主文脈の下で呼ぶ。これは、スケジュールデータを初期化して、プロジェクトの初期費用を計算する。この主文脈では事実ベース上で新しい外部の出来事を検出すると、それぞれ設計変更か外部事象の主文脈に遷移する。

4) 設計変更主文脈(DesignChange MC): 設計やその変更(決定)が行われると、PMAの文脈はこの主文脈へ遷移する。この主文脈において、PMAは設計変更がプロジェクトに与える影響に対処する。この主文脈には、設計変更がプロジェクトに悪い影響を与えるか決定し、もし悪い影響があればその問題を解決する下位文脈あるいは副文脈(Sub Context)と呼ぶ知識CxG-1を持つ。各文脈知識の主要素は行動知識つまり動作知識(Action knowledge)である。行動知識は文脈(状況)に対処する動作を規定する制御知識(Control Functions: CFi)と下位文脈(Minor Context)への遷移のための行動規則(Action rules)から成る。これらはif-then型のルールなどでも表現できるが、本ケーススタディではこの知識CxG-1を文脈グラフ(contextual Graph: CxG)として表現する。CxG-1は設計変更の到達高度、軌道などへの影響や費用・スケジュールへの影響を決定し、問題があれば、解決しようとする。

文脈グラフ CxG は以下の基本構成要素を持つ。i) 文脈ノード(Contextual nodes)はグラフ中の大円で表現する。通常それらのノードは人間のユーザまたは呼出す関数に質問をする。その答えは問題の文脈の更なる定義を可能にする。即ち文脈ノードは2つ以上の方向に分岐できる。ii) 動作(Action)は正方形によって表現され、何かを達成あるいは学ぶために人間のユーザやソフトウェアエージェントによって実行される。結果は後続の文脈ノードで処理する。iii) 活動(activity)は連続した動作をまとめるものである。

activityは角の丸い正方形として示される。iv)最後に、再結合ノードは小さい円で、上記の文脈ノードで作られたいくつかの分岐を再結合するためのものである。これは大きいグラフ中にサブグラフを作るのに役立つ。

CxG-1がする最初のことは、設計変更が“うまく動く”か、即ち、ロケットが与えられた使命を遂行できるかを決定することである。変更が行われたら、PMAは設計変更が他の既存の設計と互換性を持つか決定するために、RocketModelerというシミュレーションエージェント(AA-1)を含むこの副文脈 CxG-1を活性化させる。設計変更されたパラメータを変更されていない他のパラメータと一緒にシミュレーションすることにより、AA-1は、設計が“うまく動く”時は値1を、そうでない時は値に0を返す。ここで、AA-1は、技術的な問題だけを考え、スケジュールも費用も考えない。

RocketModeler、つまりAA-1が設計変更を、言わば「承認する」なら、次にPMAは、AA-2とAA-3を使用することでスケジュールや費用への影響を決定する。しかし、AA-1の結果が設計矛盾を示すなら、全ての制約を満たす解を見つけるのに、直ちにAA-4を呼ぶ。

7) 外部事象主文脈(ExternalEvent MC) 新しい外部事象が事実ベースに登録されたら、PMはこの主文脈を活性化させる。但し、既述の通り、これらの外部事象はプロジェクトのスケジュールと費用だけに影響するものと定義する。外部事象主文脈は、活性化されると文脈グラフ記述した副文脈CxG-2を活性化し実行する。

