

Progress in Artificial Intelligence manuscript No.
(will be inserted by the editor)

Moving Towards a New Paradigm of Creation, Dissemination, and Application of Computer-Interpretable Medical Knowledge

Paulo Novais · Tiago Oliveira · José Neves

Received: date / Accepted: date

Abstract Computer-Interpretable Guidelines (CIGs) exploit the scientific strength of evidence-based medicine to make recommendations available in Clinical Decision Support Systems. However, systems that deploy them have not been widely successful, in part due to the limitations of CIG frameworks in the adoption of inclusive and open technologies and the use of Artificial Intelligence techniques as tools to make their systems stronger and more adaptable. In this work we propose a web-based CIG framework to tackle some of these challenges and facilitate the integration of CIG-based advice not only in the everyday activities of health care professionals but also in the lives of whoever may need it.

Keywords Clinical Decision Support Systems · Computer-Interpretable Guidelines · Computer-Assisted Decision Making · Ambient Assisted Living

Paulo Novais
Algoritmi Center
Department of Informatics
Universidade do Minho
Braga, Portugal Tel.: +351 253 604 437
Fax: +351 253 604 471
E-mail: pjon@di.uminho.pt

Tiago Oliveira
Algoritmi Center
Department of Informatics
Universidade do Minho
Braga, Portugal Tel.: +351 253 604 437
Fax: +351 253 604 471
E-mail: toliveira@di.uminho.pt

José Neves
Algoritmi Center
Department of Informatics
Universidade do Minho
Braga, Portugal Tel.: +351 253 604 437
Fax: +351 253 604 471
E-mail: jneves@di.uminho.pt

1 Introduction

Clinical Decision Support Systems (CDSSs) are one of the most prominent examples of the role Medical Informatics may have in the improvement of health care (Musen et al 2006). The ability to provide advice at different moments and in various situations of the clinical process, may it be to a physician, a nurse, or an assistant caregiver, tailored to the condition of a specific patient, has the potential to significantly improve the quality of care delivery. Although this is not guaranteed, there is increasing evidence of the positive effects CDSSs may have on aspects such as medication safety (Kaushal et al 2003), practitioner performance (Adhikari et al 2005), and decision quality (Sintchenko et al 2004).

CDSSs may perform a wide variety of roles, depending on the situation for which they were conceived. Yet it is not an easy task to build systems capable of responding to all of the different moments of the clinical process, from diagnosis and treatment to follow-up. Ultimately, this ability of CDSSs to accompany the different stages of handling a patient depends greatly on the type of support they use for their knowledge base and whether that support is capable of modelling those different situations accurately. Computer-Interpretable Guideline (CIG) (de Clercq et al 2004; Oliveira et al 2013a; Peleg 2013) systems are regarded as the best choice when it comes to supporting health care professionals in their practice, not only for their reliance on Clinical Practice Guidelines (CPGs) for the recommendations, which is solid from the medical point of view, but also for their adaptability to the different stages of patient management. CPGs are systematically developed statements, based on the best available medical evidence, that provide recommendations

to health care professionals for specific clinical circumstances, and are developed for a wide variety of categories (e.g., diagnosis, management, treatment, follow-up, etc) and specialities (e.g., general practice, oncology, cardiology, etc) (Miller and Kearney 2004). CIGs, as machine-interpretable structured versions of CPGs, bring that knowledge diversity to CDSSs, along with the ability to act in different settings (Latoszek-Berendsen et al 2010).

The development of CIG technologies took off in the mid to late 1990s (Ten Teije et al 2008), with the emergence of numerous modelling languages and tools that went along with them and deal with the basic aspects of CIG development, namely (de Clercq et al 2004):

- **Creation and Editing:** this aspect refers to the representation of CPGs using a specific modelling language, the construction of clinical algorithms and the ability to change them as needed;
- **Dissemination:** this aspect is related with the ability to spread CIGs across different institutions and professionals, to make them easily available and deployable;
- **Execution:** this aspect refers to the computer-based execution of CIGs and deals with the ways in which CPGs can be automatically applied to patients and integrated in the daily practice of health care professionals.

However, CIG development and implementation has not quite fulfilled its anticipated potential just yet. Most of the existing projects in this area are still at the academic level, with little implementation in real world settings. There are but a few implemented case studies that prove the feasibility of the approaches rather than their actual contribution to the improvement of health care outcomes. Moreover, most projects do not show the openness nor integrate the technologies that would allow them to have a wider reach in the health care domain.

