

# Visualization of Patient Behavior from Natural Language Recommendations

Jonathan Siddle  
Teesside University, UK

Alan Lindsay  
Teesside University, UK

João F. Ferreira  
Teesside University, UK  
HASLab/INESC TEC, Universidade  
do Minho, Portugal

Julie Porteous  
Teesside University, UK

Jonathon Read  
Ocado, UK

Fred Charles  
Bournemouth University, UK

Marc Cavazza  
University of Kent, UK

Gersende Georg  
Haute Autorité de Santé, France

## ABSTRACT

The visualization of procedural knowledge from textual documents using 3D animation may be a way to improve understanding. We are interested in applying this approach to documents relating to patient education for *bariatric surgery*: a domain with challenging textual documents describing behavior *recommendations* that contain few procedural steps and leave much commonsense knowledge unspecified. In this work we look at how to automatically capture knowledge from a range of differently phrased recommendations and use that with implicit knowledge about compliance and violation, such that the recommendations can be visualized using 3D animations. Our solution is an end-to-end system that automates this process via: analysis of input recommendations to uncover their conditional structure; the use of commonsense knowledge and deontic logic to generate compliance and violation rules; and mapping of this knowledge to update a default knowledge base, which is used to generate appropriate sequences of visualizations. In this paper we overview this approach and demonstrate its potential.

## CCS CONCEPTS

•**Computing methodologies** → *Nonmonotonic, default reasoning and belief revision; Planning and scheduling;*

## KEYWORDS

NLP, Default Reasoning, Deontic Modality, AI Planning, Serious Games, Interactive Narrative, Virtual Reality

### ACM Reference format:

Jonathan Siddle, Alan Lindsay, João F. Ferreira, Julie Porteous, Jonathon Read, Fred Charles, Marc Cavazza, and Gersende Georg. 2017. Visualization of Patient Behavior from Natural Language Recommendations. In *Proceedings of International Conference on Knowledge Capture, Austin, Texas, Dec 4–6, 2017 (K-CAP 2017)*, 4 pages.

DOI: 10.1145/3148011.3148036

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

K-CAP 2017, Austin, Texas

© 2017 ACM. 978-1-4503-5553-7/17/12...\$15.00

DOI: 10.1145/3148011.3148036

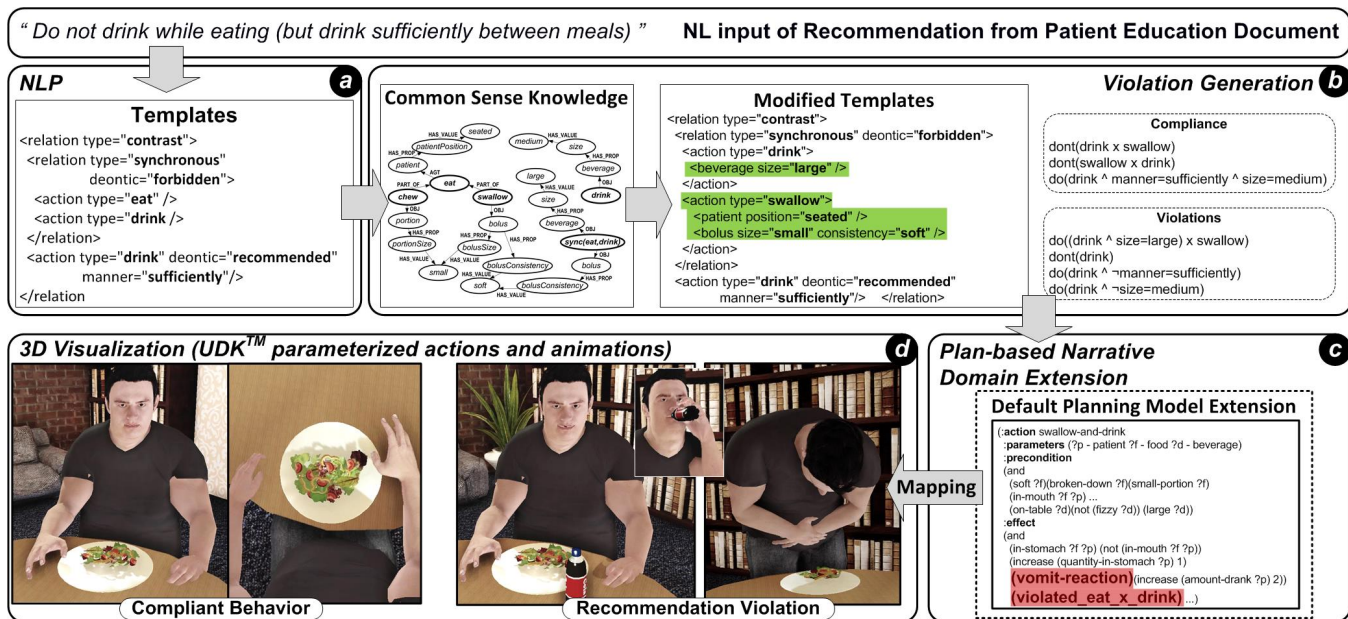
## 1 INTRODUCTION AND RATIONALE

The understanding of instructions from complex documents aimed at a general audience is a widespread problem, and one that can be assisted through the use of 3D animations. Patient education documents are a typical example, as research has suggested that dissemination of information to patients is limited by understanding difficulties. There is growing evidence of the benefits of various visual representations to facilitate patient education, including interactive systems [7], serious games [13] and 3D animations [3].

