Automated dilemmas generation in simulations

Azzeddine Benabbou, Domitile Lourdeaux, Dominique Lenne

Automated dilemmas generation in simulations

Azzeddine Benabbou[1], Domitile Lourdeaux[2], Dominique Lenne[3]

Université de technologie de Compiègne, CNRS UMR 7253 Heudiasyc
57 avenue de Landshut – 60203 Compiègne Cedex, France

[1]azzeddine.benabbou@gmail.com; [2]domitile.lourdeaux@hds.utc.fr; [3]dominique.lenne@hds.utc.fr

Abstract- **Our ultimate purpose is to train individuals, in virtual environments, to handle critical situations. One of these critical situations are dilemmas. They refer to situations that lead to negative consequences whichever is the choice made by the protagonist. In critical contexts, it is crucial to know how to handle this kind of situations in order to prevent disastrous consequences from happening. Thus, people need to be exposed to various training situations in which they put in play and develop the appropriate skills. However, in complex domains, it is difficult – sometimes impossible – to write all the possible training scenarios. To address this problem, an automated generation approach is considered. In this article, we present KOBA, a scenario engine that automatically generates dilemma situations without having to write them beforehand. This engine uses knowledge models to extract the necessary properties for dilemmas to emerge. In this article, we present this approach and expose a proof of concept of the generation process.**

Keywords- Scenario generation; virtual environment; knowledge models; dilemmas

1 INTRODUCTION

There is no consensual definition of what a "critical situation" is across the various disciplines. A first meaning of this notion refers to the severity or dangerousness of an event. Sometimes, it can also refer to the likelihood of an event (Westrum 2006). In risk assessment, the classical approach consists in putting these two parameters together to calculate a criticality value: *criticality = severity * the likelihood*. More recent approaches have considered it to be related to different types of cognitive dissonances (Vanderhaegen and Carsten 2017). In a totally different meaning, a critical situation refers to situations where *critical incidents* occur in the context of the Critical Incidents Technique (Flanagan 1954). The term here is not related to severity, but it refers to the fact that an incident makes a significant contribution to an activity. In a recent work, J-M. Burkhardt et al. (2016) brought these different meanings together and determined seven dimensions that characterise a critical situation: severity, unpredictability, dilemma, ambiguity, sociocognitive load, novelty and lack of skills. These dimensions are not only critical because they can lead to negative consequences, but also because they are essential for the development of technical and non-technical skills.

The occurrence of critical situations can have several explanations. However, since the 80's, the analysis of major incidents in aviation (Helmreich 2000) stated that human factors play a major role in the occurrence of such incidents. Similarly, in healthcare, the study of adverse events on the patients revealed that they were not caused by a technical failure (lack of knowledge, technical skill failure), but mostly because of human factors and organizational issues (Cooper et al. 1982; Newbower, Cooper, and Long 1981). These factors cover a particular type of skills, which are the non-technical skills. Flin, O'Connor, and Crichton (2008) define these skills as "the cognitive, social, and personal resource skills that complement technical skills, and contribute to safe and efficient task performance". In order to prevent negative consequences from happening, individuals need to be trained to use these skills, especially in critical domains such as healthcare and car driving. However, due to several factors such as dangerousness, costs, staff availability or ethical considerations, it is sometimes impossible to ensure training sessions in genuine situations. To address this problem, the use of virtual environments was considered in several domains (Baker et al. 1993; Reznek, Harter, and Krummel 2002; Scott, Soria, and Campo 2017; Walker, Giddings, and Armstrong 2011) as it comes with several features (Jean-Marie Burkhardt, Lourdeaux, and Mellet-d'Huart 2006). Besides, it fosters a situated and constructivist learning[1] (Chittaro and Ranon 2007; Huang, Rauch, and Liaw 2010). Our ultimate objective is to contribute to the design of a generic architecture intended to conceive a virtual environment for non-technical skills training. More specifically, our contribution in this architecture consists in the *scenario generation engine* that produces the critical situations. To be efficient, this engine has to meet several criteria that are detailed in the following paragraph.

In order to facilitate the transfer of the skills, it is necessary to vary the conditions (situations) in which a task is performed (Tardif and Meirieu 1996). Thus, a generation engine has to be able to produce a large panel of situations.

---

[1] The constructivism is a learning theory that suggests that people construct their own understanding and knowledge, through experience and reflecting on those experiences (Piaget 1948; Vygotsky 1978).
Situated learning is a theory that suggests that learning is a function of the activity, context and culture (Lave and Wenger 1991)

Also, to foster a constructivist and situated learning/training, the learners must have a freedom of action in order to experience the consequences of their acts. Thus, a generation engine has to be able to take into consideration all the possible deviations from the initial scenario. Moreover, in order to produce the best possible experience for the learners, the generation engine has to control the simulation. It means that it has to be able to select the most relevant situations and adapt them to the learners. As far as critical situations are concerned, it must be able to adjust the severity and/or the intensity of the situation according to the learner's profile. The main challenge of a generation engine is to conciliate the above objectives (variability of situations, learner freedom and control of the simulation). In classical approaches, the developers have to anticipate all the learner's actions beforehand, and propose the appropriate scenario adjustments for each case. In complex domains, this approach is doomed to fail due to the great number of entities and the complex relations between them. Indeed, writing all the possible scenarios will lead to an authoring bottleneck (Spierling and Szilas 2009). To address this problem, an alternative approach would be to automatically generate these situations. In the literature, several works have adopted this approach, and used it to generate situations that are characterized by one or more of the seven critical dimensions stated earlier. However, most of these works focused on numerical dimensions and those that can be straightforwardly generated. In Death Kitchen for example, Lugrin and Cavazza (2006) used a Danger Matrix to represent numerical values of *severity* in order to generate dangerous kitchen events. This same dimension was used by Barot (2014) along with an unpredictability-like value to model a scenario planner for risk-management training in virtual environments. Other dimensions such as the *novelty* and *lack of skills* were addressed in other models. In TAILOR for example, Carpentier et al. (2013) attribute a degree of belief on the capacity of the learner to handle classes of situations. Using these beliefs along with a situations history, the system determines which situations have never been presented to the learner and which have not been mastered yet. However, we noticed that few works have addressed the other dimensions: *socio-cognitive load, ambiguity* and *dilemma*. Therefore, our work focused on these particular dimensions. *Dilemma* is the main subject of this paper. Should anyone need more detail about *ambiguity* generation, please refer to (Benabbou et al. 2018). Future work should address the remaining dimension.

In this paper, we focus on a particular type of critical situations, namely the dilemmas. According to the online Cambridge dictionary, a dilemma is "a situation in which a choice has to be made between possibilities that will all have results you do not want". Our interest in this dimension is twofold. First, our will is to tackle the challenge brought forward by the automated generation of a 'non-numerical' dimension; and second, because of the critical nature of the dimension itself. In fact, dilemmas are faced in a daily basis in several sensitive areas such as healthcare, hence the need to train the staff to handle them (Lecomte 2006). Also, dilemmas have widely been discussed with the emergence of autonomous cars, especially the case of unavoidable collisions (Bonnefon et al. 2016). We believe that automated generation of dilemmas can benefit these two areas and can ease the development of virtual environments for non-technical skill training. We argue that confronting individuals with these situations improves their non-technical skills (J-M. Burkhardt et al. 2016), and we make the hypothesis that it can be generated automatically. To validate this hypothesis, we designed KOBA: a scenario generation engine that automatically generates dilemmas without having to write them beforehand. This will contribute to creating dilemma scenarios with less authoring effort. Therefore, we are interested in the following research questions: how do we model dilemmas? How do we qualify them? and finally, how do we generate them automatically without having to write them beforehand?

This paper is organized as follows. In section 2, we expose some related work. In section 3, we present the global architecture in which KOBA is integrated. Section 4 details the dilemma generation process which is validated using a car driving simulation presented in section 5.

2   RELATED WORK

The trolley dilemma is one of the most famous dilemmas in the literature. It was stated for the first time by Foot (1967). The dilemma says: a runway trolley is moving toward five men working on the main track. The protagonist can use a lever to redirect the trolley onto the side track where there is only one man working. What should he do? This dilemma was widely discussed in the literature. It has been the subject of several studies that indicated that the vast majority of people agree that it is permissible to pull the lever (Greene et al. 2001; Hauser et al. 2007; Valdesolo and Desteno 2006). In 2012, Navarrete et al. (2012) investigated if the people would react similarly in an immersive environment. To do so, the authors exposed the participants to a 3D version of the trolley problem in a virtual environment, giving them the opportunity to pull the lever or not. Built on this study, a more advanced experiment was conducted by Skulmowski et al. (2014) who exposed the participants to different configurations of the same dilemma. The trolley dilemma and its variants were also transposed in several domains. In car driving for example,

one could face a situation where he has to choose between sacrificing his own life or killing the pedestrians. Due to the emergence of autonomous vehicles, this kind of situations is more and more studied in order to investigate which behaviours are admissible in this case. Sütfeld et al. (2017), for example, conducted a study in which they exposed the drivers to various virtual situations of inevitable collisions. Dilemmas are also used in training simulations. For instance, the researchers of the Institute for Creative Technologies in Southern California use virtual environments to simulate complex military scenarios. In the most recurring scenario, the player plays the role of a lieutenant who faces a dilemma at a particular point of the simulation. He has to choose between achieving his initial mission which is helping a platoon in trouble, and securing a landing zone for a medevac helicopter in order to rescue a civilian boy injured by the platoon (Rickel et al. 2001).

