AN AGENT-BASED APPROACH TO MODELLING PEDESTRIAN BEHAVIOUR

NICOLE RONALD, LEON STERLING, MICHAEL KIRLEY

Department of Computer Science and Software Engineering,
The University of Melbourne, Victoria 3010,
Australia

Abstract: The modelling of pedestrian behaviour in a real-world environment is a complex problem, mainly due to the unpredictable nature of human decision making. Agent-oriented simulation moves away from traditional all-knowing and "controlling" simulations and towards reality, where pedestrians exhibit different behaviours depending on their knowledge of the environment and other personal characteristics. We investigate the behaviours that pedestrians may exhibit, the different techniques used for pedestrian modelling, and the appropriateness of each technique for particular domains. The classification framework developed will play a role in the decision making process for planning and design of pedestrian areas. We then explore the agent-based approach in more detail, in particular the belief-desire-intention (BDI) architecture, by presenting the development of a sample model using Prometheus, an agent-oriented design methodology, and JACK, an agent-oriented programming language. Although the BDI architecture is useful for high-level decision making, further work is required in representing and updating the environment.

Keywords: Transport modelling, agent-based simulation, BDI architecture

INTRODUCTION

There is a need to model pedestrian behaviour for a range of applications including event planning, resource usage, and urban planning. For example, the organisers of a large sporting event require information on what areas are likely to be congested so that management strategies can be developed and tested before the event. Similarly, the designers of a shopping mall might be interested in how people move around their intended design so they can place shop entrances and seating in useful locations.

Most of the models developed so far fall in one of two categories: large-scale models producing aggregate results and smaller, disaggregate models. Less work has been done for pedestrian modelling in between these two extremes. It is useful to have an understanding of how people will behave in certain situations, such as at a large sporting event or after a change to their environment, during the planning process.

Agents are a recent development in software engineering. An agent can be defined as "a piece of software that is:

- *Situated* - exists in an environment
- *Autonomous* - independent, not controlled externally
- *Reactive* - responds (in a timely manner!) to changes in its environment
- *Proactive* - persistently pursues goals
- *Flexible* - has multiple ways of achieving goals
- *Robust* - recovers from failure
- *Social* - interacts with other agents"
  [Padgham and Winikoff, 2004, p3]

Using agents as the main objects in simulation is known as multi-agent based simulation. This field has many applications, but many researchers are using real-world systems such as transportation to demonstrate simulation techniques and theory. Parunak et al. [1998] states "ABM [agent-based modelling] is most appropriate for domains characterized by a high degree of localization and distribution and dominated by discrete decisions. EBM [equation-based modelling, or what we call mathematical modelling] is most naturally applied to systems that can be modeled centrally, and in which the dynamics are dominated by physical laws rather than information processing."

Agent-based simulation appears to have potential for pedestrian modelling. Each pedestrian could be modelled as an autonomous agent with its own knowledge and goals. This representation is closer to reality than traditional simulation methods (eg. discrete event simulation) as it requires less abstraction. The purpose of our research is to compare approaches to modelling pedestrian behaviour, focusing on agent-based approaches and the benefits to disaggregate modelling. The agent-based approach we discuss in this paper uses the belief-desire-intention architecture.

This paper is divided into several sections. Firstly, we examine at the need for pedestrian modelling and the properties of the pedestrian system. We then review some existing models and techniques and present recommendations for

selecting an approach given a particular environment. Following this, we investigate the usefulness of the agent-based approach, more specifically the BDI architecture, for a small domain. The design and implementation of our model is covered, followed by a discussion.

BACKGROUND

We discuss some of the interesting properties of pedestrian systems and the history and need for transport modelling, followed by a survey of pedestrian modelling approaches.

System properties

The Australian Pedestrian Council defines a pedestrian as "any person wishing to travel by foot, wheelchair or electric scooter, throughout the community" [Australian Pedestrian Council, 2004]. There are many reasons for walking, and the manner in which we walk changes depending on the purpose.

> "Commuters scurry; shoppers meander; bushwalkers trek; power-walkers stride; lovers stroll; tourists promenade; protesters march ... But we all walk."
> [Australian Pedestrian Council, 2004]

Transport systems are constrained, sometimes weakly. For instance, people cannot cross the road whenever they feel like it - they should find a suitable place (such as an intersection) and wait until it is safe. They also should travel on the pedestrian network (eg. designated paths) at all times, however if it becomes too congested, pedestrians may overflow onto the road or surrounding parkland. A stricter constraint is that pedestrians cannot walk through solid objects or on water.

Pedestrian behaviour is usually individual-based and autonomous. In most cases, we decide where we want to go and how to get there without being told explicitly. Individually, in the system, there are cooperative elements (letting someone go through a door first, moving out of the way for a faster person) and competitive elements (pushing to get out of a stadium quickly).

Pedestrians often make unconscious decisions that are difficult to explain or measure. They move at a much smaller scale and in a less constrained manner than other vehicles, meaning techniques developed for modelling other modes of transport cannot be translated to pedestrians easily. Computational modelling of pedestrians is therefore difficult due to the complex and random nature of their movement.

Transport Modelling and Planning

Transport planning is a decision-making process in which the problem is identified, strategies are developed, modelled and evaluated, and the most preferable solution is recommended for implementation [Ort´uzar and Willumsen, 1994]. Figure 1 shows the steps in the planning process. The evaluation of strategies involves examining the effects on stakeholders and the environment and can be undertaken in many ways. Several strategies could be selected for a trial run, however physical tests are not always feasible. For example, it is impractical (not to mention expensive) to build several versions of a pedestrian bridge in order to evaluate the option with the most benefit. In these cases, computers are used to set up an "artificial reality" - a computer model or simulation - which is used to test different strategies.

The inputs to transport models usually include demographic data (age, sex, place of residence, type of work), land use data (assists in determining attractability of certain locations), and demand drivers (what locations are popular, what times are people travelling). This data is sourced from public data, such as census and land-use data, and data collected specifically for the model, such as the results of an observation or a questionnaire survey. The results of transport models can include economic, environmental and social data.

Pedestrian modelling is frequently used for making decisions regarding the planning, design, and management of pedestrian areas. The outputs of these models can include flows on certain routes, entry and exit counts, and level-of-service graphs. Other factors, such as costs and environmental effects, are combined with the model outputs to help management make decisions.

