ExpertCare

Product Name	ExpertCare
Owner	Ilón S.A.
Developers	Dan Rozenfarb, Argentina [1]
	Osvaldo Lanza, Argentina
Smalltalk dialect	Dolphin X6
URL	www.ilon.com.ar/expertcare/
Keywords	Triage, Expert System, Healthcare, Dolphin
License information	Commercial product

Abstract

ExpertCare is a decision support tool for medical telephone triage and ambulance dispatch. ExpertCare analyzes symptoms reported by patients and suggests questions pointing to new symptoms, in order to make a presumptive diagnosis and assess if the call requires an ambulance and/or urgent attention. Besides this primary decision, the system can suggest a particular type of ambulance, medical specialty and an upper limit of arrival time. The application has a knowledge base of syndromes defined in terms of symptoms and a system of rules to determine which questions should be asked.

Context

A few years ago, I met a small healthcare consulting company that wanted to create a system to assist a human operator in a medical phone triage. They had actual experience leading a medical call-center with doctors and advanced medicine students. In that opportunity there was no system, and each operator improvised his questions. Despite written guidelines and training were provided, they couldn't get the operators to act consistently. They even changed their behavior from day to day. With this experience, they decided to go for some automation, or at least a better assistance.

Other forces also drove their search for this kind of systems.

Argentina's health system has some differences compared to what it's found in most countries we know of. One of these is that, mostly in the private segment, there are companies that send physicians to the patient's home, even when not in an urgency situation [ii]. Private companies have a great cost when unnecessarily sending physicians on some green situations where a self care measure would be more adequate. The problem is: how to accurately assess the gravity, and the need of a physician at home. Currently, companies and also public institutions (when have a call-center) just ask a simple question or at most two and take a decision. But statistics show that there is an important mistake rate. In about 30% of the cases, they are sending a resource with a higher complexity, and thus cost, than needed. The worst thing, however, is that there is about 25% of calls that goes undervalued, 60% of which are actually emergencies (15% of the total). This is somewhat coincident with the statistics shown by UK's Direct NHS [iii].

Trees

The temptation for using a tree was there whispering at our ears. From a programmer standpoint it is such an easy algorithm… So we fell on the trap.

Problems arose when trying to fill the tree with actual branches and leaves. The complexity is enormous and the maintainability a nightmare, besides that was almost impossible to make the different specialists to agree on a common interrogatory.

But there are also non technical, but functional limitations that made us change the strategy completely.

Imagine a working decision tree. The patient says *stomach pain*, the system prompt us for *fever*, the patient says *yes, but now that you are asking, I recall that I also vomited blood*. What should we do now? Start again with *blood vomiting* as a new root? Or just ignore it and keep going through the current branch?

Also there is the ordering problem. We must assure that we arrive at precisely the same diagnostic no matter the permutation of the symptoms. Considering we have over 1000 diagnostics and more than 3500 different symptoms, the combinations rules the tree out.

With a tree is quite difficult to follow more than one path simultaneously and combining them. Even worst, typically the patient initially mentions more than one symptom. So what the operator should do? Choose one and start walking the tree? Which one? What if the branch doesn't include the other symptoms? Do we ignore them?

Expert System

This and other problems lead us to implement an expert system.

Essentially we created an application that has a deduction engine, fed with a knowledge base. The engine deduces the next question to be done based on the symptoms entered and also the age, gender and antecedents of the patient. It doesn't require any particular order of the symptoms nor cares about having one or many, given initially or later. When new evidence is entered, it reevaluates everything and generates a new question or else suggests a presumptive diagnostic and a corresponding behavior.

In a nutshell, the knowledge base includes an ontology with diagnostics, symptoms, categories, implications, systems, etc. It also contains a set of deduction rules and action rules that drive system behavior such as the questioning and diagnosing.

The knowledge base can be changed on-line. When a new consultation is created, the system detects the change and loads dynamically the modified knowledge base. It is not needed to retrain the operators, or even notify them.

