BDI versus FSM agents in social simulations for raising awareness in disasters: a case study in Melbourne bushfires

Carole Adam, Univ. Grenoble Alps, LIG, Grenoble, France,
Patrick Taillandier, MIAT, Univ. Toulouse, INRA, Castanet-Tolosan, France
Julie Dugdale, Univ. Grenoble Alps, LIG, Grenoble, France
Benoit Gaudou, Univ. Toulouse Capitole, IRIT, Toulouse, France

ABSTRACT

Each summer in Australia, bushfires burn many hectares of forest, causing deaths, injuries, and destroying property. Agent-based simulation is a powerful tool to test various management strategies on a simulated population, and to raise awareness of the actual population behaviour. But valid results depend on realistic underlying models. This paper describes two simulations of the Australian population's behaviour in bushfires designed in previous work, one based on a finite-state machine architecture, the other based on a belief-desire-intention agent architecture. It then proposes several contributions towards more realistic agent-based models of human behaviour: a methodology and tool for easily designing BDI models; a number of objective and subjective criteria for comparing agent-based models; a comparison of our two models along these criteria, showing that BDI provides better explanability and understandability of behaviour, makes models easier to extend, and is therefore best adapted; and a discussion of possible extensions of BDI models to further improve their realism.

Keywords: *agent-based social simulation; human behaviour modelling; BDI agents; GAMA platform; crisis management; bushfires*

INTRODUCTION

Each summer in Australia, bushfires burn many hectares of forest, causing deaths, injuries, and destroying property. Societies can manage such crisis and emergency situations in several ways: adopt urban and territory planning policies to reduce the risks (e.g. forbid construction in exposed areas); raise awareness and prepare the population in advance; or create efficient emergency management policies to deal with crises when they happen. Modelling and simulation offer tools to test the effects and complex interactions of these different strategies without waiting for an actual crisis to happen, without putting human lives at risk, with limited cost, and with a great degree of control on all conditions and the possibility of reproducing exactly the same situation as many times as needed.

When modelling human behaviour, mathematical, equation-based models are too limited (Parunak, Savit, & Riolo, 1998) whereas agent-based models offer many benefits (Bonabeau, 2002). They allow capturing emergent phenomena that characterise such complex systems; they provide an intuitive and realistic description of their behaviour; and they are flexible, offering different levels of abstraction by varying the complexity of agents. Agent-based modelling and simulation platforms over various architectures of different complexity for the agents: reflex or reactive agents are very simplistic, reacting to environmental stimuli without any long-term reasoning; finite-state machines require scripting all of the possible states of the agents and the corresponding transitions; cognitive agents offer a more flexible description of behaviours in terms of goals and plans.

In particular the BDI (Belief, Desire, Intention (Rao & George, 1991)) architecture is very sophisticated and realistic, grounded in the philosophy of human rationality (Bratman, 1987), and linked with emotions (Adam, Herzig, & Longin, 2009). Such realism of the human

behaviour model is important for the simulation to produce valid results (van Ruijven, 2011). For these reasons and as previously discussed in (Adam & Gaudou, 2016a), the BDI architecture is therefore more adapted for crisis situations that involve complex individual decision-making, influenced by emotions (sometimes causing irrational actions), and by the social context (effect of group, family, etc.). According to (Norling, 2004), BDI also provides an adapted level of abstraction to describe human behaviour in terms of folk psychology, which is the preferred level of description for humans. It therefore addresses the problem of the scarcity of (quantitative) behavioural data by allowing the use of qualitative data such as witness statements or expert reports.

Despite these advantages that make it very suitable for social simulation, BDI has had limited use in this field due to the lack of adapted tools to harness its complexity (Adam & Gaudou, 2016a). In previous work (Adam, Danet, Thangarajah, & Dugdale, 2016; Adam, Beck, & Dugdale, 2015; P. Taillandier, Bourgais, Caillou, Adam, & Gaudou, 2016) we have described how two new tools could be used to de-velop BDI agent-based models from interviews. We illustrated how to use these tools by turning an existing model of the behaviour of the Australian population in bushfires (with a finite-state machine architecture (Adam & Gaudou, 2016b, 2017)) into a BDI model. In this paper we now want to com-pare these two models, addressing the same problem with different architectures, using both objective and subjective criteria. We believe that such model comparison is important to further justify the interest of BDI models.

This paper is structured as follows. We first describe the context of our case study (Melbourne bushfires) and our initial model of population behaviour using a finite state machine (FSM) architecture (Adam & Gaudou 2017). We then discuss BDI agents as a more complex alternative, expose some of the obstacles preventing their use in social simulations,

propose some technological solutions and illustrate them by providing a BDI model of population behaviour. The next section is dedicated to comparing this BDI model with the initial FSM model: we discuss the literature about model comparison, derive our own list of comparison criteria, and present the results of comparing our models following these criteria. The last section discusses future prospects and concludes the paper.

INITIAL MODEL OF THE POPULATION IN BUSHFIRES

Context

We focus on the so-called Black Saturday, 7th February 2009, when particularly strong bushfires hit the state of Victoria in Australia, killing 173 people and destroying hectares of bush and many properties. The official recommendation to the population was to "prepare, stay and defend, or leave early". However, reports (Alan Rhodes, 2014) written after these fires showed that emergency management policies were designed based on an (ideal) expected behaviour that differed from the residents' actual behaviour on the day. It is therefore important to provide deciders with a simulator to raise their awareness about residents' actual decision making, and let them try different communication strategies. Currently, the available data is mostly in the form of witness statements (Exell, 2009) and police hearings. The Royal Commission (Teague, McLeod, & Pascoe, 2009) also gathered the following statistics about the victims:

• Preparation: 58% of the victims had made no preparation at all; many prepared to leave but expected a warning before going; 20% intended to stay and defend and were well prepared; 14% had made limited preparation.

• Awareness: the fire took 30% of those who died by surprise; 24% of the victims were unaware that they lived in a bush fire prone area; 38% had no basic knowledge about what to do.

