Application of context-mediated behavior to a multi-agent pedestrian flow model (PEDFLOW)

Robert Kukla*, Alexandra Willis, Jon Kerridge

Transport Research Institute
Napier University
Redwood House
66 Spylaw Road
Edinburgh
EH10 5BR, UK

email: {r.kukla, a.willis, j.kerridge}@napier.ac.uk
telephone: +44-(0)131-4555161
fax: +44-(0)131-4555141

submitted: 9/11/2002
words: 6558

ABSTRACT

Developers of pedestrian models often overlook the fact that pedestrians are subjected to a multitude of influences when walking. The majority of existing models only focus on a single aspect, typically the avoidance of obstructions or other pedestrians. PEDFLOW, a microscopic, agent-based model, uses an implementation of context-mediated behavior to enable the agents to deal with multiple cause-effect relations in a well-defined and flexible manner. This paper explains the basic idea behind the approach and illustrates it through examples.

INTRODUCTION

Computer models have been used for a long time and in many different application areas to simulate complex, dynamic processes. Typically this is done to reduce expenses. Other reasons are timing issues with very slow or very fast systems and safety aspects. With sinking cost and increasing computational power of today's computer systems, process modeling penetrates new areas and helps to provide scientifically justified solutions where otherwise decisions could only be based on experience or guesswork. User-friendly model software can run on the office desktop computer of a designer and is no longer a tool for the use of the highly qualified engineer only.

Examples for this trend are traffic models. Strategic town planning has become a requirement in today's urban design process to avoid the creation of routes that will become bottlenecks in the future. Capacity demand will need to be predicted and possible designs compared with regard to their efficiency. One particular aspect of traffic has gained increased interest in recent years - pedestrian traffic. This is due to the fact that vehicular traffic has become overwhelming and cities lose their attractiveness as a place of living. Many governments have addressed this problem by creating walking strategies (1) or similar guidelines to encourage walking. One way to achieve this is to provide attractive walking spaces that allow pedestrians to reach their destination efficiently and without interference. On the other hand the design should allow people to walk at their personal requirements without hindering others. This includes walking together in groups, walking at individual speeds and even stopping to watch a street musician or shop display. Interfacing with other means of transport is another requirement, be it public transport like a bus stop or access to a car park.

The town planner, faced with the task of designing such spaces, needs a tool that will allow different designs to be compared in terms of their attractiveness as well as their effectiveness. A shopping area has different influences on pedestrian movement than a public park and hence different layout requirements. Ideally a pedestrian model should allow describing the purpose and size of the place and the population of people that is expected to walk there, including their purpose and destination. It would then run simulations for different layout designs and give an evaluation based on several criteria.

PEDFLOW is an attempt to create such a tool. It is a microscopic model of pedestrian flow where virtual pedestrians navigate a virtual environment. On their way towards a goal the agents, representing pedestrians, interact with features of the environment and with other agents. A variety of mobile and immobile entities can be modeled by objects in an object-oriented environment.

The model is informed by the empirical study of pedestrian behavior (2). Observational studies of naturalistic movement behaviors (based on filming pedestrians unobtrusively in a range of different urban settings) have enabled us to measure individuals' preferred walking speeds and distances people like to maintain around themselves as they avoid obstacles and other pedestrians. They also allow us to determine the overall make-up of the pedestrian population associated with a particular environment in terms of different types of people (including a range of ages, group sizes, levels of mobility and trip purposes), and thus calculate a range of walking speeds and avoidance distances that closely reflects that of the natural population. This is important if the model is to be flexible enough to describe behavior in a range of different settings. The analysis of video footage also allows us to explore what decisions people make when faced with various obstructions: most people, for example, prefer to remain on the pavement than step out into the road when avoiding a lamp-post – even in quiet streets. Questionnaires and in-depth interviews allow us to explore how the layout of urban space affect people's behavior within it, and what kinds of factors afford them the most enjoyable walking experience. This more qualitative information is critical for practitioners interested in creating urban spaces that invite pedestrians and encourage people to walk more.

