

拡張現実空間概念に基づく オフィスワーク支援環境の試作

今野将† 原英樹† 藤田茂† 菅原研次† 木下哲男†† 白鳥則郎††

†千葉工業大学情報工学科
〒275 習志野市津田沼2-17-1
††東北大学電気通信研究所
〒980 仙台市青葉区片平2-1-1

あらまし

人間同士が空間を共有するサイバースペースが企業や研究機関で提案・開発されている。しかし、それらサイバースペースは実世界に対する影響力が低く、現実から遊離した存在といえる。また、増大・分散化しつづける情報資源の獲得や操作は非常に困難であり、それに伴うネットワークにかかるトラフィックの増大も問題となってきている。これらの問題の解決法として、本研究室で研究しているADIPS、やわらかいネットワーク等のエージェント技術があげられる。本稿ではこれらエージェント技術を用いて、人間とエージェントが互いに協調しあう空間、すなわち拡張現実空間の概念提案を行う。また、その適用例であるサイバーオフィスについて述べる。

An Approach to Developing a Cyber Office based on a Concept of Augmented Reality Space

Susumu Konno †, Hideki Hara †, Shigeru Fujita †, Kenji Sugawara †, Tetsuo Kinoshita ††, Norio Shiratori ††

† Dept. of Computer Science, Chiba Institute of Technology
2-17-1, Tudanuma, Narashino 275

†† Research Institute of Electrical Communication, Tohoku University
2-1-1, Katahira, Aoba-ku, Sendai 980

Abstract

CyberSpace which is shared among humans have been proposing and developing in several research organizations. But these CyberSpaces proposed have little influence to the real world, therefore, they are not the augmented real world, but just only CyberSpace. Furthermore, It's getting more difficult to get or operate the information resources which are increasing and being distributed day by day, and also exploring of network congestion becomes a serious problem. To solve these problems, the agent technology which is called ADIPS and Flexible Information Network which is based on the ADIPS have been proposed. In this paper, firstly we describe space of the human cooperation with agent which is called Augmented Reality Space(ARS) with these agent technologies. Next, CyberOffice which is application of HASS is illustrated.

1 はじめに

ネットワーク環境を新たな社会的活動の場とすることにより、物理的制約を乗り越えて、個人が多様なそしてより効率的な活動を行えるサイバース社会の到来が期待されている。われわれは、ネットワーク技術や協調技術を用いて、たとえばオフィスワークのような活動を支援するための場の実現を目指したアプローチとして、現実社会の機能を少しずつ強化して得られる拡張現実空間という概念を提案している。

これまでのバーチャルリアリティ技術を用いて、コンピュータシステムやネットワーク環境の中に仮想空間を構成するさまざまな技術が提案されている。しかしながら、このような仮想空間と現実空間の関係が希薄であれば、電子商取引などの経済活動を仮想空間で行った場合、その結果が現実世界に反映しないことがあれば、従来の現実空間での経済活動を仮想空間に導入することに問題が生じる。また、仮想空間での経済などのさまざまな状態は、現実空間の反映としてのリアリティを持つ必要がある。

現実空間と仮想空間が強いリアリティで関係づけられ、この二つの空間を統合したものを拡張現実空間と呼ぶことにする。拡張現実空間では、人間が活動する上での時間的・空間的制約がコンピュータとネットワークの力で乗り越えられる一方、現実空間での人間活動のメタファをそのまま利用することが出来る。現実空間のメタファで成立する拡張現実空間での活動は、現実空間と仮想空間の規約に基づいて、現実空間と仮想空間に反映される。このように、人間の現実空間での社会活動を支援するためには、現実空間の機能の拡張が実現できる場としての拡張現実空間を構成する技術が必要になる。

2 拡張現実空間の構成モデル

2.1 拡張現実空間ARS

図1に拡張現実空間ARS(Augmented Reality Space)の構成モデルを示す。ARSは以下のように記述される。

ARS=<RSM, SSIF, SSB>

RSM(Real Space Model): 現実空間モデル

SSIF(Symbiotic Space Interface):

共生空間インタフェース

SSB(Symbiotic Space Base): 共生空間基盤

現実空間モデルRSMとは、本モデルが支援の対象とする現実世界の人間とその活動を形式化したモデルである。 共生空間インタフェース

SSIFは、人間の間の協調や人間とエージェント間の協調を実現する場のモデルである。共生空間インタフェースSSIFには、図1に示すように、人間およびエージェントが化身（アバタと呼ぶことにする）として参加し、その場で定義された規約に基づいて、相互作用を行う。共生空間における規約とは、その共生空間で支援対象とする作業に関するワークフローなど人間の活動様式、手順に基づく制約などである。