• Causes of death: 14% died while escaping (4% by car and 10% by foot); 69% died while passively sheltering in a building; the others died while trying to defend.

• Vulnerability: 44% of the victims were considered vulnerable because of their age (under 12 or over 70), illness or disability; 32% died on properties whose defendability was questionable.
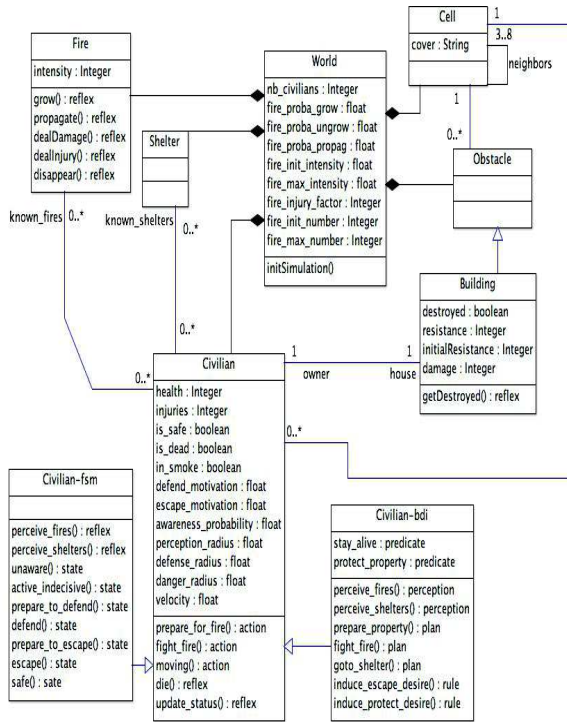
To facilitate comparison, we have implemented a general model in the GAMA simulation platform (Grignard. et al., 2013) that allows choosing between two models of the civilians' behaviour: the first one is based on a FSM (finite state machine) and the second one on BDI (belief desire intention). Figure 1 presents the class diagram of the global model. For the sake of comparison, we defined a generic Civilian class that regroups all the common properties, actions and reflexes of both behaviour models.

Below we describe the model of the environment, buildings and fires; then the generic model of civilians and finally the FSM architecture for civilians. The following section describes the BDI architecture and the tools that were used in its design and implementation. We try to give enough details about each model to allow the reader to understand the comparison provided in the next section.

MODEL OF THE ENVIRONMENT AND THE FIRE

For the sake of simplicity, the environment is represented as a grid containing the different types of agents (houses, shelters, fires, and residents). This simplistic environment is not realistic, but was proven sufficient to simulate the residents' decision-making behaviours in reaction to fires (Adam & Gaudou, 2017).

Fire. Very complex and detailed models of the spread of fire already exist (Du, Chong, & Tolhurst, 2013; Miller, Hilton, Sullivan, &

Figure 1: UML class diagram of the model

Fire
- intensity : Integer
- grow() : reflex
- propagate() : reflex
- dealDamage() : reflex
- dealInjury() : reflex
- disappear() : reflex

Cell
- cover : String

World
- nb_civilians : Integer
- fire_proba_grow : float
- fire_proba_ungrow : float
- fire_proba_propag : float
- fire_init_intensity : float
- fire_max_intensity : float
- fire_injury_factor : Integer
- fire_init_number : Integer
- fire_max_number : Integer
- initSimulation()

Shelter

Obstacle

Building
- destroyed : boolean
- resistance : Integer
- initialResistance : Integer
- damage : Integer
- getDestroyed() : reflex

Civilian
- health : Integer
- injuries : Integer
- is_safe : boolean
- is_dead : boolean
- in_smoke : boolean
- defend_motivation : float
- escape_motivation : float
- awareness_probability : float
- perception_radius : float
- defense_radius : float
- danger_radius : float
- velocity : float
- prepare_for_fire() : action
- fight_fire() : action
- moving() : action
- die() : reflex
- update_status() : reflex

Civilian-fsm
- perceive_fires() : reflex
- perceive_shelters() : reflex
- unaware() : state
- active_indecisive() : state
- prepare_to_defend() : state
- defend() : state
- prepare_to_escape() : state
- escape() : state
- safe() : sate

Civilian-bdi
- stay_alive : predicate
- protect_property : predicate
- perceive_fires() : perception
- perceive_shelters() : perception
- prepare_property() : plan
- fight_fire() : plan
- goto_shelter() : plan
- induce_escape_desire() : rule
- induce_protect_desire() : rule

known_fires 0..* known_shelters 0..*

Prakash, 2015), but realistic fire behaviour is not the focus here. With the goal of not adding unnecessary complexity, we have designed a very simplistic fire model that is sufficient to trigger and visualise the reactions of the population in which we are interested. The fire is composed of fire agents (each with a location and an intensity representing its radius of action), having a reflex architecture, i.e. the following reflexes are triggered at each step of the simulation:

- **Grow (increase or decrease intensity)**: probabilities are parameters.
- **Propagate** to a non-burning neighbour cell, creating a new fire agent. The probability of propagating, and starting intensity of new fires, are parameters.
- **Deal damage** to buildings in its radius of action (based on its intensity). The amount of damage is picked randomly between 0 and a maximum value, a function of intensity and a "damage factor" parameter.
- **Deal injuries** to residents in its radius of action, also a random number between 0 and the maximum value based on its intensity and on an "injury factor" parameter. If the person is in their house the injury is moderated by its resistance weighted by a "protection factor" parameter.
- **Disappear** when its intensity is null.

The different parameters involved allow the user of the simulation to make the fire more or less dangerous (growing and propagating more quickly, dealing more damage and injuries, etc.), in order to observe the desired behaviours. Actions are also available to start new fires or stop all fires (and thus the simulation).