In this paper we provide an overview of the current state of the art of CIG development regarding the three above-mentioned aspects and identify which points should be improved for the further advancement of computer-interpretable medical knowledge. This is mostly done in section 2. Then, after these points are clearly identified, we propose a CIG framework in section 3 that aims to provide solutions for the identified issues and discuss their implications in CIG creation, editing, dissemination and execution. Finally, in section 4.1 we make some concluding remarks regarding the proposal and point future directions for CIG development.

2 Trends in Computer-Interpretable Guideline Development

Table 1 shows a summary of the main CIG approaches headlined by the models that give them shape, namely Arden Syntax (Kim et al 2008), GLIF3 (Boxwala et al 2004), PROforma (Button and Fox 2003), GLARE (Terenziani et al 2004), SAGE (Tu et al 2007), and NewGuide (Ciccaresea et al 2004). While there are other important CIG approaches, these selected few follow similar guiding principles that make them relevant for this discussion.

2.1 Guideline Creation and Editing

Most CIG models represent CPGs as networks of intertwined basic units which can assume various forms, namely *modules*, *steps*, *tasks*, *roles*, *actions*, *nodes* or *recommendations*, as summarized in the basic elements of Table 1. While this form of task network model offers greater flexibility in the definition of clinical careflows by allowing one to display the respective building blocks as needed, it is also true that the existing models hardly converge when it comes to the meaning of their basic units. To give an example, it is easy to establish a parallel between *step* and *task*, but a relationship is not as easily established between either of the previous two and the concepts of *role* or *node*. Furthermore, it is possible to notice a tendency of the existing models to provide uneven attention to different aspects of CPG representation, of which the representation of temporal constraints is a clear example. In Table 1, it is possible to observe that the completeness of their temporal representation, defined as the number of different temporal patterns they include, differs greatly, with arguably the most complete being GLARE (Terenziani et al 2004), which does not even include temporal constraints about the state of the patient. Thus there is a clear need for an encompassing guideline model based on a common framework, whose syntax and semantics are widely known, that is expressive enough to allow the representation of the basic units of CPGs and the relations between them.

Tools such as Arezzo (Isern and Moreno 2008), the SAGE Workbench (Tu et al 2007), or the GLARE Acquisition Module (Terenziani et al 2004) use a graph-like representation of the clinical careflow with the possibility of dynamically adding the building blocks necessary for the definition of clinical protocols. This form of guideline creation and editing, although with different degrees of customization, is followed by most CIG frameworks and enables the users to have a better control over the objects they are manipulating and a bet-

Table 1 Summary of the main CIG approaches, featuring the basic elements of their models, their underlying support format, and their main tools.

| CIG Model | Guideline Basic Elements | Representation of Temporal Constraints | Support Language | Creation and Editing Tool | Execution Engine |
|--|---|---|------------------|---------------------------|-----------------------------------|
| Arden Syntax | Medical Logic Modules (encoding of one decision) | Durations, waiting times | XML | Arden Syntax IDE | Arden Syntax Rule Engine |
| Guideline Interchange Format (GLIF3) | Steps: decision, action, patient state, branch, synchronization | Durations, patient state | RDF | Protégé | Guideline Execution Engine (GLEE) |
| PROforma | Tasks: plan, action, decision, enquiry | Durations, repetitions, waiting times, repetition conditions | XML | Arezzo | Arezzo, Tallis |
| Asbru | Roles: preferences, intentions, conditions, effects, plan body | Durations, repetitions, periodicities, waiting times, patient state | XML | AsbruView | Spock Engine |
| GuideLine Acquisition, Representation and Execution (GLARE) | Actions: query, work, decision, conclusion | Durations, repetitions, periodicities, waiting times, repetition conditions | XML | GLARE Acquisition Module | GLARE Execution Engine |
| Standards-based Sharable Active Guideline Environment (SAGE) | Nodes: context, action, decision, routing nodes | Durations, repetitions, periodicities, repetition conditions | RDF | SAGE Workbench | SAGE Execution Engine |
| NewGuide | Careflow recommendations | Simple definition of a temporal sequence of recommendations | XML | NewGuide Editor | NewGuide Execution Engine |