Current status of model and relevance to practitioners. The PEDFLOW model provides a potential tool for urban planners to test the effects of any design changes on pedestrian behavior before their implementation. It provides an interface through which designers can construct a putative urban space (such as a wide pavement next to a busy road) and manipulate the position and nature of various items (such as bus stops, rubbish bins, railings) within it. The area can then be populated with the appropriate mix of different pedestrian "types" (mimicking pedestrians of different ages, trip purposes and levels of mobility; singletons or groups), and individual agents' movement behaviors modeled. The design idea can then be evaluated in terms of the extent to which it provides pedestrians with their "optimal" movement through the space (in terms of time taken and distance traveled). We are currently in the process of refining this output stage to include more qualitative aspects of "walkability", informed by a series of in-depth interviews. Finally, we plan to validate the PEDFLOW model by testing the extent to which it predicts real movement behavior in an urban setting, as compared with video footage of that new setting.

MODELLING PEDESTRIANS IN COMPLEX SITUATIONS

The increased interest in understanding and predicting pedestrian behavior has resulted in a number of interesting research projects and, as a result, publications in this area. A variety of microscopic models exist, based on different approaches such as cellular automata, space-syntax and of course autonomous agents (3,4,5,6,7,8). Looking at them in more details reveals their individual strengths but also a common, major shortcoming: they are designed to model just one aspect of pedestrian movement. This can be the navigation in a crowded room, route-choice in a complex layout or simply movement dynamics at a doorway. If it is the stated aim of the model, this is a valid approach to take. However with PEDFLOW we wanted to create an expandable design that allows for the inclusion of additional aspects which affect the routes pedestrians take, until a sufficiently close approximation of the corresponding situation in the real world is achieved. A simple example is that for the majority of movements the modeled pedestrians would only be concerned with the avoidance of objects and other people. In case of a shopping street however, the dominating factor would be to walk close to windows and shop displays and also to be able to stop and examine things more closely or possibly to enter the shop. Another group of people, not interested in shopping, will exhibit a completely different behavior. We shall look at how other models tried to incorporate such additional effects into their framework and if not, how easy it would be to expand it.

Blue and Adler (3) use a basic cellular automata (CA) approach to model pedestrian movement. A cellular automaton consists of a regular 2-dimensional lattice, where each cell of this lattice has a discrete state. It can be empty or occupied. The dynamical behavior of the CA is described by a formula describing the state of a cell for the next time step depending on the states of its neighboring cells. Movement is modeled as the vacation of one cell and the occupation of a new cell. The researchers have expanded the basic CA concept to include a directional component. If the destination cell is already occupied, a random neighboring cell is chosen. With the intended aim to create a model of minimal complexity, they have achieved some interesting results in creating emergent, collective behavior of modeled pedestrians. The modeled environment however is limited to moving pedestrians only: no interactions with other objects exist, nor can they be easily added. The authors state: "By 'designing' the CA-based pedestrian from the bottom-up at the interface with one another, higher-level functions, like route selection and trip behavior, can be added later without fundamentally changing the inter-pedestrian dynamics." This seems to imply that extension to include more elaborate rules is possible. However, there do not seem to be any specific facilities to achieve that. An additional problem with the model is the fact that parameters like walking speed are assigned randomly and do not relate to measurable attributes of the modeled environment. The current Blue/Adler model is therefore not suitable for expressing the complexity of multi-influence environments.

Dijkstra et al. (4) have extended the CA model even further and have combined it with a network approach. The aim is to create a 3-dimensional visualization of simulated pedestrian activity in the retail environment. They use agent technology to better simulate autonomous individuals and the interactions between them. Agents have an "activity agenda", which is updated according to rescheduling of activity decisions, perceptions of the environment and adaptation of time-budget. The update however can only occur at network nodes or on completion of an action. Between nodes the activity is restricted to obstruction avoidance. While the software is fit for the purpose of evaluating retail layouts, the design makes it hard to include more rapid responses to other influences in the environment or even to deal with multiple influences concurrently.

The Choice Behavior Approach employed by Hoogendorn et al. (7) promises great flexibility. By way of breaking down the decision making process into strategic, tactical and operational decisions, the model allows to deal with different influences at different levels. It is not clear however what the solutions exist to deal with concurrent opposing influences. Also the decision to base the wayfinding on a cost-based algorithm does not seem well justified. Calibration has been performed by trial and error. There is a danger that an optimal pedestrian behavior is created and not one that reflects reality.

Another model that uses agent technology to simulate pedestrian behavior is described by Shelhorn et al. (5). The STREETS model is a mesoscopic, agent-based model of large urban areas. Features of the environment (e.g. buildings or pavements) and pedestrians are modeled as agents and their attributes are automatically derived from several GIS data sets (e.g. socio-economic data, street networks). Pedestrian agents emanate from gateways and move between pre-assigned way points. On their way they are "distracted" by other agents and their route is modified by attributes like "walkability" or "fixation". While the agent approach lends itself to incorporate complex behavior, the scale of the modeled area makes it difficult to deal with microscopic effects. Further, rules and attributes used by the agents are abstract: they don't have a real correspondence to aspects of pedestrian decision making.

