スマートアシストを実現するための行動支援ソフトウエアの開発

Design of Human Support Software for Smart Assist

西山　裕之[1)]　大林 真人[1)]　嶺　行伸[1)]　山崎　航[1)]
Hiroyuki NISHIMAYAM　Makoto OBAYASHI　Yukinobu MINE　Wataru YAMAZAKI

1) 東京理科大学　情報メディアセンター（〒278-8510 千葉県野田市山崎 2669-1 Email: nisiyama@imc.sut.ac.jp）

ABSTRACT.
In this research, we develop the software for realizing the smart assistance environment in a house and office space. The system supports human using people's position and action for individualization. We strengthen the ability of sensing in the environment using camera robot and laser sensor for detecting human and its motions. By using such hardware environment, the exact coordinates of the human being and robot can be detected by the laser sensor, and individual human face recognition using the camera picture is enabled. In order to perform recognition of each people and trace of action, the programs of monitoring and offering support for each human are executed.

1　背景

近年，ネットワークの発達により，計算機間だけでなく各種家電製品を情報家電とすることでネットワークに接続し，インターネットからの制御が行われるようになった．また，無線通信の発達により，移動ロボットなどの移動型情報機器もオフィスや家庭内で活用できるようになった．

このような背景の下，人間の生活や仕事をサポートするためのスマートハウスやスマートオフィスを設計する研究が盛んに行われている．例えば，家庭における全ての家電製品を情報家電化することで，外部からのアクセスはもちろん，快適な室温調節や各家電の効率的な電流の出力調節を行うなどのスマートハウスの研究が，家電業界を中心に進められている．また，オフィスにおけるスマート化としては，上記のスマートハウスの機能以外に，移動ロボットなどを用いた手紙や印刷された用紙，書籍などを配達するサービスを実現するなど，オフィスにおける作業を支援するための研究が行われている．

しかしながら，これらの研究では人間の存在する位置などの情報はほとんど考慮しておらず，タスクの発生および目標としての座標としか用いられていない．簡単な赤外線センサなどを環境内に設置することにより，人間の有無を認識することは可能となるが，環境内に存在する正確な人数や個別認識などは行われていない．すなわち，人間は単なるタスクの発生者という役割を担うのみであり，作業の過程においては障害物として認識される以外の影響は及ぼさないものとなっている．そのため，環境内の個人に合わせた支援や，支援における作業の効率性などは配慮することが不可能であった．

このような問題に対し本研究では，オフィスや家庭空間内で作業や生活する人間の支援を個別化および効率化すべく，環境内すべての人間の位置情報や行動内容を詳細に把握し支援する，スマートアシストを実現するためのソフトウエアを開発する．その具体的な方法としては，オフィスに存在する家電機器を情報家電化するだけでなく，オフィス環境内に人感センサ，カメラロボット，そして広範囲レーザーセンサなどを配置し，環境内における感知機能を強化する．このようなハードウエア環境を用意することで，レーザーセンサにより環境内で移動する人間・ロボットの正確な座標の把握や，カメラ画像からの特徴抽出による固体認識を可能にするほか，環境内で移動するロボットと人間の識別も人感センサにより実現可能となる．また，各個人の認識や行動のトレースを行うべく，環境内に認識された固体ごとに監視及び支援を行うためのソフトウエアを起動し，その人物の現在位置，行動内容，行動予測を実現する．本研究により開発された各ソフトウエアは，東京理科大学情報メディアセンターをオフィスの実験環境として使用し，その性能を評価する．

2　目的

本研究の目的は，オフィスや一般家庭などで仕事や生活している人間の行動を詳細にトレースすることで，各個人の作業内容や目的そして状態を把握し，環境内の機器制御や情報提供によりその行動をスマートに支援するというものである．

本研究においては人間に直接センサ系や装置などを備え付けることなく，環境内に備え付けられた各種センサ情報の統合により各個人の行動，状態を把握することを前提とする．これにより，環境内で仕事や生活する人間は，機械的な拘束を一切感じることなく，要求によらない環境からの自動的なサービスのみを提供されることになる．想定されるサービスとしては，端末上における情報提供から始まり，環境内の情報家電制御，移動ロボットや腕方ロボットを用いることによる自動的な配達サービスなど，各機器の擬人的な複合的なサービスが考えられる．すなわち，オフィスや建物全体が一つのロボットとして内部の人間の行動を支援するためのソフトウエアを開発することになる．

以上より，必要とされるソフトウエアは，複数種類のセンサ系の制御および情報の統合による現在の環境認識，個人認識，行動認識に始まり，各個人の過去の行動パターンを認識することによる行動予測，作業内容の推定，目的把握である．そして，環境内に設置された情報家電や用意さ