A common use for pedestrian models is in the organisation of large, usually once-off events. These events include Olympic and Commonwealth Games, other´international sporting events such as tennis, grand prix and world championship events, and street festivals. Planning for these events is a difficult task, as there is little historical information about pedestrian behaviour and the only opportunity to collect data is at the event itself. The organisers often have several planning issues, such as the location of security barriers and food stands and whether to build or upgrade infrastructure. Modelling can assist in developing mitigation plans or decide between two infrastructure scenarios.
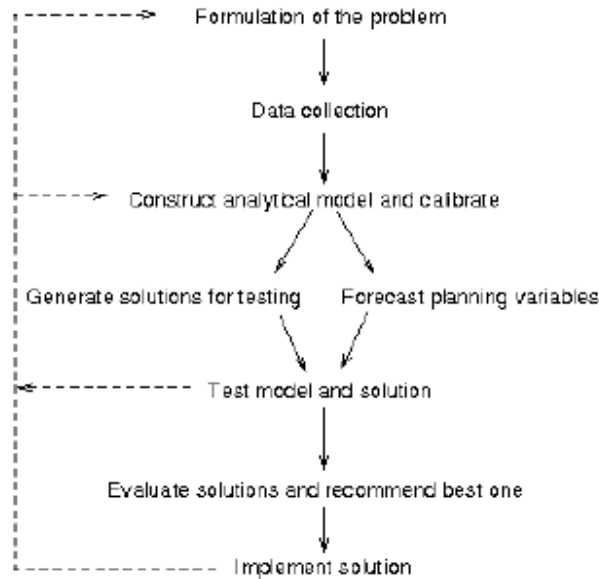
Figure 1: Key steps when using models in the decision making process for transport systems
[Ortúzar and Willumsen,1994].

In another example, the designer of a new shopping mall would be interested in what locations people are likely to be attracted to. This information could be used to place artwork or amenities such as rubbish bins or seats in appropriate positions. Designated footpaths and garden features may be redesigned in a way that people have to pass less attractive shops or alternatively as many shops as possible.

Most transport modelling techniques have focused on
the modelling of cars and vehicles on the road network [Batty, 2001], due to the detail required for a realistic pedestrian model. The recent interest in environmentally sustainable transport modes however, has led to an interest in providing better infrastructure and facilities for cyclists and pedestrians and therefore a need for improved methods of modelling their behaviour.

Review of Modelling Approaches
Many approaches have been used for modelling of pedestrian behaviour [Harney, 2002], which can be divided into two schools. The first school is the "civil
engineering" approach. This is concerned with forecasting demand so that decisions can be made about provision of new infrastructure. The main outputs of these models are numbers of people travelling along various routes and the algorithms used are frequently based on traditional vehicle modelling algorithms. They are generally macroscopic or aggregate models, where the smallest detail of a pedestrian's movement

is the locations they visited and the paths they used to get there, similar to the detail of vehicle models.

The second school is the "architecture/urban geography" approach. This group is interested in how people move around areas, in particular how design and location of certain attractions influence their movements. These models are usually microscopic, in that they model a pedestrian's path in more detail, usually in terms of steps or small grid squares. They are usually developed for small areas only, although some have been expanded to cover entire cities. Some models combine both approaches and as a result are very flexible regarding the type of areas they can model.

In terms of classification, we have identified four main approaches to modelling pedestrians: mathematical models, cellular automata/swarm models, traditional time-based simulation/microsimulation, and agent-based simulation.

Mathematical models are based on mathematical and physical equations. One form of the mathematical model is based on physical formulae of motion. The modelled pedestrians have properties, including current position and speed. These models use the notion
of force to move pedestrians around. Some of the force parameters include acceleration force, repulsive effects of boundaries and objects, repulsive effects of other people, and attraction to groups and objects. models. Helbing used the notion of attraction and repulsion to model microscopic behaviour and has developed complex equations to model a range of pedestrian behaviours, com-

monly referred to as the "social force" model [Helbing et al., 2001]. He observed that streams formed in the crowds and resembled fluid flow. Some of the basic equations are shown in figure 2. Hoogendoorn and Bovy [2004] used the same starting point of basic mechanics formulae and developed a three-layered model encompassing activity choice, way finding, and walking. This model attempts to minimise the cost of walking and was applied to a multi-modal transfer station. Another mathematical model is the the use of statistical methods to estimate the flows on particular links. This method has been used with success by Intelligent Space [2005] and Space Syntax [2004], mainly for la ger models of pedestrian movement in cities.

A similar approach is the use of cellular automata (CA), where pedestrians occupy cells on a grid and move according to some simple rules. This is a variant of the traditional CA models (eg. Conway's Game of Life) where cells have a state which changes depending on the state of the surrounding cells and there is no explicit movement involved. These models gene ally use a grid-based model where one person can occupy a cell at once (as shown in figure 3), hence the representation of large areas requires a large number of cells. Most of the models based on this approach used the Schreckenberg-Nagel approach to modelling vehicle traffic using CA as a starting point [Nagel and Schreckenberg, 1992]. Recently CA were used to model pushing behaviour in crowds leaving a sporting event [Henein and White, 2004]. The environment representation included two layers of information: a static layer pointing to the nearest exit and a dynamic layer containing the general direction of the crowd. Each pedestrian uses the information at their particular cell to decide where to move next. An issue with CA models is what to do when a collision occurs, as this behaviour is not present in the traditional CA model. This has been explored for basic movement in a corridor but with the pedestrians learning what to do when a collision occurs [Narimatsu et al., 2004]. Cellular automata have been shown to be useful for disaggregate models with minimal activity choice. AlpSim [Gloor et al., 2004] combines a cellular automata approach with aggregate representations of the environment to take advantage of the benefits of multiple map representations, specifically higher-level planning which is very complex using only a grid.