The "social force model" developed by Helbing et al. (6) uses a physics approach to modeling pedestrians. By exploiting analogies of crowd movements with gases, fluids and granular media, the researchers have been able to express "behavioral forces" which determine the amount and direction of pedestrian movements as gradients of dynamically varying potentials. In the social force model it is not important which individual pedestrian performs a certain action, it is only the percentage of the people that is considered. This makes it difficult to model isolated interactions like people walking in groups. The researchers have successfully applied the concept to modeling pedestrian movement in corridors (to show the emerging lane formation), on doors (to demonstrate oscillation at bottlenecks) and at junctions (illustrating self-organized roundabout traffic). They also used the model to simulate panic situations and optimize layouts. While the model is well suited to study such situations in isolation, there has been no attempt to simulate a complex scenario. It is assumed that this is due to the complexity of the mathematical expressions that would be required to express the necessary forces. The model doesn't lend itself well to model responses to changing environments such as traffic lights.

CONTEXT-MEDIATED BEHAVIOUR

In PEDFLOW, pedestrians are modeled as autonomous agents moving in a virtual environment. Objects in an object-oriented environment are used to represent "entities" in the real world (e.g. buildings, lamp posts). The location of these entities is described as their index in a two-dimensional array that can be interpreted as a grid or co-ordinate system with the smallest unit of 0.5 meter (1.64 feet). A similar, discrete approach is used for time modeling with the smallest time unit being 50 milliseconds. Movement is a change of position to an adjacent grid co-ordinate and an associated delay time, which is considered a step or activation. Unlike in other models, agents will not skip grid elements. For diagonal movements, the step size becomes grid size multiplied by the square root of two and the time is adjusted accordingly. Agents will re-evaluate their environment with every activation; only minimal state information is kept. A complete description of the mechanics of the environment modeling can be found in (9).

Entities only contain a limited number of features of the modeled object. Features that do not contribute to the decision making process of a pedestrian will be omitted in the representation of objects and multiple features collapsed into an abstract attributes. The decision of which features to include is based on the empirical work that supports the PEDFLOW model (2) and on requirements of particular modeling algorithms. When agents navigate their environment they can only observe it within the limits of the model.

In the basic mode of moving towards a goal while avoiding obstructions agents consider a rectangular area in the movement direction. The observation area is three grid elements wide (a left, a straight and a right lane) and DD grid elements long. DD is the deviation distance (specific to each agent) which refers to the maximum distance to an object at which the agent will start to deviate. If another entity is positioned in the middle lane, the agent will deviate to the left or to the right, choosing a lane that is unobstructed. Now it is possible that both lanes are free of obstructions and the agent has a 50% chance of choosing either one. Observation of pedestrians in the urban environment shows however that this split is not random; in fact, pedestrians will choose directions for any number of reasons. For example, if they walk with a partner they might consider stepping in front or behind the person to not get separated; if they determine the intention of an on-coming person to pass them on one side they will chose the other; or if one choice would bring them closer to a dangerous area (e.g. road) they will choose the safer direction.

The model needs a way to capture these influences in a well-defined and flexible manner. This is of particular importance during development of the model, when it is necessary to be able to test ideas and evaluate the impact that the introduction of new contexts has on the overall behavior and performance of the model. A complete rule base even with optimizations as laid out in (9) which covers all combinations of influences quickly becomes too large and unmanageable. On the other hand an incomplete rule-set where individual rules only cover a subset of the influences as input is difficult to keep consistent. Often the introduction of new rules will create ambiguity or leaves combinations unconsidered.

As a consequence we introduced the concept of context-mediated behavior. The idea is to consider selected, crucial influences independently and treat each one as a context that contributes to the decision of a pedestrian. Every context will be evaluated in isolation and a weight assigned to the outcomes that describes the importance of the influence and the likelihood of the particular action occurring. Contexts can result in multiple possible actions - in the example above the basic context can result in moving straight ahead or a 50/50 likelihood of moving left or right. Different contexts can have actions in common. FIGURE 1 illustrates this case - the "basic" context as well as the "walking with a partner" context could require an action of moving right. The action to be performed is chosen according to the sum of weights from contribution contexts. This method of managing contexts has the advantage that influences can be added or removed easily during model development and their effects investigated.