れた各種ロボットと個別に，協調的に制御するためのソフトウエアも必要となる．

そこで本プロジェクトでは，次の 4 つのソフトウエア開発を行う．

- (1) 各種センサ系の制御・認識ソフトウエアの開発
  本研究では認識対象を人間とするソフトウエアを各センサごとに開発する．特に重点を置くものとしては，カメラロボットとレーザーセンサに対する開発である．カメラロボットを用いた顔画像の認識による個人認識は，スマートアシストを個人ごとに行うためには必須のものであり，また，レーザセンサを用いた形状認識による人間の位置座標や移動速度，および歩幅などの特定は，人間の行動を認識する際に用いられる．
  ソフトウエア開発はシリアルケーブルを介した制御や情報の受信，画像解析を中心とすることから，主として C 言語を用いた実装を行う．
- (2) 情報および制御を統合するための言語開発
  分散されたシステムを統合し，システム間の情報共有を記述するための言語設計を行うとともに，本言語により記述されたプログラムを，最終的に Java プログラムに変換するためのコンパイラを Java 言語により開発する．これは，各システムが異なる OS を備えた計算機上で動作していたとしても，本言語で開発したソフトウエアを介して統合することが可能であることを目的にしている．
- (3) (2) を用いたセンサ系情報の統合及び個人支援ソフトウエアの開発
  (1) で開発した各種センサ系のソフトウエアを個別に管理するソフトウエアを (2) の言語により開発することで，各センサ系ソフトウエア間の情報共有を行うことが容易となる．これにより顔画像の解析に必須となる個人の座標情報，移動中の人物特定など，単体のセンサ系では不可能であった個別の行動認識が可能となる．これに伴い，環境内で把握された個人ごとの行動認識，座標特定が可能となることから，オフィス内に存在する個人ごとの行動を支援するためのソフトウエア開発を行う．本ソフトウエアは担当する個人の行動履歴を蓄積することで，その行動パターンを認識し，支援に役立てるものとする．
- (4) (2) を用いた機器・ロボット群制御ソフトウエアの開発
  各種ロボットの制御ソフトウエアを管理するソフトウエアを (2) の言語により開発することで，機器・ロボット群の統合ソフトウエアを実現する．本ソフトウエアは (3) で開発したソフトウエアと統合され，センサ系との協調作業を可能とし，さらには個人を支援するソフトウエアからの依頼により，作業を遂行する．

以上のように本研究では，センサ情報の収集，認識，制御におけるソフトウエアを開発すると共に，それらのソフトウエアを統合するための言語の設計と統合ソフトウエアの開発を行う．

3 使用する機器

本プロジェクトは，各種ロボット群やセンサ系を統合することにより，人間に対してサービスを行うことを目的としている．本目的の実現に際し，東京理科大学情報メディアセンター内の環境，および既に用意されている各種ロボットやセンサ系を用いた実験を行った．以下に，使用するロボットおよびセンサ系を示す．

3.1 各種ロボット

使用するロボットは，図 1 の 3 種類の移動ロボットと腕型ロボット（マニピュレータ）からなる．それぞれを簡単に紹介すると，次のようになる（括弧内は使用可能なロボット台数）．

図 1: 本プロジェクトで使用するロボットたち

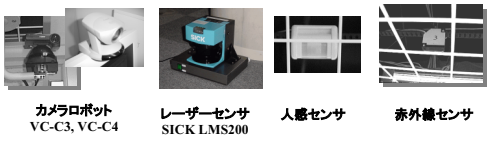

図 2: 本プロジェクトで使用可能なセンサ系

- 移動ロボット NomadScoutII（4 台）
  2 つの車輪を備え，軽量なものなら運搬が可能な円型の小型ロボット．周囲に備え付けられた 16 個のソナーセンサにより，全方向の障害物との距離を把握し，移動時における自律的な障害物回避を可能とする．また，車輪のギヤの部分にエンコーダセンサを備え付けていることから，初期位置から移動した位置までの座標を計測することができる．これらの機能を用いることで，実験環境内で書類の配達やプリンタから印刷された用紙の運搬が可能である．
- 移動ロボット PioneerII（3 台）
  NomadScouotII と運搬能力はほぼ同様であるが，ソナーセンサ以外に下記のレーザーセンサとカメラロボットを備え付けている．主として運搬よりも備え付けられたセンサ系を用いた，環境内の情報収集が中心となる．
- 移動ロボット XR4000（2 台）
  カメラロボット，レーザーセンサおよびマニピュレータ PA-10 を備えた汎用型移動ロボット．センサ系の性能としては上記の PioneerII と同様であるが，上部にマニピュレータを備え付けているため，より複雑な作業が実行可能である．
- マニピュレータ MoveMaster（2 台）
  5 つの間接を持つ腕型ロボット．爪の部分を指定された座標，角度に移動させ，爪の開閉により，ものを掴み，もしくは，放す．主としてプリンタから印刷された用紙を，移動ロボットに受け渡すために使用される．

3.2 センサ系

使用可能なセンサ系は，図 2 の 4 種類のセンサ群からなる．それぞれを簡単に紹介すると，次のようになる（括弧内は使用可能な台数）．

- カメラロボット VC-C3, VC-C4（20 台）
  主として視覚センサとして用いる．カメラの視点角度を上下および左右に変更でき，また，望遠機能を持つ．ビデオケーブルを介して計算機にカメラの映像を送信可能．使用用途として，各研究ブースの端末上に備え付けることで，端末を使用している個人の顔画像の抽出を行う．また，テレビ会議システムにおけるカメラとしても使用される．その他，網天井に備え付けられたカメラロボットは，移動ロボットによる印刷用紙の配達作業を実現するために，移動ロボットを特定の位置まで誘導を行う．
- 広範囲レーザーセンサ SICK LMS200（3 基）
  前方 180 度方向に対して 361 本のレーザーを照射し，

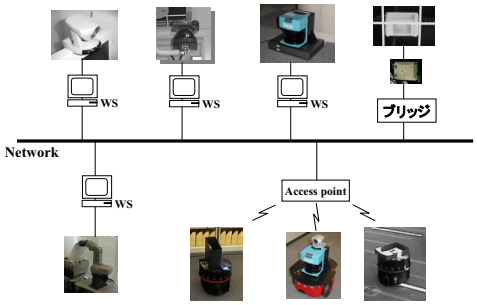

図 3: LAN ネットワークを介した各機器の接続

障害物までの距離を正確に計測する（有効射程距離 50 メートル）．障害物の照射面の形状をほぼ認識することができる．その使用目的は，環境内の人間を含む移動物体を認識することにある．