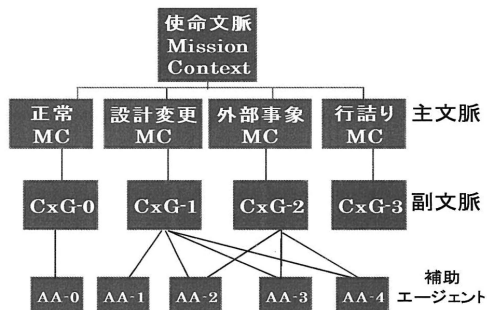


図 2. PMA のための使命/主/副文脈と専門知識の階層

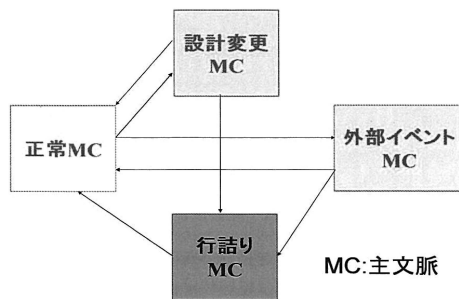


図 3. 主文脈間の水平遷移のトポロジー

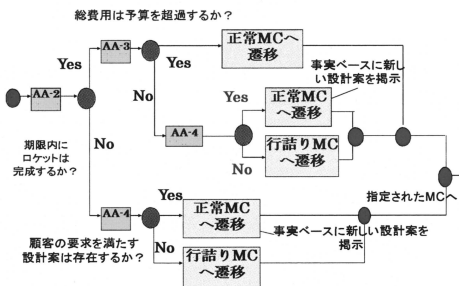


図4. 副文脈 CxG-2 の文脈グラフ

CxG-2 は、最初に AA-2 と AA-3 を呼ぶことでスケジュールもしくは費用へのマイナスの影響があるかどうか検査する。なにもなければ、正常文脈に戻される。どちらか一方でもマイナスの値、つまり許容範囲を超えるスケジュール遅延量や予算超過値などを返すなら、この問題の解決のため、CxG-2 は AA-4 を呼ぶ。

AA-4 によって修正動作が施され、問題が解決するようなら解決された値、つまり許容範囲のスケジュール(余裕)値と予算を持って正常文脈に戻る。さもなければ、システムは問題解決の解法を探すのをあきらめ、下記の行詰り主文脈に遷移する。

行詰り主文脈(Impasse MC): 状況が過剰制約のとき、または新事象や新設計が納品日か最終コストあるいはその両方の許容範囲超過を引起すとき、PMA はこの主文脈を活性化する。

この主文脈では、最終納品日(FDD)か最終承認予算の拡大が人間つまりユーザや上司に要求される。さもなければ、プロジェクトの中止が必要になるかもしれない。即ち、PMA の権限の範囲内で設計矛盾が解決できない時、行詰り主文脈が活性化されるともいえる。高レベルの権限をもつ管理者かユーザの介入以外の目的でこの MC を動作させることはない。

図 2 は様々な文脈間の階層関係を表わす。図 3 は主文脈(MC)の遷移の可能性を示す。

3. プロトタイプによる実験と評価

PMA がプロジェクトの状況を確認して、適切な修正措置を提供する能力を評価するための本プロトタイプの実験セットと、得た結果(詳細は[12])を以下に説明する。実際、プロジェクトの状況認識のために PMA が適切な主文脈知識をどのようにうまく選択・使用し、適切な対応・動作(action)を実行したかにより、これを評価した。即ち PM の状況認識の知能を実現できたかを評価した。

テストセット#1 では(製造)コスト増加や部品供給者の納入遅延のような外部イベント(External Event)の発生時の状況認識を評価した。PMA はこれらの外部イベントに反応し、プロジェクトに対する悪い影響が無いかを決め、もしある場合は AA-2 や AA-3 などの修正動作オペレータを用いて修正しなければならない。テストセット#2 では設計変更(Design Change)を導入する。設計変更にも矛盾があるか、または設計変更がコスト、スケジュールに悪い影響を与える場合の管理を PMA は必要とする。テストセット#3 は、より複雑・困難な場合の評価のためのものであり、外部事象と設計変更の両方を含む。

表 2 に、文脈の遷移系列としてテスト結果をまとめる。この系列などは省略表記する。大文字は認識(選択・使用)した主文脈(MC)を示す。"N"は正常主文脈、"EE"は外部事象主文脈、"DC"は設計変更主文脈、"I"は行詰り主文脈を表す。小文字は状況に対応して実行した動作を示す。"ca"は、制約矛盾など問題を解決するのに使用した修正動作オペレータを示す。"ca1"は AA-3 が使用