ter sense of the processes they are defining as a whole. However, there are different views on how the CIG creation process should be conducted. Platforms such as DeGeL (Shahar et al 2004) are based on the conversion of textual documents of CPGs into CIGs, providing a well defined staged workflow for the iterative mark-up of the documents and mapping of the tagged objects to guideline elements. However, this kind of approach is not free of error and the process may be hindered by the ambiguity and vagueness of CPG documents, which are later the object of subjective interpretation by CIG encoders. An alternative is the creation of CPGs directly in a CIG format, as proposed in (Goud et al 2009), with a workflow that mimics the development process of their document counterparts in a platform that provides version control and collaborative editing features for the guideline development group. This was found to have positive effects on the quality of both the narrative and machine-interpretable versions of CPGs (Goud et al 2009).

2.2 Guideline Dissemination

Models such as GLIF3 (Boxwala et al 2004) and SAGE (Tu et al 2007) place a strong emphasis on the use of information standards, namely medical terminologies and information exchange specifications, in order to make CPGs encoded in their respective models easily shareable. Most CIG frameworks feature a guideline repository from which an execution engine draws information. The problem is in the accessibility of their tools. Most CIG editing and execution tools are not easily found and the access to CIG repositories is difficult. This is a point in which CIG frameworks would greatly benefit from making their technologies more web-based,

using the web as an application and guideline deployment platform in order to build a truly collaborative and interactive environment. Distributed systems such as DeGeL (Shahar et al 2004) and most recently PICARD (Shalom et al 2015), based on service-oriented architectures, provide APIs to access their guideline deployment tools and seem to have taken in the above-mentioned principles in order to increase the reach CIGs have in the health care community.

2.3 Guideline Deployment and Execution

Guideline execution engines such as GLEE (Wang et al 2004), the Spock Engine (Young and Shahar 2005) or the GLARE Execution Engine (Terenziani et al 2004) were specifically developed for the application of guidelines to patients in health care settings. Their main function is to run CIG instructions against data from patients and produce relevant recommendations, according to the patient state. Most of them, including the above-given examples, exist in the form of client-server applications, with the intelligence engine placed on the client side. Furthermore, these applications are mostly available as desktop apps, which is an obstacle to their reach and ease of deployment.

The role of these execution engines is straightforward and consists in following the constraints of the clinical careflow, comparing items of the patient state with conditions stated in rules dictating whether a recommendation should be provided or not. The data are acquired either from an electronic medical record or on-the-fly from data entry points in the careflow. Thus, if a situation outside of the constraints defined in the careflow arises, the execution engine will be unable to decide and therefore produce a recommendation. Common ex-

amples of this are: the existence of missing or conflicting information; the emergence of situations requiring a group decision and whose logics are not completely defined; or the need for predictive features such as survival prediction, very common in the management of cancer patients. Hence, execution engines ought to include modules that extend the careflow logics and implement Artificial Intelligence (AI) techniques do deal with situations requiring a more dynamic type of knowledge (Oliveira et al 2013a) and collaborative decision frameworks such as the one described in (Lima et al 2011).

As stated above, the development of CIG engines is driven towards a clinical context, meaning that they are tailored to a specific setting and to perform a rigid role, which, in turn, results in few possibilities of integrating their technology in other applications that deviate a little from the clinical in-hospital scope. Yet, an area such as Ambient Assisted Living (AAL), which is dedicated to the development of intelligent systems of assistance for a better and healthier life, would greatly benefit from the existence of such integration facilities. Particularly when we imagine a scenario in which a patient leaves the hospital and has to proceed treatment at home. If an AAL system such as the VirtualECare, described in (Novais et al 2010), is set up with a CIG module connected to the health care institution application, then the patient can continue the guideline execution started at the previous location when he is home. This would mean that the reach of CIGs would stretch far beyond health care institutions.

3 The CompGuide CIG Framework for Integrable Guidelines

Fig. 3 shows an abstract view of the CompGuide framework we are developing for integrable CIGs, which aims to provide solutions to some of the challenges mentioned in section 2. The description of its basic components, namely the *CompGuide Protégé Plugin*, the *Core Server*, and the *Assistant Applications*, will be used to showcase some of the defining features of the framework.