- 人感センサ（20 基）
  人体から発する赤外線を検知するセンサ．設置した半径 2 メートル以内に人間が接近することで反応する．通路上と各部屋内に設置することで，各ポイントにおける人の存在の有無を把握できる．
- 赤外線センサ（80 基）
  反射鏡を床・壁に設置するか，移動物体（移動ロボット等）上に備え付けることで，特定の座標上を物体が通過したことを認識する．主として，移動ロボットの内部エンコーダの誤差修正を行うために用いる．

3.3 各機器の接続

各種ロボットおよびセンサ系の制御形式は，それぞれが異なっている．例えば，移動ロボットは内部に備え付けられた計算機から制御を行い，マニピュレータやカメラロボット，そしてレーザーセンサは，RS-232C ケーブルやビデオケーブルを介して端末に接続されている．これに対して，多数ある人感センサや赤外線センサは，LON ユニット群により通常のネットワークとは独立したネットワークを形成している．

本プロジェクトでは，これらの様々な機器を統合的に扱うべく，図 3 のように，各ロボット群およびセンサ系を LAN ネットワークを介して接続している．各センサ系の端末はそのまま LAN に接続し，各移動ロボットは無線 LAN のアクセスポイントを介することで，そして LON ネットワークは HEXABINE をブリッジとして LAN ネットワークに接続されている．これにより，互いの端末間での情報のやり取りが可能となる．

4 各種センサ系の制御・認識ソフトウエアの開発

センサ系制御および認識ソフトウエアとしては，顔画像の解析による個人認識システム，レーザーセンサを用いた環境内監視システムを中心に開発を行った．また，顔画像認識システムを開発する過程において，卓上に備え付けられたカメラを用いた，計算機どうしのテレビ会議システムの開発も行った．その他，端末上のキーボードやマウスを操作を行った場合にイベントを発生させるシステムも開発した．

4.1 顔画像認識システム

本システムは，カメラから入力された画像を使用して，予め登録されている個人の識別を実現するためのソフトウエアである．使用する機材は，画像処理用のサーバマシン (PentiumIII 800Mhz, OS: Linux 2.0.39)，画像処理ボード IP5005（日立ソフトウエアエンジニアリング），そして，各部屋に配置されたカメラ VC-C4(Cannon) およびカメラに接続されている画像入力ボードである．なお，使用するカメラは，図 4 のように，各部屋に設置された計算機の端末，もしくはディスプレイ上に備えられていると仮定する．また，顔画像の認識に際して，被験者は認識されることを望んでいることを前提とする．それゆえ，被験者の顔は常に正面を向いており，カメラと顔の距離は極端に変化することは無いものとする．

図 4: 端末前に設置されたカメラロボット

本システムにおける画像処理の流れは次の通りである．

ステップ 1　画像の取得
端末の前に被験者が座り，顔認識プロセスが呼び出されると，端末上に設置されたカメラから 4 枚の画像が取得される．画像データは，ネットワークを通じて画像処理サーバに送信される．画像処理サーバでは，送信されてきた画像を 512×440 の YUV 画像フォーマットに変換する．なお，各端末から送信される画像データは 640×480 である．その後，予め取得されていたテンプレート画像を使用して被験者の特定を試みる．

ステップ 2　計算領域の特定
一般的に，個人認識の計算量のコストは非常に大きいため，これを最小に抑えるために，様々な前処理を行う必要が生じる．そこで，取得されたカラー画像から肌色を基準にして顔画像領域を抽出し，個人認識処理を抽出された領域にのみ適用することによって，処理時間を低減させる．

- 肌色による顔画像領域の抽出
  顔認識における計算量の低減と精度向上のために，取得画像から認識対象となる被験者の顔領域を抽出し，以後，抽出された画像のみを処理対象とする．顔領域の検出には，肌色に相当する色を持つピクセルを基準として使用する．しかし，取得画像内には，顔領域以外にも肌色として認識されるピクセルが多いだけでなく，屋内環境においても，照明変動による色彩の変化を避けることは困難である．本システムでは，顔画像領域の抽出過程における照明変動による影響を低減するために，色相ヒストグラムを用いて閾値を動的に決定する．色相ヒストグラム内の特定範囲内において，最も高いピクセル値を持つ色相を中心として，一定の画素数を含む範囲を決定する境界を，それぞれ上下閾値 (A, B) として使用する．
- 2 値化処理
  肌色に該当する閾値を決定した後は，それをカラー画像データに適応し，肌色部分は 1，それ以外の色は 0 として 2 値画像を構築する．これによって，処理の対象となる領域を明確化する．
- ノイズ処理
  通常，2 値化後の画像には，処理に必要となる領域と共に多くのノイズが混在している．これを除くためにノイズ処理を行う．ノイズ除去には，4