Traditional time-based simulation has also been used in industry. In this approach, all pedestrians are controlled by an object who tells them where and when to move. It is useful for aggregate models as all the information can be easily combined. These models generally use a graph-based representation of the environment, where possible paths are represented as edges and decision points as nodes. PAXPORT, developed by the consulting firm Halcrow (http://www.halcrow.com), has been used to model pedestrian movements in airports, train stations, and sporting venues. It provides aggregate measures of flow and level-of-service in a graph-based environment. It was recently used to model behaviour in the Sports and Entertainment Precinct, Melbourne in order to select a design for a new bridge to be built for the 2006 Commonwealth Games [Ronald, 2004]. Figure 4 shows example output from PAXPORT that displays the level-of-service in the area at a given time. The block structure of the model, a common feature of aggregate models, can be seen.

Agent-based simulation is frequently used for models where there are distinct entities who are interacting with each other in a environment. The entities in the model are software agents, who have the ability to perceive, make decisions, act, and learn from their environment. The approach has been used for economic, social, business and logistics modelling, where there is a lot of interaction with the environment and other agents and also complex decision making involved. Legion [2004] is proprietary software specialising in modelling crowd behaviour and was first developed as a model of ingress and egress from events [Still, 2000]. It treats each person as a "virtual person" who senses their environment and makes decisions about where to move accordingly. Applications of Legion include train stations, sporting events, and evacuation scenarios. It was successfully used for decision making for the Sydney Olympic Games [Legion, 2004].

REQUIREMENTS ANALYSIS

There are several stakeholders in the development of a pedestrian model. The roles of people involved in the development of the model have not been researched in depth, but are of interest to agent-based modelling [Drougoul et al., 2003].

The client has a need for forecasts and is likely to be a planner or event organiser. They are likely to have or have access to the most information about the domain, including the problem and the environment. They will have observed the environment and would provide information about the current situation in the form of current usage or environment layout. They can sometimes provide their opinion on the cause of particular behaviours. They will also usually provide a set of scenarios, which will assist them in making their final decision.
The practitioner is likely to be an engineer or planner and needs to provide a service to the client, including provision of forecasts and analysis of scenarios.

Current location: ˜r

Equation of motion: d˜r(t) dt = ˜v(t)

Equation of acceleration: d˜v(t) dt = ˜ f(t) + _(t)

Behavioural force: ˜ f(t) = ˜ f0(v) + ˜ fB(r) +X_6=s ˜ f_(˜rs, ˜vs, ˜r_, ˜v_) +X ˜ fi(˜rs, ˜ri, t) + _(t)
where:
˜ f0(v): desired speed

˜ fB(r): repulsive force from borders

˜ f_(˜rs, ˜vs, ˜r_, ˜v_): repulsive force from another pedestrian _

˜ fi(˜rs, ˜ri, t): attractive forces to attraction i

Figure 2: Some of the mathematical equations used in the social force models [Helbing et al., 2001].

Figure 3: A cellular automata model showing pedestrians forming lanes through two doorways [Burstedde et al.,2001].

They interface between the client and the developer and therefore require some knowledge of both those roles.

The developer creates a model that is representative of the reality. They may develop a model from scratch (eg. in Microsoft Access or .NET) or use an existing package (eg. PAXPORT). They need to have good understanding of the features of the package/language, so that they can suggest modelling methods to the practitioner. The model needs to meet the requirements of the practitioner and the client.

A person may perform one or more roles, eg. the practitioner and developer may be the same person. Each role has different requirements for the model.

The client's needs are:
- an understanding of the model scope: they need to understand (at a high-level) the behavioursthe model can create, in order to make a judgement about the validity of the model. They also need an understanding of the environment constraints. If they are interested and/or experienced in modelling,

they would also like an understanding of the model parameters.
- results in a variety of formats: these include charts and maps for reports. 2D and 3D animations are also useful for presentations to senior stakeholders or the community, as it provides a more realistic feel for the effects of the scenario.

The practitioner's requirements are:
- ease of use: the package should follow current software engineering principles and should be straightforward to use. Graphical user interfaces should be intuitive. Input and output data should be read and written to appropriate locations and in an easy-to-manipulate format.
- a clear understanding of parameters: the parameter should have some resemblance to the real world. Ideally, it should also be clear what the intended effects of each parameter are, if changed in isolation. However, with non-linear or emergent modelling approaches, this is not feasible.
- the ability to modify parameters quickly and easily: again, the GUI should enable parameters to be changed easily. It should not take long to

make changes to several parameters to set up a new scenario.

the ability to read in environment data from various sources, including temporal constraints: environment data is extensive and there are various representa-

tions that can be used (eg. GIS, CAD, structured text files using XML etc.).
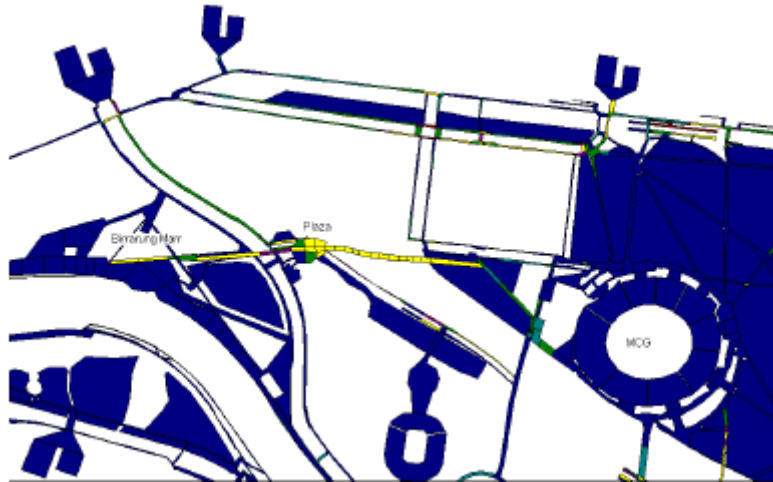
•

Figure 4: Graphical output from a PAXPORT model of the Sports and Entertainment Precinct in Melbourne [Ronald, 2004].

- flexibility with output: the practitioner should be able to select different outputs to suit the project and the client's needs. This involves selecting different statistics and at different environmental scales (eg. block, street, area, model) and at different times (peak/offpeak).
- a reasonable running time: often a large number of scenarios need to be developed and analysed, so model running time is a key factor. Often practitioners will not have access to a dedicated model-running machine.