All the simulations presented above integrate the dilemmas in the initial scenario. They are scripted and planned in advance. Certain works such as (Skulmowski et al. 2014; Sütfeld et al. 2017) introduce some variability in the dilemmas. However, it is also pre-scripted and established in advance. This approach enables the authors to have a strong control on the simulation. Its use is relevant if the authors target a limited number of situations as it will not require a huge authoring effort. Otherwise, this approach is doomed to fail, especially in complex environments. To address this problem, an alternative approach would be to automatically generate dilemmas without having to script them beforehand. To our best knowledge there are only two systems in the literature that adopt such an approach.

GADIN is an interactive narrative engine designed by Barber and Kudenko (2007) that unfolds a story based on a user's response to dilemma situations. The dilemma generation process takes into consideration the social relationship between the characters – friends or enemies – and the utility of the actions. For example, the user could be confronted with a situation in which he has to decide whether to take or not an action which outcome would be positive for him but negative for his friend. According to the authors, this situation refers to a *Betrayal* dilemma. They identified a total of five dilemma categories according to the social relationship and the actions utility: Betrayal, Sacrifice, Greater Good, Take Down and Favour. GADIN is more oriented to generate a soap opera kind of stories. Thus, it does not cover some dilemma situations that can occur in real life. Indeed, the main limit of this system is that it fails to generate dilemmas when the decision-maker character has no social relationships with the others, or simply when he is the only character in the virtual environment. To address this problem, Harmon (2016) proposes EGAD: a more expressive dilemma generation system. Each character in this system has a list of cherished values, a list of despised values and a list of values of unknown importance. These values can either be preserved of rejected by the user's action. In fact, when a choice is presented to the user, each option of the choice serves to preserve or reject a value or a set of value. There are three basic types of choice: help, harm, or ignore. A help choice supports one or more values. A harm choice reject one or more values. An ignore choice gives the user the opportunity to refuse all the presented choices. Using these three basic types of choices the system is able to generate a wide panel of dilemmas. Given two actions, an example of dilemma would be a choice in which the user has to choose which cherished (despised) value to preserve (harm). The system also produces forced scenarios in which the user has to choose which cherished (despised) value to harm (preserve). Although this system improves the expressiveness of GADIN, it is still not suitable for 3D virtual environments. The reason is that the exclusivity of the choice – which is a main characteristic of a dilemma – is guaranteed at the extradiegetic level. It means that it is handled in graphical/textual elements that belong to an external layer that is independent from scenario. Concretely, in both systems, the choice is presented in a textual menu. This latter is closed once the user selects an option. The exclusivity of the choice is guaranteed, but this breaks the immersion of the user if he is using a 3D virtual environment. Moreover, EGAD could benefit from more expressiveness by considering also the outcomes of not doing an action. Finally, to our best knowledge, the author only validated that the system was more expressive than GADIN, but no further evaluations were conducted to validate that the system truly generates dilemmas. To overcome these limits, we propose KOBA, a generic dilemma generation engine. The purpose of KOBA is to improve the state of the art expressiveness and to guarantee the exclusivity of the choice at the intradiegetic level. Moreover, since the objective is to conceive virtual environments for training, KOBA takes into account several pedagogical constraints in the generation process.

3   ARCHITECTURE AND MODELS

KOBA is a dilemma generation engine that is integrated in a more global architecture as depicted in Fig. 1. In this section, we expose this architecture and present the different knowledge models used by KOBA to generate dilemmas.

3.1   Global architecture

During a training/learning session, the learner is immersed in a *virtual environment.* This latter consists of a digital environment in which the learner navigates and interacts using standard (e.g. mouse and keyboard) or virtual reality

(e.g. Vive headset[2]) devices. In this conceptual architecture, the virtual environment is an external and fully independent module. This modularity makes a clear separation between the semantic representation of the entities and their visual instantiation. It also enables the authors to design and orchestrate the scenario in a distinct layer without worrying about the visual and graphical aspects. In this virtual environment, the learner is exposed to a wide panel of critical and non-critical situations. His actions in these situations are observed by the *learner diagnosis* module (Bourrier et al. 2017). This module is responsible of establishing the learner diagnosis by checking the level of acquisition of each targeted skill. In each iteration, once the diagnosis is established, the module constructs a pedagogical instruction which is sent to the scenario orchestration system (SOS). This instruction is composed of three elements:

- A pedagogical intention: verify, strengthen or destabilize a skill.
- The skill or the set of skills targeted by the pedagogical intention.
- A level of criticality

This instruction guides the generation of the next situation. It goes without saying, that the generated situation varies according to the pedagogical intention of the instruction. For example, a situation which intention is to verify the state of knowledge of the learner is naturally different from a situation that aims to destabilize the learner (Luengo 2009). In the first case, the system tends to comfort the knowledge of the learner, while in the second case, the system tends to destabilize him in order to warn him about an invalid use of his knowledge. This instruction is received by the *generation engine.* This latter is responsible of automated scenario generation. It processes the pedagogical instruction and infer the properties that shall characterize the proposed situations. These properties include the critical dimension of the situations (dilemma, ambiguity, unpredictability, etc.). In our work, we cover the case when this property is set to *dilemma*[3]. This case is handled by KOBA, which is the submodule responsible of generating the dilemmas. It represents our main contribution and it is the core subject of this paper. KOBA uses knowledge models to extract the entities –characters, objects, actions, etc.– with properties that could produce a dilemma situation if put together. This is formulated as *scenario goals* that are sent to the *planner.* A scenario goal consists in a state of the world that the simulation needs to reach in order to produce the desired situation (in this case, a dilemma). The *planner* takes these *scenario goals* as inputs and tries to find a path to reach these goals without losing the scenario consistency. It uses directives to command the *world manager* to change the current world state. The *world manager* is responsible of managing the world model. It notifies the *virtual environment* when any change occurs in this model, and vice versa.
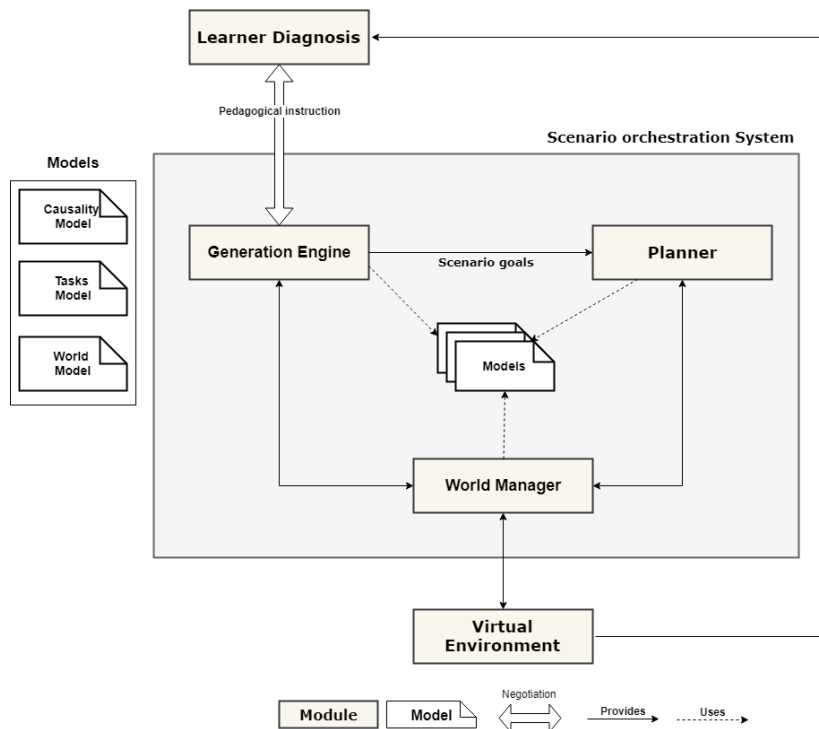

Fig. 1 Architecture

---

[3] The inference process is out of the scope of this paper

3.2   The knowledge models

The design of virtual environments for training is a collective task that involves several collaborators: domain experts, ergonomists, instructors, computer programmer, etc. These persons use different models according to their field of expertise. Ideally, each type of knowledge should to be represented in a separate model, or at least in a specific view of a central model. This modularity eases the use of the same knowledge in different contexts. In order to generate dilemmas, KOBA needs three types of knowledge. First, it needs the knowledge about the world in order to determine which entities (characters, objects, actions, etc.) can be involved in a dilemma. Second, it needs the knowledge about the activity in order to identify the potential incompatibilities between the actions and the semantic relations between them. Finally, it needs the knowledge about the causality in order to identify the actions that lead to negative consequences. In the following section, we describe the models used to represent these three types of knowledge.

