An ontology-based, model-driven approach for dialogue simulation underlining communication skills in medical consultation

Abstract

Background: Communication and interpersonal skills for general practitioners in medical consultations play an important role in providing high-quality health care, and in establishing good doctor-patient relationships. Serious games are a good vector to enforce simulation exercises to develop effective communication strategies. In such games the learner plays the role of a health care provider and the patient is simulated by a computer program.

Objective: The purpose of this study is to develop a dialogue engine that can simulate a doctor-patient dialogue during a consultation encounter process.

Methods: We define a model of the medical consultation process and of the patient profile and use ontologies to feed a dialog engine.

Results: The proposed model is called the medical consultation process model (MCPM) and is implemented with an ontology called MSCOnto (Medical Consultation Session Ontology) and described in the Web Ontology Language (OWL) syntax. We explain the architecture and workflow of the dialogue engine and show how the engine can be set up to define different types of patient profiles, and to generate conversations depending on doctor choices.

Conclusion: The model allows the representation of the processes and actors of a medical consultation. Processes are break down into phases, micro-sequences, dialogues session and phrases. Patient profiles and exchanged information can also be expressed. Tests have shown that the presented work lays the foundation of a system able to generate a realistic dialogue between a doctor and a patient.

Keywords
Professional-Patient Relations; medical ontologies; serious games; role playing; ehealth

Introduction

Background

The general practice consultation interview is a process that identifies a disease, a syndrome, or other health problems and health conditions of patients. The doctor-patient relationship plays a very important role in medical practice. As an important factor of the medical consultation interview in general medicine, communication is essential to develop successful doctor-patient relationships [1–6]. An effective doctor-patient communication is essential to the delivery of high-quality health cares, which allows doctors to detect problems earlier, prevent medical crises and expensive interventions, and provide better support to their patients.

As mentioned in the research work of [4,7] during a consultation interview, communication skills are used for two main objectives: “Information Gathering” and “Explanation and shared-decision making”. To be successful, general practitioners (GPs) should encourage patients to express their concerns and ideas, and carry out a list of questions and examinations to acquire relevant information. With the acquired
information, GPs can come out with a diagnosis, a treatment plan or an assessment of the patient’s health. To prevent misunderstandings or to get more detailed and reliable information, GPs need to develop good communication strategies. Furthermore, they should explain to the patient the details of diagnosis and treatment such as their importance and reasons. They also must verify the patients’ understanding and listen to their opinions to reach a shared decision with them. These training programs require expert teachers. Training is most effective when conducted in small groups, in which individual physicians get direct supervision. Learning is more likely to improve physician communication if there is reinforcement over time, in repeated educational sessions.

The purpose of this study is to develop a dialogue engine that can simulate a doctor-patient dialogue during a consultation encounter process. It is based on a domain model of the doctor-patient encounter process described by an ontology.

This study is part of a research project that aims to create and implement a simulation game to teach communication skills to general practitioners.

We should mention that our work focuses only on medical communication skills. Clinical reasoning and therapeutic skills learning are excluded from this study.

**Teaching Communication skills**

Doctor-patient communication is central to clinical practice [8]. It can improve patient satisfaction, understanding and outcomes of care. Effective clinical communication can be defined as the ability to integrate knowledge, communication, physical examination and problem solving [9]. Although clinical communication teaching has become increasingly accepted as a formal component of the medical curriculum, it still often appears to be a peripheral element [10].

However, communications skills can be learned and effective teaching methods have been identified [11–13] that rely on the following components:

- Small group of students or one to one learning,
- Observation of learners,
- Video or audio recording and review,
- Well-intentioned feedback,
- Rehearsal.

For example, the General Health Department of the University of Toulouse is using this method to conduct experiments where medical consultations are video recorded to debrief trainees as well as their directors. However, the exposure of the student to live patients, although required to acquire the necessary skills, can be problematic to provide optimal treatment and to ensure patients’ safety and well-being [14]. Therefore simulation techniques [15,16] are often use to replace real experiences with virtual ones, where learners are immersed in interactive and scripted situations. Role-playing is one of these simulation techniques where educators and students engage in a communication encounter playing the role of patients and doctors [17]. Role-play is usually conducted in small groups to help learners cultivate the skills required to engage in nuanced and often difficult conversations with patients [18].

Serious games (SG) are a good vector to enforce simulation exercises [19]. SG are very popular in the health field [20] and, when used for education, allow a student-oriented approach [21]. Game-based simulation can be distributed to an unlimited number of learners. Game sessions where learners proceed at their own pace can be repeated any number of times. While most educational SG focus on the learning of procedural skills or clinical reasoning [22–24], a few address the development of communication skills [25,26].
The concept of Virtual Patient [27] is central for the design of serious games targeting communication skills. In such games the learner often play the role of a health care provider and the patient is simulated by a computer program [28]. Virtual Patient are basically clinical scenarios played on a computer screen. The learner interacts with the patient (the computer) by typing or selecting actions and the computer supplies response to these actions [29]. These conversational abilities greatly depend on the algorithm used to implement the virtual patient. Most of them are prerecorded copies of the behavior of real standardized patients, while others rely on artificial intelligence methods like conversational agents [30,31].

The purpose of this study is to describe a new method to build a computer program that can simulate a conversation between a user and a virtual patient. This method is based on a model of the medical consultation process and of the patient profile and use ontologies to feed the dialog engine.

Methods

MCPM: A Medical Interview Process Model

Student's handbooks are usually based on the Calgary-Cambridge approach where two types of model are used to describe the medical interview. The first is the framework of information obtained from the patient and used to formulate a diagnosis. This is the content of the medical interview. The second is a process model that describes how doctors should conduct the medical interview [32–36]. As proposed in [37] and [4], and from knowledge and resources gathered from interviews with general practitioners and intern students in medicine, we have chosen to married content and process in designing a model where both types are merged. We called this model the Medical Consultation Process Model (MCPM).

