A Reasoning Module for Distributed Clinical Decision Support Systems

Tiago Oliveira, Ken Satoh, Paulo Novais, José Neves, Pedro Leão, and Hiroshi Hosobe

Abstract One of the main challenges in distributed clinical decision support systems is to ensure that the flow of information is kept. The failure of one or more components should not bring down an entire system. Moreover, it should not impair any decision processes that are taking place in a functioning component. This work describes a decision module that is capable of managing states of incomplete information which result from the failure of communication between components or delays in making the information available. The framework is also capable of generating scenarios for situations in which there are information gaps. The proposal is described through an example about colon cancer staging.

1 Introduction

Computer-Interpretable Guidelines (CIGs) encode medical knowledge and process knowledge in step-by-step algorithms which, in CIG systems, are interpreted by execution engines [10]. They are machine-readable versions of Clinical Practice Guidelines (CPGs) CIG systems are a kind of distributed decision support system that draw information from other information systems, scattered across organizations. This information may be provided by: a human agent, such as a physician

Tiago Oliveira, Paulo Novais and José Neves
Algoritmi Centre/Department of Informatics, University of Minho, Braga, Portugal, e-mail: toliveira@di.uminho.pt

Ken Satoh
National institute of Informatics, Sokendai University, Tokyo, Japan e-mail: ksatoh@nii.ac.jp

Pedro Leão
ICVS/3B's - PT Government Associate Laboratory, Braga/Guimarães, Portugal e-mail: pedroleao@ecsaude.uminho.pt

Hiroshi Hosobe
Department of Digital Media, Hosei University, Tokyo, Japan e-mail: hosobe@hosei.ac.jp

or a nurse, interacting with the system; or extracted from other clinical information systems. However, if there is no knowledge about the state of a patient, it becomes impossible to respond with actions to medical events. This might be due to delays in medical tests or failures in the communication. One of the major issues in these careflow management systems is the occurrence of exceptions or deviations of what is established in the protocols, of which incomplete information is a case. This calls forth the need for higher functions in clinical decision support systems that go beyond the display of information and inference. This work focuses on the development of a Speculative Module for CIG systems that allows them to cope with incomplete information. The decision process treated in the module follows a structure based on a logical framework of Speculative Computation [5], which uses default constraints to advance the computation of decisions. As such, the main contributions of this work are the following: i) a decision module for CIG systems capable of coping with missing information; ii) a default generation method for clinical decision criteria; and iii) a procedure for the generation of clinical scenarios.

2 Related Work

Existing CIG models are mainly task-network models, which means that they organize clinical procedures in networks of mutually dependent tasks. Models such as GLIF3 [2], PROforma [3], and SAGE [13] follow this representation paradigm, although with different foci. While GLIF3 is more focused on the procedural logic of CPGs, PROForma is focused on the argumentation of clinical decisions, and SAGE on the contextual elements of unfolding clinical processes. Yet, the execution engines for these CIG languages only provide straightforward reasoning and do not provide mechanisms to deal with uncertainty [10]. These mechanisms would allow health care professionals to devise scenarios, to anticipate the specific needs of patients, and to mobilize resources beforehand for the clinical tasks ahead . There are techniques derived from probability theory, such as Bayesian Probabilities [7], Certainty Factors [11], Dempster-Shafer Theory [6], and Fuzzy Sets [12], which provide a way to deal with different types of uncertainty and have been used to some extent in the medical domain. However, incomplete information, which is the object of study in this work, deviates from the type of process uncertainty treated by these techniques. In the setting presented herein, the process is established by the procedural logic of CPGs, and a decision module with a logic programming component and a machine learning component is employed to serve as an interface between guideline procedural knowledge and a data-driven default generation method.