3.2.1   The world model

The world model enables the experts to represent the knowledge related to the world entities, their properties and the relations between these entities. In the literature, there are several models that are used to represent the world knowledge (Badawi and Donikian 2004; Bille et al. 2004; Chevaillier, Querrec, and Septseault 2009; Kallmann and Thalmann 1999). In our work, for practical reasons, we used the WORLD-DL meta-model proposed by Carpentier (2015). However, any other model can be considered as long as it enables the experts to represent the entities and the semantic relations between them. In WORLD-DL, the entities with their properties are represented through ontology concepts and relations. This representation is not static since it is dynamically updated using operating rules. Fig. 2 illustrates an example of a fragment of a world model.
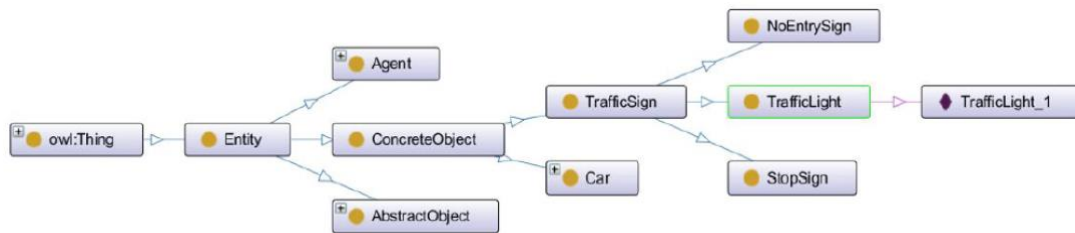
Fig. 2 Fragment of a world model using WORLD-DL

In this example, the world is composed of a Concrete Object "TrafficLight_1" which is an instantiation of the concept TrafficLight. This latter is a subclass of a more general concept which is "TrafficSign" that also includes no-entry and stop signs. Along with the cars, these concepts compose the concrete objects of the world. In addition to that, WORLD-DL also enables the authors to define abstract objects (AbstractObject) and agents (Agent) that are all subclasses of the main concept: "Entity".

3.2.2   The activity model

The activity model enables the experts to describe not only the prescribed activity, but also the actual activity as observed in the field. In the literature, there are several tasks models such as (Amokrane 2010; Edward 2011; Gerbaud et al. 2008; Rouillé et al. 2008). In our work, we used the ACTIVITY-DL meta-model (Barot 2014) first for practical reasons, and second, because it integrates some useful features comparing to the state of the art. ACTIVITY-DL represents the tasks through a hierarchy. Each task has a set of preconditions and postconditions. The formers refer to the conditions that make the realization of a task relevant (contextual preconditions) and physically possible (nomological preconditions[4]). The latter refer to the conditions that have to be satisfied in order to consider that the task is achieved. These conditions consist in a set of assertions. Each assertion is represented by a triplet *subject-predicate-object*. Moreover, each task – except the leaves – has a temporal constructor. This constructor informs about the temporal relation between the subtasks. It can be sequential, independent of parallel. Furthermore, the tasks can be tagged with any relevant information. To illustrate all these features, an example is presented in Table 1.

---

[4] According to Goel (1995) "Nomological constraints are constraints dictated by natural law".

Table 1 - Example of a task model

Task		Handle a red light
Subtasks		Slow down Brake
Constructor		Sequential
Conditions	Preconditions	(?light :has-color :red)
	Postconditions	(?vehicle :is-stopped true)
Tags		Conformity: 5 (out of 5) Technical skill: true

This example defines a main task "Handle a red light" that is composed of two subtasks: "Slow down" and "Brake". These two subtasks have to be achieved sequentially as specified by the temporal constructor. Moreover, the realization of the task is only relevant if the color of the traffic light is red (precondition). When the task is achieved, the vehicle stops (postcondition). Finally, the task is tagged with two items. The first one informs the system that the person who achieves the task has a high degree of compliance with rules. The second one informs the system that this task may mobilize non-technical skills.

3.2.3   Causality model

The causality model enables the experts to represent the relevant causality links between the world events. CAUSALITY-DL (Barot 2014), for example, is a meta-model based on graphical risk assessment formalisms. Events and actions are represented through graph nodes. The nodes are linked by causality or subsumption links. Also, they can be connected by logical gates (*AND* or *OR*). A key element of this model is the notion of *barrier.* A barrier is an event (or action) that serves to prevent the consequences of an anterior event from happening. It can refer to a task of the activity model. Fig. 3 shows an example of a fragment of a causality model.
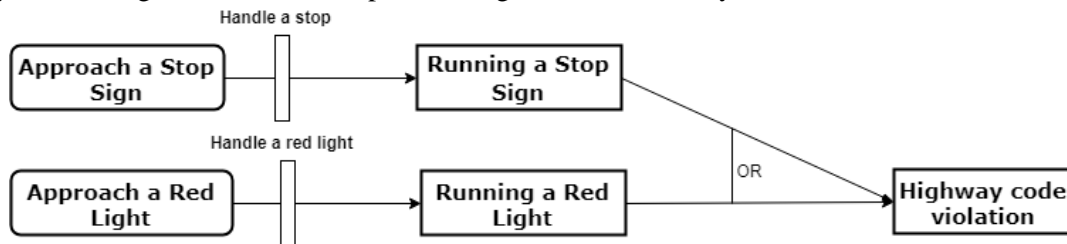
Fig. 3 Example of a causality model

This fragment describes the fact that there could be a *highway code violation* if the user runs a *stop sign* or a *red traffic light.* This can be prevented by activating the barriers: *handle a stop* and *handle a red light*.

4   AUTOMATED DILEMMA GENERATION

4.1   Approach

Our objective is to propose a generation engine that automatically generates dilemmas without having to write them beforehand. In a dilemma, all the possible actions (or not action) lead to negative consequences. We distinguish negative consequences in terms of *severity*, *points loss* and *normative violations.* This shall be detailed in 4.3. Also, we distinguish the prohibition dilemmas from the obligation dilemmas. This is not only a semantic distinction; it is also a structural distinction as the dilemma generation process is not the same. Besides, this prevents the generation engine from missing certain dilemmas due to the different ways in which the knowledge models are filled in by the experts. This shall be detailed in 4.2. Another specificity of our approach is that the exclusivity of the choice is inherent to the scenario and not handled at a graphical layer. To do so, KOBA uses the knowledge models to identify contradictory actions, that cannot be realized simultaneously. Finally, in order to propose the most relevant situation, KOBA takes into consideration a set of pedagogical and scenario constraints. These two last points shall be detailed in the generation process in 4.4.

4.2   Formalization of a dilemma

Our approach consists in generating automatically dilemmas using knowledge models that are not predestined, *a priori*, to model dilemmas. To do so, it is necessary first to represent formally a dilemma situation. This enables the generation engine to identify clearly what are the properties of a dilemma situation, and thus, to extract the elements that can produce the situations that meet these properties. Our formalization is based on the distinction made by Vallentyne (1989) between obligation and prohibition dilemmas.