The originality of this model is to allow the representation of fine grain elements like phrases or elementary actions made by the patient or the doctor. This granularity is necessary for the dialogue engine operation that is described later.

The following section presents the main concepts on which is built the MCPM model.

Domain Knowledge Concepts

We present the concepts starting from the smallest elements (the phrase) and finishing with the more general ones (the scenario). They are in that order: Phrase, Dialogue Session, Dialogue Flow, Micro-Sequence, Phase, Information and Scenario.

Phrase

Phrase is a single unit of interaction including a verbal or a non-verbal expression. Non-verbal expressions play a very important role in interpersonal communication, as they can reveal information or help to understand how the patient feels. A non-verbal expression can be a gesture like nodding or a facial expression like a frown. Verbal expressions are simply sentences pronounced by dialogue actors. We used the symbol $\rho$ to represent this concept of Phrase.
Phrase is produced by an actor who is playing a role in the dialogue. We use the symbol $\alpha$ to represent this actor. In the medical consultation interview process, a participant can be a patient, a doctor, an intern student, or someone else.

Phrase has a primitive type that we note $\mathcal{F}$ and that can take 11 different values, classified into four categories: Question, Answer, Statement, and Requirement (Table 1).

### Table 1. Phrase type and their definitions

<table>
<thead>
<tr>
<th>Class</th>
<th>Primitive Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question</td>
<td>Open_Question</td>
<td>A question that cannot be answered by yes or no but that requires a more developed answer.</td>
</tr>
<tr>
<td></td>
<td>Closed_Question</td>
<td>A question that can be answered with either a single word (yes or no) or a short phrase.</td>
</tr>
<tr>
<td></td>
<td>Questioning</td>
<td>A sentence questioning the doctor sentence or action.</td>
</tr>
<tr>
<td>Answer</td>
<td>AnswerWithInfo</td>
<td>An answer to an open question giving one or several information.</td>
</tr>
<tr>
<td></td>
<td>Agree</td>
<td>A positive answer to a closed question or a confirmation.</td>
</tr>
<tr>
<td></td>
<td>Disagree</td>
<td>A negative answer to a closed question.</td>
</tr>
<tr>
<td></td>
<td>DUnderstand</td>
<td>An answer to a “don’t understand” phrase type.</td>
</tr>
<tr>
<td>Statement</td>
<td>Normal Stat</td>
<td>A normal statement</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>A statement to summarize something.</td>
</tr>
<tr>
<td></td>
<td>Explanation</td>
<td>A statement to explain something.</td>
</tr>
<tr>
<td>Requirement</td>
<td>Require</td>
<td>An order</td>
</tr>
</tbody>
</table>

Finally, each Phrase has a vector of three attributes that we note $\chi$, representing the phrase characteristics. In our implementation, we have specified three factors that describe the aggressiveness, clarity and length of the phrase, evaluated by an integer value ranging from 0 to 100. These independent characteristics simplify the complex psychological factors and behavioral profile that affect sentences' styles. Their combination can represent complex elements, like for example the mood and confidence of a patient, or the completeness of information deliverance.

In summary, a Phrase can be characterized by the formula: $PHrase \equiv (\rho: f,)^{\alpha}$

### Dialogue Session and Dialogue Flow

A Dialogue session (DS) is a set of Phrases where different participants exchange ideas about a single topic/purpose. Dialogue sessions are organized in several rounds of exchanges between two participants $P_1$ and $P_2$. A round contains a pair of Phrase $PhraseInit^{P_1}$, $PhraseReply^{P_2}$, where $PhraseInit^{P_1} = (\rho: f,)^{P_1}$ and $PhraseReply^{P_2} = (\rho: f,)^{P_2}$.

Thus, the representation of a Dialogue Session is:
 Dialogue Session ≜ Set\{\textit{Round}\_j\}=\text{Set}\{\langle \text{PhraseInit}^P\_i,\text{PhraseReply}^P\_i\rangle\}_1^l\}

where \(l\) is the number of rounds in the Dialogue Session.

For each round, we define a state that can be:\(\{\text{DU},\text{Q},\text{R},\text{N}\}\), where DU refers to the “Don’t understand” state, Q to the “Questioning” state, R to the “Refuse” state, and N to the “Normal” state. The meaning of these states is explained below:

- **DU**: Don’t understand
  The patient didn’t understand what the doctor has said. The reason for entering this state can be circumstantial, for example if the patient was distracted or the context was too noisy. Perhaps the patient knowledge isn’t enough to understand what the doctor said, or the doctor has used an unclear phrase, etc. Whatever the reason, this state is considered like a state where the patient needs more information.

- **Q**: Questioning
  This state means that the patient has doubts or wants more information. Most of the time, the patient will enter this state in asking a question to verify that he has correctly understood what the doctor just said. Chances to enter this state are high when the patient’s trust level is low. A high value in the aggressiveness characteristic will also increase chances to enter this state.

- **R**: Refuse
  In this state, the patient does not agree with the doctor. For example, the patient will refuse to be examined, or will disagree the doctor’s idea. Like for the questioning state, the reason for entering this state can be a low trust level or a high aggressiveness. The refuse state is considered a more problematic state than the questioning one.

- **N**: Normal
  In this state, the patient will do his best to give information to the doctor by answering his questions (when the doctor phrase is of the “question” type) or will follow the doctor’s instructions when the phrase is of the “request” type. The number and details of the given information will depend of several patients’ characteristics, like his loquacity and actual doctor trust.

In our model, we have considered that for each initial phrase, there is a reply phrase.