3 Setting for Clinical Decision Support

3.1 Clinical Example

The clinical situation used as an example to demonstrate the Speculative Module is that of colon cancer staging and selection of adjuvant therapy. In most cases of

colon cancer, surgery, or more specifically a colectomy, is performed. After surgery, it is necessary to assess the patient based on the most recent findings, which typically occurs in a group appointment of physicians. A decision is made based on the TNM staging system for the classification of malignant tumors, which consists of three criteria: T (tumor), which stands for the degree of tumor invasion into the wall of the colon; N (nodes), the number of metastases in regional lymph nodes; and M (metastasis) represents the detection of distant metastases in other organs such as the liver or the lungs. In Figure 1, the clinical situation described above is represented according to the CompGuide model [9] for CPGs. It is an ontology defined in Ontology Web Language (OWL) following the task-network model. In Figure 1 first there is a *Question* task which aims to obtain the values for the T, N and M parameters. This is a data entry point in the guideline for the retrieval of patient information. This *Question* task is linked to five other tasks through the *hasAlternativeTask* property, which means that these tasks should be executed alternatively to one another, according to the validation of specific trigger conditions, also represented in Figure 1. The tasks are of the *Action* type for they recommend the execution of specific clinical procedures by health care professionals. Depending on the stage of the cancer, it is necessary to decide the best adjuvant therapy, which involves choosing between different observation and chemotherapy schemes. There are cases in which it is difficult to accurately assess the degree of tumor invasion. Furthermore, to know if there are metastases in regional lymph nodes, it is necessary to wait for a pathology report which, in turn, depends on the completion of laboratory tests. The same may happen with the evaluation of distant metastases. For these reasons, there may be cases in which the values for T, N and M are unknown. The whole clinical situation was extracted from the Clinical Practice Guideline in Oncology for Colon Cancer from the National Comprehensive Cancer Network [1]. The medical content is greatly simplified for the sake of conveying the key features of the Speculative Module.

Conditions	T (t)	N (n)	M (m)	Task	Recommendation
A	t0 *or* tis	--	--	Action 1	No adjuvant therapy, colonoscopy in 1 year
B	t1 *or* t2	n0	m0	Action 1	No adjuvant therapy, colonoscopy in 1 year
C	t3	n0	m0	Action 2	Clinical trial or observation or consider capecitabine or 5-FU/leucovorin
D	t4	n0	m0	Action 3	Capecitabine or 5-FU/leucovorin or FOLFOX or CapeOX
E	t1 *or* t2 *or* t3 *or* t4	n1 *or* n2	m0	Action 4	FOLFOX or CapeOX or FLOX
F	t1 *or* t2 *or* t3 *or* t4	n0 *or* n1 *or* n2	m1	Action 5	Colonoscopy, chest and abdominal/pelvic CT, platelets, chemistry profile

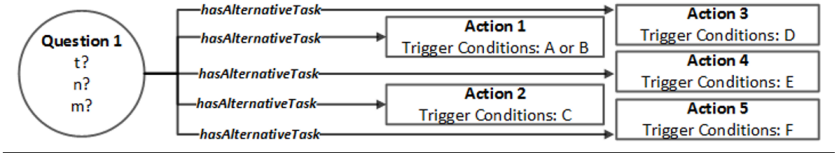

Fig. 1 Clinical setting for the example regarding colon cancer represented according to the CompGuide model.

3.2 Structure of the CIG System

CIG systems have a similar architecture to that of the CompGuide system, represented in Figure 2. The main component is the CIG Engine. It is responsible for executing CIGs encoded in the Knowledge Base, containing situations like the one described in Section 3.1, against patient information retrieved from distributed sources such as other clinical information systems. The system keeps a local repository of previous guideline executions, with patient information fed to the system in previous installments. The CIG Engine is responsible for providing the recommendations about the next clinical task. When there is a decision point, a Speculative Module is deployed to deal with cases of incomplete information. Another cause of incomplete information might be the failure of communication between the CIG system and the information sources. The setting is the following: (1) the CIG Engine will recommend the next task in the clinical careflow; (2) the transition from one task to another is only possible if the first is connected to the latter through the *hasAlternativeTask* property (just like in the example); (3) to move to one of the alternative tasks, the trigger conditions of such task must be met; (4) the information necessary to verify the trigger conditions will be acquired from an external information source, the oncology information system (*ois*) in this case; and (6) the system has a Speculative Module which uses default values to continue the execution and generate clinical scenarios in the event that there is no answer from the (*ois*).