In this paper we introduce a novel approach based on the automatic generation of 3D animations corresponding to patient education documents, using AI planning based narrative techniques for both representation and generation. One novelty is that, starting with a baseline representation of the application domain, it is possible to single out a specific sentence (e.g., from a patient education leaflet) and automatically analyze it so that the corresponding knowledge will be emphasized in the generated 3D animation.

Our domain of application is patient education after bariatric surgery: a life-changing intervention for severe obesity that involves a permanent modification of the digestive tract and where patients need to permanently modify their eating habits post-surgery by following recommendations (such as “do not drink while eating” in Figure 1). Patient education is thus an essential aspect and is supported by various leaflets and educational documents. In particular we were motivated to explore mechanisms by which natural language (text) recommendations can be clarified through the visualization of compliant, as well as violating behavior. This stems from the need to visualize relevant information when patients are presented with negative instructions (“do not drink while eating”) as part of a recommendation. Visualizing violating behavior has the advantage of presenting implicit information (e.g. on the amount or timing of drinks) and, more importantly, to show the adverse consequences of violation. Because not complying with recommended behavior after bariatric surgery results in severe discomfort, it is essential that patients have understood the recommendations.

In the paper we overview a fully implemented system which generates virtual narratives that illustrate a range of different eating habits for after bariatric surgery. The system integrates multiple AI techniques including Natural Language Processing (NLP), commonsense reasoning and AI planning-based narrative technologies into the visualization engine. In the paper we give an overview of the



**Figure 1: System Overview:** **a** analysis of input NL recommendation to extract desired behavior; **b** use of common sense knowledge and deontic logic to generate compliance and violation rules; **c** rules used to generate new actions to extend default planning model (red highlight); **d** updated planning model used to visualize input recommendation (either compliant or violating scenarios).

key components of the system and then illustrate system behavior via an end-to-end example. We also report on evaluation of user recognition of recommendations from system-generated animations.

## 2 PREVIOUS AND RELATED WORK

There has been significant interest in the use of natural language (NL) input to generate 3D scenes and animations, ranging from automatic scene creation [4] to visualization of complex procedural instructions [14]. Text-to-Scene visualization systems aim primarily at using NL for image and graphics generation [2] or at rendering spatial information conveyed in a text (such as road accidents description [8]). Other systems have aimed at generating 3D animations from NL, driving virtual agent behavior from text. The AnimNL system [14] was dedicated to the visualization of procedural instructions using a virtual agent with the aim of ultimately using instructions to control the behavior of virtual humans. SceneMaker [6] generated animations from film scripts, including some staging elements; this followed work on virtual agent animation from NL [9].

The approach we introduce in this paper, is inspired by early work of Badler et al. [12], but differs by taking advantage of significant progress in the field of plan-based narrative generation and animation [10]. Its main objective is to clarify specific instructions, namely recommendations aimed at patients, by modifying narrative generation so that it emphasizes the recommendations’ contents, if necessary by showing what happens when these are not complied with. At an implementation level, we use Planning Domain Definition Language (PDDL) 3.0 [5] so the formalism can be used for action representation as well as inference and commonsense reasoning. We also use PDDL predicates to ground NL semantics, facilitating the integration of NLP despite limited coverage.

## 3 SYSTEM OVERVIEW

Figure 1 gives an overview of the different components of our system. System input is NL sentences from patient education documents and output is a visualization illustrating this recommendation in the context of a typical meal. There are two modes of operation for the system: (i) illustrating behavior that is compliant with respect to the input recommendation; or (ii) illustrating behaviors that violate it.

For generation of behaviors our approach uses an AI planning formalism. This is because this “eating habits” domain is action-centric which enables patient behaviors to be naturally represented as planning actions. Additionally AI planning can be used to generate multiple different eating behavior scenarios ready for visualization. Our prototype system features a default planning domain model containing key actions, such as food selection, putting food in the mouth, chewing and swallowing along with basic commonsense knowledge taken for granted in the application domain, such as the basic physics of food properties (substance, size, and physical transformations) and how they accumulate. For plan-based scenario generation in our system, an initial state was created that involved a selection of foods, which are appropriate for the recommendations (e.g. soft and small-portion). Since narrative generation requires a complete plan to be produced, a bespoke goal was used to drive plan generation, namely that the patient has completed their meal. This default domain model and planning problem enabled generation of a range of eating habit scenarios.

The key stages of the system, as shown in Figure 1, are as follows: **STAGE (A)**: an input NL recommendation is analyzed to produce a deontic structure in which the target actions of the recommendation are embedded (e.g. eating and drinking for the recommendation “do not drink while eating”). Actions are represented as templates

using feature structures. Processing is based on a cascade of Finite State Transition Networks (FSTNs), the first layer being dedicated to uncovering deontic structures from deontic verbs or imperative statements [1].

