An Agent-oriented Approach to Support Multidisciplinary Care Decisions

Liang Xiao
Hubei University of Technology
Wuhan, China
lx@mail.hbut.edu.cn

John Fox
University of Oxford
Oxford, UK
john.fox@eng.ox.ac.uk

Hong Zhu
Oxford Brookes University
Oxford, UK
hzhu@brookes.ac.uk

***Abstract*—Patient care is becoming increasingly complex and multidisciplinary for many conditions, notably cancer and chronic diseases, in which a care team participates in and shares responsibility for the patient's care. Providing IT support for joint clinical decision making in an open and distributed environment raises some challenges that are worth our attention: 1) new clinical evidence and guidelines, published by healthcare authorities and subject to continuous revision, need to be shared and enacted by the care team, as automatically as possible; 2) clinical specialists, located in their own working environments, need to be able to group together wherever necessary; 3) decision points, distributed in the environment, need to refer consistently the same set of guidelines and unless these are well-coordinated across the care team, safe delivery of care will be hard to guarantee. In this paper we propose an open and adaptive agent architectural model to resolve these challenges. This is based on an Agent-oriented Model Driven Architecture and a decision support management model, which are integrated to support joint clinical decision-making.**

Keywords—Multidisciplinary Care Pathways; Multi-Agent Systems; Open and Adaptive Software Architecture

1 INTRODUCTION & MOTIVATION

1.1 Multidisciplinary healthcare and collaborative decision-making

It is now recognised that *Clinical Decision Support Systems* (CDSS) could effectively retrieve up-to-date medical knowledge and help to interpret clinical data at the point of care, and this may assist clinicians in keeping their knowledge up to date and improving compliance of clinical decision-making with evidence-based guidelines. The need and demand for this capability from both managers and patients are increasing [5]. The work on CDSS has been reported in a number of literatures, some are found more successful within the limit of specific sites and for specific needs but many fail to achieve the promised improvement in clinical outcome in routine use. The design, development and delivery of CDSS remain a grand challenge, as identified by leading experts in this field [10].

To complicate matters, modern healthcare practice has seen *joint decision-making* becoming more common, as groups of specialists are increasingly involved in and share responsibilities for care. In complex diseases such as cancer, there can be many significant decision points, the responsibilities for which may be distributed among specialist doctors, nursing staff, GPs and even patients themselves. In modelling a breast cancer diagnosis and treatment pathway, for example, we have found that there may be 65 or more significant decision points, e.g. selection of imaging (mammogram, ultrasound, both or none) or biopsy methods, leading on to distinct tasks and workflows [7] [8].

As the actors participate in a patient's care from different places and at different times, decision-making often needs to be supported in a coordinated and collaborative way. A *referecne architecture* is required in order to guide CDSS under development to flexibly choreograph workflows and decision-making throughout a "patient journey", from detection and diagnosis to treatment planning, management and follow-up. Major challenges arise in delivering this kind of architecture, including

1) Clinical evidence and guidelines for addressing any given clinical problem are continuously increasing and improving (see the controversy and criticism about NICE guidelines in [11] [12]). Such new knowledge needs to be rapidly disseminated and used – though often isn't at present.
2) Clinical collaboration is required as some clinical objectives may be difficult to achieve by a single clinician working alone.
3) Collaborations are "learning opportunities". Recurrent successful practices, spanning over multiple clinical disciplines and decision points, should be made reusable and referred consistently, to ensure safe and sound practice.

A major goal of our research is to design an open and adaptive architectural model that addresses the above challenges. The CDSS built with this architecture will be more versatile than its conventional counterparts, capable of dynamic incorporating new clinical guidelines or collaboration structures to support coordinated decision making. This is in contrast with the usual engineering method of one system per clinical problem with implemention of a fixed set of guidelines or pre-agreed interactions, which is not sustainable regarding the more and more complex nature of the problems (e.g. co-morbidity and polypharmacy) [20] and the growing number of problems emerging in that same domain. We will come back to the challenges raised here later in Section 3, justifying the techniques choosed for addressing them and in Section 6, concluding how well they have been addressed.

In the remainder of this section, we give a brief overview of the representation and enactment of clinical guidelines. We introduce in Section 2 the PROforma language for guideline modelling and decision-making as well as some current issues about centralised management. This leads us to the introduction of agent-oriented approaches, where an agent architecture and a classic decision support theory are described in Section 3; a generic agent paradigm that integrates the architecture and the theory towards collaborative decision support is proposed in Section 4; and a set of agent-oriented design models supporting such a unified framework are discussed in Section 5. In Section 6 we demonstrate the application of the approach using a breast cancer referral example and finally we conclude the paper in Section 7.