Houses. The environment initially contains a number (parameter) of houses each inhabited by exactly 1 resident (in future work we plan to consider families and their relationships). Each house is an agent with the following attributes:

- **Owner**: the resident of that house
- **Resistance**: random initial value between 100 and 200 to simulate house resistance. This value is increased by the resident preparing the house for a fire, or decreased by fire damage. Resistance also offers some protection from fire injuries to its resident.
- **Damage**: the damage received from fire

The houses collapse from fire damage when their resistance drops to 0. They then cease to offer protection and the resident's motivation to defend them also disappears. Houses stay in the environment as ruins for final visualisation.

Shelters. Shelters are safe places offering total protection from the fires (no injuries can be received while in a shelter). Once a resident has reached a shelter, it stays inside until the end of the simulation.

GENERIC CIVILIAN MODEL

Civilian agents are heterogeneous agents, each having their own values of attributes:

- **defend motivation**: random initial value between 0.0 and 1.0 to simulate the propensity to defend their home.
- **escape motivation**: random initial value between 0.0 and 1.0 to simulate the propensity to escape in case of danger.
- **awareness probability**: random initial value between 0.0 and 1.0 to simulate the attention towards dangers.
- **perception radius**: random initial value between 0.0 and 20.0 to simulate the maximal distance of perception of fires.
- **defence radius**: random initial value between 0.0 and 10.0 to simulate the area of defence.
- **danger radius**: random initial value between 0.0 and 10.0 to simulate the area of danger.
- **velocity**: random initial value between 0.2 and 1.0 to simulate the moving speed.
- **location** on the grid.
- **house id**: each agent is initially in a house.
- **health**: the health of the civilian. It is increased by preparing for fire and decreased when injuries are received. A health of 0 means death.
- **injuries**: the injuries received from fire. It decreases the health.
- **is safe**: defines if the civilian is in a shelter.
- **is dead**: defines if the civilian is dead.
- **in smoke**: defines if the civilian is in smoke (slow movement).

In addition, each civilian agent has the following actions:

- **Prepare for fire**: consists in increasing the resistance of the house (watering, removing vegetation, etc.) and health (wearing appropriate clothing, etc.).
- **Fight fire**: consists in decreasing the intensity of nearby fires by a value.

- **Moving**: consists in heading towards the closest shelter (amongst the known shelters); the agent might get injured if travelling too close to the fire.

Finally, they have reflexes:

- **Update status**: is activated at every step; updates the agent speed according to the health and the presence of smoke; it also updates the agent's motivation to defend the property according to the agent context.
- **Die**: is activated when the health of the agent reaches 0; it kills the agent (the attribute *isdead* is set to true).

FINITE-STATE MACHINE MODEL OF BEHAVIOUR

Civilian-FSM agents have a finite-state-machine architecture (Adam & Gaudou, 2017) with the following states and transitions (c.f. Figure 2), inspired by the states observed in the interviews:

- `Unaware`: initial state where the agent is (rightly or wrongly) unaware of any danger, and does nothing; agents can become aware by spotting fires in their perception radius (see flames, smell smoke, etc.), with a probability based on their objective abilities; they up-date their value of subjective danger based on their perceptions and motivations;
- `Indecisive`: the agent is aware of some fires but has not yet made a decision about how to react; agents stay indecisive for a varying amount of time, until they have enough motivation to either fight or escape; initial motivations are individual and then vary based on the evaluation of the situation (subjective danger);
- `Preparing to escape`: the agent has decided to leave and starts preparing. The agent does this until ready, or surprised by the fire before being ready (transition to `Escaping`), or blocked by the fire and forced to stay (transition to `Preparing to defend`);

- `Escaping`: the agent is evacuating towards the closest shelter (call the *moving* action); travel efficiency de-pends on objective abilities; injuries can be received from fires on the way. Unless the agent dies during travel, its next state will be Safe when reaching the shelter;
- `Preparing to defend`: the agent has decided to de-fend, or was forced to stay because the fire blocks escape; it prepares the house and itself (call the *prepareforfire* action) until the fire is close enough, which triggers the transition to Defending;
- `Defending`: the agent is actively fighting the fire around its house (call the *fightfire* action); if that fire is extinguished, the agent transitions back to Preparing to defend until another fire comes; if motivations change (e.g. subjective danger increases when actually seeing the fire, or subjective abilities decrease after failing to fight) and evacuation becomes more urgent, the agent transitions to `Escaping`;
- `Safe`: the agent is (and will stay) in a shelter, and can-not be injured anymore;
- `Dead`: final state of all agents whose health dropped to 0 as a result of injuries received from the fire (from any state other than Safe);
- `Survivor`: final state of all agents who did not die during the fires (e.g. successful

defenders, lucky passive, and all sheltered agents).

Figure 2: states of the FSM for civilian agent
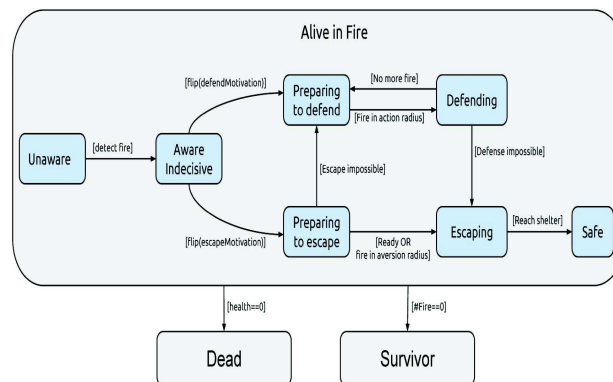
This simple model is sufficient to highlight the role of subjective, irrational determinants of the decisions and behaviours of each resident, and therefore captures the discrepancies shown by the data. Indeed the objective value of danger influences injuries and damage, and the objective value of capability influences the success of actions. But these objective values are inaccessible to the agents, whose decisions are based on their subjective values of danger and abilities, and on their motivations.

users. To further prove our point, we implemented a BDI version of the civilian agents in this bushfire simulation, in order to then compare it with the FSM version.