3.1 The CompGuide Protégé Plugin

The CompGuide model for CIGs (Oliveira et al 2013b) uses Ontology Web Language (OWL) Description Logics (DL) as its base language. OWL possesses greater machine interpretability of web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary for the definition of relationships and properties, along with a formal seman-

tics. This ultimately increases the expressiveness of the model. Additionally, there are widely known APIs that facilitate the access to content of ontologies defined in OWL, which makes the development of tools for ontology interpretation faster and easier. CompGuide follows the task network model similarly to GLIF3 (Boxwala et al 2004) and PROforma (Button and Fox 2003), as this seems to be the most inclusive and flexible organization of CPG recommendations. As such, every recommendation in a CPG is represented as one of the following tasks: *Plans*, *Actions*, *Questions*, and *Decisions*. Typically, *Actions* describe tasks that should be carried out by the health care professional, such as exams, observations and so forth. *Questions* are used to feed information about clinical parameters to the execution engine and derive new task recommendations. *Decisions* use that information to perform diagnosis or update the state of the patient. Finally, *Plans* contain instances of any other tasks and are defined to achieve specific goals. CompGuide also provides representation primitives to define different types of conditions, including trigger conditions to select one amongst alternative tasks, preconditions to execute tasks, and expected outcomes. Tasks are represented as instances of classes and the relative order between them is established through object properties connecting them. In this case, it is possible to establish sequential tasks, simultaneous tasks, and alternative tasks over which one can define temporal constraints such as durations, periodicities, repetitions, waiting times, and repetition conditions. Additionally, one can specify temporal constraints about the state of a patient in order to express for how long he had been experiencing certain signs or symptoms, or the period of time in which some sign should manifest as a result of a carried out task. In (Oliveira et al 2015), this ontology was shown to satisfy the representation needs of a wide collection of CPGs.

Given the expressiveness of the model and the fact that it is based on OWL, Protégé was chosen as the tool for CIG creation and editing. However, there was a clear need to customize this tool, based on the classes and properties of the ontology, in order to make the experience of guideline encoding more user-friendly. As such the *CompGuide Protégé Plugin* was developed to provide an interface on which guideline developers can dynamically pick ontology elements to instantiate, and fill in their properties as predefined fields. At the same time, they can visualize the careflow as an oriented graph. Upon guideline creation, a local *owl* file is created and edited locally. Then the user may choose to upload it to the *CompGuide Guideline Repository*. If the user wants to edit an existing CIG in the repository he can pull the corresponding *owl* file and edit

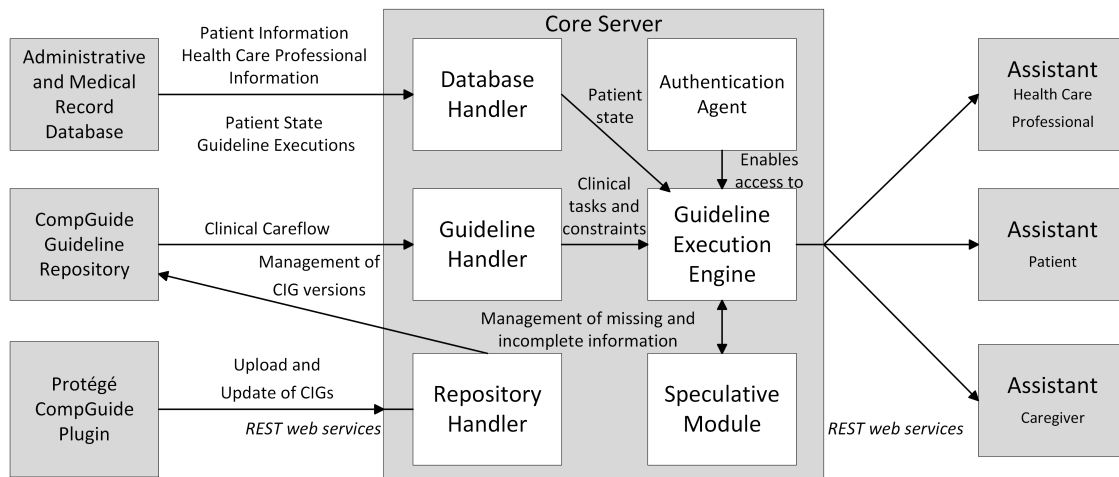


Fig. 1 Abstract view of the CompGuide CIG framework architecture for integrable guidelines with its main components: the CompGuide Protégé Plugin, the Core Server, and the Assistant Applications.

it locally, uploading it later. The plugin sets the basic components for collaborative CIG development.