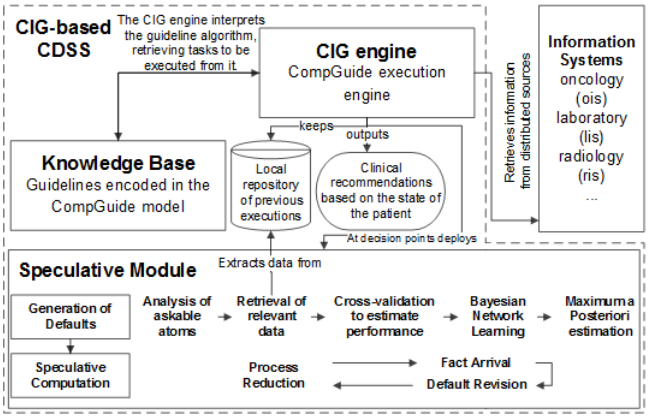

Fig. 2 CompGuide system architecture with its main components: *CIG Engine*, *Knowledge Base*, and *Speculative Module*.

4 Description of the Speculative Module

4.1 Generation of Defaults

The Generation of Defaults is a part of the Speculative Module and consists in a procedure that seeks to acquire the most likely values for the decision parameters,

taking into account possible dependence relationships between them. Bayesian Networks (BNs), for their set of characteristics [15, 14], provide an ideal support for a default generation model. The Generation of Defaults is depicted in Figure 2, inside the Speculative Module. It comprises five sequential stages. The first is the identification of askable atoms, in which the clinical parameters for the decision are identified and isolated. Next, the module retrieves relevant data about previous guideline executions regarding the isolated parameters from the local repository. In the following stage, six different BN learning algorithms are used in cross-validation. The purpose is to select the one with the lowest log likelihood loss (*logl*), which is a measure of the entropy exported by a model in order to keep its own entropy low [4]. So, low values are synonymous with a good fit between the model and the data. After the best performing algorithm is selected, a BN is generated. The last stage consists in a Maximum a Posteriori (*MAP*) estimation, which is a query to the BN that provides the most likely values for the parameters, given the evidence, along with a probability value [8]. It is possible to perform a MAP query with no evidence, which provides the most likely configuration for all the parameters in the network. Figure 3 shows the results of the procedure when applied to data of 515 patients who received treatment for colon cancer at the Hospital of Braga, Portugal. In Figure 3 (a), it is possible to see that the algorithm that produces the lowest *logl* is the *iamb*. Figure 3 (b) also shows the structures learned from the different algorithms. In the chosen structure (the one learned with the *iamb*) it was only possible to establish a dependence relationship between T and N. This is particularly useful for it indicates to the CIG Engine that tumor invasion and local lymph nodes metastases might be correlated, but distant metastases are not dependent on the other two parameters. After performing the MAP estimation on the network, the values obtained for T, N, and M were respectively $T = t3, N = n2$, and $M = m1$, with $T, N, M_{MAP} \approx 0.4395$. These values are used as initial defaults in Speculative Computation. MAP queries will be performed as information from the sources arrive in order to adjust the remaining default values. The Generation of Defaults was developed using the *bnlearn* package for R and the *inflib* Java library for inference in BNs from the SamIam project.

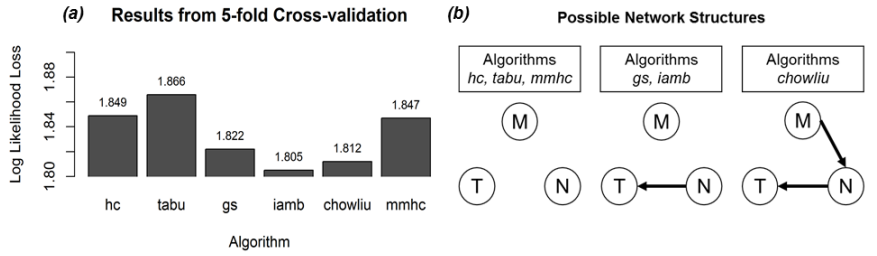

Fig. 3 (a) Results from 5-fold cross-validation of six structure learning algorithms; (b) Bayesian Network structures for the TNM parameters learned with the different algorithms.

4.2 Speculative Computation

The Speculative Computation part of the Speculative Module is based on a logic programming framework with constraint processing.