BDI MODEL OF THE POPULATION BEHAVIOUR IN BUSHFIRES

As discussed above, BDI agents offer many interesting advantages for social simulation, in particular when applied to raising awareness. However, they are little used yet, due to their higher computational complexity and the lack of dedicated supporting tools and methodologies. In this section, we describe a methodology (TDF) and tool (GAMA-BDI) to support the integration of BDI agents in simulations, and illustrate them by designing a new version of the civilian model

FSM model discussion

This model was implemented in GAMA, and our experiments showed good results in highlighting the role of subjective factors in the "irrational" behaviour of the population (Adam & Gaudou, 2016b, 2017). However, we have previously argued for the use of BDI agents in social simulations (Adam & Gaudou, 2016a); their main advantage, and even more so when aiming at raising awareness, is that their behaviour is encoded in terms of concepts from folk psychology (Norling, 2004) and are therefore more easily understood by human

above, with a BDI architecture instead of the FSM architecture.

Methodology: Tactics Development Framework (TDF)

When designing a conceptual model for computational simulation, UML is the most widely used tool, mostly because of its generality and ease of use, allowing one to describe entities in terms of attributes and actions; but it is not well suited for modelling human behaviour which is what is often required in disaster management and evacuation simulations such as our case study. On the other hand, agent-based software development methodologies that develop systems using mental attitudes of goals, events, plans, beliefs, capabilities etc. are well suited for these systems. This is particularly relevant when transcribing behaviours described by human witnesses, as is the case here, since humans naturally tend to explain their behaviour in terms of mental attitudes. For instance, consider this extract: "*I looked out the window and saw some hazy smoke to the north-west. Gary said that he thought it was just dust but we went outside and straight away we noticed that we could smell smoke. It was about 12.45pm when we smelt the smoke and as soon as that happened, Gary agreed to go and get the fire pump*". We can make the mental attitudes involved more explicit: Gary (wrongly) **believed** for a while that the smoke was just dust, but **planned** to get more information; after going outside they perceived smoke and realised that it was coming from a fire (**belief update**). As a result, Gary adopted the **goal** to get ready for the fire, and started on their **plan** whose first action was to get the fire pump.

Whilst there are several agent-oriented software engineering methodologies such as Prometheus, Tropos, O-MaSE, GAIA and others (DeLoach, Padgham, Perini, Susi, & Thangarajah, 2009), we introduce a more recent methodology that has been specifically built for eliciting and encoding tactical/strategic behaviours in dynamic domains: TDF (Tactics Development Framework) (Evertsz, Thangarajah, Yadav, & Ly, 2015). TDF is based on the Prometheus methodology (Padgham & Winiko, 2002), a mature and popular agent-oriented software engineering methodology. A pilot study has shown that TDF significantly improves comprehension of behaviour models, compared to UML (Evertsz et al., 2015). Although TDF was initially designed to capture and model military behaviour, we have shown in previous work (Adam, Beck, & Dugdale, 2015) that this framework can be adapted to model civilians' descriptions of their behaviour in crisis situations. The TDF methodology proceeds in following 3 phases: *System specification*: Identification of system-level artefacts, namely goals, scenarios, percepts, actions, data, actors and roles; *Architectural design*: Specification of the internals of the system, namely the agents that play the different roles, the interactions between the agents (via protocols) if any, and messages between agents; and *Detailed design*: Definition of the internals of the agents, namely capabilities, plan diagrams and internal messages/sub-goals.

Tool: GAMA-BDI plugin

GAMA (Grignard. et al., 2013; GAMA, n.d.; P. Taillandier, Grignard, Gaudou, & Drogoul, 2014) is an open source platform for agent-based modelling and simulation of complex spatialised systems. It provides built-in functions for using Geographical Information Systems data such as OpenStreetMap for fast and precise mapping of the environment. Simulations built with GAMA are scalable, since the platform can deal with several thousand agents, depending on the level of complexity of their architecture. Furthermore, GAMA provides a very simple and high level programming language called GAML, which allows even non-programmers to simply build and maintain their own models. As a result, it is widely used by designers from many different fields: urban growth (P. Taillandier, Banos, et

al., 2016), geo-historical reproduction of past crises (Gasmi et al., 2014), socio-environmental models (Gaudou et al., 2014), impact of floods on land planning (F. Taillandier, Adam, Delay, Plattard, & Toumi, 2016), etc. Finally, it is supported by an active development team that is progressively improving the software. GAMA was recently extended with a BDI plugin (Caillou, Gaudou, Grignard, Truong, & Taillandier, 2015; P. Taillandier, Bourgais, et al., 2016) to allow designers to easily create BDI agent models in the GAML language. They can specify logical predicates, initialise their agents with beliefs and desires, describe the effect of new percepts on the agent's beliefs, and provide them with a plan library. The BDI engine then lets the agents perceive their environment, update their beliefs and desires, select an intention based on the relative priorities of their goals, and choose and execute an adapted plan to reach that goal. We have previously shown that this BDI plugin connects well with the TDF methodology as it uses matching concepts, and have successfully used it to implement the conceptual BDI model designed with TDF (Adam et al., 2016). The next paragraph describes the resulting model in enough detail for the reader to understand the comparison of the 2 models.

The BDI model

In GAML (the programming language of the GAMA platform), the designer first needs to describe the different logical predicates that will be manipulated by the agent. This is basically the ontology of the domain. Agents can be endowed with an initial knowledge base with different beliefs and de-sires. These predicates can be associated with a priority. We define two predicates for the civilian-bdi agents:

• **stay alive**: desire to flee fires to stay alive. Its priority is based on the agent's danger aversion (escape motivation).

• **protect property**: desire to protect property. Its priority is based on the agent's danger determination (defend motivation).

Agents can then be given **perceptions**, which explain how they interpret the stimuli coming in. We define two perceptions for the civilian-bdi agents:

• **perceive fires**: perceive new fires (that are not yet in their list of known fires) and add them to their list.

• **perceive shelters**: perceive new shelters (that are not yet in their list of known shelters) and add them to their list.