An obligation dilemma is a situation where all the possible actions are mandatory (Vallentyne 1989). The necessary condition that have to be satisfied in this case in order to guarantee the dilemma is the following: the actions cannot all be done. In other words, the choice presented to the agent has to be exclusive. In fact, if the agent can realize all the actions, the dilemma is compromised. The challenge of a generation system in this case is to prescribe a state of the world that meets this necessary condition. Let us consider a situation with two actions. Let a_1 and a_2 be two actions that can be realized by an agent. For certain reasons, the agent has to do a_1. In other words, not doing a_1 leads to negative consequences NC_{a_1}. For other reasons, the agent has to do a_2. In other words, not doing a_2 leads to negative consequences NC_{a_2}. The necessary condition in an obligation dilemma is that the agent cannot realize both of the actions simultaneously. If it is not the case, the dilemma can still be guaranteed if the realization of both actions lead to negative consequences too. Given all this, an obligation dilemma is formalized as follows:

> *S(A,NC)* is an obligation dilemma situation if $\exists\, a_1, a_2 \in A$ such as:
> $$\neg a_1 \to NC_{\neg a_1}$$
> $$\neg a_2 \to NC_{\neg a_2}$$
> $$\big((a_1 \lor a_2) \land \neg(a_1 \land a_2)\big) \lor\ (a_1 \land a_2 \to NC_{a_1 \land a_2})$$

A prohibition dilemma is a situation where all the possible actions are prohibited. In this case, the necessary condition that has to be satisfied in order to guarantee the dilemma is the following: one action, at least, has to be done absolutely. The challenge of a generation engine is to push the agent to make a choice even if all the actions lead to negative consequences. When all the actions are prohibited, the only positive way out is to do nothing. In order to guarantee the dilemma, this option has to be removed. Thus, it is necessary that the non-choice leads to negative consequences too. Let us consider a situation with two actions. Let a_1 and a_2 be two actions that can be realized by an agent. For some reasons, the agent ought not to do a_1. In other words, doing a_1 leads to negative consequences NC_{a_1}. For other reasons, the agent has to do a_2. In other words, doing a_2 leads to negative consequences NC_{a_2}. The necessary condition in a prohibition dilemma is that the agent has to make a choice anyway. To do so, we prohibit the non-choice too. Given all this, a prohibition dilemma is formalized as follows:

> *S(A,NC)* is a prohibition dilemma situation if $\exists\, a_1, a_2 \in A$ such as:
> $$a_1 \to NC_{a_1}$$
> $$a_2 \to NC_{a_2}$$
> $$\neg a_1 \land \neg a_2 \to NC_{\neg a_1 \land \neg a_2})$$

4.3   Negative consequences

In the previous section, we detailed the formalization of the obligation and prohibition dilemmas. One of the key elements of this formalization is the negative consequences. In our model, we distinguish three types of negative consequences: the *severity*, *points loss* and *normative violations*. We shall detail these three types in the following paragraphs.

The *severity* corresponds to the degree of damage caused by an action or an event (e.g. hitting a pedestrian). In our model, it is a numerical value associated to the events. It varies from 0 to 10. We consider that an event is negative if this value is strictly greater than 0.

The second type of negative consequences is the *point loss.* In a simulation, a score can be attributed to the user. In general, it is a performance score that is increased if the user successfully achieves the tasks, and decreased otherwise. This score varies from one context to another. In car driving for example, it can correspond to the driving licence points. In video games, it could refer to life points or the quantity of resources. In our model, the points are numerical values associated to the events. They can refer to anything relevant to the context of the simulation. We consider that an event is negative if this value is strictly lesser than 0.

The last type of negative consequences is the *normative violations*. Each individual has his own hierarchy of human values. He cherishes some and despises others. In certain situations, realizing an action leads to the violation of one or more of these values. In our model, the values are tags associated to the events. We consider that an event is negative if the violated value is cherished by the user. This situation corresponds to what it is called a *moral dilemma*.

4.4   Dilemma generation process

Now that we have formally defined prohibition and obligation dilemmas, we shall detail in this section the whole process of automated generation. This process is illustrated in Fig. 4.
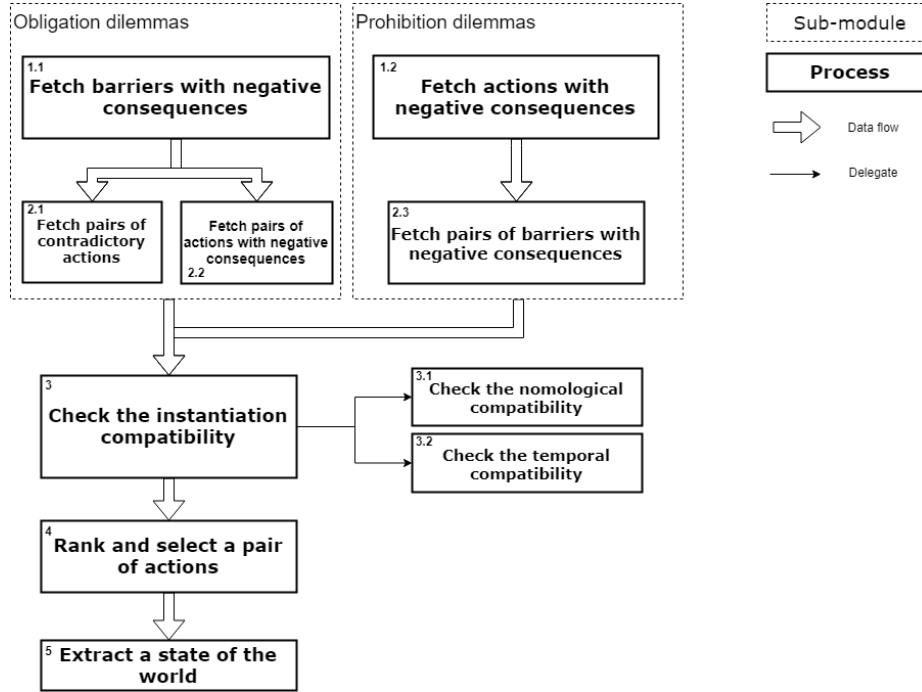

Fig. 4 Dilemma generation process

First, depending on which type of dilemmas is concerned, KOBA delegates the fetching to the appropriate sub-module. As far as obligation dilemma are concerned, KOBA constructs a list of barriers that lead to negative consequences (1.1). Then, it groups them in pairs if the corresponding actions are contradictory (2.1) or lead to negative consequences (2.2). As far as prohibition dilemmas are concerned, KOBA constructs a list of actions that lead to negative consequences (1.2). Then, it groups them in pairs if the corresponding barriers lead to negative consequences (2.3). At this stage, KOBA has two lists of pairs of actions: one for the obligation dilemmas and one for the prohibition dilemmas. First, it checks if the actions of each pair are compatible in terms of physical instantiation (3.1). Then, it verifies if there is no temporal incompatibility between the actions of each pair (3.2). Once these two verifications are done, KOBA ranks the pairs according to several criteria and selects the highest ranked one (4). Finally, it extracts a state of the world that is conducive to the realization of the pair of actions. All these steps shall be detailed in the following subsections.

4.4.1   Generation of obligation dilemmas

In order to generate obligation dilemmas, KOBA has to identify first the actions that lead to negative consequences if they are not realized. To extract this information, KOBA uses the causality model. An action – or an event – leads to negative consequences if there is a path between this action's node and at least one of the negative consequences nodes (severity, points loss or normative violations). But here, we are interested in the consequences if an action is not realized. Thus, rather than considering the "normal" nodes, KOBA examines the barriers. Let us recall that a barrier prevents its posterior events from happening. Thus, the process of identifying the action that leads to negative consequences if they are not realized, consists in identifying the barriers that lead to negative consequences. Let us take the example illustrated in the Fig. 5.
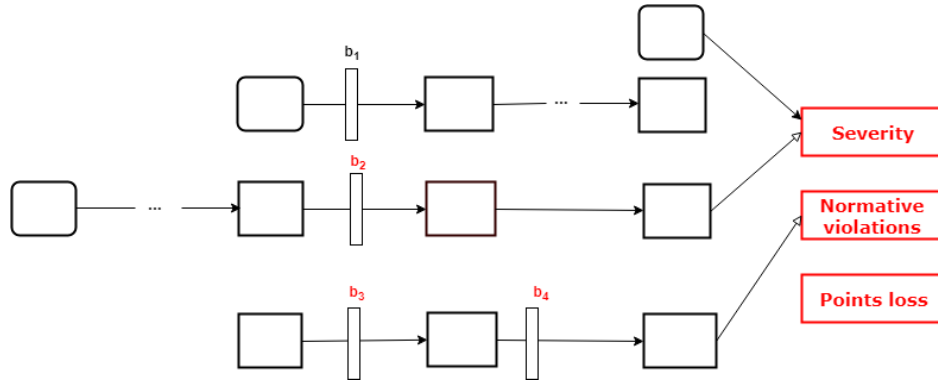
Fig. 5 Searching for barriers that lead to negative consequences

The barrier b_2 is selected because there is a path between this barrier and the negative consequences node *Severity*. Likewise, the barriers b_3 and b_4 are selected because there is a path between each one of them and the negative consequences node *Normative Violations*. As for the barrier b_1, it is discarded because there is no path between this barrier and any of the negative consequences node. At this stage of the process, KOBA possesses a list of barriers that lead to negative consequences.