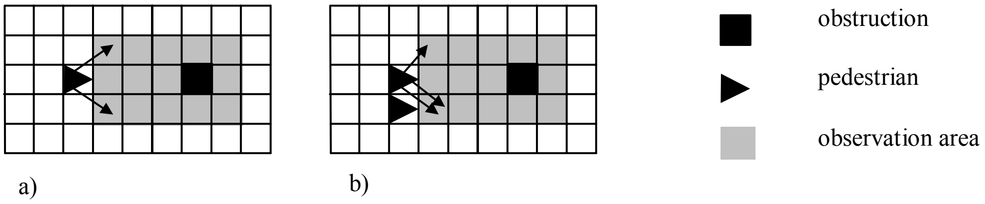

FIGURE 1 Agent objected to one (a) or two (b) contexts.

So far only movements have been considered as possible actions, but pedestrians can "act" in other ways. A special case of a movement would be a pause – the agent doesn't change position, but re-evaluates its options again after a certain time has passed. Other non-movement actions are possible as well. Typically they have a time associated with them and therefore fit in with the concept (e.g. listen to a street musician for a while). Some actions are instantaneous (signal to another agent) or include waiting for an external event (e.g. green traffic light). The former is trivial to implement as it is the action followed by an immediate re-activation, the latter can be realized either by polling of the entity responsible for the event or by means of a re-activation queue within the event-entity. The choice of implementation will depend on the type and frequency of the event and the number of entities involved.

PROBLEMS AND SOLUTIONS

Although this approach is very flexible, it has a number of problems, which need to be addressed:

Weighted random choice. When selecting an action from all possible alternatives according to weight, one option is to always pick the one with the highest weight. This makes the model deterministic. Experience shows that this is not always desirable as it leads to some actions dominating and others being ignored. Additionally it frequently happens that actions have very similar weight totals which results in an unfair disregard for the runner up. A solution is to select an action randomly such that actions with a higher weight have a greater chance of being selected. It is currently investigated if a linear dependency is sufficient or other transformations which lead to a preference of higher weighted actions (e.g. quadratic) lead to more realistic results.

Generating the weights. There is no universal way to derive the values for the weights of the different contexts. The solution is to start with few contexts where the importance can be roughly determined by experience and/or deduced from empirical study of pedestrian behavior. By tweaking the weights a modeled behavior that approximates how real pedestrians behave can be achieved. Continue the process incrementally by adding new contexts and adjusting the values. When introducing new contexts they will be validated with a small set of contexts first before combining them with the whole set. A good idea is to roughly prioritize the contexts in groups: e.g. essential, alternative, irrelevant contexts and adjust weights within these groups more precisely.

Absolute actions. If the agent has reached its goal, there is no reason to consider other possible actions - the action of goal update is the only choice. In this case and with similar contexts the context-evaluation can be aborted and the action selection process omitted.

Disable actions. Certain contexts do not result in an action, but rather block other actions. For example, a red traffic light should prevent an agent stepping onto the road by disabling actions from other contexts that would require just that. All contexts need to be evaluated as there is the possibility they will lead to unproblematic actions. Actions will then be post processed to eliminate actions that must not apply. If no possible action remains the default action of pause will be applied.

Every context requires up to two independent activities by the agent: context detection and context evaluation. Context detection is the process where the agent determines whether a certain context is applicable or not. This can take the form of an observation of the virtual environment, an interrogation of an attribute of another entity, the check of an internal flag or a combination of two or more conditions. The internal flag can be set by an external entity or be a counter field that is updated under certain conditions. An internal flag is also used to keep track of re-occurring conditions. For instance an attractor should only attract an entity once, as it would otherwise result in constant attraction.

Contexts can be classified as external (set from the outside of the entity) or internal (detected from the value of an internal flag or timer), and persistent (lasting over a number of activation cycles, until turned off) or transient (only valid for the current activation cycle).

Once it has been determined that the context applies, possible actions will need to be derived. Typical actions are: move to adjacent grid element, swap position with entity at adjacent grid element, pause (un-schedule), wait for event (queue), send signal to agent, insert new sub goal, adjust sub goal, remove sub goal. While "move", “swap” and “pause” incorporate a time period after which a new activation is scheduled (and thus contain the concept of speed of the action) the last three actions can happen instantaneously. Additional actions can be added as required without breaking the design.