Agents are also endowed with a number of rules for updating their mental attitudes. Each rule allows inferring new beliefs or desires according to a specific condition (that can be a specific belief or desire). For the civilian-bdi agent, we define two rules:

• **infer escape desire**: infer the *stayalive* desire according to the *awarenessprobability* if the civilian knows that there is at least one fire (the agent refers to its known fires list).

• **infer protect desire**: infer the protect property desire according to the *awarenessprobability* if the civilian knows that there is a least one fire (known fires list).

The agents are endowed with a library of **plans** to achieve their goals. Each GAML plan is defined with several optional features: the goal it achieves (keyword: intention); a context condition (keyword: when) describing when this plan is applicable; and a success condition (keyword: finished_when). We define three plans for the civilian-bdi agents:

• **prepare property**: this realises the goal *protectproperty*, by executing the *prepareforfire* action it is applicable when the fire is not too close (i.e. the distance is higher than defence radius).

• **fight fire**: this realises the goal *protectproperty*, by executing the *fightfire* action; it is applicable when the fire is close

enough (distance is lower than or equal to defence radius).

- **go to shelter**: this realises the goal *stayalive*, by executing the *moving* action.

BDI model discussion

This methodology and tool are generic and not limited to this particular case study. TDF was also used for military training simulations. GAMA has been used in such various domains such as agronomy, epidemiology, civil engineering, and geosciences. We are currently using GAMA-BDI for simulating social attachment in earthquake evacuations (Bangate, Dugdale, Adam, & Beck, 2017), as well as cognitive biases in bushfires (Arnaud, Adam, & Dugdale, 2017). We believe that the combination of these tools can help more designers to use BDI agents in their models and simulations.

COMPARING THE SIMULATIONS

In this section, we compare our two simulations (FSM and BDI) of the population behaviour in bushfires. We start by discussing the literature about model comparison, and then propose our own list of comparison criteria, finally using them to perform this comparison.

Model comparison: state of the art

In crisis management as well as other application fields of social simulation, many ad hoc simulators are created, with different agent architectures, different underlying models, and different tools; making comparisons hard. However model comparison is essential to determine which model is most appropriate for which application.

Model comparison has been the topic of many research works. Some of these works evaluate and compare the actual modelling platforms (e.g. Netlogo, Repast, Mason, etc.) (Railsback, Lytinen, & Jackson, 2006; Laclavík et al., 2011; Daudé & Langlois, 2007; Bajracharya & Duboz,

2013). Others focus on comparing the performance of the resulting systems. For instance (Bartish & Thevathayan, 2002) have compared the use of BDI and FSM architectures in games: evaluating complexity in terms of the number of behaviours, finding that it was linear for BDI agents and quadratic for FSM. On the other hand, run-time performance, using a small number of agents, degraded more quickly for BDI agents than for FSM agents in bigger systems.

Comparisons have also been made by focusing on the underlying models. In this context, most of works base their comparison on the simulation results and use two types of metrics: a fitness function - often computed by estimating the error between the observed data and simulated ones - and a computation time. For example, (Truong et al., 2016) compare three land-use change models based on three different architectures for the farmer agents: probabilistic, multi-criteria and BDI. The comparisons between the three models are achieved by using the fuzzy kappa coefficient (Hagen, 2003) that allows evaluating the local similarity between the observed data and the simulated ones, and the percent absolute deviation.

Another technique that has been used by the agent-based community is "Docking", sometimes also known as "replication", or Model to Model comparison (Carley, 2002). Docking attempts to align multiple models in order to investigate if they yield similar results. The compared models may all use an agent-based approach but be implemented in different platforms or languages for example (Axtell, Axelrod, Epstien, & Cohen, 1996; Arifin, Davis, & Zhou, 2010; Xiang, Kennedy, & Madey, 2005), or they may use completely different approaches, by specifying their models using, for example, symbolic mathematical expressions or agents (North & Macal, 2002; Rank, 2010). The benefits of docking are well documented (Arifin et al., 2010; Rouchier & Tanimura, 2016) and include ensuring the validity of simulation

results, increasing the quality of the model, and assessing if one model subsumes another.

Some works go further and propose measures to compare the complexity of models. Thus, (Mandes & Winker, 2015), propose measures divided in three groups:

1. Difficulty of description:
   - number of parameters
   - number of lines of code
   - maximum cyclomatic complexity[1]
   - average cyclomatic complexity
   - maximum nested block depth level
   - average nested block depth level
2. Difficulty of creation:
   - computational time
   - memory usage
3. Difficulty of organization:
   - approximate entropy
   - fractal dimension

Our comparison metrics

This work aims at comparing two agent architectures through the comparison of two models. Contrary to (Mandes & Winker, 2015), the models will be very complex (integration of cognitive agents) and share many elements. As a result, some of the previously proposed metrics are not very well adapted. In addition, these authors do not propose any specific metric concerning the ease of appropriation of the models by users, which is a very important criterion for the reusability of models. We therefore propose the following metrics:

1. Difficulty of description:
   - number of characters in the code: we choose to use the number of characters rather than the number of lines as the length of lines can be very heterogeneous
2. Difficulty of creation:
   - computational time
   - memory usage
3. Difficulty of appropriation:
   - understandability
   - explainability
   - extensibility
4. Model credibility:
   - error between observed data and simulated data

Note that as stated by (Müller et al., 2014) the common use of a modelling platform (such as Netlogo, GAMA or Repast) can facilitate model comparison. This is why in this work we have used the GAMA platform to implement both models.

Comparison results

Difficulty of description. Table 1 shows the comparison of code length. We can observe that the code is far more compact with the BDI model (more than 24% more compact). This is due to using specific features, like perceptions, that simplify writing the model, and to the fact that the FSM architecture requires explicitly specifying all the possible existing states and their transitions. This second explanation also shows a limitation of the FSM architecture in terms of modularity: enriching the model requires adding new states and specifying new transitions, which becomes increasingly complicated as the number of states increases, whereas adding new desires and plans in the BDI architecture is straightforward (see Extensibility, below).