According to the formalization, the necessary condition in an obligation dilemma is that the agent cannot realize both actions simultaneously. Thus, to ensure this condition, the actions that are proposed to the agent have to be contradictory. To identify such actions, KOBA uses the activity model. It scans by pairs the set of barriers selected previously – i.e. actions that lead to negative consequences if they are not realized –, then, for each pair of actions, it examines the postconditions of its actions. If the postconditions are incompatible, then the pair of actions is contradictory. We consider that two conditions are incompatible if they have the same subject and predicate but a different object (e.g. "Vehicle is-stopped true" and "Vehicle is-stopped false"). This process can be compared to the rule-based system proposed by Vanderhaegen (2016) that detects inconsistencies in knowledge bases. In this system, the conflict emerges if the conclusion of one rule is the opposite of the conclusion of the other (i.e. $conc(R_1) = NOT(conc(R_2))$). In our models, the NOT is not explicit. It is implied in the semantic of the action (i.e. "Slow down" implies not "to accelerate"). Thus, compared with the previous system, KOBA realizes the conflict detection in a deeper level.

Moreover, according to this same formalization, if the actions are not contradictory, it is still possible to produce a dilemma if the realization of both actions leads to negative consequences too. To identify such actions, KOBA uses the causality model. First, it scans by pairs the set of selected barriers and checks if they also exist as actions in the causality model. Then, it verifies if the actions have an *AND* gate as a common descendant. If it is the case, the pair is retained. Otherwise, it is discarded. Let us take the example illustrated in Fig. 6.
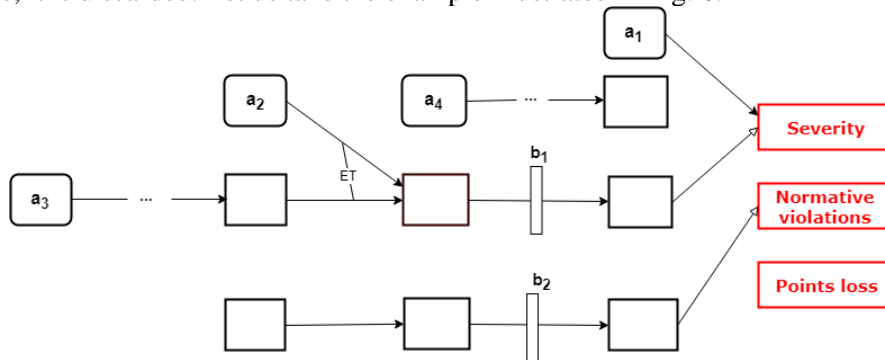

Fig. 6 Search for actions that are connected with *AND* gate

Let us suppose that the selected barriers correspond to the actions a_1, a_2 and a_3. The pair $\{a_2, a_3\}$ is retained because its actions are connected by an *AND* gate that leads to a negative consequence. As for the pairs $\{a_1, a_2\}$ and $\{a_1, a_3\}$, they are discarded because they are not connected by an *AND* gate.

4.4.2   Generation of prohibition dilemma

In order to generate prohibition dilemmas, KOBA has to identify first the actions that lead to negative consequences if they are realized. To extract this information, KOBA uses the causality model. It scans the set of

actions and checks if there is a path between the action nodes and the negative consequences nodes. According to the formalization, the necessary condition in a prohibition dilemma is that the non-choice leads to negative consequences too. The non-choice here corresponds to not doing both actions. To identify such actions, KOBA uses the causality model. It scans by pairs the actions previously selected and retain the ones that exist also as barriers. Then for each pair, it verifies if the barriers have an *AND* gate as common descendant. Let us take the example illustrated in Fig. 7.
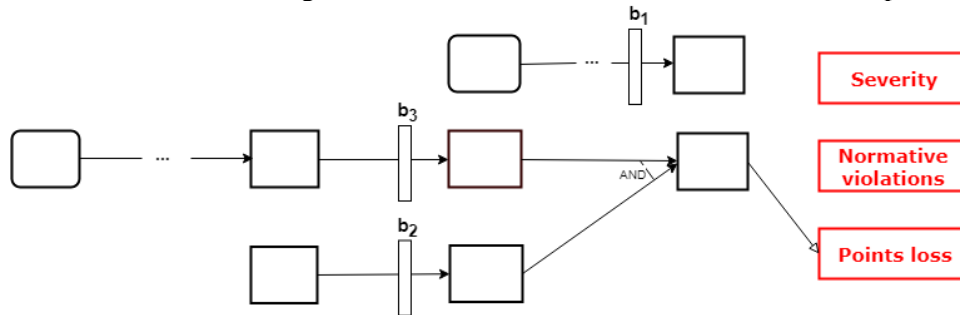
Fig. 7 Search for barriers that are connected by and *AND* gate

In this example, the pair $\{b_2, b_3\}$ is retained because the two barriers lead to a negative consequence and are connected by an *AND* gate. As for the pairs $\{b_1, b_2\}$ and $\{b_1, b_3\}$, they are discarded because b_1 does not lead to negative consequences.

4.4.3   Instantiation compatibility

At this stage, KOBA has two lists of pair of actions: one list for the obligation dilemmas and another one for the prohibition dilemmas. However, in order to be able to generate a situation that involves one of these pairs of actions, KOBA has to verify that these pairs are temporally and nomologically compatible.

Two actions are nomologically compatible if they can "physically" be instantiated in the same situation. For example, the actions *Open a door* and *Close a door* are nomologically incompatible because this would require the door to be physically open and closed in the same time, which is impossible. To identify such actions, KOBA uses the activity model. In particular, it checks the nomological preconditions of the actions. If they are compatible, the pair is retained, otherwise – i.e. if the object is different for the same predicate and subject – it is discarded.

Moreover, the actions of the pair have to be temporally compatible. It means that the realization of an action has to be independent from the other. To identify such actions, KOBA uses the activity model. It identifies the actions that have a common ancestor which has a *sequential* temporal constructor. When two actions are related in some point of the hierarchy by a sequential temporal constructor, it means that the realization of one of them is only possible if the other is already done. Thus, KOBA retains only the pair of actions that are connected by an *independent* or *parallel* constructor.

4.4.4   Ranking and selecting a pair of actions

At this stage, KOBA has two lists of pair of actions that are instantiable. However, in order to select the most relevant pair, the engine has to rank them. To do so, KOBA attributes a score to each pair according to several constraints. We define two categories of constraints: the pedagogical constraints and the scenario constraints. The formers refer to the properties that a situation has to verify in order to achieve the pedagogical objectives. In our work, we take into account the following constraints:
- The **maximum severity** of the situation. This constraint is specified if we want to set a limit of severity in order not to frustrate the user.
- The **minimum severity** of the situation. This constraint is specified if we want to keep the situation challenging for the user
- The **difference of severity**. This constraint is specified if we want to increase (decrease) the intensity of the dilemma. We believe that the more this difference is close to zero, the more intense a dilemma will be.
- The **type of negative consequences**. This constraint is specified if we want to target a specific type of negative consequences. It is useful if we want to generate a moral dilemma for example.

As for the scenario constraints, they refer to the condition that a situation has to meet in order to achieve the scenario objectives. In our work, we defined the following constraints:
- The **probability of instantiation**. It refers to the minimum degree of certainty that the generation engine must have regarding the possibility of instantiating a situation involving the pair of actions.

- The **time of instantiation.** It refers to the maximum theoretical time that the scenario engine can take to instantiate – and/or direct the simulation to – a situation that involves the pair of actions.

The ideal pair of actions is the one that fully satisfies these constraints. However, in practice, there is hardly such a pair. Thus, it is interesting to be able to specify the importance of each type of constraint. In some cases, for example, we would like a dilemma that absolutely respect the pedagogical constraints even if it has little chances to be instantiated. On the contrary, in other cases, we would like a dilemma that has strong chances to be instantiated even if it does not satisfy completely the pedagogical constraints. In order to model such preferences, each type of constraints has a weight. KOBA uses these weights to calculate the global score for each pair of actions as follows:

$$s_{total} = s_{pc} * \omega_{pc} + s_{sc} * \omega_{sc}$$

with s_{pc} being the score of satisfaction of the pedagogical constraints and s_{sc} being the score of satisfaction of the scenario constraints. ω_{pc} and ω_{sc} are respectively the weight attributed the pedagogical and scenario constraints.