Here’s an example of the content of a dialogue session named DS\_a. The topic of the session is “invite to sit” and it’s composed of three rounds:

\[
\text{DS\_a = \{Round}\_1, \text{Round}\_2, \text{Round}\_3\} // @Annotation-Objective: Invite to sit
\]

**Round\_1**: <PhraseInit^P\_1 (“Come in, please sit down”), PhraseReply^P\_1 (“So can I sit here?”)> // @Annotation-state: Q

**Round\_2**: <PhraseInit^P\_2 (“Yes please.”), PhraseReply^P\_2 (“Ok.”)> // @Annotation-state: N

**Round\_3**: <PhraseInit^P\_3 (“Ok.”), PhraseReply^P\_3 (“Very well...”) > // @Annotation-state: N

The beginning of the dialogue session is called the entry point, and the end is called the exit point. One dialogue session can have several exit points determined by the last reply Phrase of the last round. For example, a reply Phrase with the Disagree type will lead to a “refuse” exit point, while a reply Phrase with the “DuUnderstand” type will lead to a “don’t understand” exit point. These kinds of exit points disrupt the normal flow of
conversation. Therefore, they lead to dialogues sessions planned to try to solve communicational deadlocks.

For example, let’s specify a variant of the previous session called DS_b:

\[
\text{DF_inviteSit} = \{ \text{DS}_b, \text{DS}_c \} \quad // \text{@Annotation – dialogue flow}
\]
\[
\text{DS}_b = \{ \text{Round}_1 \} \quad // \text{@Annotation-Objective: Invite to sit}
\]
\[
\text{Round}_1: <\text{PhraseInit}^b \{ \text{"Come in and sit down."} \}, \text{PhraseReply}^p \{ \text{"No, I don’t want to."} \}> \quad // \text{@Annotation-state: R}
\]
\[
\text{DS}_c = \{ \text{Round}_1 \} \quad // \text{@Annotation-Objective: persuade to sit down}
\]
\[
\text{Round}_1: <\text{PhraseInit}^b \{ \text{"Do you want to remain standing the whole consultation?"} \}, \text{PhraseReply}^p \{ \text{"Ok."} \}> \quad // \text{@Annotation-state: N}
\]

This time the exit point of DS_b leads to a new session called DS_c, so we say that the two dialogue sessions are linked. A sequence of several dialogue sessions that accomplish a single purpose is called a Dialogue Flow (DF). In this example, DS_b and DS_c are forming a dialogue flow named df_inviteSit.

In the meantime, a dialogue session can have an entering condition, which prevents the session to start if the condition is not met. For example, the entering condition of the dialogue session about physical examination states that the patient agrees to be examined. Dialogue sessions without an entering condition can be accessed at any time.

**Micro-sequence**

Micro-sequences are the general steps of the medical consultation process. These steps allow GPs to conduct a thorough and logic interview. A micro-sequence (MS) is simply a non-empty set of dialogue flows.

\[
\text{MS} = \{ \text{df} : \text{df} \in \text{MS}, \text{df} \text{ is a dialogue flow} \}
\]

A micro-sequence can occur several times in a given medical consultation or can totally be absent. However, the quality of a consultation process can be assessed by judging the selection of micro-sequences and their orders. This shows whether the process complies to the rules for a good consultation. Furthermore, the decomposition of the medical consultation process in micro-sequences allows experts to define specific challenges for training and education purposes. The player’s performance can then be checked sequence by sequence. Training can also be conducted on a subset of the micro-sequences instead of going through the whole consultation process. We have identified a total of 43 micro-sequences in a typical consultation process, such as history taking or inviting to the physical exam.

**Phase**

Micro-sequences are grouped into containers called Phase. Phase segments the whole consultation process into 4 steps. These steps are: “Initializing phase”, “Information Gathering phase”, “Conclusion phase” and “Ending phase”.

\[
\text{Phase} = \{ \text{ps} : \text{ps} \in \{ \text{Initializing}, \text{Information gathering}, \text{Conclusion}, \text{Ending} \}, \text{ps is a set of ms (micro sequence)} \}
\]
Information

As already mentioned, one major moment in a general practice consultation is patient-related information gathering, especially when the patient is met for the first time [CITATION Bru01 \l 1036]. Based on the acquired information, GPs can express diagnosis judgments and treatment plans, and can explain their conclusions and decisions. Our proposed model divides information into two classes: information obtained from the patient and doctor-inferred information. Doctor’s communication skills are especially useful to get the first type of information. His medical skills are mostly used for elaborating the second one.

Patient-related information is what should be known about the patient at the end of a consultation session. It also consists of two elements: medical related information and non-medical related information.

Medical related information is for example the patient’s medical history, his family medical history and his current signs and symptoms like:

- Patient’s health habits: drinking, smoking, and exercising.
- Patient’s biometrics index.
- Patient’s medical history: diagnosis, treatment, surgery (risk factors).
- Patient’s family medical history (risk factor)
- Patient’s vaccination history.
- Health problems that have been diagnosed and current treatment
- Reason for this medical consultation interview.
- General descriptions of patient’s problems (diseases).
- Symptoms, signs and findings related to patient’s diseases.
- Details of symptoms, signs and findings.

Non-medical related patient information consists of the patient’s administrative information, and personal information like:

- Patient’s name, address, age, telephone number, etc.
- Patient worries.
- Facts about family members, work, lifestyle etc.

We use a single concept to categorize patient related information and we call it “information unit”. Information units can be organized in a hierarchical way. For example, information about the patient “Lifestyle” can be divided into sub-classes specified as “Smoking”, “Drinking” and “Exercising”. More detailed information about the amount and the frequency of drinking can be provided in subclasses.