4.2.1 Formalization of the Clinical Example According to the Framework

A Framework for Speculative Computation in clinical decision support systems which features disjunctive constraint processing is a tuple $\langle \Sigma, \mathscr{E}, \Delta, \mathscr{P} \rangle$. This formulation is based on the work presented in [5]. Σ is a finite set of constants. An element in Σ is a system component identifier, an information source with which the CIG system communicates. $\mathscr{E}$ is a set of predicates called external predicates, representing the decision criteria, i.e., the clinical parameters in the trigger conditions. When Q is an atom with an external predicate and S is the identifier of a remote information source, $Q@S$ is called an askable atom. Δ is the default answer set, consisting of set of default rules, obtained from the Generation of Defaults, called default rules with respect to $Q@S$, of the following form: $Q@S \leftarrow C \parallel$, where $Q@S$ is an askable atom and C is a set of constraints called default constraints for $Q@S$. As for $\mathscr{P}$, it is a constraint logic program of the form: $H \leftarrow C \parallel B_1, B_2, \ldots, B_n.$, where H is a positive ordinary literal called a head of rule R; C is a set of constraints; and each $B_1, B_2, \ldots, B_n$ is an ordinary literal, or an askable literal. The situation described in Section 3.1 regarding the staging of colon cancer can be represented according to the framework. The predicate $nt(a,b)$ indicates that b is the task that follows a and is used in the initial query in order to know what procedure should be performed next. $alt(a,b)$ indicates that b is an alternative task linked to a. $tcv(b)$ means that the trigger conditions for task b are validated. The complete representation is:

- $\Sigma = \{ois\}$
- $\mathscr{E} = \{t, n, m\}$
- Δ is the following set of rules: $t(T)@ois \leftarrow T \in \{t3\} \parallel .$ $\quad n(N)@ois \leftarrow N \in \{n2\} \parallel .$ $\quad m(M)@ois \leftarrow M \in \{m1\} \parallel .$
- $\mathscr{P}$ is the following set of rules:
  $nt(X,F) \leftarrow \parallel alt(X,F), tcv(F).^{rule\ 1}$
  $tcv(F) \leftarrow F \in \{action1\}, T \in \{tis, t0\} \parallel t(T)@ois.^{rule\ 2}$
  $tcv(F) \leftarrow F \in \{action1\}, T \in \{t1, t2\}, N \in \{n0\}, M \in \{m0\} \parallel t(T)@ois, n(N)@ois, m(M)@ois.^{rule\ 3}$
  $tcv(F) \leftarrow F \in \{action2\}, T \in \{t3\}, N \in \{n0\}, M \in \{m0\} \parallel t(T)@ois, n(N)@ois, m(M)@ois.^{rule\ 4}$
  $tcv(F) \leftarrow F \in \{action3\}, T \in \{t4\}, N \in \{n0\}, M \in \{m0\} \parallel t(T)@ois, n(N)@ois, m(M)@ois.^{rule\ 5}$
  $tcv(F) \leftarrow F \in \{action4\}, T \in \{t1, t2, t3, t4\}, N \in \{n1, n2\}, M \in \{m0\} \parallel t(T)@ois, n(N)@ois, m(M)@ois.^{rule\ 6}$
  $tcv(F) \leftarrow F \in \{action5\}, T \in \{t1, t2, t3, t4\}, N \in \{n0, n1, n2\}, M \in \{m1\} \parallel t(T)@ois, n(N)@ois, m(M)@ois.^{rule\ 7}$
  $alt(question1, F) \leftarrow F \in \{action1\} \parallel .$ $\quad alt(question1, F) \leftarrow F \in \{action2\} \parallel .$
  $alt(question1, F) \leftarrow F \in \{action3\} \parallel .$ $\quad alt(question1, F) \leftarrow F \in \{action4\} \parallel .$
  $alt(question1, F) \leftarrow F \in \{action5\} \parallel .$

Speculative Computation starts with a top goal. The notion of goal is central for it represents what is necessary to achieve in the computation, or, in other words, it is the outcome of the decision. In the framework, a goal has the form of $\leftarrow C \parallel B_1, \ldots, B_n$ where C is a set of constraints and each of $B_1, \ldots, B_n$ is either an askable atom or an atom that is unifiable with the head of a rule in $\mathscr{P}$. During the computation the framework keeps a set of beliefs about the askable atoms, i.e., the clinical parameters in the decision, in a current belief state (*CBS*). In the beginning, the *CBS* assumes the default rules in Δ and, afterwards, the top goal is

reduced according to *CBS* and $\mathscr{P}$. Goals and the product of their reduction are kept in processes. They are structures that represent the different alternative computations in the framework. In this regard, there are two types of processes: active and suspended. Active processes are those whose constraints C are contained in, and therefore are consistent with, the *CBS*. They are regarded as the valid scenarios. Processes that do not fulfill this condition are suspended. In order to keep track of the different processes, active processes are kept in the active process set (*APS*), whereas suspended processes are kept in the suspended process set (*SPS*).

