Evacuation Simulation through Formal Emotional Agent based Modelling

Ilias Sakellariou[1], Petros Kefalas[2] and Ioanna Stamatopoulou[2]
[1]Dept. of Applied Informatics, University of Macedonia, 156 Egnatia Str., 54636, Thessaloniki, Greece
[2]Dept. of Computer Science, The University of Sheffield International Faculty, City College, L. Sofou 3, 54624, Thessaloniki, Greece

Keywords: Emotional Agents, X-Machines, Formal Modelling, Simulation, Evacuation Modelling.

Abstract: Evacuation Simulation is recognised as an important tool for assessing design choices for urban areas. Although a number of approaches have been introduced, it is widely acceptable that such simulation scenarios demand modelling of emotional aspects of evacuees, and how these affect their behaviour. The present work, proposes that formal agent modelling based on ^{e}X-machines can rigorously define but also naturally lead to realistic simulations of such scenarios. ^{e}X-machines can model agent behaviour influenced by emotions, including social aspects of emotions, such as emotion contagion. The developed formal model is refined to simulation code, that is able to visualise and simulate crowd believable behaviour.

1 INTRODUCTION

Assessing the evacuation capability of an area under emergency conditions is a crucial aspect to the design of modern urban areas, such as buildings, stadiums, metro stations, etc. Computer based simulation has been identified as an important tool for such evacuation analysis and assessment of design choices. A large number of evacuation models have been proposed (Santos and Aguirre, 2004; Kuligowski, 2004) that follow different approaches (Zheng et al., 2009) with respect to the method used and granularity (scale) of the simulation.

There is a remarkable range of crowd evacuation models, that includes *fluid dynamics* models, *social force* models, *cellular automata* and *gas lattice* models. To our interest, *Agent Based Modelling* (ABM) has been widely adopted as a promising approach to evacuation modelling, due to a number of advantages (Bonabeau, 2002), such as emergence, flexibility and natural description of the model under study. ABM can easily accommodate the diversity of population with respect to walking speed, age, behavioural changes depending on psychological stress, and disabilities (Christensen and Sasaki, 2008). Successful case studies using ABM have been reported, such as a metro station evacuation (Zarboutis and Marmaras, 2004), validation of real data in the "Garuda Indonesia Airways Accident" (Miyoshi et al., 2012), and evacuation plans of the Castello Ursino (Camillen et al., 2009). However, non of these approaches employs a formal modelling of the behaviour of evacuating agents. The significant amount of research in the area of evacuation modelling and simulation could not possibly be reported in the context of the present paper; the reader should refer to reviews, such as (Kuligowski, 2004; Schadschneider et al., 2009; Zhou et al., 2010; Zheng et al., 2009).

In order to achieve a realistic simulation, agents must be able to demonstrate believable behaviour, the latter reflecting behavioural changes under stress conditions, i.e. considering emotions as part of the agent reasoning process. So far, there is not yet a widely accepted definition of emotions supported by a complete theory that can describe how emotional processes affects reasoning in general (Kleinginna and Kleinginna, 1981; Frijda, 2007). Most commonly used psychological theories in agent design today refer to the reactions to three types of stimuli (OCC model) (Ortony et al., 1988), while a number of computational models of emotions, logically formalised as BDI agents have been proposed, most recent being (Marreiros et al., 2010; Steunebrink et al., 2010; Steunebrink et al., 2011). The role of emotions as well as the type of agents in emergency evacuation has been receiving increased attention (Tsai et al., 2011; Zoumpoulaki et al., 2010). Rather recently, more complex issues, such as emotion contagion (Hoogen-

Sakellariou I., Kefalas P. and Stamatopoulou I..
Evacuation Simulation through Formal Emotional Agent based Modelling.
DOI: 10.5220/0004824601930200
In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence* (ICAART-2014), pages 193-200
ISBN: 978-989-758-016-1

doorn et al., 2010), have been addressed by different models.

The purpose of this paper is twofold: demonstrate that formal state based modelling can rigorously define complex agent behaviour influenced by emotions, and that such models can be refined to executable code, thus leading to realistic simulations of evacuation scenarios. Thus, the main contribution of this work is the use of ^{e}X-Machines with a complex emotion model involving emotion contagion, demonstrating (a) its power as a formal method to be used in cases such as agent based evacuation modelling and (b) a refinement of the former that can produce an executable simulation.

The structure of this paper is as follows: Section 2 presents the ^{e}X-machines formalism. Section 3 discusses the evacuation scenario used as a working example, whereas section 4 describes how that scenario is modelled as a formal model. Section 5 presents discusses the mapping between modelling constructs and the simulation platform, as well as presents initial experimental results. Finally, Section 7 concludes the paper and suggests direction for future work.