4.4.5 Extraction of a state of the world

At this stage, KOBA has a ranked list of pairs of actions. It selects the pair with the highest score and prescribes a world state that is conducive to the realization of the actions of the pair. The prescribed world state corresponds to a partial description of the world in the form of an aggregate of assertions. Concretely, the prescribed world state ws corresponds to the aggregation of the preconditions of the actions of the pair: $ws = P_{a_1} \cup P_{a_2}$, with P_{a_1} and P_{a_2} the respective preconditions of the actions a_1 and a_2. The Fig. 8 illustrates an example of this extraction.
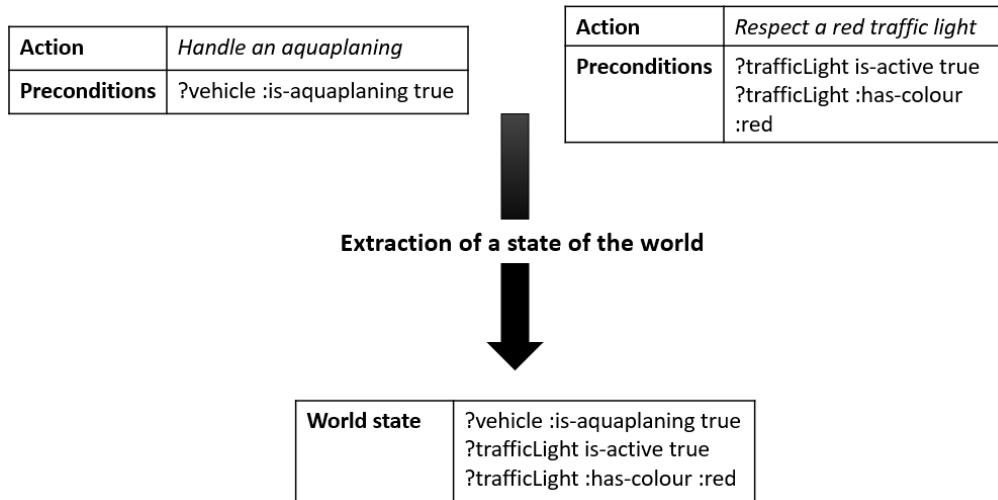
Fig. 8 Extraction of a world state

In this example, the pair is composed of the following actions: *<Handle an aquaplaning, Respect a red traffic light>*. Each action has its own preconditions. To handle an aquaplaning, the vehicle has to be aquaplaning (?vehicle :is-aquaplaning true). To respect a red traffic light, there must be a red traffic light (?trafficLight :is-active true AND ?trafficLight :has-colour :red). A world state that is conducive to the realization of these actions is described by the aggregation of their preconditions. In this case, it is a situation where there is an active red traffic light and where the vehicle is aquaplaning. This world state is then transmitted to the planner, which is in charge of directing the simulation towards this *goal state*.

5 EVALUATIONS AND RESULTS

As we said before, our ultimate objective is to train individuals to use their non-technical skills to handle critical situations, such as dilemmas. However, the purpose of this paper is not to validate that exposing individuals to dilemmas will improve their non-technical skills. We suppose that this assumption is true. The main objective of this work is to validate that it is possible to automatically generate dilemmas using knowledge models without having to write them beforehand. The use of this automated approach would be relevant for complex domains in which the authoring of all the possible situations is impossible.

In order to validate the capabilities of KOBA, it was necessary to realize an experiment in a virtual environment to investigate if the generated situations were perceived as dilemmas. The experiment consisted in exposing the participants to situations of different nature. We wanted to see if in the middle of these situations, the generated ones were perceived as dilemmas. More precisely, we wanted to check if the generated dilemmas were perceived in the

same way as the scripted ones. Our hypothesis is that it is true: *KOBA generates dilemmas*. To validate this assumption, we divided it into three sub-hypotheses:

- H_1: *the participants perceive that they have to choose between at least two options in both of the conditions (generated and scripted).*
- H_2: *the participants perceive that there is no option without negative consequences in both of the conditions (generated and scripted).*
- H_3: *the participants consider that they were confronted with a dilemma in both of the conditions (generated and scripted).*

In order to maximize the immersion, this experiment was conducted in a virtual reality room composed of three large screens. In the middle of the room, we put a Playseat and a Logitech racing wheel as shown in Fig. 9.

Fig. 9 Virtual reality room

5.1   Protocol of the experiment

We recruited a total of 20 participants. This number might be relatively low due to the complexity of setting up the experiment and the difficulty in recruiting more participants in a short period of time. All of them played two preliminary scenarios to get familiar with the experiment setup. The preliminary scenarios were pretty simple. They consisted in car driving situations in which the participants have to brake and avoid some obstacles. After that, the participants were exposed to four scenarios: a generated dilemma scenario, a scripted dilemma scenario, a critical scenario and a normal scenario. The last two scenarios served as a camouflage for the others. All the scenarios started by exposing an objective to the participant. We detail below these four scenarios:

- Normal scenario: the player has to join a colleague at the end of the road. The triggered event in this scenario is a visible pedestrian who crosses the road from the pedestrian passage.
- Critical scenario: the player has to pick up a colleague who is waiting for him just after the first stop sign. The triggered event is a hidden pedestrian that suddenly crosses in the middle of the road. If the player does not slow down in time, there is a risk of collision.
- Generated dilemma scenario: the player has to pick up a colleague waiting for him at the end of the road. In this scenario, the user is confronted to situations of inevitable collision.
- Scripted dilemma scenario: the player has to follow his friend's car because he does not know the place of the rendezvous. In his way, his friend takes a non-entry road. The player has to choose between respecting the highway code and fulfilling his personal objective.
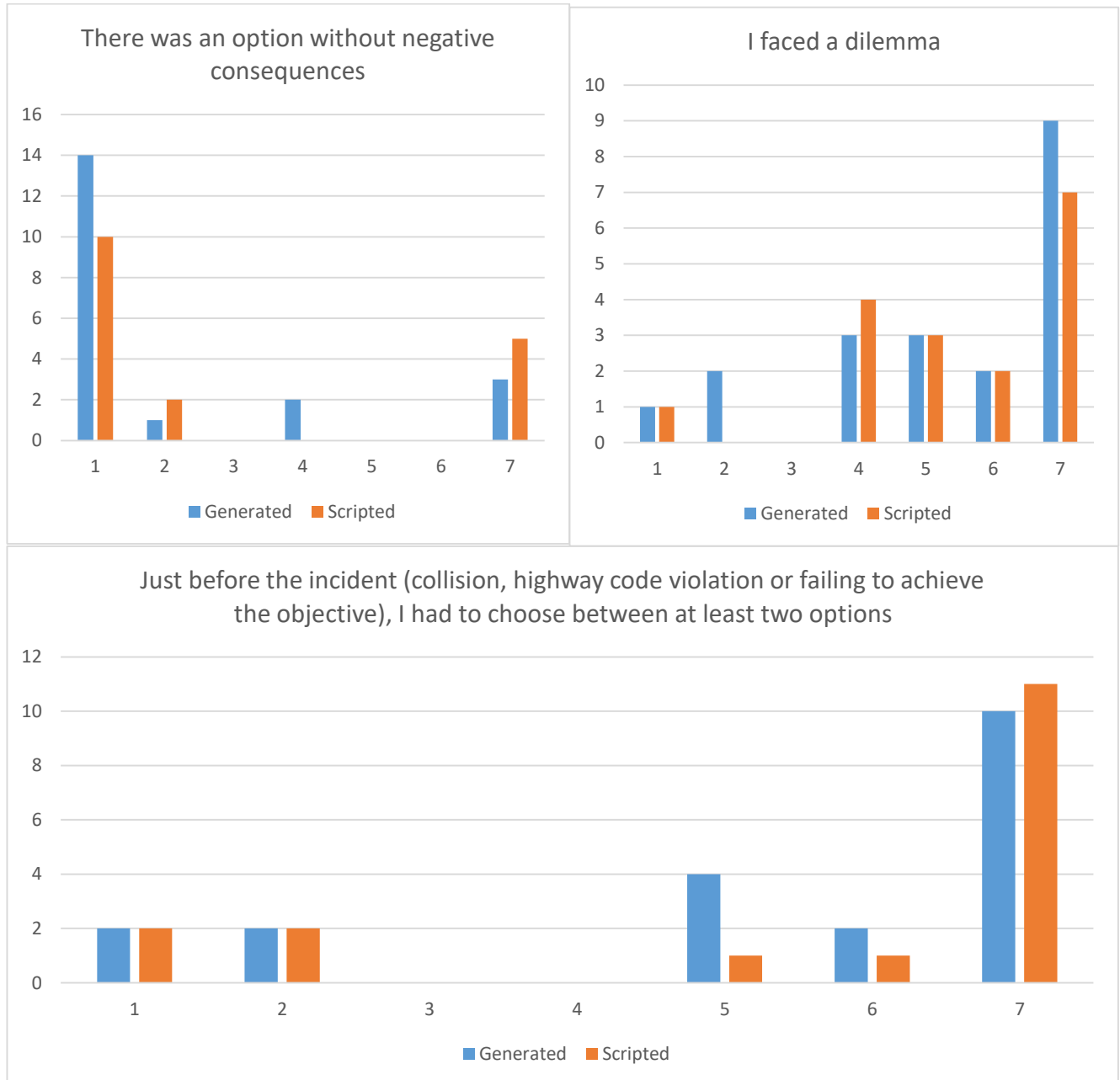
The scenarios were presented to the participants in a random order. A scenario terminates if the participant fulfils his objective or if there is an accident. At the end of each scenario, the participants had to fill in a questionnaire, which purpose was to see how they perceived the situations. Its key items were:

- I had the choice between at least two options.
- There was an option without negative consequences.
- I faced a dilemma.