4.2.2 Phases of Speculative Computation

Speculative Computation includes: the *process reduction* phase, the *fact arrival* phase, and the *default revision* phase. Throughout these phases active processes are represented according to the tuple $\langle \leftarrow C \parallel GS, UD \rangle$, where C is a set of constraints, GS contains the goals in the form of literals to be proved, and UD is the set of used defaults. If an atom is reduced through a default, it is added to the UD of the process. Suspended processes have a similar structure, they are represented by a tuple $\langle SP, \leftarrow C \parallel GS, UD \rangle$, where SP is a set containing the atoms that were responsible for suspending the process and whose constraints are not consistent with the *CBS*.

The first phase is *process reduction*, which is depicted in Figure 4 (a). The initial query to the system becomes the initial goal set in the first active process. The process is further reduced according to its goals. If an atom in the goal set is unifiable with a rule in $\mathscr{P}$ then that atom is replaced in GS with the body of the rule, and the constraints of the rule are added to the constraints of the process. This originates as many active processes as there are rules with which the atom in GS is unifiable. This ensures that different execution traces are explored, keeping all possible scenarios open. But, if the atom in the goal is an askable atom $Q@S$, with and attached constraint C, for which there is a default constraint C_d, the module verifies the consistency of $C \wedge C_d$ and $C \wedge \neg C_d$. If $C \wedge C_d$ is consistent, then a new active process is created from the initial process with this constraint, and $Q@\mathrm{S}$ is added to UD. If the same happens with $C \wedge \neg C_d$, a suspended process is created as well, and $Q@S$ is added to SP. With this disjoint constraint processing, the objective is not to exclude any possible scenario and keep active only the processes that are consistent with the *CBS*. While this takes place, the real value of the atom is asked from the information source. These procedures are applied to active processes until they have an empty goal set, which means that there is nothing left to prove and they are valid scenarios according to the default constraints. *Fact arrival* occurs when a constraint $Q_r@S \leftarrow C_r \parallel$ is returned from an information source S for a question regarding an askable atom $Q_r@S$. After a step of process reduction is finished, the new constraint replaces the default in the *CBS* for the corresponding askable atom. It is then necessary to revise the processes in *APS* and *SPS* that have $Q_r@S$ either as a used default (in UD) or a suspended atom (in SP). The revision of constraints occurs as shown in Figure 4 (b). Since the replies from the information sources are facts and, thus, regarded as definitive, there are execution traces that are deleted when they are inconsistent with the *CBS*. The objective with this phase is to remove

scenarios that can no longer occur. The *default revision* phase consists in changing all the processes according to the new default rules provided by the method for the Generation of Defaults. The arrival of a new fact may change the remaining default rules. As such, a new MAP estimation is done using the BN with the available facts as evidence. For every changed default, *default revision* is performed. Processes in *APS* and *SPS* are revised in order to determine if their constraints are consistent with the new default constraint $Q_d@S \leftarrow C_{newd} \parallel$. According to the position of the atom attached to the new default, i.e., if it is a suspended atom (in *SP*) or a used default (in UD), the existing processes may generate new active and suspended processes, but their execution trace is never erased, unlike what happens in the *fact arrival* phase. The revision of processes takes place as shown in Figure 4 (c). This keeps the scenarios coherent and increasingly aligned with the newly arrived facts. It is a dynamic mechanism because it is triggered by changes in the default rules. These changes, in turn, are caused by the arrival of facts from the information sources. The treatment of processes in all of these three phases is implemented in Prolog.