3.2 The Core Server

The *Core Server* is the central component of the framework and was developed as a RESTful service application. The usage of web services as the means to access the *Core Server* offers expandability and the possibility to improve selected services without compromising others. For their lightness and ease of access, and to stay aligned with the overall goal, REST web services were the chosen service model. This grants greater flexibility when integrating functionalities of the framework in third party applications (Oliveira et al 2014a).

The *Core Server* has six modules. The first is the *Authentication Agent* which provides authentication and authorization to the different types of users of the system, which can be administrators, health care professionals, or patients. The *Repository Handler* is responsible for managing updates to the *CompGuide Guideline Repository* requested by the *Protégé CompGuide Plugin*, keeping different versions of the same CIG organized by authorship and by date. In order to use a guideline for execution, the *Guideline Handler* accesses the selected CIG in the *CompGuide Guideline Repository* and pulls the corresponding careflow, delivering it to the *Guideline Execution Engine*, which, in turn, uses information about the patient state provided by the *Database Handler* to fill in the data entry points of the careflow and produce recommendations. The *Guideline Execution Engine* interprets all the scheduling constraints on the tasks and controls enactment times, which means that starting and ending

times are verified and delivered to the *Assistant Applications* which, in turn, produce notifications to users. These mechanisms to follow the application of procedures over time and to verify the execution of tasks are absent from most CIG frameworks (Shalom et al 2015), but they are essential to have a decision support that is truly capable of following up on the application of a CPG. As for the *Speculative Module*, it is used whenever the *Guideline Execution Engine* has to reason about parameters of the patient state and the values for these parameters are not available. This may be due to delays in getting the results from exams or simply to the impossibility of knowing them. The *Speculative Module* uses a framework of Speculative Computation combined with a Bayesian Network based on previous executions of the CIG. The network is used to derive default values for the missing items, which are then given to the Speculative Computation framework for the generation of the most likely scenarios for the decision point in the careflow (Oliveira et al 2014b). The scenarios are revised as information arrives, adapting themselves to new information states. This module ensures that the execution engine is always able to produce a recommendation.

The purpose of the *Core Server* is to make available CIG services that anyone can integrate in their own applications, with a special focus on AAL applications, where these functionalities would help the improvement of care to sick and elderly people in their home environment (Costa et al 2007).

3.3 The Assistant Applications

The *Assistant Applications* are a set of web-based and mobile implementations that use the services of the

Guideline Execution Engine to execute CPGs for patients. According to their scope, i.e., if they are deployed by health care professionals, patients, or caregivers, they provide different advice based on the target of the clinical recommendation specified in the CIG.

Besides providing recommendations of clinical tasks along with their description and allowing the users to fill in data entry points of the guideline, they implement a notification system that alerts the users to when they should start a task, when they should end it, and the number of repetitions of the task. The applications build calendar views with the recommendations each user should adopt, integrating these recommendations into the daily practice of health care professionals and the daily life of patients and caregivers. In this way, it is possible to build a schedule for them to follow, endowed with notification features about the correct time to enact clinical tasks. The system displayed herein is innovative in the sense that it is unlike other CIG execution tools. It maps clinical protocols, as they are being applied, onto an agenda. The impact of this is that it enables medical personnel to keep a better track of the clinical processes they are managing, and patients, as well as caregivers, to accurately follow medical advice.

4 Final Remarks

4.1 Conclusions

There is a clear need to make CIG technologies more adaptable and context-sensitive. By making use of semantic web technologies, distributed easy to access frameworks and AI techniques, it is possible to achieve just that and tackle the current challenges in CIG and CDSS development. It is necessary to think of machine-interpretable formats not only as supporting tools but as privileged carriers of that knowledge, making it available at any time in our everyday devices.

The features of the CompGuide framework mentioned in this work aim to achieve this goal, building up on the strengths of existing approaches and proposing a new way to share and execute CPGs. These features include:

- An intuitive interface for the definition of CIGs;
- A set of services enabling the integration of CIG-based clinical decision support in any application;
- A set of assistant applications that build an agenda for health care professionals, patients and caregivers based on guideline recommendations; and
- A scenario engine that fills in information gaps in the clinical process.