Each information unit is associated with four elements. One is how the doctor can get the information, for example from questioning the patient, from the patient medical files, or from a physical exam. Another element is the importance of this piece of information, evaluated and predefined by domain experts. Besides this value of importance defined by medical experts, a priority value from the patient point of view is also specified. This element reflects the patient’s will to communicate the information to the doctor. The last element is a threshold value of trust, which reflects the sensitivity of the information for the patient. It determines whether the patient is comfortable to reveal this information. All these values will be used by the dialogue engine, to compute at each step, the amount of information about the patient obtained by the doctor.

Doctor-inferred information is the information owned by the GP. It is the result of deductions made from the patient information, medical knowledge and experience, and can be seen as the outcome of the consultation session. Doctor-inferred information can be categorized as follows:
- Patient’s diagnosis.
- Explanations about the diagnosis.
- Patient’s treatment plan including drugs prescription, goals to achieve, next appointments, etc.
- Explanations about the treatment plan.

Scenario

A scenario is a full instance of the above model where participants and dialogue scripts are instantiated. The scenario can be represented by a finite tree of dialogue flows where each node is a dialogue session. The exploration of this tree forms a dialogue script that represents what happened during the consultation session.

We have identified five main actions that can be performed by the doctor during the consultation process: “Interview” (patient questioning), “Observation” (eventually with a physical examination), “Document Reading” (medical record consultation such as test lab results), “Phone Call” (interruption of the current dialogue session), and “Input Record” (information recording like adding a prescription in the patient medical record). The first three actions provide possibilities for gathering information while “Phone call” and “Input record” are suited to train some specific communication skills, such as managing interruptions or paying attention to the patient while handling another task.

The information units of a scenario can be obtained by one or more actions. For example, the patient’s name can be known by the actions “Interview” and “Document Reading”. On the contrary, symptoms related information can be acquired only through the “Interview” action. Table 2 shows the mapping between information categories and actions required to get this kind of information.

<table>
<thead>
<tr>
<th>Information Category</th>
<th>Action to get this kind of information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrative information</td>
<td>Interview, Document reading</td>
</tr>
<tr>
<td>Life style</td>
<td>Interview, Document reading</td>
</tr>
<tr>
<td>Biometrics Testing</td>
<td>Interview, Document reading, Observation</td>
</tr>
<tr>
<td>Medical history</td>
<td>Interview, Document reading</td>
</tr>
<tr>
<td>Finding</td>
<td>Document reading, Observation</td>
</tr>
<tr>
<td>Sign</td>
<td>Observation</td>
</tr>
<tr>
<td>Symptom</td>
<td>Interview</td>
</tr>
<tr>
<td>Non-Medical Information</td>
<td>Interview</td>
</tr>
</tbody>
</table>

MCPM Model

Based on the previously exposed concepts, we define our MCPM model as:

\[
MCPM = \left[ Info, MS \left| DF \left| \left\{ PhraseInit^P : f, PhraseReply^P : f, i \right\} \right| k; i, j = 1, 2, \ldots, n, i \neq j, k = 1, 2, 3, 4 \right. \right] \\
\]

Where:

- \( Info \) is the set of information included in the interview.
MS | DF | \( \langle \text{PhraseInit}^{P_i}: t, \text{PhraseReply}^{P_j}: t \rangle \) \_k \)

represents the Phrase exchanged within a dialogue flow DF belonging to a micro-sequence in Phase k.

\( P_i \) and \( P_j \) denote the participants of the interview, with \( n \) is the number of participants.

Let's illustrate the model with some scenarios examples. The context of these scenarios will be a diabetic patient who is visiting his doctor for the first time.

Three micro-sequences \( MS_1, MS_2, MS_3 \) are defined to compose the initializing phase (phase 1). Each contains only one dialogue flow, \( DF_1, DF_2, DF_3 \) with respectively the purposes ‘SayHello’, ‘InviteToSit’ and ‘TalkAboutGeneralTopics’. The model representation is:

\[
MCPM = \begin{cases} 
\text{Info}, \\
MS_1 | DF_1 | \langle \text{PhraseInit}^{P_i}: t, \text{PhraseReply}^{P_j}: t \rangle \_1, \\
MS_2 | DF_2 | \langle \text{PhraseInit}^{P_i}: t, \text{PhraseReply}^{P_j}: t \rangle \_1, \\
MS_3 | DF_3 | \langle \text{PhraseInit}^{P_i}: t, \text{PhraseReply}^{P_j}: t \rangle \_1 
\end{cases}
\]

Where \( \text{Info} = \{ i \in I | I = \text{Just moved to the region}, \text{Retired}, \text{Married} \} \)

A scenario based on this model can be express as:

\[
\begin{align*}
\text{DF}_1: & \langle \text{PhraseInit}^{D} \text{["Hello Mr Jones"]}: \text{f=} \text{Statement}, \text{PhraseReply}^{P} \text{["Hello doctor"]}: \text{f=} \text{Statement} >, \langle \text{PhraseInit}^{D} \text{["Follow me please."]}: \text{f=} \text{Require}, \text{PhraseReply}^{P} \text{["Ok."]}: \text{f=} \text{Agree} > \\
\text{DF}_2: & = \{ \langle \text{PhraseInit}^{D} \text{["Please sit down, Mr Jones"]}: \text{f=} \text{Statement}, \text{PhraseReply}^{P} \text{["Thank you"]}: \text{f=} \text{Statement} >, \langle \text{PhraseInit}^{D} \text{["How are you doing?"]}: \text{f=} \text{Closed_Question}, \text{PhraseReply}^{P} \text{["Well, you know, not too bad"]}: \text{f=} \text{Agree} > \\
\text{DF}_3: & = \{ \langle \text{PhraseInit}^{D} \text{["It’s the first time I see you, isn’t it?"]}: \text{f=} \text{Statement}, \text{PhraseReply}^{P} \text{["Yes, I just moved here a month ago. I’m recently retired."]}: \text{f=} \text{AnswerWithInfo} > \\
\end{align*}
\]