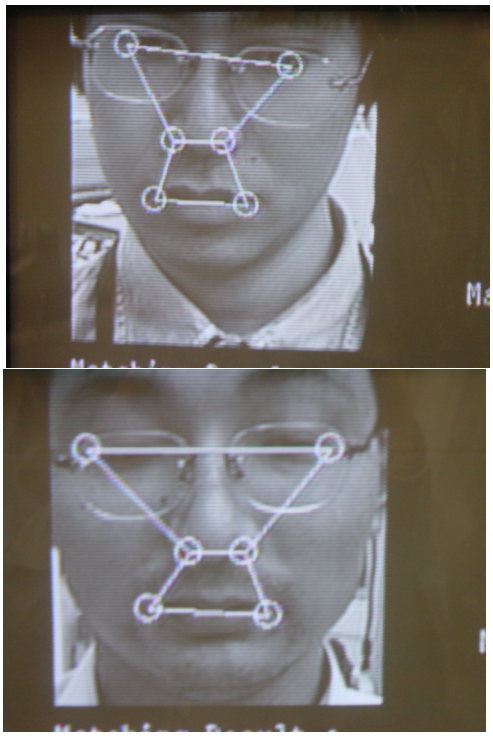

図 5: 認識結果

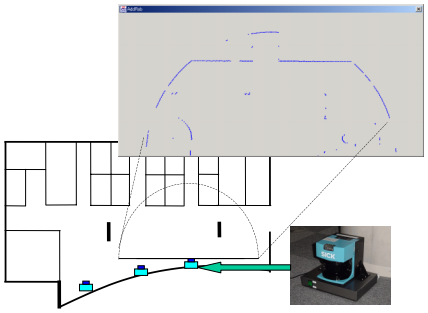

図 6: レーザーセンサによる環境内の監視

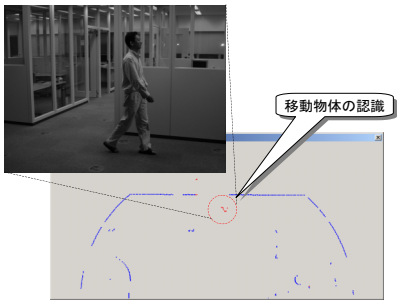

図 7: レーザーセンサによる移動物体の認識

連結ノイズ処理，収縮・膨張処理を使用する．

- 2 値画像データの簡略化
  本システムでは，処理の簡略化と高速化のために，16x16 ピクセルを処理の 1 単位として処理する．これによって，2 値化画像の全領域（512x440）は 864 個のブロックに分割される．このそれぞれについて，白のピクセル数（肌色）と黒のピクセル数（肌色以外）の比をとり，一定値以上の値をとるものを 1，それ以下を 0 として，画像データを 32x27 の 2 次元データに変換する．
- 顔領域の検出
  本システムでは，カメラの位置と被写体となる人間の位置がほぼ決まっているため，画像内における顔画像の大きさは，一定の範囲に収まると仮定することが可能である．このため，顔画像の大きさを 192x208 ピクセル（12x13 ブロック）と定めた．前ステップで生成された 32x27 の 2 次元データを，この矩形領域で走査（ラスタスキャン）し，矩形内におけるブロックの値の総和が最も多い点を顔画像領域として出力する．

ステップ 3　個人認識前ステップによって出力された顔領域に対して，あらかじめ登録されていたテンプレートを使用してマッチングを行なう．テンプレートには，被験者の目尻，鼻，口の両端が用いられる．テンプレートと抽出画像とのマッチングには，正規相関が用いられる．認識による結果を図 5 に示す．個人の特定には，各テンプレートの相関値および，各マッチング座標間距離を評価値として使用する．

以上により、本システムでは顔画像を利用した個人の認識システムを実現した。本システムは実験環境である情報メディアセンター内において利用されており、現在、利用者人数は 10 名となっている。本システムの認識に要する平均時間は一人あたり 4.32sec であり、認識の成功率は 94.4 %となった。ただし、他人と間違える誤認識率は 0 %となっており、以下で実装する様々なサービスの対象者を間違えるといったエラーは生じさることはない。

4.2　環境内監視システム

本システムは，環境内に設置された広範囲レーザーセンサ（SICK LMS200）の情報を用いて，環境内の障害物（人間を含む）の位置，および，その動作を監視するソフトウエアである．使用するレーザーセンサは床上 10cm 前方 180 度方向に対して 361 本のレーザーを照射し，障害物までの距離を正確に計測する機能を有する．その有効射程は 50 メートルであり，近距離の物体であれば照射面の形状を認識することも可能となる．本システムではレーザーの照射を繰り返すことで，照射範囲内に侵入した障害物の形状および位置を計測する．

本プロジェクトでは，情報メディアセンター 2F に，図 6 に本レーザーセンサ 3 台配置を行った．各レーザーセンサは 361 本のレーザー照射と障害物までの距離の計測を行う．図 6 におけるウインドウ画面は，その照射座標をプロットしたものである．これより，通路における壁面がレーザーセンサに反応している様子が伺える．また，図 7 は，この通路上を人間が移動した時のレーザーセンサの反応を示したものである．人が歩行により移動することで両足がレーザーセンサに照射され，図 7 のような反応が得られる．このとき，環境構造物までの照射距離を予め計測しておけば，反応している物体が，人間を含む障害物であるか否かの判定が可能となる．

本システムにおける障害物の認識方法は，次のステップにより実現されている．

ステップ 1　環境のデフォルト値の測定
システムの起動段階で一定時間レーザー照射を繰り返し行い，環境の壁や柱などの構造物までの距離を各角度ごとに測定を行う．

ステップ 2　障害物の測定
レーザー照射により各角度の照射距離において，ステップ 1 で測定した距離以下が計測された場合，障害物を測定したとしてステップ 3 へ進む．計測されない場合はステップ 5 へ進む．

ステップ 3　障害物の認識
測定された各角度の照射距離より，それぞれの照射位置の座標を計算する．反応した角度と障害物までの距離より，環境内に侵入している障害物の数とそれぞれの中心座標および大きさを認識する．このとき，人間が照射範囲に侵入した場合，移動方向によっては両足が個別に認識されてしまうため，大きさが 200mm 以下の 2 つの障害物が 1500mm 以内で認識された場合，同一の障害物として認識している．

ステップ 4　新たな障害物の判定
ステップ 3 で認識された障害物が，既に認識された障害物と同一の物であるかを判定する．このとき，新たに判定された障害物は，それらの座標および形状が記録される．