2 EMOTION X-MACHINES

Formal methods allow the mathematically rigorous modelling of complex systems and their behaviour. Although, formal modelling can be viewed as a pedantic step towards development, it offers the ability to prove correctness with respect to the specification. Correctness can be achieved by verification that certain properties hold in the original model and that the system implementation behaves as expected under a complete test set. This work employs X-machines (XM) (Holcombe and Ipate, 1998), that are state-based machines extended with a *memory structure*, that makes the machine more compact compared to memory-less state machines, and *functions* that guard transitions between states. The great advantage is their strong legacy of theory and practice, including testing methods that prove correctness (Ipate and Holcombe, 1997).

The state based orientation of XM provides an intuitive modelling approach of agents, as for instance in the case of biology-inspired MAS (Kefalas et al., 2009). X-machines have been formally extended to model emotional agents, leading to the *emotions X-machines*. The new method introduces an *emotional state*, represented as a vector E containing emotion identifiers.

Definition 1. *An emotions X-machine ($^e\mathcal{X}$) is defined as (Kefalas et al., 2012):*

$$^e\mathcal{X} = (\Sigma, \Gamma, Q, M, \Phi, F, q_0, m_0, E, {}^e\Phi, e_0)$$

where:

- Σ *and* Γ *are the input and output alphabets.*
- Q *is a finite set of states.*
- M *is a (possibly) infinite set called memory.*
- Φ *is a set of partial functions* φ*; each such function maps an input, a memory value and an emotional states to an output and a possibly different memory value,* $\varphi : \Sigma \times M \times E \rightarrow \Gamma \times M$.
- F *is the next state partial function,* $F : Q \times \Phi \rightarrow Q$*, which given a state and a function from the type* Φ *determines the next state.* F *is often referred to as a state transition diagram.*
- q_0 *and* m_0 *are the initial state and initial memory.*
- $E = (\varepsilon_1, \ldots, \varepsilon_n)$ *is a vector containing emotion identifiers.*
- ${}^e\Phi : E \times M \times \Sigma \rightarrow E \times M$ *is the emotions revision function.*
- e_0 *is the initial vector of emotion identifiers representing the initial emotional state.*

Input triggers the emotions revision function thus changing the emotional state and the memory. The same input triggers a transition function which will return a new state. Thus, the emotions vector can change the computation path by affecting the applicability of functions. More formally, computation is defined as follows.

Definition 2. *An* $^e\mathcal{X}$ *computation state is defined as the tuple* (q,m,e)*, with* $q \in Q$ *and* $m \in M$ *and* $e \in E$*. A computation step, which consumes an input* $\sigma \in \Sigma$ *and changes the computation state* $(q,m,e) \vdash (q',m',e')$ *is essentially composed of two substeps:*

- $(q,m,e) \overset{^e\varphi}{\vdash} (q,m'',e')$ *with* $q \in Q$*,* $e,e' \in E$ *and* $m,m'' \in M$*, such that* ${}^e\varphi(e,m,\sigma) = (e',m'')$
- $(q,m'',e') \overset{\varphi}{\vdash} (q',m')$ *with* $q,q' \in Q$*,* $m',m'' \in M$ *and* $e' \in E$*, such that* $\varphi(\sigma,m'',e') = (\gamma,m')$ *and* $F(q,\varphi) = q'$.

A computation defined as the series of computation steps that take place when all inputs are applied to the initial computation state (q_0,m_0,e_0).

Although $^e\mathcal{X}$ seems to provide an elegant way to model agents under emotions, further investigation of its expressive power and how it can accommodate the large number of theories proposed in the area is necessary. Thus, one of the aims of the current paper, is to demonstrate the effectiveness of $^e\mathcal{X}$ in representing an artificial emotion contagion model.

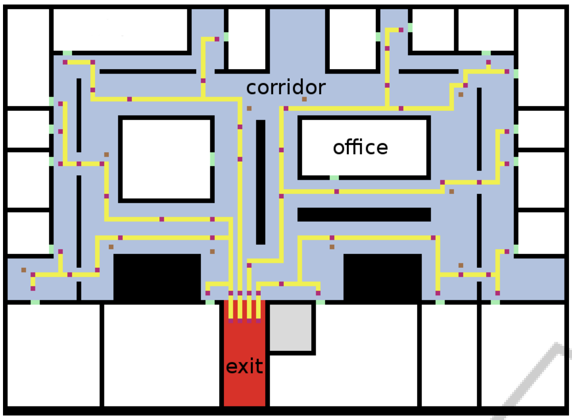

Figure 1: Example blueprint of an office floor, indicating the plans from each room to the exit. For illustrative purposes evacuation paths are depicted in yellow in the figure.