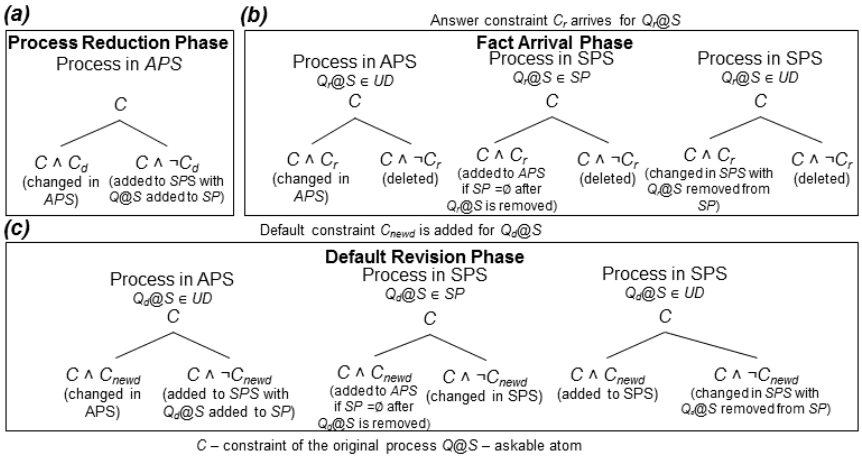

Fig. 4 Diagram of Speculative Computation. Each bifurcation represents two new processes with the displayed constraints.

4.3 Example and Discussion

According to the situation described in Section 3.1 and the formalization in Section 4.2.1, the question to pose to the Speculative Module would be $nt(question1, F)$, which seeks to determine which task should follow *Question1* in the management of the patient. The initial *CBS* assumes the values in Δ. Then, process reduction occurs with the unification of the goals in the *GS* with the head of rules and facts in $\mathscr{P}$. This will happen through rules 1 and 2. By the time process reduction reaches rules 3, 4, 5, 6, and 7, one will have five active processes representing the five possible alternative tasks. At this point, it becomes necessary to reduce askable atoms. For instance, if the system needs a value for the goal $t(T)@ois$, since there is only a default constraint for that goal in the *CBS*, that constraint

is used in process reduction, yielding active processes which are consistent with the default and suspended processes which are not. As such, a process representing *action5* is split into two processes, an active process using the default constraint $t(T)@ois \leftarrow T \in \{t3\} \parallel$, and a suspended process using the negation of the default constraint. Meanwhile, a question is sent to *ois* so as to know the real value of $t(T)@ois$. The procedure is similar whenever an askable atom is found. By continuing process reduction, it is possible to arrive at a state in which there is a process with an empty goal set, achieved purely by relying on default constraints. As such, the process is a tentative scenario. By outputting constraints C and used defaults UD, the system provides the task that will most likely follow *question1*. In this case $C = \{F \in \{action5\}, T \in \{t3\}, N \in \{n2\}, M \in \{m1\}\}$ and $UD = \{t(T)@ois, n(N)@ois, m(M)@ois\}$, which means that *action5* should be the next task, featuring a series of workup exams such as a colonoscopy, an abdominal/pelvic CT, and so forth. Based on this, the health care professional may start preparations. There might be cases in which more than one process with an empty goal set are produced. In such situations, both are presented as scenarios.

In *fact arrival*, answers from the *ois* trigger the revision of active and suspended processes. Assuming that the fact $(n(N)@ois \leftarrow N \in \{n0\} \parallel)$ arrives, it becomes the new rule for $n(N)@ois$ in the *CBS*, and all the processes consistent with this and the remaining *CBS* become active, whereas the others are deleted. Additionally, a new MAP estimation is submitted to the BN with the new fact as evidence. It produces the result $T, M_{MAP} = argmax_{T,M}\, P(T, M \mid N = n0) \approx 0.9126$ with $T = t1$ and $M = m1$. The default value for $t(T)@ois$ is the only one that changes. Through *default revision*, $t(T)@ois \leftarrow T \in \{t1\} \parallel$ replaces the old default in the *CBS* and all the processes are revised accordingly, but no execution trace is deleted. In the end of these steps there is an active process representing the most likely scenario given the configuration of the information, achieved through fact and default constraints. Outputting constraints and used defaults would give $C = \{\leftarrow F \in \{action5\}, T \in \{t1\}, N \in \{n0\}, M \in \{m1\}\}$ and $UD = \{t(T)@ois, m(M)@ois\}$. Curiously, the scenario recommends the same task, but achieved through a different route. In this case, the new default is included in the process and $n(N)@ois$ in no longer in UD because it is not a default anymore. As information arrives for the other askable atoms, these procedures are repeated. The Speculative Module provides a filter for rule-based clinical decision support systems, managing different information states.