And a second one as:

\[
\begin{align*}
\text{DF}_1: & \langle \text{PhraseInit}^{D} \text{["(Hello.")]}: \text{f=} \text{Statement}, \text{PhraseReply}^{P} \text{["Hello doctor."]}: \text{f=} \text{Statement} > \\
\text{DF}_2: & = \{ \langle \text{PhraseInit}^{D} \text{["Take a sit"]}: \text{f=} \text{Statement}, \text{PhraseReply}^{P} \text{["So I’m going to sit here"]}: \text{f=} \text{Statement}, \langle \text{PhraseInit}^{D} \text{["Good, let’s begin”]}: \text{f=} \text{Statement}, \text{PhraseReply}^{P} \text{["Ok."]}: \text{f=} \text{Agree} > \\
\text{DF}_3: & = \{ \langle \text{PhraseInit}^{D} \text{["Are you a newcomer?"]}: \text{f=} \text{Closed_Question}, \text{PhraseReply}^{P} \text{["Yes.”]}: \text{f=} \text{Agree}, \langle \text{PhraseInit}^{D} \text{["Ok, tell me more about you"]}: \text{f=} \text{Require}, \text{PhraseReply}^{P} \text{["Well I don’t know. I…”]}: \text{f=} \text{Disagree} > \\
\end{align*}
\]

In conclusion, our domain model is based on seven concepts: Phrase, Dialogue session (DS), Dialogue Flow (DF), Micro-Sequence (MS), Phase, Information and Scenario. In the next section, we present the construction of an ontology built on these concepts, which implement the proposed model.

MSCOnto Ontology
Overview

Ontology is “hierarchical structuring of knowledge about concepts by sub-classing them according of their properties and qualities” [38]. It forms a declarative model of a domain and represents the concepts of this domain [39]. A single ontology to represent all the concepts exposed in the previous section will be confusing and limited. Therefore, we have developed four ontologies sometimes linked to already defined ontologies. Indeed, there have been a lot of works on knowledge bases about clinical terminology as well as clinical information of patients. For example, resources like the Human Disease Ontology [40], SNOMED [41] and ICD-10 [42] have identified concepts and terms about standardized disease, symptoms, signs and laboratory findings. As mentioned before, our study focuses on the delivery of communication skills, and therefore our own ontologies exclude the medical knowledge needed to make health care decisions but can be linked to some of these research results. The four developed ontologies are the Patient Profile Ontology, the Consultation Result Ontology, the Scenario Ontology and the Phrase Ontology (Figure 1).

The Patient Profile Level Ontology includes most information related to the patient and his health records. The Consultation Result Level Ontology represents the outcome of a medical consultation session, such as the diagnosis, the prescription and the treatment [4]. These two modules define all the information handle during the consultation. The Scenario Level Ontology describes the MCPM structure and elements such as Micro-sequences and Dialogue Flows, and represents a session by defining individuals of the Scenario class. Finally, the Phrase Level Ontology represents the Phrases concept.

In addition to the concepts (class), relations between concepts (object properties) and data properties (attributes of a concept) are defined as well. Some restriction are added to ensure that the model is consistent and logic.

In the following, we present the global structure of each ontologies and give some examples of the more detailed elements.

Figure 1. The global ontological model

Patient Profile Ontology

This ontology is made of several classes that are relevant and necessary to express information about the patient.

In the current version, the Patient Profile Ontology T-Box consists of 38 classes, 20 object properties and 18 datatype properties, where 5 classes come from external
concepts like “Human disease”, “Vaccination”, “Allergy”, “Surgery” or “Treatment”. These classes have the prefix “Ext_”. provides a list of the core classes (concepts) represented by our model. Figure 2 shows the core classes and subclasses represented by our model and the relations between them. Figure 3 shows instance examples as viewed through the OntoGraf user interface of the Protégé software.

Figure 2. Classes, subclasses and relations between classes in the Patient Profile Ontology.

Figure 3. Some instances of the Patient Profile Ontology.

Class axioms and property axioms are also defined to restrict the model and define extra concepts base on existing ones. For example, the axiom defining the concept Family_Medical_History is noted with a DL expression (Descriptive Logic) as shown in Figure 4.
Consultation Result Ontology

The Consultation Result Ontology consists of classes that contain data for different possibilities of diagnosis and treatment plans, as well as explanations for these diagnosis, treatment plan and recommendations (Figure 5).

Its T-Box is made of 5 classes. Like for the Patient Profile Ontology, external concepts are marked with the prefix “Ext_”. This is the case of Ext_Therapy class that represents treatment decisions, where the class Treatment_Plan defines data properties to evaluate several characteristics of a treatment plan, such as its risk, cost and efficiency.

Two separate concepts have been defined to represent a disease: Chronic_disease and Diagnosed_Disease. They are subclasses of the Human_Disease class. This decomposition allows the Chronic_disease class to be part of the Patient Profile Ontology because it describes diseases recorded on the patient’s record. In contrast, the Diagnosed_Disease is a member of the Consultation Result Ontology because it represents the concept of a diagnosed disease as the result of a consultation session.

Scenario Ontology

The Scenario Ontology describes the MCPM structure and its core elements. Domain knowledge concepts like Scenario, Phase, Micro-sequence, Dialogue Flow, and Dialogue Session are represented in this ontology. In addition, other essential elements of a medical consultation session are also included, such as Participants and possible actions for a participant (Figure 6). A complete simulated session can be modeled by defining individuals of the Scenario class.
Figure 6. Classes, subclasses and the relations of the Scenario Ontology.