3 THE EVACUATION SCENARIO

An office floor evacuation scenario was selected as a working example for this work. The floor (Figure 1) consists of several offices, connected though corridors and one exit. Lines in the figure represent evacuation paths, with red dots indicating points that form the evacuation plan (see section 4). The simulation concerns evacuee behaviour from the moment danger is perceived until the evacuee exits the office floor. The scenario assumptions are:

- Initially, all evacuees are located inside offices.
- Evacuees have limited visibility, i.e. their knowledge (sensing) about the environment is limited to a neighbourhood.
- There are parents accompanying children.
- Evacuation plans are posted in each office door, that can be consulted by evacuees. Plans come in the form of a path from the office position to the exit.
- Security officers, positioned in strategic locations, provide evacuees with directions to exits, similar to evacuation plans.

Informally, agents are initially in a “No Emergency” state, and upon perceiving danger (alarm bell), they proceed to the room door, obtain the evacuation plan and follow the latter to reach the exit. However, during the evacuation, agents might get disoriented due to an increased emotional level, “forget” their original plan and wander, until they receive instructions either from a security officer, or obtain a new plan from a spot where it is posted. Parents ensure at each step of the evacuation that their child is close; in the case that is not they drop temporarily their current plan and engage in a searching behaviour to find their child before resuming evacuation. Finally, emotional levels affect how fast an agent is moving; higher emotional levels lead to an increased walking speed.

4 MODELLING USING ${}^{E}\mathcal{X}$

Modelling in the ${}^{e}\mathcal{X}$ formalism requires specifying agent behaviour in terms of states, functions, input, output and emotions (Definition 1). Input Σ concerns the agent’s *percepts* (P hereafter) the latter being:

- The set of available “empty” positions the agent can move to, in the form of $(Pos, empty)$, where Pos is the point’s coordinates.
- Positions of interest, as for example the location of a room door ($door(Pos)$).
- Plan information, obtained only at specific points (doors), in the form $(seq(Pos_i), plan)$, or by security officers, $(seq(Pos_i), officer)$ where $seq(Pos_i) : [P_1, \ldots P_n]$ is a sequence of points in space or ε to denote the empty plan.
- Location and status of the agent’s child ($\{child\text{-}close, see\text{-}child, child\text{-}exited\}$).
- Emotional values, expressiveness and distance of other agents in its vicinity, in order to compute emotion contagion strength (section 4.2).

The ${}^{e}\mathcal{X}$ holds a number of evacuee (agent) characteristics, some of which determine agent behaviour, while others hold information regarding evacuation. For instance, the fact that the agent has a child (Ch) or its personality trait are memory values that allow to easily model crowd diversity using a single ${}^{e}\mathcal{X}$, without deviating from the formal definition. Other elements of memory hold more “dynamic” information, as for example the current speed (S) of the agent, the current evacuation plan $seq(Pos_i)$ or the position Pos.

The *state transition* diagram of the model, i.e. F in Definition 1, is depicted in Figure 2. “No Emergency” is the *initial state* in which evacuees are found, and state “Exiting” is the *final state*, which when reached, agents are considered to have evacuated the floor successfully and disappear from the simulation. Note that in the figure, transitions are marked by functions. In the following, some of the functions are discussed in order to show how agent behaviour is encoded. For instance, function *perceive-danger* simply acts as a guard for state transition between the “No emergency” state and that of “Looking for Plan”:

$\varphi_{perceive\text{-}danger} : (P, (\varepsilon, Pos, S, Ch), E) \mapsto$

$(\text{“}DangerPerceived\text{”}, (\varepsilon, Pos, S, Ch))$
$if\ danger \in P$

Function *exiting-room* is slightly more complicated and implements the strategy of the evacuee to move closer to the room door. The function moves the agent to a new "empty" position. Since positions in percepts are given as sets of coordinates it is easy to compute whether the distance of the new point is indeed closer to the door. The boolean function $canMove(Pos, NewPos, S)$ returns true if $NewPos$ is reachable by the agent in a single simulation step given the agent's speed S.