Regarding the last item, by dynamically revising default constraints about clinical parameters, it becomes possible to ensure that the information has the most likely configuration at all times and, in this way, produce the most likely scenarios. As far as we know this kind of on-line framework is innovative in CIG systems and it may have implications in the outcome of the clinical process. The best possible recommendation is given every time and the CDSS is able to continue the execution of the careflow based on plausible assumptions.

The above-mentioned features enable the adaptation of CIG execution to different contexts, making them highly applicable in areas such as eHealth and AAL, and the tailoring of that same execution to a patient by anticipating changes in his state, thus helping to keep his treatment on track. An additional effect of frameworks such as CompGuide is an increasing adoption of CPGs in multiple environments and the dissemination of best practices throughout different settings.

4.2 Future Directions

As mentioned previously, further integration of CIGs with specialized decision frameworks and more dynamic, adjustable models is necessary and shall be further explored in order to make CIG-based clinical decision support more adaptable and context-sensitive. The path for this may be found in the adaptation and merging of Logic Programming and Machine Learning techniques in order to produce updatable and ever-changing knowledge to fill in the gaps found in the rigid rules of CPGs. In order to do this, CIG components have to be modular, in the sense that it should be easy to displace them in a careflow and connect them to said models.

The ultimate goal for CIG-based CDSSs is to make them self-explanatory by developing information dashboards presented at each decision point in a careflow, showing what information is known about the patient, what information scenarios are derived by the system and their respective likelihood scores, what recommendations are suggested based on them, and the predicted evolution of the patient expressed in careflow paths that will likely be followed. Through this, one hopes to help health care professionals, caregivers, and patients making better-informed decisions regarding the course of the clinical process.

Acknowledgements This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope UID/CEC/00319/2013. The work of Tiago Oliveira is supported by a FCT grant with the reference SFRH/BD/8-5291/ 2012.