共生空間基盤SSBとは、拡張現実空間において、人間にサービスをし、人間の活動を協調的に支援するソフトウェアシステムの集団であり、エージェントプログラムの集合APとオブジェクトプログラム/データの集合OPから構成される。エージェントプログラムとは自律性を有する能動的なプログラムであり、オブジェクトプログラムはメッセージに反応する受動的なプログラムである。共生空間基盤はネットワークにより実現される分散処理環境を表す。以下拡張現実空間ARSの3つのサブシステムについて述べる。

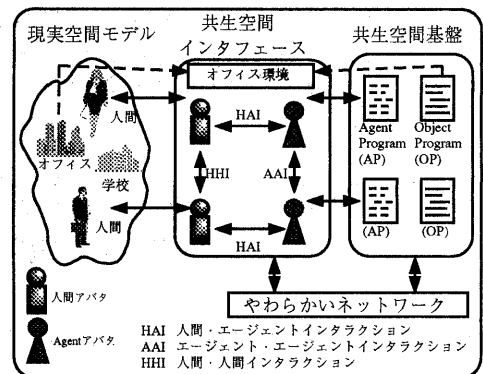


図1 現実世界の拡張

2.2 現実空間モデルRSM

現実空間モデルRSMとは、拡張現実空間で支援する対象の人間の活動や、必要な資源や処理に関するモデルである。

RSM=<DOMAIN, TASK, USERS, OBJECT>

DOMAINは拡張現実空間ARSを構成する対象となる領域の仕様である。各領域ごとに様々なスクリプトが用意される。本稿ではオフィスワーク領域の作業プロジェクトが例題として設定されている。この他、教育コースウェアなど、さまざまな領域のモデル化が必要になる。TASKは作業プロジェクトで実施されるタスクの目標、作業内容、

作業計画、作業制約などの情報である。USERは、作業プロジェクトに参加する利用者のリストである。利用者のロケーション情報、役割、制約などの情報が記述される。OBJECTは、TASK実行において利用者により必要と認識される資源、データや処理プログラム（システム）のリストである。

2.3 共生空間インタフェースSSIF

(1) 共生空間インタフェースSSIFの構成

共生空間インタフェースSSIFは、現実空間モデルRSMでおこなう作業TASKを処理・実現するための活動を強化するための場を構成する。RSMとSSIFを統合した場が拡張現実空間になる。このような協調作業を行う場の最小単位を空間ユニット(Space Unit)と呼ぶ。一つの空間ユニットは複数の人間とエージェントにより共有される。図2に示すように、空間ユニットの集合を空間モジュール(Space Module)という。空間モジュールの集合も同様に空間モジュールと呼ぶ。共生空間は空間モジュールにより階層的に構成される。

SSIF = < SU, SM, S-REL >

SU : SSIFを構成する空間ユニットの集合

SM : SSIFを構成する空間モジュールの集合

S-REL : SSIFの空間モジュールの階層構造の関係の集合

空間ユニットおよび空間モジュールへのアクセスは、ゲートと呼ばれるアクセス点のみにより行われる。空間ユニットおよび空間モジュールに対して、これ以外のアクセス点は存在しない。ゲートは人間用のゲートHGとエージェント用のゲートAGの2種類ある。ゲートでは空間ユニットあるいは空間モジュールの性質・機能により定まるセキュリティが定義されている。