IMPLEMENTATION CASE STUDIES

Incorporating the context-mediated concept into the existing PEDFLOW model consisted of replacing the sequence of

pre-processing / observation / transformation / decision / action-execution / post-processing
with
pre-processing / context-action-creation / action selection / action-execution / post-processing

Like in the original model, *pre-* and *post-processing* deal with program-related housekeeping and statistics, and *action-execution* performs the action. *context-action-creation* is a sequence of several context detection and action evaluation steps. It can be considered as a framework or skeleton to which new contexts can be added as long as they obey the interface definitions. For every context there is context detection as a condition check. This can be a simple *true* for the basic context or a check of an internal flag, but typically involves observation of the grid and/or the interrogation of other entities. Once it is established a context applies it is evaluated and possible actions are derived. An *action* object holds all the information related to a particular action, namely *type, direction* (if applicable), associated *delay* and *weight.* Action objects are aggregated in a collection class for easy access during action selection. First the collection is processed for disabled actions and then the random choice is performed. The fact that all actions are pre-evaluated makes their processing (e.g. to transform the weight distribution) simple.

For long lists of contexts computing efficiency can be improved in several ways: 1) Context checks can be extended to include a flag *more_contexts.* This is set to false and will as a result skip remaining contexts as soon as an absolute context is encountered. Absolute contexts are placed early in the list. 2) The area in front of the person is only observed once and the processed results stored in a local structure, which can be accessed for all contexts. This cuts down on the overhead involved in translating the relative direction into grid co-ordinates.

The following contexts have been implemented so far:

Context	Detection Method	Possible Actions	Comment
basic	none (applies always)	straight, left, right	if there is nothing within DD, move straight, otherwise left/right
undecided	observation: straight is blocked, left and right free	left, right	predict movement of blocking entities by evaluating their position at the next activation and apply basic context
crowded	observation: all three lanes are blocked	pause, avoid left, avoid right	currently all three actions at equal likelihood; a more sophisticated approach is desired, but hard to develop & validate
sub goal reached	observation: associated sub goal area reached	remove current sub goal and continue to next	absolute action (highest priority)
facing an oncoming person	interrogation: person in front is walking in the opposite direction	swap position with person	swapped person is flagged - see just swapped
just swapped	internal flag	pause to account for swapping	absolute action (highest priority)
walking with a partner	interrogation entity (partner)	move towards partner	detailed explanation below
pedestrian crossing	interrogation: red light	don’t enter road	detailed explanation below
queuing	observation, interrogation	move towards last person in queue	access to a goal is blocked by agents with the same goal; queue shape emerges from interaction between agents
attraction	observation, flag, interrogation	sub goal, pause	detailed explanation below
walking around corners	observation	new sub goal	under development
crossing the road	observation	new sub goal(s)	under development

TABLE 1 Currently implemented contexts.

It becomes clear from the table that some contexts are simple and can be implemented with little performance overhead whereas others require complex algorithms. Three of the more involved will be discussed in more detail:

Walking with a partner. Our research shows that the majority of non-commuting journeys are performed not as single person, but in groups and that group behaviors are significantly different from these of singletons. These differences are difficult to capture in the model. Some features, like reduced average walking speed, can be realized with adjusted parameters; others require elaborate algorithms. One typical group feature is the desire of the group to stay together - even if that means individual disadvantages like pauses or detours are required. For pairs of pedestrians (the dominating group size) we try to capture this behavior in the context "walking with a partner". For bigger groups, more complex algorithms are required to implement "leader-followers" behavior. The prerequisite for "walking with a partner" is that both agents have the same desired speed and the same goal. Each agent in the group contains a reference to the other agent. Under undisturbed circumstances (basic context) they will walk next to each other. If however an additional context requires the agent to pause, slow down or deviate from the common path (e.g. it is faced with an obstruction), special action is required. One part of the solution is to reduce the likelihood of such situations occurring. Each agent flagged as "walking with a partner" will reduce the likelihood of actions that move it away from that partner. If such an action happens anyway, either or both partners will initiate actions to come together again. Depending on the situation the partner will pause or (more likely) move towards the other agent. This does not necessarily happen immediately, as it will take a certain minimum distance (upper limit) to become aware of losing the partner. Once it does happen, the agent will only stop moving towards the partner once a distance is under a certain maximum range (lower limit). The limits can be different for both partners and reflect the affinity of the pedestrian to its partner. This is illustrated in FIGURE 2. In part a) agent A has detected the loss of its partner while B has not. A will therefore initiate action to bring itself close to B. If it doesn't succeed but instead gets further away at some point B will detect the loss as well. In part b) agent B will stop any activities while agent A still needs to get closer.