Let’s note that the Micro-sequence class has a data property “hasPurpose” that describes the goal of the micro-sequence. It also contains information acquired by the doctor and coming from the Patient Profile Ontology, and information inferred by the doctor coming from the Consultation Result Ontology. This distinction is important for the functioning of the dialog engine described later.

Phrase Ontology

This ontology is built to meet the training requirements at the phrase level. Practitioners must know how to use the correct expressions as they conduct a consultation, such as adapting language styles and using appropriate non-verbal expressions. The concepts on this level allow modeling the expressions discussed in section 3.1. Three core classes are defined: Phrase, PhraseCommCategory and PhraseType. They represent the Phrase concept with its characteristic χ and primitive type ℓ. Individuals of the class Phrase can provide several possibilities to express things with the same meaning. Figure 7 shows a general representation of the Phrase Ontology and its relationship with other ontologies.
**Figure 7.** Overview of the Phrase Ontology and links with the Patient Profile Ontology and Scenario Ontology.

Instances of the class Phrase are textual utterances. Figure 8 and Figure 9 give an example of such individuals in the context of a medical consultation where the doctor tries to obtain information about the smoking habits of his patient. The four instances dPh1, pPh1, dPh2, pPh2 compose the "query_smoking_normal" dialogue flow.

**Figure 8.** One instance of the class Phrase: *TopPhrase* is a subclass of the class Phrase.

**Figure 9.** Another instance of the class Phrase: *RefinePhrase* is a subclass of the class Phrase.

In the next section, we present an application using this ontology to implement a dialogue engine between a patient and a doctor. Generated conversations will be used to evaluate the quality of the ontology.
**Dialogue Engine**

The MCPM model has been designed to build a program where a player can play the role of a doctor facing a virtual patient. The heart of this program is a dialogue engine able to organize dialogue sessions stored in the database into a logical conversation scenario.

The engine principle is shown Figure 10. Instances of the MCPM model supply possible actions for the doctor and his patient. Other information for the system is specified as well, such as parameters values for engine functioning. The dialogue engine uses this information to organize dialogue sessions into dialogue flows, which form the whole scenario.

![Figure 10. Proposed system structure](image)

To make its decisions, the algorithm uses a numerical representation of the patient state and a similar representation to evaluate the effect of doctor actions. For each doctor action, a match is made between the current patient state and the effect of the action, to compute the next state of the dialogue session and the evolution of the patient state. These two new states (session and patient) are used to retrieve the information given to the doctor by the patient, and the effective response used by the patient to do that.

To be more specific, five vectors are defined:

- $\vec{PProfileInit}$
  This vector specifies the initial values of the patient traits.

- $\vec{PProfileCoeff}$
  This vector describes the change factors of the patient traits according to the characteristics of phrases used by the physician. It acts as an amplifier and reflects the patient's sensitivity to the doctor's actions.

- $\vec{DActEff}$
  Each physician's action is characterized by this vector that represents the effect of the action on the current state of the patient.

- $\vec{PStatus}$
  This vector represents the current state. It is calculated in applying actions performed by the player with $\vec{DActEff}$ on $\vec{PProfileInit}$ with $\vec{PProfileCoeff}$.

- $\vec{PActC\text{a}r\text{a}c}$
This vector is the equivalent of the vector $\mathbf{DActEff}$ for the patient. It therefore represents the characteristics of the patient’s actions.

The composition of vectors is defined by an expert according to the area covered by the dialogue engine. In our case, we chose to use four values: the patient trust to his doctor, the level of confusion of the patient, his degree of aggressiveness and finally his loquacity.

shows how the next patient action is selected from the doctor's action.

The dialogue engine carries out the phrase selections in a loop that consists of the following five steps:

**Step 1**: The player chooses one of the possible doctor actions. This action is characterized by a $\mathbf{DActEff}$ vector. From this vector, the engine calculates two values:

- The new $\mathbf{PStatus}$ vector value, which represent the patient’s status after the selected action.
- The new session’s state.

**Step 2**: According to the current session state, the dialogue engine identifies all possible actions for the patient.

**Step 3**: The engine chooses a patient’s action by considering the values of the vectors $\mathbf{PStatus}$, $\mathbf{PProfileInit}$ and $\mathbf{PProfileCoeff}$.

**Step 4**: The engine computes the possible actions for the doctor from the new session state of the current micro-sequence.

**Step 5**: If the end condition isn’t met, the system loops to step 1.

A more detailed description of these steps is available in [CITATION guo-2016-serious-games-pour-la-e-santé:-application-à-la-formation-des-médecins-généralistes | 1036].

**Dialogue traces**

The behavior of the dialogue engine has been evaluated in running autonomous simulations where both the patient and the doctor are played by the computer. The scenario used for these experiments is about a male patient in his fifties, diabetic for 8 years, who is coming to visit his doctor to renew a prescription. This scenario has been tested under three difficulty levels, that is three different patient profiles, and three types of doctor actions computed respectively by a best, worst and random choices strategy.

Because our model contains a great number of classes, relations, properties and individuals, it isn’t possible to show all the details of every element of the simulation. Instead we illustrate the whole process in giving a sample of the resulting dialogue traces.

Trace 1: “welcome” micro-sequence with a talkative, confused and aggressive patient, with a low level of understanding.

<table>
<thead>
<tr>
<th>Scenario parameters</th>
<th>Loquacity</th>
<th>Aggressiveness</th>
<th>Confusion</th>
<th>Trust in the doctor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variation coefficient</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>-</td>
</tr>
</tbody>
</table>
Initial Value | 80 | 80 | 80 | 20 |
---|---|---|---|---|
Don't understand state probability | 0.2 | | | |
Refuse state probability | 0.3 | | | |

***** Best choice algorithm *****
Doctor: Hello Madam, I am Dr. Dupont
Patient: Hello doctor.
Doctor: How are you?
Patient: I am fine.
Doctor: Follow me, please sit down.
Patient: (sitting down) You know I have been waiting for more than an hour and... well... here I am now, finally.
Doctor: Yes indeed, I had to spend more time than expected with my previous patient and had an emergency in the afternoon.
Patient: All right.