**STAGE (B):** the system generates a violation of this recommendation by applying a set of logical rules to the deontic structure resulting in a modification of the recommendation structure. This stage also uses default and commonsense knowledge about the basic physics of eating when “negating” specific actions (i.e. not chewing properly, which requires default knowledge on chewing speed and size of food bites).

**STAGE (C):** the output of the knowledge capture is used to update the default planning domain. The first mapping step selects the most relevant planning operator for each action template taking into account the specific features added during the default reasoning stage. In addition, whenever actions have been “negated” either as part of the recommendation or as part of a violation of the recommendation, this negation is applied to relevant planning operators using rules that modify default parameters.

**STAGE (D):** finally the system generates an eating scenario corresponding to the input recommendation using the extended planning domain (i.e. the set of operators includes the default ones as well as those modified in previous stages to reflect the recommendation). This scenario (a generated plan) is then visualized using animations (UDK™ game engine) attached to planning operators that emphasize the selected recommendation (compliance or violation).

## 4 WORKED EXAMPLE: SYSTEM BEHAVIOR

As an illustration of our system, in this section we work through an end-to-end example, using the following recommendation:

“Sit down to eat your meals and do so in a calm environment”

**STAGE (A)** builds a representation of the text of the recommendation. Firstly, *Deontic Analysis* recognises “Sit down to eat” as the deontic in the imperative form. The different scopes of the sentence along with conjunctions are also recognized: an action is found in the deontic scope; and the remainder of the sentence contains properties related to the action and a conjunction. The second part of this stage is *Action Recognition*, in which a number of FSTNs are run against the sentence to detect actions, slot values, and special instances (e.g. synchronicity). The input sentence is segmented on the conjunction and generates the following segments:

(1) “Sit down to eat your meals and” (2) “do so in a calm environment.”

This process is necessary for action analysis. The issue posed in this example is that there are properties related to eating in both segments of the sentence, but patterns that will cause an eating template to be instantiated only occur in the first segment. The template is instantiated from a trigger FSTN match in the first segment. After the initial action recognition, the template goes through a number of processing stages. For this example, the process is executed on both segments as shown below:

**Segment 1:** “Sit down to eat your meals and”

**Basic templates:** A successful trigger FSTN causes the “eat” template to be instantiated. This is the only concrete template instantiated at this stage.

**Slot Recognition:** This stage identifies the property of being seated attached to the “eat” template.

**Segment 2:** “do so in a calm environment.”

**Basic templates:** No template will be recognized in this segment.

**Slot Recognition:** No action is recognized in this segment, but since an action template was successfully instantiated in the first segment, slot analysis is carried over to the second segment to capture additional properties occurring after the conjunction. Here, the property of “calm environment” is recognized.

The final post-processing step in this stage merges all the information computed into the final action template and the slot values are normalized (e.g. the property “sit down” is transformed to “seated”).

**STAGE (B):** the first step is extending the generated templates with commonsense knowledge. However, this isn’t needed for this recommendation as the commonsense knowledge inferred by the default reasoning engine is already present. The second step is generation of *compliant* and *violation rules*. A single *compliant* rule is extracted from the instantiated action template:

do(eat & position = seated & environment = calm)

*Violations* are automatically generated by implementing a number of rules expressed using dynamic deontic logic (similar to [11]). The rules list possible shapes of recommendations with corresponding sets of violations. For example, the shape of a rule for some action,  $\alpha$  (e.g. “swallow”), which is “forbidden”, is  $F(\alpha)$ , and this has a single corresponding violation  $\{DO(\alpha)\}$ . Using the set of definitions all possible violations are generated. For this example, this is:

dont(eat)

do(eat & NOT position = seated & environment = calm)

do(eat & position = seated & NOT environment = calm)

do(eat & NOT position = seated & NOT environment = calm) \*

**STAGE (C):** a mapping is established between instantiated action templates and actions in the default planning model based on semantic relevance of the default actions to the template, with relevance measured using a bag-of-words approach (BoW): words are normalized using the Porter Stemmer; the BoW for each operator contains the name and precondition predicates; operators are ranked (linearly scaled to reduce bias to smaller BoW); the top ranked operator is selected. For example, in the planning model the conditions on eating, being seated in a calm environment, are encoded in an operator called *begin-meal*. Although the instantiated template is named *eat*, the added context in the BoW leads to the mapping selecting the *begin-meal* operator.

At run-time one or more rules can be selected in the compliance or violation mode and a new planning model is generated in each case. For this example the violation mode is illustrated, using the violation rule marked \* above. Then a plan for this example of compliance/violation is generated using the newly extended domain model.

**STAGE (D):** for output plans the planning operators are mapped to parameterized scripts in UDK™, which modify and select animations and the objects present in the environment. For example, the noisy environment property is mapped onto physical devices in the scene with properties such as volume levels (e.g. TV): thus a quiet environment is an empty room with the TV off, while a noisy environment has multiple agents and high TV volume. Figure 2 shows examples of visualizations: part (a) shows part of a plan for compliant behavior with someone seated in a quiet room whilst eating; in contrast, part (b) shows a busy room.