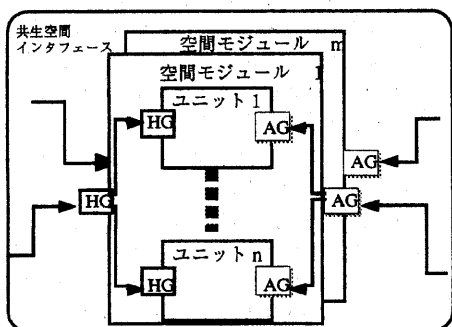


図2 空間の階層的構成

(2) 空間ユニットのモデル

空間ユニットSUは以下の記述で表される。

SU = < WF, HAA, SAA, OBJ, CONST, ACTION, HG, AG >

WF : ワークフローの記述

HAA : 人間(USERS)のアバターの集合

SAA : ソフトウェアエージェントのアバターの集合

OBJ : 空間ユニットのオブジェクトの集合

CONST : 空間ユニットの構造

ACTION : 相互作用の記述

HG : 人間用ゲート

AG : エージェント用ゲート

ただし、WFは空間ユニットで処理するタスクを実現するためのワークフローの記述である。ワークフローWFの詳細は4章で述べる。

HAAはRSMのUSERSのアバターの集合である。

HAA = { h : h = < id, ability, responsibility, av > }

id : 個人のネットワークID

ability : タスク処理能力や役割の記述

responsibility : 空間ユニットで行われるタスクプロセス (TASKで定義) の責任を持つプロセス名

av : 共生空間インタフェースに対する視覚化情報

SAAは共生空間基盤で定義され、WFの中で利用者に認識されるエージェントのアバターの集合である。

SAA = { s : s = < ag-id, ability, responsibility, av > }

ag-id : エージェントのID情報

以下はHAAと同じ

OBJ = { obj : obj = < id, access, position, functions > }

id : ネットワークでの識別子

access : アドレスやアクセスするための情報

position : 共有空間に占める位置情報

functions : オブジェクトの機能

CONST = { comp : comp = < class, position, struct, member, object, communi, view > }

class : 構造物の種類 (クラス)

position : 構造物の位置情報

struct : 他の構造物との関係

member : 構造物を利用するメンバー

object : 構造物に含まれるオブジェクト

communi : 他の空間ユニットの人間やエージェントの通信手段

view : インタフェースへの視覚化情報

ACTION = < ACT, STATE, RULE >

ACTは、その空間ユニットにおいて人間やエージェントの間で可能な相互作用の集合である。STATEはアバタの内部状態である。RULEは相互作用や事象が発生したときの制約や内部状態の変化規則を表すルールの集合である。

人間やエージェントが活動する場は空間ユニットであり、空間ユニットをボトムアップに組み上げて、上位概念の場を構成するのが空間モジュールである。空間ユニットあるいは空間モジュールでは一つのオントロジーが定義され共有される。HGは人間が空間ユニットにアクセスするアクセスポイントであり、HAAに登録されたメンバーであれば、そのゲートに自由にアクセス（入室）ができる。登録されていない人間のアクセスは、HAAのメンバの承認、あるいはゲストとして登録されていることが必要である。エージェントのアクセスポイントAGについても同様である。

(3)空間モジュール間の移動と通信

人間およびエージェントは、自分の目標を達成するために、他の空間ユニットあるいは空間モジュールに移動することができる。移動はアクセスポイント(HG,AG)を通じて行われる。アクセスポイントでは認証の確認が行われる。

また、空間ユニットにおいては、その中に存在するメンバー同志であれば、自由に通信（相互作用の一つ）を行うことができる。しかし、空間ユニットを超えた通信はできない（相互作用の定義がない）。空間ユニットを超えた通信は、オブジェクトとして定義されている、電話メタファなどの通信機器オブジェクトを利用して行われる。

2.4 共生空間基盤SSB

共生空間基盤SSBとは、ソフトウェアエージェント（以下略してエージェントと呼ぶ）、オブジェクト（計算手続き、データ/知識）の集合である。共生空間基盤では、さまざまな役割や機能を持つエージェントが定義され、共生空間インタフェースのアバタを通して利用者の要求を協調的に処理する。エージェント間には協調・交渉のプロトコルが定義されており、必要に応じてこれらのプロトコルを用いて、協調処理を行う。また、エージェント化されない計算手続き、データ/知識などはオブジェクトとよばれ、エージェントにより利用される。

共生空間基盤のエージェントはADIPSフレームワークを用いて、開発される。ADIPSフレームワークでは、オフィスワーク支援要のクラスエー

ジェントが定義されており、共生空間インタフェースで定義された空間ユニットモデルに基づいて必要なエージェントが生成される。

3. ワークフロー記述

強化すべき機能は、ワークフローに基づき決定され、この機能により作業を支援する場として拡張現実空間が設計される。この節ではソフトウェア開発プロセスに限定したワークフローのモデルを考える。

3.1 ワークフローの基本モデル

本節では作ワークフローを記述するための基本モデルとして定型作業プロセスのモデルを定義する。

WF =< task-id, WP, PG, AGENTS, OBJECT >

task-idはRSMにおけるTASK情報のポイントであり、WPはワークフローの要素となるプロセス集合である。PGはプロセス間の関係から構成されるプロセスグラフpgの集合である。AGENTSは、プロセスpの実行において人間の協力者として認識されるエージェントの識別子のリストである。例えば秘書エージェントや店員エージェントなどがその例である。利用者より認識されないバックグラウンドで処理を行うエージェントはここでは記述されない。AGENTSの実体は共生空間基盤SSBに存在するエージェントプログラムである。OBJECTは、プロセスpにおいて必要とするリソースの情報である。