This is mostly useful under epidemic situations where we can shorten some interrogatories in order to copy with the higher call rate. In this situation, one can even define rules to automatically detect an epidemic situation and change the knowledge base accordingly.

Other features

Support for *Don't know* as a valid answer

You can have rules that predicate on it and take actions such as using alternative questions to obtain similar information.

Tolerates contradictions

Conflict policies can be defined: first input rules, last rules, prompting the operator, etc.

Synonyms for symptoms

Each symptom can have any number of synonyms. These alternative ways of referring to the symptom range for formal to very informal, even mixed languages. Thus the operators don't have to mentally translate the patient saying and can enter it just as told. This made it easy to adapt the software to other regions where alternative symptom names were used.

<u>Different kind of questions</u>

Besides Boolean [iv] we have multiple choice (excluding and not), numerical, etc.

<u>Ambulance dispatch</u>

ExpertCare can be integrated with an ambulance dispatch system which receives patient's info and the suggested diagnostic. It also gets the whole interrogatory and a suggestion of behavior including type of ambulance, upper limit of arrival time, recommended medical speciality, etc. All these suggestions are defined by rules in the knowledge base. This information can also be sent to the chosen ambulance via SMS, email, or some other ways.

<u>Customized advice</u>

One advantage of arriving at a presumptive diagnostic, as opposed to just assessing gravity, is that we are able to give a customized advice to the patient, according not only to the diagnostic but also taking into account the whole context, and her clinical history. The advice is dynamically composed by the expert. This is clearly different to the classical system, where we can only tell the patient "we'll be there in about 60 minutes" or, "go to the nearest hospital", without further instructions.

General architecture

We are dealing with emergencies, and thus best efforts must be done in order to have robustness.

Both the application and the knowledge base (KB) are located in each terminal, and kept synchronized with the master one, located at the application server. However, if the network fails, or the application server, or even the database, ExpertCare saves the consultations locally. When connection is regained, the database is automatically updated. During the failure, the only feature lacking is the ability to bring previous consultations of the same patient.
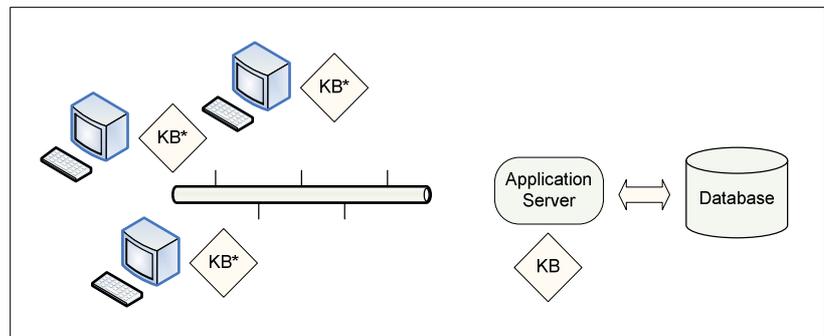
Extensive SUnit tests, a thought out architecture, and Smalltalk [v], all contributed to achieve a very robust system.

Impact on actual people

The system is in production, so it affects actual people's life. Hopefully for good ☺

One of our success stories involves Cordoba state.

State government installed our system in its call-center (35 positions) to cover Cordoba city (second largest in Argentina) and its surroundings. In total, we covered a population of about 1,500,000. The call-center was used to send ambulance, physicians for some green codes, but also to give hospital appointments or self care instructions. ExpertCare brought the best specialists' knowledge [vi] directly to people, some of which with otherwise no access to good medical knowledge. Some isolated towns even don't have doctors handy. But they do have phones, and then access to our system.

We had peaks of 70,000 per month, with a customer satisfaction near to 80% [vii].

Regarding the accuracy of the system, we obtained 85% in gravity code (green, yellow or red) matching physician opinion when arriving at place. Of the remaining 15%, 8% was

over-evaluated. The remaining 7% corresponds to under evaluation. This seems too high. However several aspects must be taken into account: patient missing some relevant info, situation aggravating after the call, difference in interpretation between the physician and the system (bear in mind that is a phone talk, meanwhile the physician actually sees and can touch the patient).