ステップ 5　既知の障害物の存在判定
既に認識されていた障害物が現在も存在するかを判定する．存在しないと判定された障害物は記録から削除される．

ステップ 6　ステップ 2 へ戻る

以上の流れにより，本システムでは障害物の認識と，認識された障害物の消滅（移動）の情報が生成されることになる．これらの情報を用いることにより，照射範囲内の人間の移動をトレースすることが可能となる．

本システムはレーザーセンサに接続された計算機上でのみ実行可能である．また，レーザーセンサの制御ライブラリは Linux 用の C 言語を中心に用意されているため，その実装には C 言語へのコンパイルが可能な MRL(Multi Robot Language) 言語 [2] を用いて行った．

本システムにより，レーザーの照射範囲内で移動中の人間の座標を正確に特定でき，環境情報を用いることにより，対象が照射範囲外に移動した場合も，いずれかの部屋に入ったか，通路を通り抜けていったかの判定が可能である．しかしながら，今回の実装では全てのレーザーセンサを同一方向から使用したため，一人の人間が障害物として認識されると，その裏側の障害物情報が未知となってしまう．そのため，複数の人間が並行に移動した場合や，通路間で交差を行なった場合，誤認識を行う場合があった．なお誤認識の状態は，顔画像認識システムの再認識により修正される．

4.3　テレビ会議システム

本テレビ会議システムは，本プロジェクトにより実現可能となるサービスの一環として開発を行った．本システムの本来の役割は，各部屋に配置されたカメラロボットからの顔画像を，要求に応じて顔画像認識システムに送信することにある．本プロジェクトでは本システムの機能の拡張を行い，映像および音声を送受信を可能とした（図 8 参照）．

本システムでは，映像と音声の送受信に RTP(Real-time Transport Protocol) と呼ばれるビデオ会議システムで標準的に使用されるプロトコルを実装している．顔画像認識システムへの画像送信を要求された場合は，512 × 440 のサイズの静止画像を RGB24bit 形式のものを送信する（画像サイズは 660Kb）．テレビ会議システムとして用いられる場合は，映像と音声の送受信をリアルタイムに行うべく，320 × 240 もしくは 176 × 144 の映像を Motion JPEG または H.263 により，音声を MPEG Audio により圧縮を行った後，送信している．

本システムの利用方法としては 3 つのパターンが用意されている[*1]．1 つ目は一般的な方法であり，指定した端末との回線を直接開くものである．2 つ目は計算機のログオン情報を用い，指定した個人の使用している計算機をブロードキャストにより捜し出した後，回線を接続する．3 つ目は，以下で説明する個人支援ソフトウエアを介したものであり，個人を指定すると，その人物の存在する部屋の端末との回線を開く．

図 8: テレビ会議の様子

本システムは Java 言語により実装されているため，JavaVM(JDK1.2 以降) が実行可能で，映像および音声が入力可能な端末であれば，基本的に実行は可能となっている[*2]．

4.4　キーボードおよびマウスの監視システム

本システムは，キーボードやマウスが操作された場合，ネットワークを介して，その事実を通知する機能を持つソフトウエアである．本システムは Java 言語により実装され，上記のカメラの接続された計算機上に実装するものとする．本システムの用途は，上記の顔画像認識システムを使用する際のスイッチとしての機能が中心となっている．

5　情報および制御を統合するための言語開発

分散されたシステムを統合し，システム間の情報の共有を記述するための言語設計として，その記述形式や操作的意味論は，我々が開発したマルチエージェント言語 MRL (Multi Robot Language) と基本的に同様のものとする．MRL で記述されたプログラムは MRL トランスレータにより最終的に C 言語に変換され，Solaris および Linux 上で実行可能となる．これに対し，本プロジェクトで設計する言語 JMRL(Java Multi Robot Language) のトランスレータは，Java 言語で開発し，JMRL プログラムを Java 言語に変換するものとする．これにより JMRL プログラムは，JavaVM の実装されている全ての計算機上で実行可能となる．ここで，MRL プログラムは C 言語に変換されることから，内部コードとして C のプログラムの組み込み，および C ライブラリの利用が可能である．これに対して JMRL は Java プログラムを組み込むことは可能であるが，C プログラムや C ライブラリをそのまま組み込むことはできない．そのため，既に作成された MRL プログラムの中で，制御系など C 言語のプログラムを組み込んでいるものは，そのまま Java 言語に変換することはできない．

JMRL プログラムから Java プログラムへの変換には，Java のパーサジェネレータ JavaCC を用いる．JavaCC におけるパージングにおける付帯動作中に，JMRL から Java への変換用のテンプレートオブジェクトのインスタンスを作成し，パージング終了時にそれらのインスタンスの情報を元に，対応する Java プログラムファイルを出力する．なお，出力される Java ファイルは実行前にコンパイルされる必要があるが，実行機種に依存した出力は行わないため，一般的にはどのホストでも実行できる．パーサは，拡張 BNF と呼ばれる JavaCC 独自の形式で記述し，JavaCC が出力する Java ファイルをコンパイルすること

[*1] 各部屋の端末に本システムが実装されているものとする

[*2] ただし，MacOS X では映像の送信は行えない．

でパーサとしての動作を行う．以下に，JMRL システムへの入力として JMRL の記述形式を，次にパーサからの出力形式及び動作について述べる．

5.1 JMRL システムへの入力

複数のエージェントを定義するために，節の集合からなるそれぞれのエージェントの定義は以下のようになる．

```
agent(AgentName){
   節 1.
   節 2.
   ...
   節 n
}
```