$\varphi_{exiting\text{-}room} : (P, (Plan, Pos, S, Ch), E) \mapsto$
$(\text{“}ExitingRoom\text{”}, (Plan, NewPos, S, Ch))$
$if\ (NewPos, empty) \in P$
$\wedge\ canMove(Pos, NewPos, S) \wedge door(DoorPos) \in P$
$\wedge\ dis(NewPos, DoorPos) < dis(Pos, DoorPos)$

Function *get-dissoriented* presents an interesting case since it also takes into account emotions, discussed in the section that follows:

$\varphi_{get\text{-}dissoriented} : (P, (Plan, Pos, S, Ch), E) \mapsto$
$(\text{“}GotDissoriented\text{”}, (\varepsilon, Pos, S, Ch))$
$if\ horror\text{-}level(E) \in \{panic, hysteria\}$

In a similar manner other functions depicted in Figure 2 are modelled.

4.1 The Emotion Revision Function

The emotional state of the evacuee is represented as the a vector E containing (artificial) emotion identifiers. Only the basic emotion *Horror* (Parrott, 2001) is included, which can be assigned a set of crisp values, i.e. *HorrorLevel*={*calm, alarm, fear, terror, panic, hysteria*}. To model the strength of the emotion and determine the horror level, a strength value SV is required, ranging from $SV = 0 \dots 100$ and the initial emotion vector is $e_0 = (calm, 0)$.

The emotion revision function updates the emotion strength and horror level, given *individual emotion strength updates* δSV_{ind} and *emotion contagion* δSV_{social} according to the following:

$SV' = SV + \delta SV_{ind}(P, M, E) + \delta SV_{social}(P, M, E)$

Individual emotion strength updates depend on the rate of change of E, different for each evacuee, since evidence suggests that there exist individual differences in affective response to emotion eliciting stimuli. *Personality trait*, for example, is one relevant factor. Some individuals have a predisposition (sensitivity response) towards experiencing certain emotions, so different personality traits are responsible for how quickly an emotional state is reached, maintained and recovered from (Dalgleish and Power, 1999), resulting to some agents reaching a state of panic or hysteria more easily. The personality trait in the current implementation ranges between 0.5 to 1.5 (normal distribution) and thus the change of emotion strength due to the former is given by:

$f_{ind}(M) = pTrait(M) * EmotionLevelInc$

where M is the Xm Memory, $pTrait(M)$ a function that obtains the agent personality trait from the $^{e}\mathcal{X}$ Memory and $EmotionLevelInc$ a constant, set as an experiment/simulation parameter. However, in a realistic situation, emotional strength levels can drop when certain situations occur, e.g. when a plan is perceived, or instructions are received from a security officer, thus S should be decreased by a factor:

$f_{percept}(P, E) = -c * eStrength(E)$
$if\ (seq(Pos_i), officer) \in P \vee (seq(Pos_i), plan) \in P$

where c is a constant set as a simulation parameter and $eStrength(E)$ a function that represents that the emotional strength is obtained by the emotional vector. Thus the individual strength update becomes:

$\delta SV_{ind}(P, M, E) = f_{ind}(M) + f_{percept}(P, E)$

4.2 Emotion Contagion

The emotional contagion theory implemented the model, is a simplification of the ASCRIBE model proposed in (Hoogendoorn et al., 2010), that has been successfully used to simulate the May 4 incident in Amsterdam, Netherlands (Bosse et al., 2011). Briefly, the ASCRIBE model introduces *contagion strength* γ_{iSj} that determines the strength by which agent j influences on some state S agent i. Since in the model under study intra agent influences that concern beliefs or intentions are not considered, but only emotional strength levels (as "fear" in (Bosse et al., 2011)), contagion strength always concerns emotional level SV and thus γ_{ij} is given by the following equation:

$$\gamma_{ij} = expressiveness_j * a_{ij} * openness_j \qquad (1)$$

In the above, $expressiveness_j$ and $openness_j$, are agent specific values, and a_{ij} the *channel strength*. The latter is determined by a linear function that depends on the euclidean distance between the agents $dis(Pos_i, Pos_j)$, in the area of influence. In equation 2 dis_{infl} is the radius of the *area of influence*, i.e. the area which contains agents, emotional levels of which affect agent i.