In any case, it is quite an improvement over the previous situation, where almost no question was done, and decision was improvised.

Screenshot (still in Spanish, sorry)
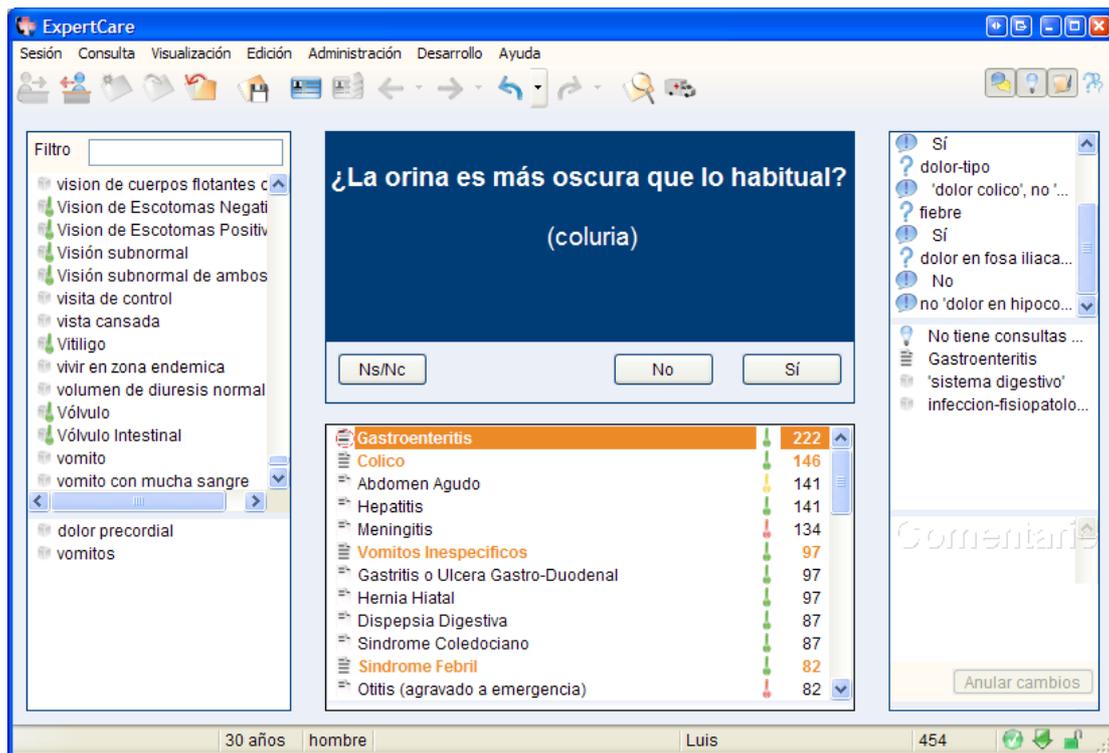
Conclusion

We successfully implemented an expert system in Smalltalk. Loaded with a medicine knowledge base, it is working right now, attending medical consultations of real people.

We are aiming to continue improving our software [viii] and also create other knowledge bases for other vertical markets.

As far as we know, this use of expert systems in medical attention is unprecedented in Argentina and also in Latin America.

---

i With the invaluable contribution of Andrés Taján
ii An international standard labels the emergency situation as red, urgency as yellow and the rest green
iii NHS Direct. NHS Direct: 4 Years On. March 2002.
http://www.nhsdirect.nhs.uk/misc/four_year_report.pdf
iv Well, actually there are three possibilities if we also consider *Don't know*
v Despite its much infamous lack of static type check ☺
vi ExpertCare has been revised by the Medicine Faculty of the University of Buenos Aires
vii Not only with our software, of course, but with the system in general
viii In his master thesis, Carlos Ferro is creating a new interrogatory engine.
He is presenting the results in this ESUG