ここで節は以下のような形式をしている．この形式は，KL1 及び MRL と同様な形式である．

$$H : -G_1, G_2 \cdots, G_n | B_1, B_2, \cdots, B_m.$$

H（頭部）, G 及び B のそれぞれはアトムと呼がれる形式で，アトムは，述語名と任意の数の引数からなる．引数は，項 (T) からなり，T は定数，変数，及びリストを含む構造体からなる．なお，以下では，:-から | までをガード部，それ以降をボディ部と呼ぶ．JMRL では，アトムのうち，特別なものとして，”new”, “run”, “java”, ’’’^’’’, ’’’!’’’ が予約されている．これらはそれぞれ，新たなエージェントの生成，エージェントの状態宣遷移ルール記述，外部の Java コードの呼出し，上位エージェントとのメッセージ通信，下位エージェントとのメッセージ通信の役割を持つ．

- new アトム
  節のうち，頭部で new アトムが出現する場合，その述語はそのエージェントが生成された際に呼び出される．また節のうちのボディ部で new アトムが出現した場合，それは新たなエージェントを生成することを意味する．例えば，root という名前のエージェントで new 述語において，新たに 2 つのエージェント robot と laser を生成する場合には，以下のような記述となる．

```
agent(root){
 new:-
  new(robot(robot1,p1,192.168.1.90,12345)),
  new(laser(laser1,192.168.1.90,54321)),
  run().
 run():-
  ....
}
```

- run アトム
  節の頭部に run アトムがある場合は，その述語がエージェントの状態遷移を記述していることを意味する．述語の引数はエージェントの内部状態を，（存在すれば）ガード部は，他のエージェントとのメッセージなどのエージェントの外部状態をそれぞれ表しており，内部状態，及び外部状態がその形式にマッチした場合に，ボディ部が実行される．ボディー部はそれぞれ成功することが期待され，prolog に見られるようなバックトラックは行なわれない．以下の記述は，内部状態が任意であり（run 述語のすべての引数が変数であることに相当），親エージェント（そのエージェントを生成したエージェント）からのメッセージが，$request(gotogoal(X), Sender)$ という形式の構造体であった場合には，run 述語の 4 番目の引数に相当するエージェントの内部状態のリストにそのメッセージを追加したものに置き換える (ボディ部における $NR = [request(gotogoal(X), Sender)|Request]$ が相当)，再帰的に実行を続けるという意味になる．

```
run(Name, Task, Point, Requested) :-
  ^request(gotogoal(X), Sender) |
  NR = [request(gotogoal(X), Sender)|Request],
  run(Name, Task, Host, NR).
```

- ^述語
  ^述語は，上位のエージェントとのメッセージ通信を行なうためのアトムである．ここで，上位のエージェントとは，agent A が agent B を生成した場合，B からみた A のエージェントのことを言う．節内のガード部，及びボディ部で用いることができる組み込み述語である，run アトムや new アトムと異なり節の頭部に，定義として記述することはできない．このアトムがガード部で出現した場合は，上位エージェントからのメッセージについての条件を表しており，ボディ部で出現した場合は，上位エージェントにそのメッセージを送信することを意味している．メッセージは，上の run 述語で示したような形式によって，通信が行なわれる．
- !述語
  上記 ^述語と対をなすアトムであり，下位のエージェントとの通信を行なうためのアトムである．ここで下位のエージェントとは，自分が生成したエージェントのことをいう．このアトムも，ガード部，及びボディ部でのみ記述することができ，それぞれの意味は，下位エージェントからのメッセージのマッチング，及び，下位エージェントにメッセージの送信を意味している．
- java 述語
  java 述語は，外部で定義された Java 言語で記述されたクラスを呼び出すためのアトムであり，ガード部及びボディ部で利用できる．現在の実装では，Java の記述自体を本言語中に直接書くようにはなっておらず，外部ファイルに後にしめすような一定の書式にしたがって記述する必要がある．java アトムでは，引数として一つの項を java コードに送ることができる．複数の項を送信したい場合には，リストや構造体を用いる．例えば，$java(foo(X))$ という呼出をすると，システムは，$JE\_foo.class$ を Java のリフレクション機能によってロードする．このロードするクラスは，抽象クラスである JEObject クラスの拡張であることが期待され，システムはその抽象メソッド exec() を実行する．ユーザは，この exec メソッドを実装することで，Java の機能にアクセスする．例えば，X にバインディングされている，値を文字列の形でとりだし，Java の機能によって表示するためには，以下のように $JE\_foo.java$ を記述する．ここで，コンストラクタの記述形式は，JE_Object クラスで定義されている，$getValS()$ メソッドやそれを実現するための設定をする $set()$ メソッドが正しく動作するために必要な記述であり，この書式に従う必要がある．

```
class JE_foo extends JEObject{
   public JE_foo(Term t, Binder b){
       set(t,b);
   }
   public boolean exec(){
       String str = getValS(‘‘X’’);
       if(str!=null){
         System.out.println(str);
         return true;
       }
       else{
         return false;
       }
   }
}
```

5.2 JMRL システムからの出力とその動作

本節では前節で述べた入力から，Java のファイルへの出力を行なうまでの変換形式と，出力される Java ファイルの動作について述べる．図 9 は，出力される Java ファイルがどのような構成になっているかを示している．定義されたそれぞれのエージェントは各々クラスとして定義され，定義されている run 述語の数だけ対応する RunObj