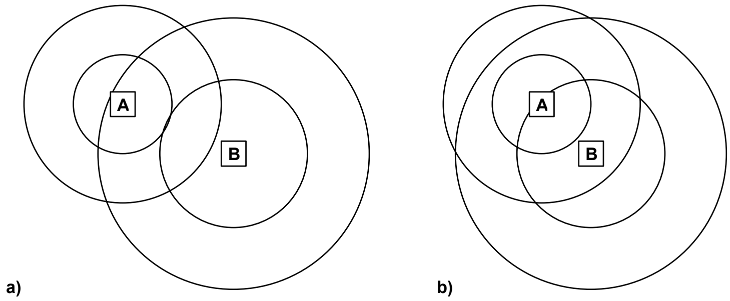

FIGURE 2 "Walking with a partner": context detection (a) and leaving the context (b).

When analyzing the resulting behavior one must keep in mind that other contexts have an influence on the agents' behavior and the choice of action is made randomly. For example an obstruction will force them to deviate from their path. The agents will move independently and the relative distance has hence to be re-established for every activation. Possible superposition effects with other contexts are shown as a screen capture from the model in FIGURE 3. In all six cases the two agents have the same parameters and want to move from the left to a goal on the right. Obstructions in their way force them to split up. Depending on the weighted random selection different routes emerge. In all cases they will walk close together again eventually, although the time until they achieve it and the route they take differ.

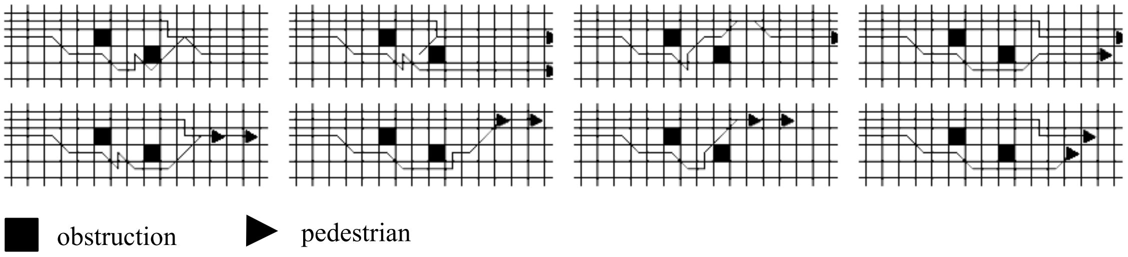

FIGURE 3 Superposition of basic and "walking with a partner" context.

Pedestrian Crossing. One common feature in the urban environment which affects pedestrian flow is the signalized pedestrian crossing. The primary effect is the red light, which will cause affected people to stop at the road. Secondary effects are that people behind the first line will start to queue, in other words they will not try to avoid them as they would ordinary obstructions but instead move in close behind, possibly filling gaps as the number of waiting people increases. Furthermore people walking at a different direction and not affected by the light are now faced with a crowd of people blocking their path.

When trying to model this situation, the first step is to describe the condition used in context detection. An agent needs to determine if a traffic light is applicable to it and if the light shows red. While the latter can easily be determined by interrogating the traffic-light entity, the former is more complicated. Implementing it via observation is inefficient, as relative physical location is not enough to determine if the red light applies, and the search area would be too large for a repeated scan. Instead a method is used, where a list of all traffic lights in a scenario is maintained. Every traffic light has an associated area of effect. By querying the traffic-light entity, a pedestrian agent can ascertain if its co-ordinates fall within this area and if the light is red. The last condition in the context detection is to compare the direction of influence to the walking direction of the agent to assure it is not affected by lights from which it is moving away. FIGURE 4 shows a screen capture of a part of a pedestrian crossing scenario. The traffic light A affects agent C, but not agent B (walking at a direction that is 90degrees different from that of the traffic light).