ENVIRONMENT CATEGORISATION

There are many types of environments that pedestrians walk in and the environment is a major factor in their behaviour. We propose that the environments can be categorised by examining the environmental features, walking behaviours, map representation, and the expected volume of pedestrians. This is a very "physical" approach to the categorisation for simplicity of the modelling. For example in crowding, psychologists claim that our experiences are influenced by our comfort with the crowding rather than the density of people, however this is harder to model [Gifford, 2002].

- **small-scale enclosed spaces:** these consist of small rooms connected by corridors and exits.

For example, buildings often have many enclosed spaces (eg. offices, meeting rooms). Multistorey buildings will have lifts, stairs, or escalators to facilitate movement between floors. The type of walking trips in this environment are short and direct (ie. little chance of distraction). The number of pedestrians is variable depending on the location and the map representation would be at a small scale.

- **large-scale enclosed spaces:** these are generally larger buildings that are open-plan. For example, sports arenas consist of an area filled with seats, aisles, and exits. Cafes and souvenir stands could also be found within the arena, so queues are likely to occur outside these shops and also at exits at the end of the event. The trips in this environment are likely to be short and direct (eg. from entrance to seat, from seat to cafe). Another example is an airport or a train station, where the main purpose of walking is to change between pedestrian and public transport mode. Another issue with these environments is that temporal constraints are present, in that one needs to catch a train/plane at a particular time (hard constraint) or one should aim to arrive for the start of a sports match (soft constraint). These environments would contain a large number of pedestrians in a small area.

- **mixed mode:** this environment consists of a area, possibly shared with cars or public transport, which connects the pedestrian to building entrances and other streets. The pedestrian has static objects (eg. public seating, rubbish bins, garden areas) to navigate around. Another element of this environment is a queue, which can be either an ordered queue of people waiting to get into a busy shop or an unordered queue of people waiting to cross the street or waiting for a bus. The trips in this environment are likely to be a mix of direct trips with a purpose (eg. those who are walking to work), purposeful but indirect trips (eg. I need a pharmacy but I don't know where one is), and purposeless/weak-purpose trips (eg. a shopping trip to the city).

- **open space:** this consists of open areas, possibly with some designated pathways. The purpose is most likely to be leisurely, so the behaviour will consist of meandering, frequent stopping, and possibly longer stops for picnics or sightseeing.

- **hybrid:** this category includes generally pedestrian areas or low-traffic areas containing several attractions, such as sports precincts or universities. It will consist of a combination of behaviours from the open space environment (eg. meandering, afternoons on the lawn), the mixed-mode environment (eg. avoiding vehicles, queueing for public transport), and the enclosed spaces environments (eg. moving around lecture theatres).

In these descriptions, several behaviours were mentioned. We consider these behaviours to be the key behaviours that occur in a model and would influence the type of model required.

- **purposeful and familiar:** the pedestrian knows where they are going, how to get there, and has a low probability of being distracted on the way.

- **purposeful and unfamiliar:** the pedestrian knows where they want to go, but is not sure how to get there and as a result may get distracted or lost on the way.

- **purposeless:** the pedestrian is in wandering mode.

- **evacuation/panic:** the pedestrian is in panic mode and will behave differently to normal. If this behaviour is required, it will be the main focus of the model and other behaviours will probably not be included.

- **forced waiting:** this behaviour occurs in environments where pedestrians have to wait for an action to happen before they can continue. For example, they may wait in a queue to buy a train ticket, but they have no control over this queue. Another example is an unordered queue at a traffic light where pedestrians must wait for the green light..

- **temporal constraints:** these occur in train stations and airports (so-called hard constraints: the train leaves at a certain time and if you miss it, you miss it completely) and also at sporting events (soft constraints: you can arrive slightly late for the game, but you can still be admitted and see most of it). Temporal constraints have not always been considered in pedestrian models, but from our experience are a requirement for accurate outputs for some environments.

The notion of purpose has been discussed in the cognitive/mental map literature. Regular, purposeful journeys (work, shops) start out as unfamiliar, however due to their frequency become more familiar. A regular path would be formed quickly and it is expected that it would be close to (if not the) shortest path and would aim to minimise the amount of concious decisions made during the trip [Golledge and Garling, 2004]. Our first three categories are also sometimes referred to by other names: commuting (purposeless and familiar), quest (purposeful and unfamiliar), and exploring (purposeless) [Weston and Handy, 2004].

SELECTING AN APPROACH

In software engineering, the language or package to be used for building a system is dependent on the requirements of the system. However, in practice, the development environment is usually chosen before the requirements are set out. This could be because the developers have extensive experience in a particular package, or the client prefers a particular package.

With modelling, the same sometimes occurs: the approach chosen is the approach one of the stakeholders is the most familiar with. The client may want a model they can reuse or adapt in the future and this puts a constraint on the selected approach. The practitioner may also be more familiar with one approach over another. This is not necessarily the best tool for the job and ideally the approach should be chosen based on the requirements of the model.

For pedestrian models, there are several key factors that can point to one approach being more suitable than another.

Earlier, we identified five areas and six behaviours that may be modelled. Not all behaviours are present in the chosen environment or are of significant interest to clients. Some approaches suit the environment or behaviour better than others. For example, an agent approach is more suitable when there are complex decisions to be made about activities and moving through the environment.

Scale is also a key factor in choosing an approach. This includes both the number of pedestrians to appear in the model as well as the size of the environment and the detail required. The choice of the scale of the model is also related to the outputs. If approximate volume counts are required, then an approach that models the exact steps of each pedestrian is unnecessary.

As the first item to be decided usually is the location (or type of location) to be simulated, this should dictate the approach chosen. From our experience, it is common for clients to decide later the exact outputs required.

Small-scale enclosed spaces

Behaviours: direct, some wandering, possibly evacuation

Office buildings consist of mostly direct trips, whereas shops and other "leisure" buildings would have a combination of direct and distractable trips. A CA approach would suit the former, as there would not be enough traffic to require a mathematical approach and the microsimulation and agent approaches would be too complicated. For "leisure" buildings, an agent approach would be preferred, in order to see the individual choices that people make when moving through the area.