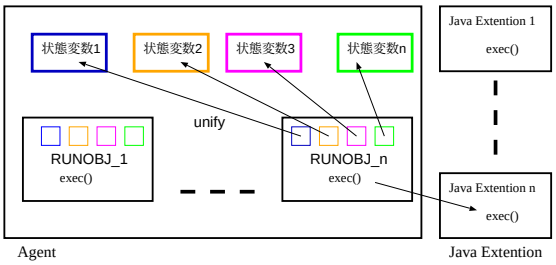

図 9: JMRL により出力される Java ファイルの構成

と呼ばれる内部クラスが生成される．以下に，出力されたエージェントの動作の概要を示す．

step1 それぞれのエージェント内では，Java のスレッドによって，RunObject がループ内で一つ選択される．

step2 現在のエージェントの状態変数と選択された RunObject 内の状態変数の制約が，マッチするかを論理プログラミングで用いられるユニフィケーションによって調べる．図では，それぞれの色分けされた変数への矢印がそれに相当している．ユニフィケーションに失敗した場合は step1 へ．

step3 ユニフィケーションが成功すると，外部からのメッセージを参照し，同様にユニフィケーションを行なう（その処理は図では省略されている）．失敗したら step1 へ

step4 ヘッドとガード，2 種類のユニフィケーションが成功すると，RunObject 内に定義されているボディー部の述語が実行される．ここで実行される述語は必ず成功するようにプログラムされていなければならない．

step5 run 述語が再帰的に呼ばれている場合に限り step5 へそうでない場合は終了．

step5 エージェントの状態を，ユニフィケーションによって代入された RunObject 内の状態で置き換えて step1 へ．

6 センサ系情報の統合及び個人支援ソフトウエアの開発

本ソフトウエアは，環境内に存在する個人の位置や状態，および過去の行動や受けたサービスなどの情報を，個別にそして動的に管理を行うシステムである．本システムは，各種センサ系とネットワークを介して接続することにより情報の送受信を行う．

本支援ソフトウエアは，役割の異なる複数種類のプロセスと，登録された各個人の経歴情報を記録するデータベースから成り立っている．

- 各種センサ系の管理プロセス．
  ネットワークを介したソケット通信により，各種センサと情報交換を行うためのプロセスである．各センサごとにプロセスの定義が行われており，同一種類のセンサ系を複数用いる場合は，同一種類のプロセスが，そのセンサの数だけ生成される（それぞれ異なる識別名を持つ）．具体的には，顔画像認識システム，環境内監視システム，テレビ会議システム，キーボードおよびマウスの監視システムの 4 つのセンサ系を管理するためのプロセスが個別に定義されており，使用するセンサの数だけ，それぞれのプロセスを生成する．
- モニタープロセス
  個人情報を動的に管理し，支援するためのプロセスである．上記のセンサ系の管理プロセスから情報を入手することにより，管理している個人の位置情報の更新を行う．また，位置情報の認識を正確に行なうための依頼をセンサ系に行う．その他，管理対象からテレビ会議システムなどに対するサービスの依頼が発生した場合，管理対象の代理人となり，各システムへの依頼を行う．
- モニター生成プロセス．
  モニタープロセスを生成するためのプロセスである．環境内に新たな人間が認識されたとき（その人間のモニタープロセスが存在しない場合），新たにモニタープロセスの生成を行う．また，センサ系から入手した情報に対して，モニタープロセス間で競合を解消するためのスーパーバイザとしての機能も有する．

例えば，本システムにおいて個人が識別されるまでの流れは，次のようになる．

ステップ 1　人間の認識
レーザーセンサの照射範囲内に人間が侵入すると，環境内監視システムからモニター生成プロセスに対して，その座標情報が伝達される．

ステップ 2　モニタープロセスの生成
その座標情報に該当するモニタープロセスが存在しない場合，新たな人間が環境内に侵入したものと判断し，その人物を管理するためのモニタープロセスを生成する．この段階で，その人物の個体名は unknown となっている．

ステップ 3　レーザー照射範囲外への移動
管理対象が全てのレーザーセンサの照射範囲外へ移動した場合，最後に認識された座標がいずれかの部屋の前であった場合はステップ 4 へ．それ以外はステップ 5 へ．

ステップ 4　顔画像による個人の特定
モニタープロセスは，管理対象が入った（と推定される）部屋の端末の，キーボードおよびマウスの監視システムに対して依頼を行い，何らかの操作が行われた場合はメッセージを受けとる．そのメッセージを受けとった場合，管理対象が端末前に座ったものと判断し，その端末のテレビ会議システムおよび顔画像認識システムに対して個体認識の依頼を行う．これにより，管理対象の個体名が unknown から認識された人物名に変更され，その個体名の登録されている経歴情報をデータベースより引き出す．

ステップ 5　環境外への移動
通路等を通り抜けて，管理対象が環境外へ移動を行った場合，そのときの個体名が unknown であったならば，そのモニタープロセスは消滅する．また，個体名が特定されていた場合，これまでの経歴情報をデータベースに格納後，そのモニタープロセスは消滅する．

以上のように本システムでは，各種センサ系との状況に応じた情報のやり取りにより，環境内の人間を個別に管理することが可能となっている．

7 機器・ロボット群制御ソフトウエアの開発

使用するロボットとして，移動ロボット NomadScoutII 4 台，移動ロボット PioneerII 3 台，そしてマニピュレータ MoveMaster 2 台を中心とする．これらのロボットの制御システムは既に C 言語により開発されている．

本ソフトウエアは，3.2 の言語を用いてロボット群制御の統合を行うためのものである．その役割としては，個々に独立に制御されているロボットやセンサ系などの各種機器を統合的に制御すべく，各機器どうしでの情報のやり取りを可能にすることにある．すなわち，本ソフトウエアは使用する機器の数だけ独立したプロセスを生成し，各プロセスが各機器の代理人として他の機器のプロセスと情報のやり取りを行う．本プロジェクトでは，このプロセスを