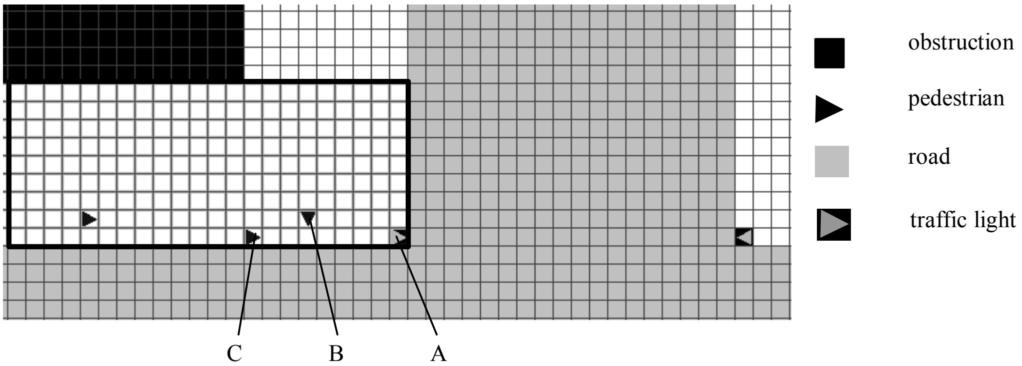

FIGURE 4 Pedestrian crossing.

An agent affected by a red light will not be able to make movements that would take it out of the associated area for that particular light. Agent C will therefore continue to move towards the road while still being subjected to other contexts (e.g. the avoidance of B). Once it has reached the road (which coincides with the border of the affected area) it will stop. If other agents have already lined up on the road, the agent will also be subjected to the "crowd" context to allow it to get close to other pedestrians. Being confined to the context area also means, it will not be possible for the agent to swerve onto the road to avoid the queue. If the light changes to green all restrictions on the agents are lifted and they will move according to other applicable contexts. Although this lends itself to an implantation as a wait action where the traffic light maintains a queue of affected pedestrians it is more realistic (but also less efficient) to realize it as agents repeatedly checking the traffic light entity. This way they can deal with other contexts while they wait for the light, for instance get out of the way of other pedestrians or move towards an attractor. Currently the model does not allow agents to ignore the red light, although it might be desirable to do so, as some people ignore it in real life.

Attraction: Attractors are used to model features of the environment that attract the attention of a pedestrian and distract it from its current goal (e.g. street musicians or shop displays). Usually the person will move towards the attractor and pause for a certain time, before continuing to move towards the original goal. Unlike traffic lights, attractors do not affect all passing pedestrians in the same way. To detect the context of attraction a similar approach to pedestrian crossings is used. An agent will check a list of available attractors for applicability. It maintains a list of attractors already detected and will only check attractor not yet contacted. If the distance to an attractor is smaller than the attraction range of the attractor and the attractiveness of it higher than the "attractivety" (likelihood of being attracted) of the agent, the context of attraction is entered. A new sub goal is inserted and the agent starts moving towards the attractor. The agent will stop moving if the distance to the attractor is equal to or closer than the attractor's attraction distance. This is similar to the "walking with a partner" context. Once this limit is reached, the agent will leave the context of attraction. The sub goal is removed and the agent is ready to pursue the original goal again. By associating a delay with this removal, the time spent by the agent in close vicinity to the attractor is represented. This time is derived from the difference of attractiveness and "attractivety", but the representative ranges are still under investigation.

EVALUATION

The original idea of context-mediated behavior developed from two needs: the decision algorithm of the original PEDFLOW model was too rigid in its design and the rules could not be validated individually. Rules had to conform to the specified format and the addition of new inputs required the re-creation of all rules. As a result, new functionality could only be added by means of hard coded exceptions to the rules, leading to code that was difficult to read and had unwanted side effects in the execution.

In the new implementation of the PEDFLOW model these limitations don't exist. Additional functionality can be implemented by means of new contexts. Adding a new context means to describe its detection and produce an algorithm to derive actions from observations. New flags, interfaces to interrogate other entities and even new actions can be added as all contexts work independently and do not interfere with each other. Contexts can be tested in selected combinations to eliminate interference.

While the flexibility exists as far as implementation is concerned, contexts need to remain meaningful if combined with other contexts and reflect actual pedestrian behavior. Let us consider the example of an attractor near a pedestrian crossing. From the two contexts it would be quite possible for agents to assemble in the middle of the road during the green phase of the traffic light and stay there when the light turns red. This is not realistic. It is important for the designer to understand the implications of the layout and the placement of entities. Fortunately new designs can be tested and the visualization will immediately show conflicts.