$$a_{ij} = 1 - \frac{dis(Pos_i, Pos_j)}{dis_{infl}} \qquad (2)$$

The overall contagion strength is determined by:

$$\gamma_i = \sum_{i \in AF} \gamma_{ij} \qquad (3)$$

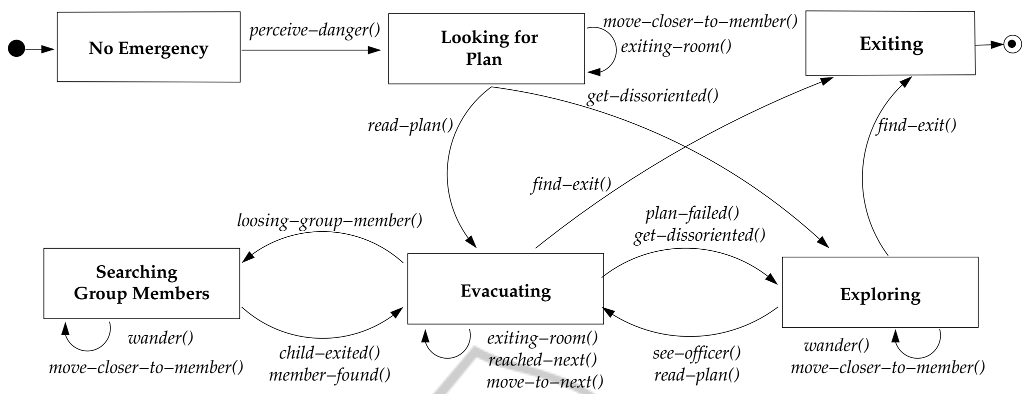

Figure 2: The XM state transition diagram. "No emergency" and "Exiting" are the initial and final state respectively.

where AF is the set of agents currently located in the area of influence of agent i. Since the present work adopts a simpler model for determining emotional contagion than that of ASCRIBE, we simply compute the influence of agents by the following equation:

$$\delta SV_{social}(P, Mem, E) = \frac{\sum_{j \in AF} (\gamma_{ij}/\gamma_i) * (SV_i - SV_j)}{|AF|}$$

where SV_i is the emotional strength of the agent i, obtained by $eStrength(E)$ and SV_j are the emotion levels of other agents in the area of influence, obtained by the percept of the agent.

Given the emotion revision function described above, the horror-level and the speed of the agent are updated by simple mapping functions.

5 EVACUATION SIMULATION

Agent simulation is a valuable tool for informally verifying system properties, that can reveal a variety of desired, or unwanted and unexpected emerging behaviours. One of the main benefits of formally specifying emotional agents using ${}^{e}\mathcal{X}$, is that a simulation can be easily derived.

A large number of agent simulation platforms and tools have been proposed in the literature. NetLogo (Wilensky, 1999) has been widely accepted as a platform for agent based simulation of emerging and social phenomena. It offers an easy to use environment for the complete development and testing of the simulation experiment with strong visualization facilities and a programming language that has a strong functional flavour. These reasons led to its introduction as the tool of choice for implementing the simulation model described in Section 4.

However, direct encoding of an ${}^{e}\mathcal{X}$ model is not supported by the NetLogo language, in the sense that the user should manually encode in the Netlogo language functions (reporters) representing ${}^{e}\mathcal{X}$ functions, emotions and the state transition diagram, a task that proves to be error prone and result in non-easily modifiable code. Thus, in order to accommodate the former, a modified version of a state machine domain specific language (DSL), originally introduced in (Sakellariou, 2012), was developed that supports ${}^{e}\mathcal{X}$. The new DSL allows a direct encoding of the ${}^{e}\mathcal{X}$ model in a notation that facilitates its rapid development. Each agent is mapped to a NetLogo turtle, with a number of turtle variables holding percepts, memory, and the emotion vector. These demand the development of appropriate NetLogo code, since they are experiment depended. Probably, the most interesting part of the DSL is the encoding of the ^{e}X-Machine state transition diagram and the corresponding functions, with the DSL allowing their encoding in a form very close to the specification presented. For instance, part of the transition diagram depicted in Figure 2, is encoded as follows in the simulation:

```
to-report state-def-of-persons
report (list
 state "No Emergency"
 # x-func "perceive-danger"
      goto "Looking for Plan"
 # otherwise do "nothing" goto "No Emergency"
 end-state    ...
 state "Evacuating"
 # x-func "loosing-group-member"
      goto "Searching Group Members"
 # x-func "get-dissoriented" goto "Exploring"
 # x-func "plan-failed" goto "Exploring"
 # x-func "find-exit" goto "Exiting"
 # x-func "reach-next" goto "Evacuating"
 # x-func "move-to-next" goto "Evacuating"
 # x-func "exiting-room" goto "Evacuating"
 # otherwise do "nothing" goto "Evacuating"
 end-state    ...
end
```

${}^{e}\mathcal{X}$ functions that annotate transitions, are directly encoded as reporters, although the semantics of the results they return are handled by a meta-interpreter, discussed below. This encoding is different that that reported in (Stamatopoulou et al., 2012) that uses Prolog as the language for encoding functions, and has increased efficiency. For instance the function:

$\varphi_{perceive\text{-}danger} : (P, (\varepsilon, Pos, S, Ch), E) \mapsto$
$(\text{“}DangerPerceived\text{”}, (\varepsilon, Pos, S, Ch))$
if $danger \in P$, is now directly encoded in NetLogo as:

```
to-report perceive-danger [Perc Mem Emo]
 ifelse has-percept "danger" Perc
  [report (list true "Danger Perceived" Mem)]
  [report [false] ]
end
```

A meta-interpreter, fully developed in the NetLogo language that respects the ${}^e\mathcal{X}$ semantics is responsible for executing the agents, obtaining input from the simulation environment and updating the agent “simulation state” in the latter. At each computation step, the meta-interpreter determines the set of applicable functions to the current ${}^e\mathcal{X}$ state, and selects a single function to “fire” leading to state and memory change. In the original ${}^e\mathcal{X}$, function selection among multiple applicable functions is non-deterministic, however, such an approach creates a number of problems in the simulation. Thus, in the DSL transition definitions as presented above, an ordering on the function selection is implied, with the functions appearing higher in the state definition having priority over those appearing lower.

Having the layer for specification and execution of ${}^e\mathcal{X}$ agents, we developed a simulation experiment (available in `http://users.uom.gr/~iliass`) that allows a number of parameters to be set, such as the total number of people on the office floor, the number of parents as well as the monitoring of various parameters, such as the average time units required for evacuation and the total evacuation time. The latter is defined as the number of cycles required for all evacuees to exit the office floor. In the simulation the following assumptions hold:

- Space is discrete, with each individual occupying a 40×40 cm cell, as in (Kirchner et al., 2003). Discretisation of space is well suited both to the ${}^e\mathcal{X}$ formalism, since the latter deals with discrete events (input/output), as well as to NetLogo, since the patch size can be adjusted according to this assumption. The office floor measures 49m × 35m.
- Agent speed ranges between 1 and 4, thus agents can move with a maximum speed of approximately 1.6m/sec.
- Agents’ vision is obstructed by walls, leading to a limited perception of the environment.
- Children are modelled as simple reactive agents, that follow their parents if they have visual contact with them, otherwise stay still.

6 EXPERIMENTS

A set of experiments was conducted with a varying number of agents and security officers, using two different emotion functions, the first (no EC) without considering social aspects of emotion contagion, whereas the second taking into account emotion contagion (EC) with an influence distance of 10. All results listed in table 1 are average times of 20 runs with varying initial conditions.

Although results are preliminary and need further extensive experimentation and model validation, some initial observations can be made. A first observation is that the number of security officers present in the area decreases the evacuation time. This is expected since, (a) emotion levels decrease when agents “see” a security officer, thus the overall population remains calm during evacuation and (b) when an agent becomes “disoriented” then the presence of security officers providing evacuation instructions leads to these agents resuming evacuation sooner. A second observation is that emotion contagion seems not to play a significant role when the number of evacuees is less than 1000. In fact, evacuation time appears to be slightly less, since calm crowds help keep emotional levels within bounds. However, when the population increases, emotion contagion increases evacuation time, since increase of emotion levels “spread” within the crowd, an effect that has been observed in some real life situations as well.

In Table 1, columns indicated with “(P)” show evacuation times under the existence of 50 parents. In these experiments, simulation was terminated when the number of evacuees in the simulation area was under 10% of parent population. This terminating condition was adopted specifically for these tests, since according to the model, when parents loose visual contact with their children, they engage in a random exploration that can even, in some cases, lead to a non completed evacuation within reasonable time limits and increase dramatically evacuation times. Even with such a terminating condition, in some cases experiments failed to terminate within a bound of 13000 time units (indicated with a dash in Table 1). In all cases, the evacuation times under the presence of parents are significantly increased, since the latter have to interrupt their exit to look for their children in many cases, and thus remain on the evacuation floor for longer periods of time. In this set of experiments,

Table 1: Evacuation time in alternative scenarios; EC signifies that emotion contagion is included in the emotion revision function, and no EC otherwise. (P) indicates simulations involving 50 parents.