5 Conclusions and Future Work

Speculative Computation provides tentative scenarios for the state of a patient and, based on them, the most appropriate clinical task. Speculative Computation is used as an interface to clinical algorithms, bringing to clinical decisions a set of mechanisms to reduce and revise scenarios according to facts and newly derived defaults. As such, they ensure a convergence of all the scenarios towards the real state of the patient. This kind of mechanism does not have a parallel in other CIG systems. The computation is dynamic in the sense that it accounts for changes in the original beliefs. As future work, it is necessary to incorporate the probability provided by

the MAP estimation into Speculative Computation so as to provide a measure of the likelihood of the processes holding true.

Acknowledgements This work is part-funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-028980 and project scope UID/CEC/00319/2013.The work of Tiago Oliveira is supported by a FCT grant with the reference SFRH/BD/85291/ 2012.

References

1. Benson, A., Bekaii-Saab, T., Chan, E., Chen, Y.J., Choti, M., Cooper, H., Engstrom, P.: NCCN Clinical Practice Guideline in Oncology Colon Cancer. Tech. rep., National Comprehensive Cancer Network
2. Boxwala, A.A., Peleg, M., Tu, S., Ogunyemi, O., Zeng, Q.T., Wang, D., Patel, V.L., Greenes, R.A., Shortliffe, E.H.: GLIF3: A Representation Format for Sharable Computer-Interpretable Clinical Practice Guidelines. Journal of Biomedical Informatics **37**(3), 147–161 (2004)
3. Fox, J., Ma, R.T.: Decision Support for Health Care : The PROforma Evidence Base. Informatics in Primary Care **14**(1), 49–54 (2006)
4. Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., Tibshirani, R.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2 edn. Springer-Verlag, New York (2009)
5. Hosobe, H., Satoh, K., Codognet, P.: Agent-Based Speculative Constraint Processing. IEICE Transactions on Information and Systems **E90-D**(9), 1354–1362 (2007)
6. Hua, Z., Gong, B., Xu, X.: A dsahp approach for multi-attribute decision making problem with incomplete information. Expert Systems with Applications **34**(3), 2221 – 2227 (2008)
7. Kononenko, I.: Inductive and bayesian learning in medical diagnosis. Applied Artificial Intelligence **7**(4), 317–337 (1993)
8. Korb, K., Nicholson, A.: Bayesian Artificial Intelligence, 2 edn. CRC Press, London (2003)
9. Oliveira, T., Novais, P., Neves, J.: Representation of clinical practice guideline components in owl. In: Trends in Practical Applications of Agents and Multiagent Systems, *Advances in Intelligent Systems and Computing*, vol. 221, pp. 77–85. Springer (2013)
10. Peleg, M.: Computer-Interpretable Clinical Guidelines: A Methodological Review. Journal of Biomedical Informatics **46**(4), 744–63 (2013)
11. Shortliffe, E.H., Davis, R., Axline, S.G., Buchanan, B.G., Green, C., Cohen, S.N.: Computer-based consultations in clinical therapeutics: Explanation and rule acquisition capabilities of the mycin system. Computers and Biomedical Research **8**(4), 303 – 320 (1975)
12. Straszecka, E.: Combining Uncertainty and Imprecision in Models of Medical Diagnosis. Information Sciences **176**(20), 3026–3059 (2006)
13. Tu, S.W., Campbell, J.R., Glasgow, J., Nyman, M.a., McClure, R., McClay, J., Parker, C., Hrabak, K.M., Berg, D., Weida, T., Mansfield, J.G., Musen, M.a., Abarbanel, R.M.: The SAGE Guideline Model: achievements and overview. Journal of the American Medical Informatics Association : JAMIA **14**(5), 589–98 (2007)
14. Van der Heijden, M., Lucas, P.J.F.: Describing Disease Processes Using a Probabilistic Logic of Qualitative time. Artificial Intelligence in Medicine **59**(3), 143–155 (2013)
15. Visscher, S., Lucas, P.J.F., Schurink, C.A.M., Bonten, M.J.M.: Modelling Treatment Effects in a Clinical Bayesian Network Using Boolean Threshold Functions. Artificial Intelligence in Medicine **46**(3), 251–266 (2009)