ワークフローのプロセス記述WPは

WP={ p | p = < class, pid, ip, op, pre, post, pg-id,

mem-id, res-id >

class : プロセスのクラス

pid : プロセスの識別子

ip : プロセスへの入力情報

op : プロセスからの出力情報

pg-id : 詳細化プロセスグラフの識別子

mem-id : プロセスを処理する人間およびエージェントの識別子である

res-id : プロセスで使用される資源
(オブジェクト・データ)

作業プロセス（以下単にプロセスと呼ぶことにする）の基本タスクとは図3に示すように、与えられる入力プロダクトをある決められた手続きに基づいて出力プロダクトに変換する作業である。プロダクトとは仕様書、マニュアル等の文書、テストの結果、実行プログラムなどである。設計プロセスにおいては、入力プロダクトを要求仕様、

出力プロダクトを設計仕様と呼ぶことにする。

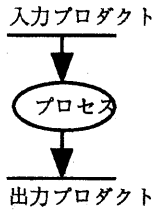


図3 基本アクティビティ

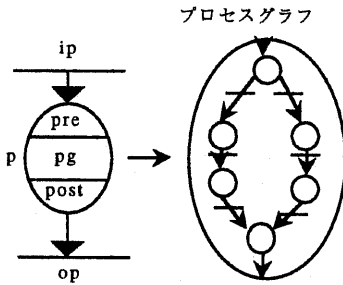


図4 プロセスの階層的定義

プロセスグラフpgは方向付きグラフであり、次のノード集合pgnとアーク集合pgaで記述される。

$$pg = \langle pgn, pga \rangle$$

但し

$$pgn = pcn \cup pdn$$

であり、pcnはプロセスノード集合と呼ばれ、pdnはプロダクトノード集合と呼ばれる。pcnは図3.1にしめされるプロダクトの変換を行うプロセスの集合であり、pdnは図3に示される処理対象のプロダクトの集合である。

アーク集合pgaは2つの部分集合in-arcとout-arcの和集合で表される。

$$pga = in\text{-}arc \cup out\text{-}arc$$

但し

$$in\text{-}arc = \{(n1, n2) \mid n1 \in pdn, n2 \in pcn\}$$

$$out\text{-}arc = \{(n1, n2) \mid n1 \in pcn, n2 \in pdn\}$$

dc-arcの要素はプロセスに対する入力アークと呼ばれ、out-arcの要素はプロセスからの出力アークと呼ばれる。プロセスpが内部にプロセスグラフを持たないとき ($pg = \emptyset$ と書く)、そのプロセスは素プロセスと呼ぶ。

3.2 プロダクト

プロセスの処理対象と処理過程であるプロダクトは以下に示すように多数の属性から構成され、属性間には様々な関係あるいは制約が存在する。あるプロダクトの属性およびそれらの関係制約を

規定するモデルをドメインモデルという[渡辺93]。ドメインモデルはそれらに属する事例の抽象化されたクラスの集合であり、逆にプロダクトはドメインモデルに定義されたクラスの具象化された事例である。クラスで定義された属性の値を与えることをインスタンスレーションという。その結果生成されたプロダクトを事例(インスタンス)と呼ぶことにする。

以下プロダクトの定義を行う。

$$prod = \langle name, domain, generated_from, provided_for, \{description\} \rangle$$

但し、nameはプロダクトの識別子、domainはこのプロダクトが属するドメインの名前である。generated_fromはこのプロダクトが出力されるプロセス名であり、provided_forはこのプロダクトが入力されるプロセス名である。descriptionは仕様記述などのプロダクトの内容であり以下のように定義される。

$$description = \langle product_name, description_class, interface, \{attribute\} \rangle$$

descriptionはドメインモデルに含まれる、記述の項目を表わすクラスdescription_classの事例であり、項目の値が具体的に書き加えられたものである。attributeは属性名と属性値の2項組である。interfaceは他のdescriptionとの関係の記述である。