	Security Officers											
	No EC			**EC**			**(P) No-EC**			**(P) EC**		
Agents	**0**	**5**	**10**	**0**	**5**	**10**	**0**	**5**	**10**	**0**	**5**	**10**
200	101	105	108	94	100	102	2532	435	185	4918	781	209
400	126	130	134	126	128	131	4036	1110	271	8260	882	304
600	252	209	192	181	194	204	5910	2015	394	6621	2703	567
800	666	353	337	627	338	325	6698	2641	819	-	3801	575
1000	968	491	423	1702	584	431	7999	3193	1230	-	5471	1316
1200	1132	690	524	2859	970	489	-	-	-	-	-	-
1400	1286	739	541	3742	1052	581	-	-	-	-	-	-
1600	1443	796	618	4426	1345	664	-	-	-	-	-	-
1800	1519	868	686	4581	1384	625	-	-	-	-	-	-
2000	1634	924	715	4816	1594	745	-	-	-	-	-	-

emotion contagion has a increasing effect in all cases, since parents have higher levels of emotion and these values are propagated to other agents as well.

By no means the results reported above constitute a validation of the evacuation model. The aim of the paper was to demonstrate that a believable model of evacuation can be obtained by formal modelling of agents with complex emotion influenced behaviour and that the latter can lead to a simulation.

7 CONCLUSIONS

In this work, formal ABM has been applied to an emergency evacuation scenario. Emotions X-Machines, a special class of state based machines, have been used to model individuals acting under the influence of emotions, the later being revised by both agent perception and a simplified version of the ASCRIBE emotion contagion model. ^{e}X-Machines, seem to provide the modelling constructs to easily accommodate such complex emotion theories and the corresponding agent behaviour. A simulation based on the described model was developed in NetLogo and preliminary results were obtained demonstrating how emotion contagion affects evacuation time.

However, the artificial emotions model is by no means complete. The model needs further enhancement to deal with emotions that affect perception appraisal, communication, etc. Thus, the present work can be extended towards a number of directions. One of them is to investigate the effect of introducing the full OCC model of emotions, as in (Steunebrink et al., 2011) and study its effects to the behaviour of agent in a number of situations. Another direction involves validating the model. Although, evacuation model validation still lacks a set of benchmarks and data sets publicly available, as well as a methodology for such a validation process, a recent proposal described in (Ronchi et al., 2013) introduces a set of validation tests that models can be evaluated against.

Finally, although the formal model was easily mapped to NetLogo, given the DSL developed for the task, a main direction towards the simulation of larger scale experiments, is to develop a compiler from ${}^{e}\mathcal{X}$ specifications to NetLogo. Such a compiler will facilitate large model development, reducing the time required to move from formal modelling to visualisation. Our plans, include a similar compiler to other agent simulation tools, including simulators build in functional programming languages, such as Erlang or Haskell, taking advantage of the efficiency and scalability of the latter.

REFERENCES

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7280–7287.

Bosse, T., Hoogendoorn, M., Klein, M. C., Treur, J., and Van Der Wal, C. N. (2011). Agent-based analysis of patterns in crowd behaviour involving contagion of mental states. In *Modern Approaches in Applied Intelligence*, pages 566–577. Springer.

Camillen, F., Capri, S., Garofalo, C., Ignaccolo, M., Inturri, G., Pluchino, A., Rapisarda, A., and Tudisco, S. (2009). Multi agent simulation of pedestrian behavior in closed spatial environments. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, pages 375–380.

Christensen, K. and Sasaki, Y. (2008). Agent-based emergency evacuation simulation with individuals with disabilities in the population. *Journal of Artificial Societies and Social Simulation*, 11(3):9.

Dalgleish, T. and Power, M. (1999). *Handbook of Cognition and Emotion*. Eds. Chichester: John Wiley and Sons.

Frijda, N. H. (2007). What might emotions be? comments on the comments. *Social Science Information*, 46(3):433–443.

Holcombe, M. and Ipate, F. (1998). *Correct Systems: Building a Business Process Solution*. Springer, London.

Hoogendoorn, M., Treur, J., Van Der Wal, C. N., and Van Wissen, A. (2010). Modelling the interplay of emotions, beliefs and intentions within collective decision making based on insights from social neuroscience. In *Neural Information Processing. Theory and Algorithms*, pages 196–206. Springer.

Ipate, F. and Holcombe, M. (1997). An integration testing method that is proved to find all faults. *International Journal of Computer Mathematics*, 63(3):159–178.

Kefalas, P., Stamatopoulou, I., and Basakos, D. (2012). Formal modelling of agents acting under artificial emotions. In *BCI 2012, 5th Balkan Conf. in Informatics*.

Kefalas, P., Stamatopoulou, I., Sakellariou, I., and Eleftherakis, G. (2009). Transforming Communicating X-machines into P Systems. *Natural Computing*, 8(4):817–832.