する知識、即ちコスト削減(あるいはコスト管理)オペレータを表し、"ca2"は AA-2 が使用するスケジュール(時間)管理オペレータを表す。"ca"エントリがないのは矛盾(これらのオペレータを必要とする問題)が起こらなかったことを示す。小文字"a"は PM の知識(AA-1)により設計変更が拒否されたことを示す。小文字"a"は、PM の知識(AA-1)によって設計変更が受け入れられたことを意味する。(上記の記号が何度も繰り返す)反復記述は、1つ以上の外部イベントや設計変更が異なる時間に起こったことを示す。表 2 の予想と実測結果は予想/実測のそれぞれの主文脈の遷移系列を示す。

テストはテストセット#1、#2 に対して成功した。しかし、テストセット#3 に対して実測結果の一部が予想結果と異なった。例えば、3.2 では、1つつまり行詰り主文脈になると予想されるのにシステムは正常の文脈を宣言した。しかし、これらの問題はコスト管理(AA-3 中の ca1)/時間管理(AA-2 中の ca2)に対する比較的容易な誤り検出と改良によってスムーズに解決された。理由はこれら ca1 や ca2 がコストや時間等と利用条件が文脈的に、つまり特定の状況下では、明確な知識であるためと考えられる。この様々な不良検出・改良の容易性によって複雑なウェブベースの CSCW 向けの PMA、あるいは、それ自身が複雑な PMA を造るのにも、CxBR の利用が効果的であると立証された。

表 2. テスト結果のまとめ

Test(可否)	予想	実測結果	note
1.1	OK Normal	Normal	初期化
1.2	OK N-EE-N	N-EE-N	コスト
1.3	OK N-EE-ca1-N	N-EE-ca1-N	
1.4	OK N-EE-N	N-EE-N	納期
1.5	OK N-EE-ca2-N	N-EE-ca2-N	
1.6	OK N-EE-ca2-N-EE-ca1-N	N-EE-ca2-N-EE-ca1-N	
1.7	OK N-EE-ca2-N-EE-ca1-N-EE-ca2-ca1-N	N-EE-ca2-N-EE-ca1-N-EE-ca2-ca1-N	
1.8	OK N-EE-ca2-N-EE-ca1-N-EE-ca2-ca1-I	N-EE-ca2-N-EE-ca1-N-EE-ca2-ca1-I	
2.1	OK N-DC-a-N	N-DC-a-N	
2.2	OK N-DC-r-N	N-DC-r-N	
2.3	OK N-DC-a-ca1-ca1-N	N-DC-a-ca1-ca1-N	
2.4	OK N-DC-a-ca1-ca1-I	N-DC-a-ca1-ca1-I	
2.5	OK N-DC-a-ca1-ca1-N	N-DC-a-ca1-ca1-N	
3.1	X N-DC-a-ca1-ca1-N	N-DC-a-N	コスト
3.2	X N-DC-a-ca2-ca1-ca1-I-DC-N	N-DC-a-ca1-ca1-N	納期

これらのテスト結果から判断すると、共同作業工程の管理に文脈を駆使する利点は以下の通りである:

文脈に注目することにより文脈的に、即ち、状況認識やその対応方法として意味のある独立性の高い単位に知識を分割できた。また、分割された知識モジュールの構造を CxGs により明確かつグラフィカルにできた。これらにより、知識の検証と改良が簡単になった。こうして、不良部や知識変更による影響を受ける範囲を局所化するのが比較的簡単になった。

4. 結論および今後の研究

本研究では、CxBR を適用した文脈利用方式により共同開発作業などの状況を確認し、状況に対応する処理をする PMA を作成するのに成功した。具体的には外部イベントと設計変更を監視し、問題を認識し、プロジェクトの設計、費用、およびスケジュールを要求範囲内に持ってくるように修正動作オペレータより汎用の問題解決器を適用することが可能であった。問題が過剰制約のときは行詰まり (impasse) を宣言し、外部の人間の介入を求めた。