Table 1: code comparison (difficulty of description)

Measure	FSM Model	BDI Model
#characters in the code	1769	1310

---

[1] Cyclomatic complexity is a metric used to indicate the complexity of a program; it is a quantitative measure of the number of linearly independent paths in its source code.

Difficulty of creation. Table 2 shows the comparison concerning the use of computer resources. For the time computation, we use the same scenario with 10 replications and the same series of seeds. The memory usage was estimated with 5000 civilian agents (and a grid of 100x100 cells) after 2 simulation steps (the time for the agents to detect fires). The memory usage concerns the memory used by all aspects of the simulation, not only the civilian agents, but also all of the other agents and the GAMA platform interface.

The computation times are relatively close. A deeper analysis with a profiling tool shows that the computation time is mostly due to the civilian agent actions and not to the computation linked to the architecture.

In the same way, the results for the memory usage are very close. Note that as our goal was to define a BDI model as close as possible as the FSM one, we did not use belief predicates that could have used more memory than basic variables (for example, for the known fires and known shelters).

Table 2: comparison of computer resource usage (difficulty of creation)

Measure	FSM Model	BDI Model
Computation time	35 s	45 s
Memory usage	103 Mo	109 Mo

Measure	Real data	FSM Model	BDI Model
Died escaping	14%	18%	13 %
Died passively	69%	72%	68 %
Died defending	17%	10%	19 %

Difficulty of appropriation. We designed a questionnaire for test subjects to compare the two models on two aspects: understandability of code (can they understand how it works and modify it), and explainability of behaviour (can they understand what the agents do and why). We asked a limited number of testers (a dozen students from a computer science laboratory, with various levels of previous exposure to FSM and BDI models) to answer this questionnaire. These are more subjective criteria so this survey only provides qualitative feedback on the models. The testers found the BDI model more understandable and had less difficulty modifying it than the FSM model. It was also easier to explain behaviour in terms of what the agents desired instead of which state they were currently in. In future work we will conduct the study on a larger scale.

Model credibility. In order to evaluate the model credibility, we used some of the data provided in the reports concerning bushfires (see Context section above), in particular the ones related to the causes of death:
- 14% died while escaping
- 69% died while passively sheltering in a building;
- 17% died while defending their property.

As the models are stochastic, we carried out 10 replications for each model with the same series of seeds.Table 3 shows that the two models produce correct results even if they could be improved by enriching them to better take into account the heterogeneity of the behaviours. The difference between the results of the two models is not really significant. This was predictable as the Civilian-bdi and Civilian-fsm agents share the same attributes and actions.

Table 3. Comparison of the model outputs (Model credibility)

Table 3: comparison of model outputs (model credibility)

Extensibility

Experiments. To assess the extensibility of both models, we performed an experiment where we asked seven researchers and students from various profiles (sociology, agronomy, computer science, civil engineering) to undertake an exercise. We gave them the BDI and FSM models, and asked them to extend both models with a new "warn neighbours" behaviour. This behaviour was specified as precisely as possible: the agents should have an attribute determining their motivation to warn neighbours; a motivated agent chooses an neighbour who is unaware of the fire, communicates to the neighbour their list of known fires, and makes them aware; this action should take exactly one time step. The subjects were asked to implement this behaviour in both architectures, and then to provide feedback about how easy it was to perform this extension. We also reviewed and marked their code to measure how well they succeeded in extending each model.

Results. The subjects' performances on both models were correlated together (i.e. those performing better on FSM also performed better on BDI, and vice versa). Also, all subjects obtained a better mark for the BDI model than for the FSM model, which seems to suggest that the BDI model is easier to extend. On the contrary, from the participants' feedback, they felt that FSM concepts were easier to understand. This was particularly true for those participants who did not have a computer science background or those not familiar with BDI. But this impression of easiness was not correlated with the quality of their code (despite finding it easy, they made errors and their models were not correct).

Many programming errors were independent of the architecture (e.g. forgetting to consider only unaware neighbours, or to remove already warned neighbours from the unaware list). Typical errors in the BDI model were syntax errors or plan-managing errors (how to properly finish a plan, remove the satisfied intention) that would be avoided if the subjects had more knowledge of BDI. Typical errors in the FSM model were related to forgetting transitions, which is inherent to the FSM architecture and increases as the model complexity increases (more states mean more transitions to add for each additional state).

Comparison conclusion. The concepts of states and transitions in the FSM model appeared to be more intuitive to the subjects who were unfamiliar with BDI. However, despite this impression of simplicity they made more errors (in particular missing transitions). Conversely, BDI is initially harder to understand due to complex concepts such as plans and intentions, but easier to use (since there is no need to exhaustively list many transitions). BDI was also found to be more flexible and less sequential than FSM.

Discussion. Extending the FSM model required specifying a new state, the action to be performed in that state, and more importantly all of the transitions to and from that new state, linking it to the previously existing states. Extending the BDI model required writing a new plan, specifying the desire to be satisfied by that plan, and the perception triggering that desire; this does not change if the complexity of the agent increases. As a result, the difference in extensibility of both architectures is not very significant for an agent of limited complexity such as the one presented here, but would grow exponentially as the agent's complexity increases. Indeed as the complexity grows, an increasing number of states would need to be added to the FSM model, also requiring ever more transitions to and from these states. Conversely, the cost of extending the BDI model would stay rather stable.

CONCLUSION AND FUTURE WORK

In this paper, we discussed the need to compare agent-based models for social simulations using both objective and subjective criteria, in order to help designers determine which agent architecture is the most adapted for their needs. Concretely, we focused on modelling the

behaviour of the Australian population in bushfires, with two very different agent-state machines, and belief-desire-intention agents. We then compared these two models on a number of objective criteria, and also asked subjects to subjectively compare their understandability and explainability, and to try to extend both models. Our results show that BDI models, despite being initially more di cult to understand, offer a gain in modularity, flexibility, understandability and extensibility. This is essential in crisis management where the goal of such models is precisely to explain behaviour, raise awareness, and explore new strategies. Extensibility is also key in facilitating the reusability of existing models.