3.3 制約記述

制約はプロダクトのdescriptionの各属性値に対して付加された条件記述である。条件記述は次のように定義される。

$$constraint = \langle const_id, description, attributes, operator, range \rangle$$

descriptionの属性名attributesに含まれる属性名の属性値 $v1, \dots, vn$ に対してoperatorを作用させた値operator($v1, \dots, vn$)がrangeに対して

$$operator(v1, \dots, vn) \subset range$$

を満足するときconstraintは充足するという。rangeの表現として不等式あるいは集合などが与えられる。

入力制約pre-constraintは条件記述の集合である。

$$pre\text{-}constraint = \langle name, product_name, \{constraint\} \rangle$$

出力記述post-constraintも同様に定義できる。

4 プロセス間相互作用

協調作業や競合解消を行うためにプロセスは様々な相互作業をおこなう。相互作用はプロセス

間での仕様の受け渡しの関係や依存関係によって発生する。図5にプロセス間相互作用の概念をグラフで表わす。

時刻 t でプロセスPとプロセスQの間に発生した相互作用を $int(t)$ と書くことにする。時刻 t はタイムスタンプと呼ばれ、相互作用が発生した順に自然数が割り当てられる。

相互作用は大きく2種類に分類される。

(1)プロダクト

仕様書、検査結果、プログラムなど

(2)メッセージ

要求、通知、依頼など

相互作用として、プロダクトの受け渡しに限定したグラフを基本相互グラフと呼ぶ。図5は基本相互作用グラフの一部の例である。

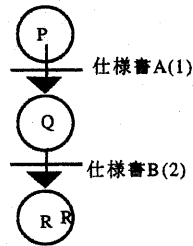


図5 基本インタラクショングラフ

基本相互作用グラフから構成されるプロセスグラフを基本プロセスグラフと呼ぶ。基本プロセスグラフは定型プロセスの骨格を決めるグラフであり、定型プロセスにおける競合解消は基本プロセスグラフに付随する拡張グラフとして定義される。

プロセス間相互作用は以下の7種類に分類される。

- (1)プロダクトの受け渡し Send
- (2)アクション要求 Request
- (3)応答 Notify
- (4)合意 Commit
- (5)拒否 Refuse
- (6)逆提案 Propose
- (7)取消し Cancel

上記7種類の相互作用は、その作用の順序がプロセス間プロトコルとして規定されている。その典型的パターンを図6に示す。

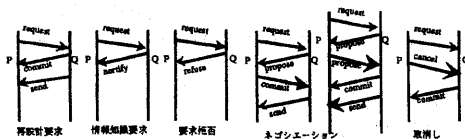


図6 プロセス間プロトコル

5 ソフトウェア開発支援環境の実験例

3章で述べた、ソフトウェア開発に関する、作業プロセスを支援するための場として、分散するオフィスで作業する個々の技術者がネットワークで協調的に作業を進める場として、サイバーオフィスを試作した。作業プロセスの各過程で、プロセス進行状況に対応して、オフィスの構造が決定され、進行と伴に変更されていく。

6 おわりに

地理的に分散したオフィスの間で、一般の利用者がネットワークを用いて協調作業を行うためには、現実のオフィスワークのメタファをネットワークを用いた作業に適用し、現実世界におけるさまざまな行為が通用する仮想的な場を構成することが効果的である。この場は、現実のオフィス環境に必要なさまざまな機能を強化し、現実の空間にこのような仮想的な場を加えた拡張世界と考えることができる。

このような場を定義するためには、作業プロセスのモデル化と、このモデルに基づいて明確かされる協調や競合の関係をj用いて、これを支援する場として、空間を構成することが有効である。本稿では、この考え方に基づいてオフィス環境の試作を行った。

謝辞

本研究の一部は、情報処理振興事業協会の創造的ソフトウェア育成事業「インターネットによる教育・会議のためのやわらかい発想支援環境」プロジェクトの成果に基づいている。

参考文献

- [1]藤田茂, 菅原研次, 木下哲男, 白鳥則郎, 分散処理システムのエージェント指向アーキテクチャ, 情報処論, Vol.37, No.5, pp.840-852, 1996
- [2]Shiratori, N., Sugarara, K., Kinoshita, T., Chakraborty, G., Flexible Networks: Basic Concepts and Architecture, IEICE Trans. Commun., Vol.E77-B, No.11, pp.1287-1294, 1994
- [3]Smith, R.G., The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, IEEE Trans. Comp., Vol.C-29, No.12, pp.1104-1113, 1980
- [4]松下温編, グループウェアの社会・文化的考察, 情報処理学会GW研究報告, Vol.1-1, Apr.1993