Large-scale enclosed spaces

Behaviours: direct, some congestion and queueing, hard and soft temporal constraints, possibly evacuation

A microsimulation approach would be good as this could model the queues and level of service at exits easily. For more detail of the crowding, a mathematical or CA model could be used. As there is minimal decision making involved in the environment and the pedestrian count may be high, an agent approach may not be appropriate, however Legion has also been used for arenas and train stations [Legion, 2004]. PAXPORT was designed for the airport and train station environments.

Mixed mode

Behaviours: direct, wandering, queues, congestion

All approaches are suitable for a mixed mode environment, however if the model is too large in area a CA model would probably be unsuitable, due to the detail of the environment. The models developed by Intelligent Space and Space Syntax would suit this environment. Agent-based simulation and microsimulation could also model the vehicles in the environment, especially public transport vehicles that are effectively an exit from the pedestrian model.

Open space

Behaviours: wandering, bushwalking (leisure), picnicking

Depending on the level of choice involved in the model, the best approaches are a microsimulation approach or an agent approach. There is not enough interaction or congestion to warrant a mathematical approach or a CA approach. Ideally the model should scale between a large block representation and smaller units. Hybrid environments are complex and it is difficult to generally recommend an approach. It may be necessary to create more than one model to retrieve the required outputs.

THE AGENT-BASED APPROACH

We now present a case study of the development of a model, using one of the approaches described earlier. Our domain of interest involves agents entering a sports precinct and moving towards a stadium. Several "distractions" were located on the way to the stadium, such as food stands and street performers. This system involves an open environment and we are interested in the decision making of pedestrians making their way through the environment. Using our classification scheme, either an agent or a microsimulation approach is recommended.

Agent-based simulation is an active research area and many applications have been developed. Transport applications and social simulation are popular domains.

There are many types of agents, however we will use the BDI architecture. This architecture is based on folk psychology, in particular the work of Bratman [Bratman, 1987] and Dennett. Rao and Georgeff [1995] presented a logical analysis which became the base for many BDI implementations. The reasons for choosing BDI over other architectures are it is commonly used for modelling human behaviour [Norling, 2004] and it is supported by a mature design methodology (Prometheus) and an implementation language (JACK Intelligent Agents).

Modelling Aspects

Vehicle systems can be broken down into three mainconcepts: user, vehicle and road [Ogden and Taylor, 1996]. The user has a perception of attributes of the road and their vehicle, and needs to guide their vehicle along the road. The vehicle interacts with andchanges the road.

The road is constantly updated with the new locations of vehicles and provides perceptions to vehicles and users. These same concepts can be extended to pedestrians, however for road we should substitute environment. For pedestrians however, the user and vehicle are essentially the same object: a human. However, most of our walking is done subconsciously and therefore it is permissible to separate these two concepts. We can define the user as the human's brain and the vehicle as the human's legs. Given the high-level decision making role of the user, the belief-desire-intention architecture would be useful for modelling user behaviour in transport systems.

The BDI Architecture

The philosophical component of BDI is based upon practical reasoning. Practical reasoning is defined as reasoning toward actions, as opposed to theoretical reasoning, which is reasoning about beliefs. Practical reasoning can be broken down further into two activities: deliberation (deciding what goals to achieve) and means-end reasoning (how to achieve a goal) [Wooldridge, 2000]. Another nice feature of the BDI architecture is the ability to act in both a reactive and proactive manner, however there is a danger of being too reactive or too proactive.

The key concepts in the BDI architecture are:

- beliefs: what I know or don't know about the world;

- desires: what I want to do;

- intentions: how I plan to do what I want to do.

BDI fits our problem well, in that:

- Pedestrians have beliefs about the environment that affect their decisions (eg. "The main street is always crowded at lunchtime - I will take another route.");

- Pedestrians have desires to do something or to visit somewhere. This is more obvious with vehicle travel as people do not drive around the city just for something to do, whereas people will sometimes walk somewhere just because it is there. If people are wandering "just because", then that is still a desire;

- Pedestrians have plans or procedures of deciding where to go first, how to get there, and how to create a path to follow;

- If a route is blocked due to congestion or temporary infrastructure, a new plan can be formulated and a new path taken to reach a location.

Designing with Prometheus

Prometheus is a methodology developed for specifying agent-oriented software systems. Although there are several design methodologies that could be used for this system [Bergenti et al., 2004], Prometheus was chosen because of its maturity (a book was recently published [Padgham and Winikoff, 2004]) and because the concepts used in the methodology tie in with the concepts used in JACK Intelligent Agents, our chosen implementation language. JACK is based on the BDI architecture, and while Prometheus is technically architecture-independent, it was designed with JACK in mind and therefore fits the BDI architecture neatly.

Prometheus consists of three design phases:

- system specification phase: identify functionalities, inputs, outputs and shared data sources;

- architectural design phase: determine agents required and their interaction; and

- detailed design phase: internal design of agents.

The resulting design is a combination of forms and diagrams, which clearly describe the percepts, action, environment, agents, capabilities and plans in the system. It would not be unfamiliar to those familiar with UML for object-oriented design [Rumbaugh et al., 2004]. Although not tied to any implementation, Prometheus fits nicely with JACK Intelligent Agents.

System specification The system specification involves identifying system goals and functionalities, developing the interface between system and environment, and developing use case scenarios.

Firstly, the system goals and possible subgoals need to be established. An example of this is shown in Figure 5, which shows three goals (visit attractions, arrive at the stadium at a reasonable time, move through the environment) and their subgoals using an oval shape. One of the goals is to move through the environment, with the subgoals of satisfying network constraints and taking a reasonable path[1]. Goals can then be grouped together to create functionalities. Scenarios can also be developed. These consist of steps such as percepts, messages, goals and actions. The system interface involves determining the actions and percepts of the interface to the environment.