The originality of our work is that we have developed and compared two models in the exact same context, with two different architectures for the same agents. Having already argued previously for the use of the BDI architecture in social simulation (Adam & Gaudou, 2016a), this work goes further by describing useful tools for creating BDI agents (TDF and GAMA). It also concretely compares BDI with another behaviour in social simulations (and even more so in crisis situations), in order to deduce valid results. In future work, we will therefore improve the BDI version of the human behaviour model by integrating various emotions and psychological factors, starting again from the interviews where many survivors describe how they felt before, during, and after the bushfires. The random fire model has also become a

REFERENCES

Adam, C., Beck, E., & Dugdale, J. (2015). Modelling the tactical behaviour of the Australian population in a bushfire. *In Proceedings of ISCRAM-Med 2015*. LNBIP 233 Springer.

Adam, C., Danet, G., Thangarajah, J., & Dugdale, J. (2016). BDI modelling and simulation of human behaviours in bushfires. *In*

architectures: finite

frequently used and seemingly simpler architecture, finite-state machines. Our experiments show that although BDI might initially seem more complex to handle, it is then more compact to implement, and makes it easier to explain behaviours and to extend complex models. This paper therefore addresses two frequently raised problems of BDI agents: the lack of tools for using them, and their complexity limiting their accessibility to non-specialists.

Finally we would like to mention that the BDI architecture can be extended with emotions (Adam et al., 2009), and this extension has now been implemented in the GAMA simulation platform (Bourgais, Taillandier, & Vercouter, 2016). Other psychological factors can also be taken into account such as social attachment (Bangate et al., 2017) and cognitive biases (Arnaud et al., 2017). Such extensions will provide agents with even more realism, which is crucial when modelling human

limiting factor. We therefore plan to increase realism by using a more credible fire simulator (e.g. (Miller et al., 2015; Du et al., 2013)). Indeed, our long-term goal is to further develop this simulator and turn it into a serious game, which requires all aspects of the simulation to be as close as possible to reality.

Proceedings of ISCRAM-Med 2016. LNBIP 265 Springer.

Adam, C., & Gaudou, B. (2016a). BDI agents in social simulations: a survey. *Knowledge Engineering Review*, 31(3):207-238.

Adam, C., & Gaudou, B. (2016b, September). Modelling human behaviours in disasters from interviews: application to Melbourne bushfires. *In Proceedings of Social simulation Conference (SSC)*. Rome, Italy.

Adam, C., & Gaudou, B. (2017). Modelling human behaviours in disasters from interviews: application to Melbourne bushfires. *Journal of Artificial Societies and Social Simulation (JASSS)* 20(3).

Adam, C., Taillandier, P. & Dugdale, J. (2017). Comparing agent architectures in social simulation: BDI agents vs finite-state machines. *In Proc. of HICSS-50, Hawaii, USA.*

Adam, C., Herzig, A., & Longin, D. (2009). A logical formalization of the OCC theory of emotions. Synthese, 168.

Alan Rhodes. (2014). Why don't they do what we think they should? In Afac. Arifin, S. N., Davis, G., & Zhou, Y. (2010). Verification and validation by docking: a case study of agent-based models of Anopheles Gambiae. In *Summer Computer Simulation Conference* (pp. 236–243).

Arnaud, M., Adam, C., & Dugdale, J. (2017, May). The role of cognitive biases in reactions to bushfires. *In the Proceedings of ISCRAM 2017.* Albi.

Axtell, R., Axelrod, R., Epstein, J. M., & Cohen, M. D. (1996). Aligning simulation models: A case study and results. *Computational and Mathematical Organization Theory*, 1, 123–141.

Bajracharya, K., & Duboz, R. (2013). Comparison of three agent-based platforms on the basis of a simple epidemiological model. *In Proceedings of the symposium on theory of modeling & simulation devs integrative m&s symposium* (p. 7).

Bangate, J., Dugdale, J., Adam, C., & Beck, E. (2017, May). A review on the influence of social attachment on human mobility during crises. *In the Proceedings of ISCRAM 2017.* Albi.

Bartish, A., & Thevathayan, C. (2002). BDI agents for game development. *In Proceedings of Automomous Agents and Multi-Agent Systems conference.*

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

Bourgais, M., Taillandier, P., & Vercouter, L. (2016). An Agent Architecture Coupling Cognition and Emotions for Simulation of Complex Systems. In *Proc. of Social Simulation Conference*.

Bratman, M. E. (1987). *Intention, plans, and practical reason*. CSLI Publications.

Caillou, P., Gaudou, B., Grignard, A., Truong, C. Q., & Taillandier, P. (2015, September). A Simple-to-use BDI architecture for Agent-based Modeling and Simulation. *In The Eleventh Conference of the European Social Simulation Association (ESSA 2015)*.

Carley, K. M. (2002). Computational organizational science and organizational engineering. *Simulation Modelling Practice and Theory* (5-7), 253–269.

Daudé, E., & Langlois, P. (2007). Comparison of three implementations of Schelling's spatial segregation model. *Agent-Based Modelling and Simulation in the Social and Human Sciences*.

DeLoach, S. A., Padgham, L., Perini, A., Susi, A., & Thangarajah, J. (2009). Using three AOSE toolkits to develop a sample design. *International Journal of Agent-Oriented Software Engineering*, 3(4), 416–476.

Du, T. J., Chong, D. M., & Tolhurst, K. G. (2013). Quanti-fying spatio-temporal differences between fire shapes: Estimating fire travel paths for the improvement of dynamic spread models. *Environmental Modelling and Software*, 46, 33–43.

Evertsz, R., Thangarajah, J., Yadav, N., & Ly, T. (2015). A framework for modelling tactical decision-making in autonomous systems. *Journal of Systems and Software*, 110, 222–238.

Exell, S. (2009). Witness statements. http://vol4.royalcommission.vic.gov.au/index03a1.html?pid=111.