References

- Adhikari NKJ, Beyene J, Sam J, Haynes RB (2005) Effects of Computerized Clinical Decision Support Systems on Practitioner Performance. *Journal of the American Medical Association* 293(10):1223–1238
- Boxwala A, Peleg M, Tu S, Ogunyemi O, Zeng QT, Wang D, Patel VL, Greenes Ra, Shortliffe EH (2004) GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. *Journal of biomedical informatics* 37(3):147–61
- Button DR, Fox J (2003) The syntax and semantics of the PRO forma guideline modeling language. *Journal of the American Medical Informatics Association* 10(5):433–443
- Ciccarese P, Caffib E, Boiocchia L, Quagliina S, Stefanellia M (2004) A guideline management system. In: *Medinfo 2004*, pp 28–32
- de Clercq PA, Blom JA, Korsten HHM, Hasman A (2004) Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial intelligence in medicine* 31(1):1–27
- Costa R, Novais P, Machado J, Alberto C, Neves J (2007) Inter-organization cooperation for care of the elderly. In: Wang W, Li Y, Duan Z, Yan L, Li H, Yang X (eds) *Integration and Innovation Orient to E-Society Volume 2, IFIP International Federation for Information Processing*, vol 252, Springer US, pp 200–208
- Goud R, Hasman A, Strijbis AM, Peek N (2009) A parallel guideline development and formalization strategy to improve the quality of clinical practice guidelines. *International journal of medical informatics* 78(8):513–520
- Isern D, Moreno A (2008) Computer-based execution of clinical guidelines: a review. *International journal of medical informatics* 77(12):787–808
- Kaushal R, Shojania KG, Bates DW (2003) Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review. *Archives of internal medicine* 163(12):1409–16
- Kim S, Haug PJ, Rocha Ra, Choi I (2008) Modeling the Arden Syntax for medical decisions in XML. *International journal of medical informatics* 77(10):650–6
- Latoszek-Berendsen A, Tange H, van den Herik HJ, Hasman A (2010) From clinical practice guidelines to computer-interpretable guidelines. A literature overview. *Methods of information in medicine* 49(6):550–70
- Lima L, P, Costa R, Bulas Cruz J, Neves J (2011) Group decision making and Quality-of-Information in e-Health systems. *Logic Journal of IGPL* 19(2):315–332
- Miller M, Kearney N (2004) Guidelines for clinical practice: development, dissemination and implementation. *International journal of nursing studies* 41(7):813–821
- Musen MA, Shahar Y, Shortliffe EH (2006) Clinical decision-support systems. In: Shortliffe E, Cimino J (eds) *Biomedical Informatics Computer Applications in Health Care and Biomedicine*, Springer, pp 698–736
- Novais P, Costa R, Carneiro D, Neves J (2010) Inter-organization cooperation for ambient assisted living. *J Ambient Intell Smart Environ* 2(2):179–195
- Oliveira T, Novais P, Neves J (2013a) Development and implementation of clinical guidelines: An artificial intelligence perspective. *Artificial Intelligence Review* pp 1–29
- Oliveira T, Novais P, Neves J (2013b) Representation of Clinical Practice Guideline Components in OWL. In: Pérez JB, Hermoso R, Moreno MN, Rodríguez JMC, Hirsch B, Mathieu P, Campbell A, Suarez-Figueroa MC, Ortega A, Adam E, Navarro E (eds) *Trends in Practical Applications of Agents and Multiagent Systems SE - 10, Advances in Intelligent Systems and Computing*, vol 221, Springer International Publishing, pp 77–85
- Oliveira T, Leão P, Novais P, Neves J (2014a) Webifying the Computerized Execution of Clinical Practice Guidelines. In: Bajo Perez J, Corchado Rodríguez JM, Mathieu P, Campbell A, Ortega A, Adam E, Navarro EM, Ahrndt S, Moreno MN, Julián V (eds) *Trends in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection SE - 18, Advances in Intelligent Systems and Computing*, vol 293, Springer International Publishing, pp 149–156
- Oliveira T, Satoh K, Neves J, Novais P (2014b) Applying Speculative Computation to Guideline-Based Decision Support Systems. In: *IEEE 27th International Symposium on Computer-Based Medical Systems 2014 (CBMS)*, pp 42–47, DOI 10.1109/CBMS.2014.32
- Oliveira T, Novais P, Neves J (2015) Assessing an Ontology for the Representation of Clinical Protocols in Decision Support Systems. In: Bajo J, Hernández JZ, Mathieu P, Campbell A, Fernández-Caballero A, Moreno MN, Julián V, Alonso-Betanzos A, Jiménez-López MD, Botti V (eds) *Trends in Practical Applications of Agents, Multi-Agent Systems and Sustainability SE - 6, Advances in Intelligent Systems and Computing*, vol 372, Springer International Publishing, pp 47–54
- Peleg M (2013) Computer-interpretable clinical guidelines: A methodological review. *Journal of biomedical informatics* 46(4):744–63
- Shahar Y, Young O, Shalom E, Galperin M, Mayaffit A, Moskovitch R, Hessing A (2004) A framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools. *Journal of Biomedical Informatics* 37(5):325–344
- Shalom E, Shahar Y, Lunenfeld E (2015) An architecture for a continuous, user-driven, and data-driven application of clinical guidelines and its evaluation. *Journal of Biomedical Informatics* (November)
- Sintchenko V, Coiera E, Iredell JR, Gilbert GL (2004) Comparative Impact of Guidelines, Clinical Data, and Decision Support on Prescribing Decisions: An Interactive Web Experiment with Simulated Cases. *Journal of the American Medical Informatics Association* 11(1):71–77
- Ten Teije A, Miksch S, Lucas P (2008) Computer-based medical guidelines and protocols: a primer and current trends, vol 139. Ios Press Inc
- Terenziani P, Montani S, Bottrighi A, Torchio M, Molino G, Correndo G (2004) The GLARE approach to clinical guidelines: main features. *Studies in health technology and informatics* 101(3):162–6
- Tu SW, Campbell JR, Glasgow J, Nyman Ma, McClure R, McClay J, Parker C, Hrabak KM, Berg D, Weida T, Mansfield JG, Musen Ma, Abarbanel RM (2007) The SAGE Guideline Model: achievements and overview. *Journal of the American Medical Informatics Association* : JAMIA 14(5):589–98
- Wang D, Peleg M, Tu SW, Boxwala AA, Ogunyemi O, Zeng Q, Greenes RA, Patel VL, Shortliffe EH (2004) Design and implementation of the GLIF3 guideline execution engine. *Journal of biomedical informatics* 37(5):305–318
- Young O, Shahar Y (2005) The Spock System : Developing a Runtime Application Engine for Hybrid-Asbru Guidelines. *Artificial Intelligence Review* 3581(1):166–170