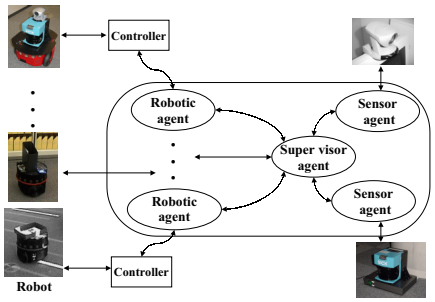

図 10: マルチエージェントシステムによる各機器の統合

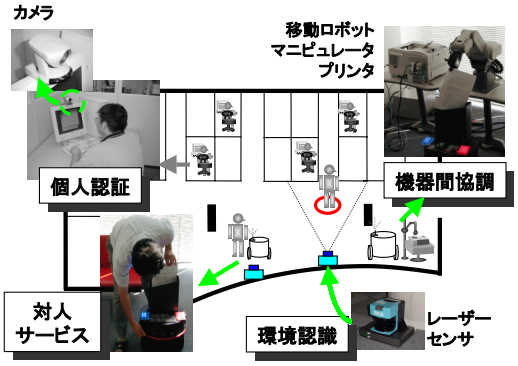

図 11: 本プロジェクトにより実現したサービスのイメージ

エージェントと呼び，複数のエージェント群からなるシステムをマルチエージェントシステムと呼ぶそれぞれのエージェントは各機器から指示を受けることで，他の機器のエージェント間とメッセージのやり取りを行い，各機器間の協調動作や競合回避を実現するための仲介を担う．

本プロジェクトにおける各機器を統合するためのマルチエージェントシステムは，図 10 のように実装されている．本システムは使用するロボットおよびセンサ系の台数分エージェントを生成する．また，途中で稼働している移動ロボットをバッテリの充電などの理由で使用不可能となる場合はエージェントを消滅させ，逆に復帰したり新たにロボットを投入する場合は，動的にエージェントを生成することで対処可能である．現在のシステムでは各エージェントは一つのサーバーマシン内で生成および動作を行なっている．

各エージェントの役割は各機器すべての情報交換を引き受けるのではなく，協調もしくは競合相手を捜し出すことにある．そのため，協調相手が確定した場合は，それぞれの機器の制御システムどうしがソケット通信により独自に回線を接続し，本マルチエージェントシステムを介することなく，情報のやり取りを行なう．これにより，本マルチエージェントシステム内における情報の集中は回避できる．

7.1 本マルチエージェントシステムによる具体的なサービス内容

本プロジェクトにより実現したサービスのイメージは，図 11 のようになっている．具体的には，次の 4 つのサービスを実現した．

- 環境内の個人認証
  端末上に備え付けられたカメラからの映像により，その端末上で作業を行っている個人特定を実現した．さらに，レーザーセンサによる環境内の移動物体のトレースを組み合わせることで，一度認識された個人の行動を，実験環境内において把握することが可能となった．
- プリンタからの印刷物の配達作業
  移動ロボットを用いた印刷物等の配達システムは，我々は既に情報メディアセンター内で開発を行っている [1]．しかしながら，従来のシステムは，印刷を行った個人の部屋の前まで運ぶだけの機能しか有していなかった．
  本プロジェクトの成果により，環境内の全ての個人の位置情報を把握できるため，印刷した個人の部屋に対してではなく，その存在するエリアへの配達が可能となった．また，その個人が実験環境外に移動してしまった場合も，改めて環境内にて認識が行われるまで，他の配達作業を優先的に行うことが可能となった．
- 個人どうしの配達作業
  従来，個人どうしの書類や本などの移動ロボットを介したやりとりは，その個人の存在する研究ブースに移動ロボットを呼び出し，荷物をロボット上に置いた後に，対象となる個人の存在する研究ブースへ移動させることで作業を完了する．これに対して，本プロジェクトの成果により，上記の印刷物の配達と同様にして，各研究ブース間の配達ではなく，個人間の場所に捕らわれない配達作業が実現できた．
- テレビ会議
  ここでのテレビ会議とは，卓上の端末を用いたテレビ電話と同様である．ここで，従来のシステムとことなるのは，通話時に通信したい相手の部屋を指定するのではなく，相手の名前を指定することにある．これにより，現在，環境内に対象となる相手が存在する場合，そこにネットワークに接続された端末が存在するならば，その端末上との通信が行なわれる．

8 おわりに

本研究では、オフィスや家庭空間内で作業さ生活するユーザの支援を個別化および効率化することを目的として、環境内すべての人間の位置情報や行動内容を詳細に把握し支援する、スマートアシストを実現するためのソフトウエア群を開発した。

本研究の今後の展望として、現在，東京理科大学野田図書館における 24 時間サービスの実現へ向けて，図書館内のスマート化およびセキュリティ強化案として，本プロジェクトの成果の適用が決定している．その目的は，オフィス空間におけるサービスを対象としたものとは異なり，深夜でも図書館が安全に利用できることにある．図書館は不特定多数の人間が出入りするため，本プロジェクトの顔画像認識システムを導入することは困難であるが，各種センサとカメラの連動により，図書館への来訪者の顔画像を抽出し保存することは可能である．また，レーザーセンサ，カメラを用いた環境内の監視を中心とし，レーザーセンサを装備した移動ロボットが，図書館内を徘徊する統合システムの構築を予定している．

参考文献

[1] F. Mizoguchi, H. Nishiyama, H. Ohwada and H. Hiraishi: Smart Office Robot Collaboration based on Multi-agent Programming, *Artificial Intelligence*, Vol. 114, pp. 57-94, 1999.

[2] 西山 裕之、大林 真人、大和田 勇人、溝口 文雄：ロボット間協調を容易に実現する並列論理型プログラミング言語の設計、日本ロボット学会論文誌、Vol.19, No.5, pp.620-631, 2001.