GAMA. (n.d.). GIS & agent-based modelling architecture. http://gama-platform.org.

Gasmi, N., Grignard, A., Drogoul, A., Gaudou, B., Taillandier, P., Tessier, O., & Vo, D. A. (2014). Reproducing and exploring past events using agent-based geo-historical models (regular paper). In E. Norling & F. Grimaldo (Eds.), *International Workshop on Multi-Agent-Based Simulation (MABS)*, Paris, France (Vol. 9002, pp. 151–163). http://www.springerlink.com/: Springer-Verlag.

Gaudou, B., Sibertin-Blanc, C., Thalrond, O., Amblard, F., Auda, Y., Arcangeli, J.-P., Mazzega, P. (2014). The MAELIA multi-agent platform for integrated assessment of low-water management issues (regular pa-per). In S. J. Alam & H. V. D. Parunak (Eds.), International Workshop on Multi-Agent-Based Simulation (MABS), Saint-Paul, MN, USA (Vol. 8235, pp. 85– 110). http://www.springerlink.com: Springer.

Grignard., A., Taillandier, P., Gaudou, B., Huynh, N., Vo, D.-A., & Drogoul, A. (2013). Gama v. 1.6: Advancing the art of complex agent-based modeling and simulation. In PRIMA.

Hagen, A. (2003). Fuzzy set approach to assessing similarity of categorical maps. International Journal of Geographical Information Science, 17(3), 235–249.

Laclavík, M., Dlugolinsky, Š., Šeleng, M., Kvassay, M., Schneider, B., Bracker, H., Hluchy, L. (2011). Agent-based simulation platform

evaluation in the context of human behavior modeling. In Advanced agent technology (pp. 396–410). Springer.

Mandes, A., & Winker, P. (2015). Complexity and model comparison in agent based modeling of financial markets. Journal of Economic Interaction and Coordination, 1–38.

Miller, C., Hilton, J., Sullivan, A., & Prakash, M. (2015). Spark: A bushfire spread prediction tool. Environmental Software Systems. Infrastructures, Services and Applications, 448, 262–271.

Müller, B., Balbi, S., Buchmann, C. M., De Sousa, L., Dressler, G., Groeneveld, J. (2014). Standardised and transparent model descriptions for agent-based models: current status and prospects. *Environmental Modelling & Software,* 55, 156–163.

Norling, E. (2004). Folk psychology for human modeling: extending the BDI paradigm. In AAMAS. New York.

North, M., & Macal, C. (2002). The beer dock: Three and a half implementations of the beer distribution game. In Swarmfest. Swarm Development Group.

Padgham, L., & Winiko , M. (2002). Prometheus: A methodology for developing intelligent agents. In Agent-oriented software engineering iii (pp. 174–185). Springer.

Parunak, H., Savit, R., & Riolo, R. (1998). Agent-based modeling vs equation-based modeling: A case study and users' guide. In Multi-agent systems and agent-based simulation (pp. 10–25). Springer.

Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-based simulation platforms: Review and development recommendations. Simulation, 82(9), 609– 623.

Rank, S. (2010). Docking agent-based simulation of collective emotion to equation-based models and interactive agents. In Spring simulation multiconference (p. 6). Society for Computer Simulation International.

Rao, A., & George , M. (1991). Modeling rational agents within a BDI-architecture. In 2nd international conference on principles of knowledge representation and reasoning (pp. 473–484).

Rouchier, J., & Tanimura, E. (2016). Learning with communication barriers due to overconfidence. what a "model-to-model analysis" can add to the understanding of a problem. Journal of Artificial Societies and Social Simulation, 19, 7. Retrieved from http://jasss.soc.surrey.ac.uk/19/2/7.html

Taillandier, F., Adam, C., Delay, E., Plattard, O., & Toumi, (2016). SPRITE: participatory simulation for raising awareness about coastal flood risk on the Oleron island. ISCRAM-med 2016. Lecture Notes in Business Information Processing, vol 265. Springer.

Taillandier, P., Banos, A., Drogoul, A., Gaudou, B., Marilleau, N., & Truong, C. Q. (2016). Simulating Urban Growth with Raster and Vector models: A case study for the city of Can Tho, Vietnam. *In N. Osman & C. Sierra (Eds.), AAMAS 2016 Workshops,* Visionary Papers, Singapore (Vol. 10003, pp. 154–171). Springer International Publishing.

Taillandier, P., Bourgais, M., Caillou, P., Adam, C., & Gau-dou, B. (2016, May). A situated BDI agent architecture for the Gama modelling and simulation platform. *In International workshop on multi-agent-based simulation (MABS workshop @ AAMAS).* Singapore, Malaysia: ACM.

Taillandier, P., Grignard, A., Gaudou, B. and Drogoul, A., 2014. Des données géographiques à la simulation à base d'agents: application de la plateforme GAMA. *Cybergeo: European Journal of Geography.*

Teague, B., McLeod, R., & Pascoe, S. (2009). Final Report, Volume I, Part 2: *The people who died.* Chapter 21: lessons learnt. (Tech. Rep.). 2009 Victorian Bushfires Royal Commission. Retrieved from http://goo.gl/ 0xcUFGTDF.

Tactics Development Framework. http:// agentprojects.com/tdf/.

Truong, Q. C., Taillandier, P., Gaudou, B., Vo, M. Q., Nguyen, T. H., & Drogoul, A. (2016). Exploring agent architectures for farmer behavior in land-use change. A case study in coastal area of the Vietnamese Mekong delta. In B. Gaudou & J. Sichman (Eds.), *Multi-agent based simulation xvi: International workshop, MABS 2015,* (pp. 146–158). Cham: Springer International Publishing.

van Ruijven, T. (2011). Serious games as experiments for emergency management research: A review. *In 8th international ISCRAM conference.* Lisbon, Portugal.

Xiang, X., Kennedy, R., & Madey, G. (2005). Verification and validation of agent-based scientific simulation models. *In Agent-directed simulation symposium (ADS).*