---

[1] It is difficult to define a reasonable path, however one criterion might be that there are no loops in the path.
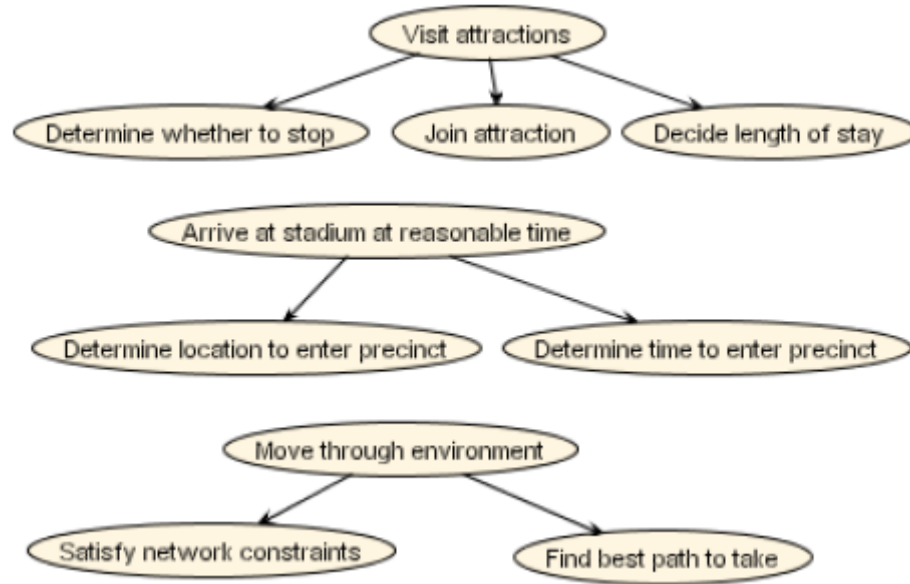
Figure 5: A system goals diagram in Prometheus.

Architectural design phase

The architectural design phase involves grouping the functionalities into similar areas, developing agents to control each area and specifying the interactions between agents.

The functionalities can be grouped into areas by investigating the data that is present in the system and which data is required by each functionality. This is then used to determine the agents required and what functionalities they should have. Figure 6 shows an example of a coupling diagram. Each data source is represented by a yellow cylinder and the functionalities by rectangles. The arrows signify whether the functionalities read (arrowhead at the functionality end) or write (arrowhead at the data end) data. As part of this phase, communication between agents can also be specified at a high level.

Detailed design phase

In the detailed design phase, the agent's capabilities, events, plans and data structures are developed in more detail. An example of a detailed design for the Pedestrian agent is shown in Figure 7. This shows the events (percepts: star shapes; actions: arrow shapes; messages: envelope shapes), capabilities (rounded rectangles), and plans that the agent requires. The arrows again signify the incoming and outgoing nature of events. Capabilities are similar to modules in that related plans, events, and data can be combined together in a coherent manner. In this model the pedestrian has a capability for each of its main activities.

Events can be actions (affecting the environment in some way), percepts (knowledge coming from the environment), and messages (to and from other agents). For each of these concepts, several parameters need to be designed including the information carried by the percept/message, the effect of the action, and what to do in case of failure.

Descriptions of data usage are also required. Plans need to specify whether they are reading and/or writing data. Figure 8 shows two plans (p Startup and p UpdateEnvironment) that use three data sources (Links, Nodes, and Attractions). p Startup has writeonly access, whereas p UpdateEnvironment has both read and write access to the data sources. In this stage, plans are described at a high level, including their name, the percepts that trigger them (as shown in Figure 8 by the star shapes) and the actions that occur during the plan. They will be designed in more detail depending on the implementation platform.

The main issue with using an agent-oriented methodology is that it doesn't "run the model" ie., it only design what the agents are doing. It cannot design the core of the simulation ie., how the clock will tick over, the graphical user interfaces required to set up the simulation, the methods to collect outputs. Therefore Prometheus needs to be combined with anothermethodology to design the whole of the simulation.

Implementation

We constructed a prototype model in JACK Intelligent Agents [Agent Oriented Software, 2005]. In our prototype, we attempted to implement the entire model archi-

tecture (user-vehicle-environment) in JACK to avoid complex interfacing.
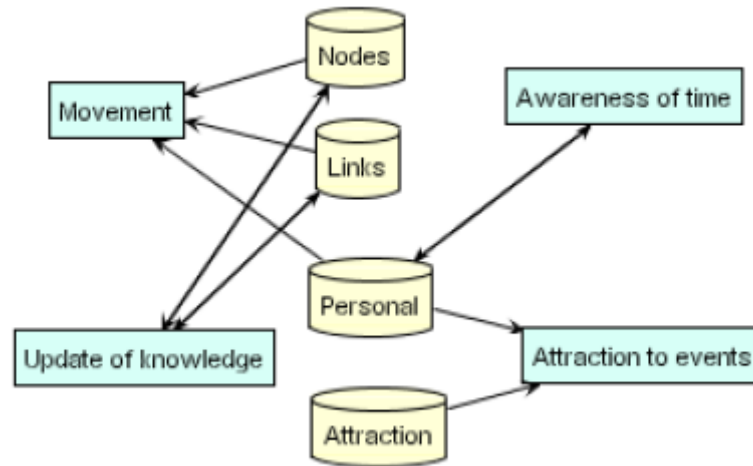

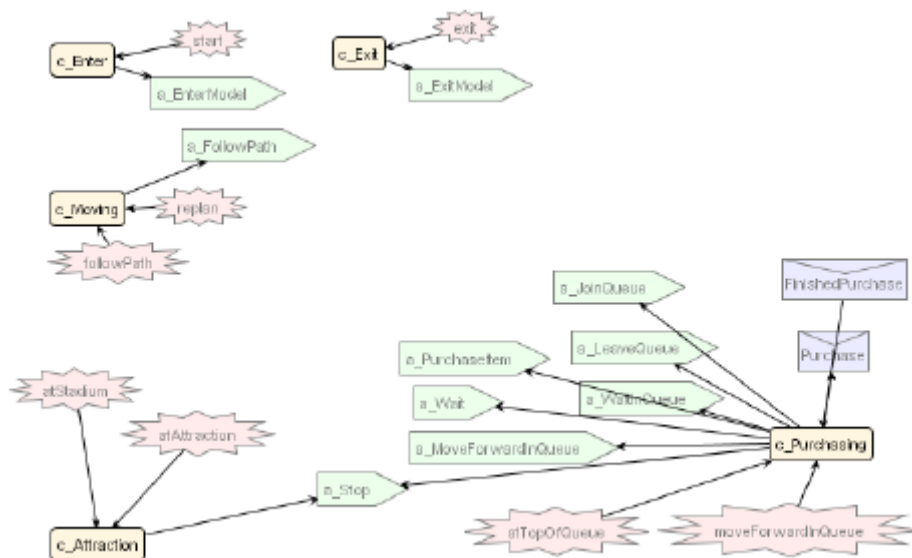Figure 6: A coupling diagram in Prometheus.