更に、共同作業工程を管理するのに文脈を駆使する効果が立証された。即ち、このように複雑な PM 知識でさえ、文脈に焦点を合わせることで、独立で意味のある単位に分割できたため、その獲得・表現・利用および検証・改良が容易であった。

この様な PMA は(他の作業者と)対面しない環境がプロジェクト管理を非常に難しくしている分散あるいはウェブベースの CSCW システムで特に効果的であると考える。

それにもかかわらず、AA-4 エージェントは、非常に難しい問題を解決するために機能的に肥大化した。この比較的簡単な応用では成功したが、現実世界への応用に対しては AA-4 は不合理に複雑になると思われる。そこで、問題をさらに文脈化し、AA-4 がより単純になるように新しい副文脈を定義するのが望ましいだろう。

将来の研究では、AA-4 に関係する状況をさらに文脈化することを計画している。更なる文脈化により、AA-4 などを過度に複雑にせずうまくいくことを検証するために、本提案方式を大規模ソフトウェア開発やビル建設管理のようなより現実的で複雑なプロジェクトへ適用することを考えている。

5. 参考文献

- [1] Wikipedia, www.wikipedia.com
- [2] Sluizer, S and Cashman, P. M., "XCP: An Experimental Tool for Managing Cooperative Activity", Proc. of the ACM 13th Computer Science Conference, New Orleans, LA, pp. 251-259 (1985).
- [3] Bardram, J. E., "Organisational Prototyping: Adopting CSCW Applications in Organisations", Scandinavian Journal of Information Systems, 8(1): p. 69-88 (1996).
- [4] Medina-Mora, R., Winograd, T., Flores, R. and Flores, F., "The Action Workflow Approach to Workflow Management Technology", CSCW 1992, Toronto CA, (1992).
- [5] Levitt, R. E. and Nissen, M. E., "The Virtual Design Team (VDT): A Multi-Agent Analysis Framework for Designing Project Organizations", Proceedings of the 2003 KIMAS conference, Boston, MA, pp.115-120 (2003).
- [6] Wu, S. and Kotak, D., "Agent-based Collaborative Project Management System for Distributed Manufacturing", Proc. of the IEEE Int'l Conf. on Systems, Man and Cybernetics, Vol. 2, pp. 1223-1228 (2003).
- [7] Lee, C-S Wang, M-H Chen J-J, and Hsu C-Y, "Ontology-based Intelligent Decision Support Agent for CMMI Project Monitoring and Control," North American Fuzzy Information Processing Society (NAFIPS), Montreal, Canada, pp. 627-632, (2006).
- [8] M. Endsley, "Design and evaluation of situation awareness enhancement", in Proceedings of the Human Factors Society 32nd Annual Meeting. Santa Monica, CA, pp.97-101(1988).
- [9] R. Nienaber, and E. Cloete, "A software agent framework for the support of software project management", Proc. of SAICSIT 2003, pp.16-23(2003).
- [10] Gonzalez, A. J. and Ahlers, R., "Context-based Representation

of Intelligent Behavior in Training Simulations", Transactions of the Society for Computer Simulation, 15(4), pp. 153-166 (1999).

- [11] Brezillon, P., "Modeling and Using Context: Past, Present and Future", <http://www.lip6.fr/reports/lip6.2002.010.pdf> (2002).
- [12] National Aeronautics and Space Administration, RocketModeler version 1.2 - undated, (2003).
- [13] Gonzalez, A. J., "Using Contexts to Control a Collaborative Process", Final Report, CSCW project, Tokyo Denki University, December 2007.
- [14] Englemore, R.S. and Morgan, A. (eds), Blackboard Systems. Addison-Wesley, (1988).
- [15] Chatzoglou, P. D. & Macaulay, L. A., "A Rule-Based Approach to Developing Software Development Prediction Models", Automated Software Engineering 5, pp. 211-243 (1998).