***** Worst choice Algorithm *****
Doctor: Hello.
Patient: Hello.
Doctor: How are you?
Patient: I am fine.
Doctor: Please sit down.
Patient: What did you say?
Doctor: I said you can sit down.
Patient: (sitting down) You know I have been waiting for more than an hour and... well... here I am now, finally.
Doctor: some patients require more time than others you know....
Patient: All right.

Trace 2: “welcome” micro-sequence with an easy going and talkative patient, trusting his doctor.

<table>
<thead>
<tr>
<th>Scenario parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loquacity</td>
</tr>
<tr>
<td>Variation coefficient</td>
</tr>
<tr>
<td>Initial Value</td>
</tr>
<tr>
<td>Don't understand state probability</td>
</tr>
<tr>
<td>Refuse state probability</td>
</tr>
</tbody>
</table>

***** Best choice algorithm *****
Doctor: Hello Madam, I am Dr. Dupont
Patient: Hello doctor.
Doctor: How are you?
Patient: I am fine.
Doctor: Follow me, please sit down
Patient: Oh, thank you (sitting down) You know I have been waiting for more than an hour and... well... here I am now, finally.
Doctor: Yes indeed, I had to spend more time than expected with my previous patient and had an emergency in the afternoon.
Patient: Ok, I understand...

***** Worst choice algorithm *****
Doctor: Hello.
Patient: Hello doctor.
Doctor: How are you?
Patient: What did you say?
Doctor: I said how are you?
Patient: (smiling) I'm fine thank you.
Doctor: Please sit down.
Patient: (sitting down) thank you. You know I have been waiting for more than an hour and... well... here I am now, finally.
Doctor: Some patients require more time than others you know....
Patient: Yes of course, I understand...

Results
The tables below show some quantitative results of the simulations. For each difficulty level, we have evaluated the quality of the simulation in considering three characteristics of the patient’s status at the end of the session: his confidence in his doctor, his degree of confusion and his aggressiveness. Each value is expressed between 0 and 100. For example, a value of 100 for Trust means that the patient totally trusts his doctor. A value of 80 for Aggressiveness means that the patient is quite aggressive. We have also counted the total number of pairs (exchanged sentences) of the conversation (last column). It’s also interesting to look at the session states obtained during the simulations (Table 6). Results shown are the average values obtained after 100 runs of the micro-sequence “Welcome”.

Table 3. Final patient’s status for the micro-sequence “Welcome” in high difficulty.

<table>
<thead>
<tr>
<th>Average Value</th>
<th>Trust</th>
<th>Confusion</th>
<th>Aggressiveness</th>
<th>Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Value</td>
<td>20</td>
<td>80</td>
<td>80</td>
<td>5</td>
</tr>
<tr>
<td>Best*</td>
<td>52,65</td>
<td>21,74</td>
<td>47,042</td>
<td>5</td>
</tr>
<tr>
<td>Random*</td>
<td>44,4</td>
<td>45,57</td>
<td>58,328</td>
<td>5,87</td>
</tr>
<tr>
<td>Worst*</td>
<td>3,25</td>
<td>100</td>
<td>99,982</td>
<td>7,34</td>
</tr>
</tbody>
</table>


Table 4. Final patient’s status for the micro-sequence “Welcome” in medium difficulty

<table>
<thead>
<tr>
<th>Average Value</th>
<th>Trust</th>
<th>Confusion</th>
<th>Aggressiveness</th>
<th>Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Value</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>Best</td>
<td>80,9</td>
<td>5,95</td>
<td>23,94</td>
<td>4,96</td>
</tr>
<tr>
<td>Random</td>
<td>68,85</td>
<td>30,1</td>
<td>36,98</td>
<td>5,42</td>
</tr>
<tr>
<td>Worst</td>
<td>31,5</td>
<td>97,5</td>
<td>80,15</td>
<td>7,43</td>
</tr>
</tbody>
</table>

Table 5. Final patient’s status for the micro-sequence “Welcome” in low difficulty
<table>
<thead>
<tr>
<th>Average Value</th>
<th>Trust</th>
<th>Confusion</th>
<th>Aggressiveness</th>
<th>Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Value</strong></td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td><strong>Best</strong></td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>4.79</td>
</tr>
<tr>
<td><strong>Random</strong></td>
<td>93.85</td>
<td>2.2</td>
<td>1,034</td>
<td>4.73</td>
</tr>
<tr>
<td><strong>Worst</strong></td>
<td>67.75</td>
<td>27.28</td>
<td>14.96</td>
<td>4.96</td>
</tr>
</tbody>
</table>

**Table 6.** Session states proportion according to the strategy and difficulty level

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Difficulty</th>
<th>Session State</th>
<th>Session Exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best</strong></td>
<td>High</td>
<td>N: 74.10 %</td>
<td>DU: 7.81 %</td>
</tr>
<tr>
<td></td>
<td>Middle</td>
<td>N: 78.83 %</td>
<td>DU: 12.10 %</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>N: 82.46 %</td>
<td>DU: 8.14 %</td>
</tr>
<tr>
<td><strong>Worst</strong></td>
<td>High</td>
<td>N: 40.60 %</td>
<td>DU: 23.43 %</td>
</tr>
<tr>
<td></td>
<td>Middle</td>
<td>N: 32.30 %</td>
<td>DU: 36.88 %</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>N: 78.43 %</td>
<td>DU: 10.89 %</td>
</tr>
<tr>
<td><strong>Random</strong></td>
<td>High</td>
<td>N: 64.91 %</td>
<td>DU: 13.46 %</td>
</tr>
<tr>
<td></td>
<td>Middle</td>
<td>N: 71.22 %</td>
<td>DU: 17.53 %</td>
</tr>
<tr>
<td></td>
<td>Easy</td>
<td>N: 84.36 %</td>
<td>DU: 7.40 %</td>
</tr>
</tbody>
</table>