Figure 7: A detailed design for a Pedestrian agent in Prometheus.

JACK is based on the BDI architecture and was purpose-built for simulations, in particular defence simulations. The aim of the package was to develop a stable, lightweight and practical agent-based programming language that would not be superseded quickly and would facilitate further research. It is based on Java with a few syntactic extensions, and when compiled compiles to Java code [Busetta et al., 1999]. The product has been used for several applications, mainly within defence, however it has a strong reputation worldwide both in research and industry.

JACK supports the concepts in the BDI architecture and Prometheus: agents, events, beliefs (data), capabilities, and plans. Appendix A shows a plan (p Stopping.plan) written in JACK code. As the JACK files are compiled into Java before execution, normal Java statements can be embedded in JACK files.

It is straightforward to implement goal-directed behaviour, such as moving towards the stadium. The belief system, however, is similar to facts in a logic programming language such as Prolog and does not handle complex beliefs well. For example, it is difficult to represent an environment in detail using JACK beliefsets. Ideally an interface to the environment should be developed and

then any environment format (eg. graph, cells, shapes) could be used behind that. The decision-making used in BDI and in JACK cannot elegantly handle continuous events, such as stepping. It is also difficult to define the subconscious decisions behind

walking. Therefore the vehicle model in our architecture would be better suited to an object representation rather than an agent one. JACK has the ability to interface with both Java and C++ code.
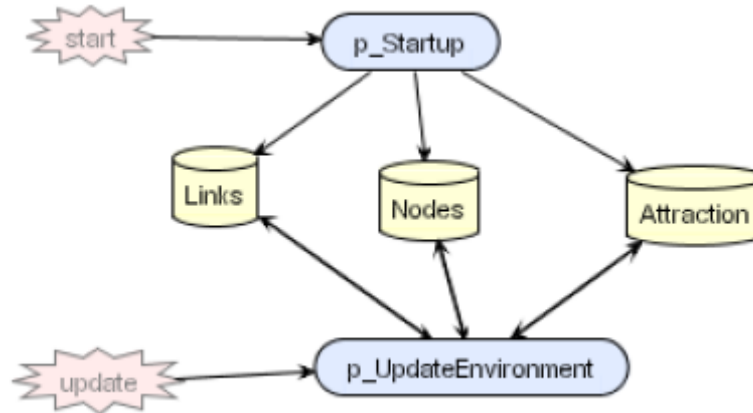

Figure 8: A data diagram in Prometheus.

An advantage of agent-based technology is that agents are capable of doing several things concurrently without trouble eg. walking and looking about. So, if you are walking to the post office and you see a shop that has a sale, you continue walking while´you make a decision whether to detour or not. If you decide to keep going, you continue with your existing walking plan. However, if you decide to detour, you stop your current walking plan and construct another to get to the sale. Using JACK, we found that the interrupted plan would sometimes continue after it had been interrupted and another plan started. In our model, this lead to agents finding themselves in two places at once. A solution is to increase the lookahead of the pedestrian agent, so that they make decisions and construct new plans earlier.

CONCLUSION

This paper presented an analysis of the requirements for pedestrian models from both the end-user's and developer's viewpoints by reviewing models developed in research and in industry. We explored the key factors in designing pedestrian models and began developing a framework for selecting a particular modelling technique based on the type of model desired.

This framework will be of use to clients, practitioners, and developers. It will play a strong role in the usefulness and reliability of pedestrian modelling in the decision-making process for planning and design of pedestrian-frequented areas.

We also investigated using the belief-desire-intention (BDI) architecture to model pedestrian behaviour using the design methodology Prometheus and the agent-oriented language JACK Intelligent Agents.

We found that the user-vehicle-environment architecture is an appropriate separation for transport models, and applies to pedestrian models even though the user and vehicle are physically the same. The BDI architecture is appropriate for the user model only, as that is where the decisions are made. Prometheus is useful for designing the BDI concepts required. For the vehicle and environment model, an object approach is more suitable than using an agent language such as JACK.

The work is continuing as part of a larger project to evaluate approaches and methodologies for modelling pedestrian behaviour. The next step is to develop the framework further by investigating the inputs and assumptions required for each approach and also developing recommendations for hybrid environments. Experimentation with some of the approaches will also be undertaken to test their suitability for different applications. This will involve learning from our prototype and using JACK Intelligent Agents to develop a model of the user, following the Prometheus specification, and connecting it to environment and vehicle
modules written in Java.

APPENDIX A: SAMPLE JACK CODE

```
package pedestrian;
import data.*;
public plan p_Stopping extends Plan {
#handles event m_Stopping att;
#posts event m_Replan replan;
#uses data b_Personal personal;
#uses data b_Attraction attraction;
#uses interface a_Pedestrian self;
body() {
logical int node, id, m, n, type,
open, close, distract, l,
s, swing;
logical String name;
attraction.get(id, att.where, n,
type, name, open, close,
swing);
personal.add("move",0);
personal.add("seenAttraction",0);
System.out.println(self.name()
+ ": " + "stopping");
personal.get("currentNode", n);
personal.get("currentLink", l);
personal.get("currSpot", s);
@subtask(replan.post(n.as_int(),
l.as_int(),s.as_int(),
att.where));
System.out.println(self.name()
+ ": " + "stopping now");
}
}
```

REFERENCES

Agent Oriented Software [2005], "Agent Oriented Software (AOS) - JACK Intelligent Agents, software agent system.", http://www.agentoriented.com/shared/home/index.html. Accessed 4 April 2005.

Australian Pedestrian Council [2004], "The Pedestrian Council: Pedestrian Council Home", http://www.walk.com.au. Accessed 9 December 2003.

Batty, M. [2001], "Editorial", *Environment and Planning B: Planning and Design* **28**, 321–326.

Bergenti, F., Gleizes, M.-P. and Zambonelli, F. [2004], *Methodologies and Software Engineering for Agent Systems*, Springer.