These statements were evaluated by the participants by giving a quantitative value based on a seven-level Likert scale (Likert 1932). Answering these items was only mandatory if the scenario was terminated by a negative consequence (collision, high way code violation or not fulfilling the objective). Once the participants played all the scenarios, we conducted a little session of debriefing in order to discuss their choices. The results of this evaluation are discussed in the following section.

5.2   Results

Fig. 10 presents the results associated to the generated and the scripted scenarios. We note that for the scripted scenario, we only have 17 answers out of 20. The three participants who did not answer these questions stated that they simply did not see the non-entry sign and that they would have acted differently if they had seen it. Thus, we did not retain these three participant's answers when comparing the generated and scripted scenarios.

Fig. 10 Results of the generated and scripted scenario

Fig. 10 shows the participants' answers to the three questions stated in section 5.2 for the scripted and the generated dilemma scenarios. The horizontal axis represents the Likert scale, while the vertical axis represents the number of participants. These results shall be discussed in section 5.3.

We recall that our purpose was to investigate if the generated dilemma scenario was perceived in the same way as the scripted scenario. In other terms, we wanted to verify if there was a significant difference between the answers of the generated and the scripted dilemma scenarios. To do so, we used the Wilcoxon signed-rank test (Wilcoxon 1945). This test compares two measures of a quantitative variable carried out on the same subject. Concretely, it verifies the hypothesis that the average difference between the two measures is null. This hypothesis is rejected if the

p-value is lesser than 0.1. We used this test to compare the answers to the questions that serves to verify our three sub-hypothesis. The Table 2 gives a summary of the test results.

Table 2 p-value result of the Wilcoxon

	H_1	H_2	H_3
p-value	0.41	0.13	0.68

The table shows the p-value related the test of the three hypothesis stated in section 5.

5.3   Results interpretation

For the generated dilemma scenario, 80% of the participants stated that they had to choose between at least two options. Thus, it may be deduced that, in absolute terms, KOBA managed to generate situations that present different options to the users. Besides, we compared this result with the scripted dilemma by using the Wilcoxon test. This latter returned a *p-value* of 0.41. Thus, there was no significant difference between the two scenarios. Therefore, the hypothesis H_1 is verified.

Moreover, 75% of the participants stated that in the generated dilemma scenario, there was no option without negative consequences. Thus, it may be deduced that, in absolute terms, KOBA managed to generate situations that lead to negative consequences no matter was the user action. Besides, we compared this result with the scripted dilemmas scenario by using the Wilcoxon test. This latter returned a *p-value* of 0.13. Thus, there was no significant difference between the two scenarios. Therefore, the hypothesis H_2 is verified.

Furthermore, 70% of the participants considered that they were confronted with a dilemma in the generated scenario. It may be deduced that, in absolute terms, KOBA managed to generate situations that are perceived as dilemmas. Besides, we compared this result with the scripted dilemmas scenario by using the Wilcoxon test. This latter returned a *p-value* of 0.68. Thus, that there was no significant difference between the two scenarios. Therefore, the hypothesis H_3 is verified.

All in all, in absolute terms, it may be deduced that KOBA managed to generate situations that present different options to the user. Also, it managed to generate situations that lead to negative consequence whichever is the option selected by the user. Besides, there was no significant differences between the generated dilemma scenario and the scripted one. Therefore, it may be deduced that KOBA managed to automatically generate dilemma situations without having to write them beforehand.

6   CONCLUSION

Our work is related to virtual environments for training. In particular, we were interested in generating critical situations in order to foster the development of non-technical skills. In this paper, we focused on a particular type of critical situations: the dilemmas. We argued that automated generation of these situations is a relevant approach to tackle the problem of "authoring bottleneck". Thus, we proposed KOBA: a generation engine that automatically produces dilemmas from knowledge models that are not intended, *a priori,* to model dilemmas. We distinguished two types of dilemmas: the prohibition dilemmas and the obligation dilemmas. For each one of them, we proposed a formalization that states the necessary conditions that need to be met in order to consider that a situation is a dilemma. Based on this formalization, we proposed KOBA, a generation engine that takes into consideration the pedagogical and the scenario constraints of the simulation. We validated this generation process with an experiment in car driving simulation. The results showed that there was no significant difference between how the generated dilemma and the scripted dilemma were perceived.

To enhance the current work, it would be interesting to improve how moral dilemmas are generated. As far as they are concerned, it is necessary to take into consideration the moral profile of the person. In our work, we proposed a simple representation of this moral profile based on the theory of universal value of (Schwartz 2017). This representation could be enhanced by using an uncertainty framework that quantify the uncertainty, the ignorance and also the conflict that the system has about each moral value. This will enable the system to identify which moral values it could be interesting to challenge. Also, in order to update this moral profile, this framework should provide the system with information fusion techniques. We believe that the theory of belief functions (Shafer 1976) would be a serious track to consider.

7   REFERENCES

Amokrane, Kahina. 2010. "Suivi de l'apprenant En Environnement Virtuel Pour La Formation à La Prévention Des Risques Sur Des Sites {SEVESO}." Université de Technologie de Compiègne.

Badawi, Marwan, and Stéphane Donikian. 2004. "Autonomous Agents Interacting With Their Virtual Environments Through Synoptic Objects." In *Proceedings of {CASA} 2004*, , 179–87. http://marwan.badawi.free.fr/en/publications.html (September 23, 2013).

Baker, David et al. 1993. "Aviation Computer Games for Crew Resource Management Training." *The International Journal of Aviation Psychology* 3(2): 143–56. http://www.tandfonline.com/doi/abs/10.1207/s15327108ijap0302_4 (February 17, 2018).

Barber, Heather, and Daniel Kudenko. 2007. "A User Model for the Generation of Dilemma-Based Interactive Narratives." *Papers from the 2007 AIIDE Workshop, Optimizing Player Satisfaction*: 13–18.

Barot, Camille. 2014. "Scénarisation d'environnements Virtuels. Vers Un Équilibre Entre Contrôle, Cohérence et Adaptabilité." Université de technologie de Compiègne.

Benabbou, Azzeddine, Domitile Lourdeaux, and Dominique Lenne. 2018. "Towards Generation of Ambiguous Situations in Virtual Environments for Training." In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Cham, 631–35. http://link.springer.com/10.1007/978-3-319-98572-5_61 (December 4, 2018).

Bille, Wesley et al. 2004. "Using Ontologies to Build Virtual Worlds for the Web." In *ICWI*, , 683–90. https://wise.vub.ac.be/sites/default/files/publications/ICWI2004_Bille.pdf (July 31, 2014).

Bonnefon, J.-F., A. Shariff, and I. Rahwan. 2016. "The Social Dilemma of Autonomous Vehicles." *Science* 352(6293): 1573–76. http://www.ncbi.nlm.nih.gov/pubmed/27339987 (March 17, 2017).

Bourrier, Yannick, Jambon Francis, Catherine Garbay, and Vanda Luengo. 2017. "A Multi-Layered Architecture for Analysis of Non-Technical-Skills in Critical Situations." https://hal.archives-ouvertes.fr/hal-01517152 (May 24, 2018).

Burkhardt, J-M. et al. 2016. "Simulation and Virtual Reality-Based Learning of Non-Technical Skills in Driving: Critical Situations, Diagnostic and Adaptation." In *IFAC-PapersOnLine*, , 66–71.

Burkhardt, Jean-Marie, D Lourdeaux, and D Mellet-d'Huart. 2006. "La Réalité Virtuelle Pour l'apprentissage Humain." *G. Moreau (Coord.), B. Arnaldi & P. Guitton (Coords). Le Traité de la réalité virtuelle* 4: 43–100.

Carpentier, Kevin. 2015. "Scénarisation Personnalisée Dynamique Dans Les Environnements Virtuels Pour La Formation." Université de Technologie de Compiègne. https://tel.archives-ouvertes.fr/tel-01120040/ (April 4, 2017).

Carpentier, Kevin, Domitile Lourdeaux, and Indira Thouvenin. 2013. "Dynamic Selection of Learning Situations in Virtual Environment." In *Proceedings of {ICAART}'13: International Conference on Agents and Artificial Intelligence*, Barcelona, Spain, 101–10.

Chevaillier, Pierre, Ronan Querrec, and Cyril Septseault. 2009. "{VEHA}, Un Métamodèle d'environnement Virtuel Informé et Structuré." *Techniques et sciences informatiques* 28(6–7): 715–40. http://tsi.revuesonline.com/article.jsp?articleId=13816 (September 23, 2013).