Through these simulations, our intent was to check how the dialogues were intrinsically linked depending both on the patients' profile and doctor's decision. For the first point, the above results show that, regardless of the player's strategy, there is a strong correlation between the parameters expressing the difficulty level of the game and the value of the chosen indicators (final patient status, number of exchanged sentences, session states distribution). For the patient's status, we observe, for example, that the final value of the properties trust, confusion and aggressiveness, is always related to the difficulty level. This correlation is also obvious when we look at the number of pairs reached in the sessions. This number is indeed always higher in high difficulty simulations. This allows us to conclude that the proposed parameters to represent the patient profile allow to adjust the difficulty level of the session.

Another conclusion is that, for a given difficulty, the choices of the player significantly influence the result of the game. A player making the best choices gets a much better patient's status and a higher rate of normal session states than when he does the worst.

These results are very important because they express the fact that the dialogue engine is properly reacting to changes on two axes: player actions and patient profiles.

**Discussion**

**Principal Results**

The purpose of this study was to present a model of the process of a medical consultation between a patient and a doctor. This model is implemented by ontologies
that are used by a dialogue engine. Our results suggest that this engine can generate conversations with different characteristics between a general practitioner and a patient.

The quality of the engine is evaluated in two ways:

- His ability to generate realistic conversations.
- His adaptability to the player’s actions and to the virtual patient profile.

The first point depends on the ability of medical domain experts to populate the ontology with correct sentences, dialogue sessions, micro-sequences, in other words, to write a story about an interesting medical consultation implementing the various communication methods needed to establish a good relationship with a patient. The proposed ontology has been built to allows the expression of such scenarios.

The second point is very important if the dialogue engine is to be used in a serious game context. How player's actions impact the course of the story? How virtual patient’s profiles challenge players? The tests we have run to answer these questions, gave us encouraging results because they have shown that:

- The quality of the player’s action is linked to the overall quality of the dialogue evaluated through the final patient condition and through the number of bad session states (a bad session state means that something went wrong in the conversation). This means that doctor's actions influence the result of the simulation with a correlation between the relevance of his choices and the quality of the consultation.
- Given a scenario, a difficult patient’s profile is harder to handle than an easier one. This means that the dialogue engine can be parametrized to propose different kind of challenges to the player. A patient’s profile characterizes the difficulty level of the consultation. It is thus possible to offer interesting challenges to both novice and experienced players that will play the role of the doctor.

This preliminary result show that the dialogue engine is sufficiently configurable to enable designers to develop scenarios with an adjustable difficulty level that can fit the learner's profile.

Limits and future works

The main limit of the presented work is that we have developed only one concrete patient profile case through a complete scenario model. Indeed, this creation phase is time consuming because a lot of data are needed to implement a complete medical consultation. The results we have presented are then based on this unique scenario. To allow a greater diversity of scenarios and training programs, the database should be enriched with many more virtual patient profiles and a larger phrase repository. For now, the lack of an authoring tool slows down this work that should map more data from our DUMG (Toulouse College General Medical Department) partner to the databases.

Another limit of our study is that, in the current state of the development of the MCSOnto ontology, a phrase is simplistically modeled with three characteristics: its length, its aggressiveness, and its clarity. On our test scenario, this choice generates realistic dialogues, but things should be refined by working with linguistic experts to get a more accurate phrase model. Because this representation is used by the phrase selection algorithm both for the doctor and the patient, it has a great impact on the behavior of the virtual patient.
Let’s also state, that so far, we have only worked on scenarios that involved one doctor and one patient. Real cases can involve the presence of additional subjects like an intern or a family member. We have not yet investigated the extension of the proposed model to generate dialogs between more than two participants.

**Conclusions**

We have presented the architecture of a simulation system for the medical consultation interview that focus on the development of communication and interpersonal skills for practitioners. This system is based on a model of the medical consultation process implemented by ontologies and on a dialogue engine using this model and able to generate a conversation between a patient and a doctor.

The model allows the representation of the processes and actors of a medical consultation. Processes are break down into phases, micro-sequences, dialogues session and phrases. Patient profiles and exchanged information can also be expressed.

The system is intended to be part of a serious game tool used to improve communications skills of medical students. It extends experimentations led by the Toulouse College General Medical Department where medical consultations are video recorded to debrief trainees as well as their directors. The added serious game aspect will allow learners to play the role of a doctor facing a virtual patient, and to discover and test many situations with the possibility to evaluate the consequences of their decisions.

Tests have shown that the presented work lays the foundation of a system able to generate a realistic dialogue between a doctor and a patient. However, questions about a more precise model of sentences and patient profiles remain. In our prototype, we answered these questions in proposing several characteristics for phrases and patients with independent dimensions. These characteristics should be further examined by linguists, psychologists and doctors because their definition is crucial regarding the validation of skills deployed by the learner.

**Acknowledgments**

The authors thank Dr. Marc Vidal for comments and feedback that greatly improved the medical consultation model. We also thank all the medical interns that have accepted to be filmed for the realization of medical scenarios.

**Conflicts of Interest**

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

**References**


3. Byrne G. Skills for Communicating with Patients Silverman Jonathan, Kurtz Suzanne and