Bratman, M. [1987], *Intentions, Plans and Practical Reason*, Harvard University Press.

Burstedde, C., Kirchner, A., Klauck, K., Schadschneider, A. and Zittartz, J. [2001], Cellular automaton approach to pedestrian dynamics - applications, *in* M. Schreckenberg and S. Sharma, eds, "Pedestrian and Evacuation Dynamics", Springer-Verlag, Berlin.

Busetta, P., R¨onnquist, R., Hodgson, A. and Lucas, A. [1999], Light-Weight Intelligent Software Agents in Simulation, Technical Report 3, Agent Oriented Software.

Drougoul, A., Vanbergue, D. and Meurisse, T. [2003], Multi-agent based simulation: Where are the agents?, *in* J. S. Sichman, F. Bousquet and

P. Davidsson, eds, "Multi-Agent Based Simulation II", Vol. 2581 of *Lecture Notes in Computer Science*, Springer-Verlag. Gifford, R. [2002], *Environmental Psychology: Principles and Practice*, Optimal Books, Canada.

Gloor, C., Stucki, P. and Nagel, K. [2004], Hybrid techniques for pedestrian simulation, *in* P. Sloot,

B. Chopard and A. Hoekstra, eds, "Cellular Automata: 6th International Conference on Cellular Automata for Research and Industry, ACRI 2004", Vol. 3305 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 581–590.

Golledge, R. G. and G¨arling, T. [2004], Cognitive maps and urban travel, *in* D. Hensher, K. Button, K. Haynes and P. Stopher, eds, "Handbook of Transport Geography and Spatial Systems", Elsevier.

Harney, D. [2002], "Pedestrian modelling: current methods and future directions", *Road & Transport Research* **11**(4), 2–12.

Helbing, D., Moln´ar, P., Farkas, I. J. and Bolay, K. [2001], "Self organizing pedestrian movement", *Environment and Planning B: Planning and Design* **28**, 361–383.

Henein, C. M. and White, T. [2004], Agent-based modelling of forces in crowds, *in* P. Davidsson,

L. Gasser, B. Logan and K. Takadama, eds, "Multi-Agent andMulti-Agent-Based Simulation", Vol. 3415 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 173–184.

Hoogendoorn, S. and Bovy, P. H. L. [2004], "Pedestrian route-choice and activity scheduling theory and models", *Transportation Research B* **38**, 169– 190.

Intelligent Space [2005], "Intelligent Space Partnership", http://www.intelligentspace.com. Accessed 1 August 2005.

Legion [2004], "Legion pedestrian simulation software for public environments", http://www. legion.biz. Accessed 23 January 2004.

Nagel, K. and Schreckenberg, M. [1992], "A cellular automaton model for freeway traffic", *Journal de Physique I* **2**(12), 2221–2230.

Narimatsu, K., Shiraishi, T. and Morishita, S. [2004], Acquisition of local neighbor rules in the simulation of pedestrian flow by cellular automata, *in*

P. Sloot, B. Chopard and A. Hoekstra, eds, "Cellular Automata: 6th International Conference on Cellular

Automata for Research and Industry, ACRI 2004", Vol. 3305 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 211–219.

Norling, E. [2004], Folk psychology for human modelling: Extending the bdi paradigm, *in* "Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems", ACM.

Ogden, K. W. and Taylor, S. Y. [1996], *Traffic engineering and management*, Dept. of Civil Engineering, Monash University, Clayton, Australia.

Ort´uzar, J. D. D. andWillumsen, L. [1994], *Modelling Transport*, John Wiley & Sons, Chichester, NY.

Padgham, L. andWinikoff, M. [2004], *Developing Intelligent Agent Systems - A Practical Guide*, John

Wiley and Sons. Parunak, H. V. D., Savit, R. and Riolo, R. L. [1998], Agent-based modeling vs. equation-based modeling: A case study and users' guide, *in* J. S. Sichman, R. Conte and N. Gilbert, eds, "Multi-Agent Systems and Agent-Based Simulation", Vol. 1534 of *Lecture Notes in Computer Science*, Springer- Verlag, pp. 10–25.

Rao, A. S. and Georgeff, M. P. [1995], BDI Agents: From Theory To Practice, Technical Report 56, AAII.

Ronald, N. [2004], The role of pedestrian modelling in planning for special events, *in* "Proceedings of the 27th Australasian Transport Research Forum". Rumbaugh, J., Jacobson, I. and Booch, G. [2004], *The unified modeling language reference manual*,

Addison-Wesley, Boston, MA. SpaceSyntax [2004], "SpaceSyntax Home Page", http://www.spacesyntax.com. Accessed 23 January 2004.

Still, K. [2000], Crowd dynamics, PhD thesis, University of Warwick, Department of Mathematics. Accessed 20 January 2004 from http://www.crowddynamics.com/ Thesis/Contents.htm.

Weston, L. and Handy, S. [2004], Mental maps, *in* D. Hensher, K. Button, K. Haynes and P. Stopher, eds, "Handbook of Transport Geography and Spatial Systems", Elsevier.

Wooldridge, M. [2000], Intelligent agents, *in* G. Weiss, ed., "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence", MIT Press.

AUTHOR BIOGRAPHIES

NICOLE RONALD graduated with undergraduate degrees in computer science and civil engineering from the University of Melbourne in 2002. She is currently a postgraduate student in the Department o Computer Science and Software Engineering at Melbourne. Between December 2002 and January 2005 she worked as a graduate transport modeller for an global engineering consulting firm in Melbourne. Her webpage is http://www.cs.mu.oz.au/˜naron and her email address is naron@cs.mu.oz.au.

LEON STERLING is Adacel Chair of Software Innovation and Engineering at the University of Melbourne. He is a co-director of the Intelligent Agent Lab (http://www.cs.mu.oz.au/agentlab) and his research interests include agent-oriented software engineering. His email address is leon@cs.mu.oz.au.

MICHAEL KIRLEY is a Lecturer in the Department of Computer Science and Software Engineering at the University of Melbourne. His research interests include complex systems science, multi-agent systems, and the theory and application of evolutionary computation. His email address is mkirley@cs.mu.oz.au.