Chittaro, Luca, and Roberto Ranon. 2007. "Web3D Technologies in Learning, Education and Training: Motivations, Issues, Opportunities." *Computers & Education* 49(1): 3–18. https://www.sciencedirect.com/science/article/pii/S0360131505000813 (October 8, 2018).

Cooper, J B, C D Long, R S Newbower, and J H Philip. 1982. "Critical Incidents Associated with Intraoperative Exchanges of Anesthesia Personnel." *Anesthesiology* 56(6): 456–61. http://www.ncbi.nlm.nih.gov/pubmed/7081730 (February 26, 2018).

Edward, Lydie. 2011. "Modélisation Décisionnelle de Personnages Virtuels Autonomes Évoluant Dans Un Environnement Pour La Présentation Des Risques Sur Les Sites {SEVESO}." Université de Technologie de Compiègne.

Flanagan, J.C. 1954. "The Critical Incident Technique." *Psychological Bulletin* 51(4): 327–58. http://content.apa.org/journals/bul/51/4/327 (January 18, 2017).

Flin, R, P O'Connor, and M Crichton. 2008. "Safety at the Sharp End: A Guide to Non-Technical Skills; Chapter 2." In *Safety at The Sharp End: A Guide to Non-Technical Skills*, , 17–40.

Foot, Philippa. 1967. "The Problem of Abortion and the Doctrine of Double Effect."

Gerbaud, Stéphanie et al. 2008. "{GVT}: A Platform to Create Virtual Environments for Procedural Training." In *{IEEE} Virtual Reality*, Reno, États-Unis, 225–32. http://hal.inria.fr/inria-00266625.

Goel, Vinod. 1995. *Sketches of Thought*. MIt Press.

Greene, J D et al. 2001. "An FMRI Investigation of Emotional Engagement in Moral Judgment." *Science (New York, N.Y.)* 293(5537): 2105–8. http://www.ncbi.nlm.nih.gov/pubmed/11557895.

Harmon, Sarah. 2016. "An Expressive Dilemma Generation Model for Players and Artificial Agents." In *The Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-16)*, , 176–82.

Hauser, M et al. 2007. "A Dissociation between Moral Judgments and Justications." *Mind & Language* 22(1): 1–21.

Helmreich, R L. 2000. "On Error Management: Lessons from Aviation." *BMJ (Clinical research ed.)* 320(7237): 781–85. http://www.ncbi.nlm.nih.gov/pubmed/10720367 (February 26, 2018).

Huang, Hsiu-Mei, Ulrich Rauch, and Shu-Sheng Liaw. 2010. "Investigating Learners' Attitudes toward Virtual Reality Learning Environments: Based on a Constructivist Approach." *Computers & Education* 55(3): 1171–82. https://www.sciencedirect.com/science/article/pii/S0360131510001466#bib63 (October 8, 2018).

Kallmann, M, and D Thalmann. 1999. "Direct 3d Interaction with Smart Objects." In *Proceedings of the {ACM} Symposium on Virtual Reality Software and Technology*, , 124–30.

Lave, Jean, and Etienne Wenger. 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge university press. http://books.google.fr/books?hl=fr&lr=&id=CAVIOrW3vYAC&oi=fnd&pg=PA11&ots=OBkGqn4LHp&sig=-FlIRtlWtvo8tTFXKS_h0QQQLpc (August 14, 2014).

Lecomte, Marie-Anne. 2006. "La Formation à l'éthique Des Étudiants En Soins Infirmiers (Belgique)." *Recherche en soins infirmiers* 86(3): 4. http://www.cairn.info/revue-recherche-en-soins-infirmiers-2006-3-page-4.htm.

Likert, Rensis. 1932. "A Technique for the Measurement of Attitudes." *Archives of psychology*.

Luengo, Vanda. 2009. "Les Rétroactions Épistémiques Dans Les Environnement Informatique Pour l'Apprentissage Humain." Université Joseph Fourier - Grenoble I. internal-pdf://hdr_luengo_retroaction.

Lugrin, Jean Luc, and Marc Cavazza. 2006. "AI-Based World Behaviour for Emergent Narratives." In *International Conference on Advances in Computer Entertainment Technology 2006*,.

Navarrete, C David, Melissa M McDonald, Michael L Mott, and Benjamin Asher. 2012. "Virtual Morality: Emotion and Action in a Simulated Three-Dimensional 'Trolley Problem'." *Emotion (Washington, D.C.)* 12(2): 364–70. http://www.ncbi.nlm.nih.gov/pubmed/22103331.

Newbower, R S, J B Cooper, and C D Long. 1981. "Learning from Anesthesia Mishaps: Analysis of Critical Incidents in Anesthesia Helps Reduce Patient Risk." *QRB. Quality review bulletin* 7(3): 10–16. http://www.ncbi.nlm.nih.gov/pubmed/6787496 (February 26, 2018).

Piaget, Jean. 1948. *La Naissance de l'intelligence Chez l'enfant*. Delachaux et Niestlé.

Reznek, M., Phillip Harter, and Thomas Krummel. 2002. "Virtual Reality and Simulation: Training the Future Emergency Physician." *Academic Emergency Medicine* 9(1): 78–87. http://doi.wiley.com/10.1197/aemj.9.1.78 (February 17, 2018).

Rickel, Jeff et al. 2001. "Steve Goes to Bosnia: Towards a New Generation of Virtual Humans for Interactive Experiences."

Rouillé, Michaël et al. 2008. *Prototype Du Langage de Modélisation de l'activité Humaine*.

Schwartz, Shalom H. 2017. "The Refined Theory of Basic Values." In *Values and Behavior: Taking a Cross Cultural Perspective*, Cham: Springer International Publishing, 51–72. http://link.springer.com/10.1007/978-3-319-56352-7_3 (February 4, 2019).

Scott, Ezequiel, Alvaro Soria, and Marcelo Campo. 2017. "Adaptive 3D Virtual Learning Environments—A Review of the Literature." *IEEE Transactions on*

Learning Technologies 10(3): 262–76. https://ieeexplore.ieee.org/document/7569100/ (July 4, 2019).

Shafer, Glenn. 1976. 1 *A Mathematical Theory of Evidence*. Princeton university press Princeton.

Skulmowski, Alexander, Andreas Bunge, Kai Kaspar, and Gordon Pipa. 2014. "Forced-Choice Decision-Making in Modified Trolley Dilemma Situations: A Virtual Reality and Eye Tracking Study." *Frontiers in Behavioral Neuroscience* 8(December): 426. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4267265&tool=pmcentrez&rendertype=abstract.

Spierling, Ulrike, and Nicolas Szilas. 2009. "Authoring Issues beyond Tools." In *Interactive Storytelling*, Springer, 50–61.

Sütfeld, Leon R., Richard Gast, Peter König, and Gordon Pipa. 2017. "Using Virtual Reality to Assess Ethical Decisions in Road Traffic Scenarios: Applicability of Value-of-Life-Based Models and Influences of Time Pressure." *Frontiers in Behavioral Neuroscience* 11: 122. http://journal.frontiersin.org/article/10.3389/fnbeh.2017.00122/full (September 29, 2017).

Tardif, Jacques, and Philippe Meirieu. 1996. "Stratégie Pour Favoriser Le Transfert Des Connaissances." *Vie pédagogique* 98(7): 4–7.

Valdesolo, Piercarlo, and David Desteno. 2006. "Manipulations of Emotional Context Shape Moral Judgment." *Psychological Science* 17(6): 476–77.

Vallentyne, Peter. 1989. "Two Types of Moral Dilemmas." *Erkenntnis (1975-)* 30: 301–18. https://www.jstor.org/stable/20012215 (June 29, 2017).

Vanderhaegen, F. 2016. "A Rule-Based Support System for Dissonance Discovery and Control Applied to Car Driving." *Expert Systems with Applications* 65: 361–71. https://www.sciencedirect.com/science/article/pii/S095741741630478X#! (March 12, 2018).

Vanderhaegen, F., and O. Carsten. 2017. "Can Dissonance Engineering Improve Risk Analysis of Human–Machine Systems?" *Cognition, Technology and Work* 19(1).

Vygotsky, L S. 1978. *Mind in Society*. Cambridge, {MA}: Harvard University Press.

Walker, Warren E., Jordan Giddings, and Stuart Armstrong. 2011. "Training and Learning for Crisis Management Using a Virtual Simulation/Gaming Environment." *Cognition, Technology and Work* 13(3): 163–73.

Westrum, Ron. 2006. "A Typology of Resilience Situations." In *Resilience Engineering: Concepts and Percepts*, , 55–65.

Wilcoxon, Frank. 1945. "Individual Comparisons by Ranking Methods." *Biometrics Bulletin* 1(6): 80.