Kirchner, A., Klüpfel, H., Nishinari, K., Schadschneider, A., and Schreckenberg, M. (2003). Simulation of competitive egress behavior: comparison with aircraft evacuation data. *Physica A: Statistical Mechanics and its Applications*, 324(3-4):689–697.

Kleinginna, P. and Kleinginna, A. (1981). A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and Emotion*, 5:345–379.

Kuligowski, E. D. (2004). Review of 28 egress models. In Peacock, R. D. and Kuligowski, E. D., editors, *Workshop on Building Occupant Movement During Fire Emergencies. Session 3.3. Gaithersburg, MD*, pages 68–90. NIST Special Publication 1032 (2005).

Marreiros, G., Santos, R., Ramos, C., and Neves, J. (2010). Context-aware emotion-based model for group decision making. *Intelligent Systems, IEEE*, 25(2):31–39.

Miyoshi, T., Nakayasu, H., Ueno, Y., and Patterson, P. (2012). An emergency aircraft evacuation simulation considering passenger emotions. *Computers & Industrial Engineering*, 62(3):746 – 754. Soft Computing for Management Systems.

Ortony, A., Clore, G., and Collins, A. (1988). *The cognitive structure of emotions*. Cambridge University Press, Cambridge, MA.

Parrott, W. (2001). *Emotions in Social Psychology*. Psychology Press, Philadelphia.

Ronchi, E., Kuligowski, E. D., Reneke, P. A., Peacock, R. D., and Nilsson, D. (2013). The process of verification and validation of building fire evacuation models. Technical Report 1822, National Institute of Standards and Technology, US Department of Commerce.

Sakellariou, I. (2012). Turtles as state machines - agent programming in netlogo using state machines. In *ICAART 2012, 4th International Conference on Agents and Artificial Intelligence*, pages 375–378. SciTePress.

Santos, G. and Aguirre, B. E. (2004). A critical review of emergency evacuation simulation models. In Peacock, R. D. and Kuligowski, E. D., editors, *Workshop on Building Occupant Movement During Fire Emergencies. Session 3.3. Gaithersburg, MD*, pages 27–52. NIST Special Publication 1032 (2005).

Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., and Seyfried, A. (2009). Evacuation dynamics: Empirical results, modeling and applications. In Meyers, R. A., editor, *Encyclopedia of Complexity and Systems Science*, pages 3142–3176. Springer.

Stamatopoulou, I., Sakellariou, I., and Kefalas, P. (2012). Formal agent-based modelling and simulation of crowd behaviour in emergency evacuation plans. In *ICTAI 2012, 24th International Conference on Tools with Artificial Intelligence*, pages 1133–1138. IEEE.

Steunebrink, B. R., Dastani, M., and Meyer, J.-J. C. (2010). Emotions to control agent deliberation. In *AAMAS 2010, 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 973–980.

Steunebrink, B. R., Dastani, M., and Meyer, J.-J. C. (2011). A formal model of emotion triggers: an approach for BDI agents. *Synthese*, 185(S1):83–129.

Tsai, J., Fridman, N., Bowring, E., Brown, M., Epstein, S., Kaminka, G., Marsella, S. C., Ogden, A., Rika, I., Sheel, A., Taylor, M., Wang, X., Zilka, A., and Tambe, M. (2011). ESCAPES - evacuation simulation with children, authorities, parents, emotions, and social comparison. In *Innovative Applications Track of AAMAS 2011, 10th International Conference on Autonomous Agents and Multiagent Systems*.

Wilensky, U. (1999). *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern Univ., Evanston, IL. http://ccl.northwestern.edu/netlogo/.

Zarboutis, N. and Marmaras, N. (2004). Searching efficient plans for emergency rescue through simulation: the case of a metro fire. *Cognition, Technology and Work*, 6:117–126.

Zheng, X., Zhong, T., and Liu, M. (2009). Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, 44(3):437–445.

Zhou, S., Chen, D., Cai, W., Luo, L., Low, M. Y. H., Tian, F., Tay, V. S.-H., Ong, D. W. S., and Hamilton, B. D. (2010). Crowd modeling and simulation technologies. *ACM Trans. Model. Comput. Simul.*, 20(4):20:1–20:35.

Zoumpoulaki, A., Avradinis, N., and Vosinakis, S. (2010). A multi-agent simulation framework for emergency evacuations incorporating personality and emotions. In *Artificial Intelligence: Theories, Models and Applications*, volume 6040 of *Lecture Notes in Computer Science*, pages 423–428. Springer Berlin Heidelberg.