The price for the flexibility is increased demand on computing performance. All contexts and resulting actions are computed for every step, even though only one action will eventually be executed. Consequently the introduction of a new context has an impact for all agents in the model, even though it might only apply to a few. For efficiency reasons the context detection is optimized to evaluate simple conditions first. In the case of the pedestrian crossing it is faster to check for a red light first and only then determine if the agent falls within the affected area. Often different contexts use similar algorithms in the action evaluation. For example, both the attractor and the pedestrian crossing contexts require the agent to find all existing traffic lights/attractors, respectively. Re-use of results however is difficult due to the de-coupling of the contexts. Despite these issues, today's typical desktop computer is more than adequate to run the PEDFLOW model.

SUMMARY AND CONCLUSION

In the PEDFLOW model of pedestrian flow, autonomous agents (representing pedestrians) re-evaluate their environment (static and dynamic entities) with every step as they negotiate an urban space (e.g. pavement). The approach of context-mediated behavior allows agents to deal with multiple concurrent influences in a well-defined manner. Contexts are detected, evaluated and possible actions generated. Additional contexts can be added as part of the development of the model to deal with not yet explored influences. Examples are: walking around corners and the unregulated crossing of roads. Although the context-mediated behavior approach is very flexible, it makes high demands on computational power, because all possible actions per step are generated but only a single one executed. The number of possible contexts, their complexity and the non-determinism of the random choice makes it hard to calibrate and validate the model. This can be mitigated by an incremental approach when adding new contexts. The model is unique in its ability to simulate complex interactions in an urban environment.

ACKNOWLEDGEMENTS

This work is funded by the UK-EPSRC grant reference GQ/M59792, awarded to Jon Kerridge and Julian Hine. It is being undertaken in collaboration with the City Councils of York and Edinburgh and the consultancy Halcrow Fox, UK.

REFERENCES

1. Department of the Environment Transport and Regions (DETR). *Encouraging Walking: Advise to Local Authorities*. London, March 2000, 99AILT0220
2. Willis, A., R. Kukla, J. Hine, and J. M. Kerridge. Developing the behavioural rules for an agent-based model of pedestrian movement. *Proceedings of the European Transport Conference*, 2000, P445: 69-80.
3. Blue, V.J and J. L. Adler. Cellular automata model of emergent collective bi-directional pedestrian dynamics. In M. Bedau, McCaskill, J., Packard, N., and Rasmussen, S. (eds.) *Artificial Life VII*, MIT Press, 2000, pp. 437-445.
4. Dijkstra, J. , H.J.P. Timmermans and A.J. Jessurun. A Multi-Agent Cellular Automata System for Visualising Simulated pedestrian Activity; Theoreteical and Practical Issues on Cellular Automata. In S. Bandini and T.

Worsch (eds.) *Proceedings on the 4th International Conference on Cellular Automata for research and Industry*, Springer Verlag, held at Karlsruhe, Germany, October 4-6, 2000. pp. 29-36.
5. Schelhorn, T., D. O'Sullivan, M. Haklay and M. Thurstain-Goodwin STREETS: an agent-based pedestrian model. Presented at *Computers in Urban Planning and Urban Management*, Venice, 8-10 September, 1999.
6. Helbing, D. , I. Farkas, P. Molnar and T.Vicsek. Simulation of Pedestrian Crowds in Normal and Evacuation Situations. In M.Schreckenberg, S.D.Sharma (eds.) *Pedestrian and Evacuation Dynamics*. Springer Verlag Berlin and Heidelberg, 2001, pp. 21-58
7. Hoogendoorn, S.P. and P.H.L. Bovy. Hoogendoorn, S.P. and P.H.L. Bovy. Microscopic Pedestrian Wayfinding and Dynamics Modelling, In M.Schreckenberg, S.D.Sharma (eds.) *Pedestrian and Evacuation Dynamics*. Springer Verlag Berlin and Heidelberg, 2001, pp. 123-154
8. Penn, A. and A.Turner. Space Syntax Based Agent Simulation In M.Schreckenberg, S.D.Sharma (eds.) *Pedestrian and Evacuation Dynamics*, Springer Verlag Berlin and Heidelberg, 2001. pp. 75-85.s
9. Kukla, R., A.Willis, J.Hine and J.Kerridge. PEDFLOW: development of an autonomous agent model of pedestrian flow. In *Transportation Research Record 1774*, TRB, National Research Council, Washington, D.C., 2001, pp. 11-16.

LIST OF TABLES AND FIGURES