1.2 Computer-interpretable clinical guidelines

A standard definition of *Clinical Practice Guidelines* (CPGs) is that of: "systematically developed statements to assist practitioners and patient to make decisions about appropriate health care for specific circumstances" [13]. A primary purpose of CPGs is to support clinical decision-making in a way that is consistent with published and peer-reviewed evidence, in order to provide a more rational basis for decision-making and reduce inappropriate variation in practice.

In the UK, the National Institute for Health and Clinical Excellence (NICE) provides national clinical guidelines, enabling timely translation of research findings into health and economic benefits [11]. However, despite great effort by NICE and many other bodies to develop clinical guidelines, it is well known that compliance with guidelines in practice leaves much to be desired. Reasons vary from unawareness of such guidelines by clinicians to the lack of a robust implementation and supporting system [12]. The medical informatics community has therefore sought new ways of bringing up-to-date scientific and clinical knowledge to the point of care in a more flexible and usable form.

Computer-Interpretable Guidelines (CIGs) are formal representations of CPGs that can be used to provide active support for improved effectiveness and safety of clinical practice. CIGs can provide reminders and alerts, assess individual risks, recommend possible treatments and give other patient-specific advices, often providing direct links to the supporting research and evidence as part of the advices. The computerisation of conventional paper-based clinical guidelines transforms the knowledge dissemination strategy and addresses many problems that hinder the adherence of evidence in practice.

1.3 Problem of central orchestration of CIGs

Representation of guidelines using formal guideline representation languages is growing but automatic enactment of CIGs based on them is typically centrally orchestrated via a monolithic hospital information service. This is inconsistent with real life situations because healthcare professionals work in quite *ad hoc* ways which are only loosely coordinated, and are often concerned to maintain local autonomy. It is therefore highly desirable to explicitly specify the interaction protocols among collaborative teams, through which the same set of guidelines can be distributed and universally respected; the patient conditions shared and tracked; individuals' roles and responsibilities clarified; institutional strategies and policies maintained; and joint decision making enabled.

2 The PROforma guideline modelling language

PROforma [1] is a computer-executable clinical process representation language (capable of describing and linking CIGs), developed at Cancer Research UK. The language recognises the complexity of computation in medicine, draws on work in Knowledge Engineering, Software Engineering, AI, and cognitive science for its theoretical foundation, and adopts a multi-paradigm form of computing [9]:

1) *Logic programming*: PROforma preserves declarative logic relationships between primitive terms for reasoning and decision-making, and its simplicity and uniformity facilitates formal analysis of systems against safety constraints.

2) *Object-orientation*: PROforma provides a small number of generic task classes for composing task flows:

- An *Enquiry* is a task for obtaining information from a source;
- A *Decision* is any kind of choice (diagnosis, risk classification, treatment selection, etc.);
- An *Action* is any kind of external operation, and
- A *Plan* is a "container" for a collection or sequence of tasks with a flexible control framework.

The class inheritance mechanism enables sub-classing over tasks where specialised clinical decision plans are required.

3) *Agent-orientation*: PROforma specifies procedural "*agent*" functionalities in terms of performing tasks, interacting with users, and making decisions and recommendations. However communication and coordination are not well supported by the language yet so it does not currently offer a satisfactory basis for developing multi-agent services.

These are the "pillars" of CDSS design [8]: 1) a knowledge theory (first-order logic with extensions); 2) a process theory (task object network); 3) an organisation theory (distribution of tasks and responsibilities to agents); and 4) a rational decision theory (detailed in Section 3.3). Each of these topics are major research areas in their own right, and there are recognised standards in some of these areas. Unique challenges arise from the interfaces between them and which we aim to address, e.g. knowledge sharing in multidisciplinary care pathways and joint decision-making.

PROforma's simple and concise task model has proved to be capable of modelling a range of clinical processes and decisions (see [1] for the syntax and semantics of the language and www.openclinical.net for use cases). PROforma and the authoring and enactment tool *Tallis*, have been used to develop a wide range of applications over the past ten years, including CREDO (a collection of decision-support applications for use in oncology) [2], ERA (early referral for suspected cancer) [3], CAPSULE (prescribing in primary care) [4], and the delivery of decision support with Clinical Evidence of British Medical Journal (BMJ) [5].

A key task supported by the PROforma language is decision making. A PROforma *Decision* has a small set of standard attributes, including *Candidates* (decision options under consideration); *Arguments* (logical conditions which can be used to generate reasons that argue for or against a candidate); and decision rules that can be used to make *recommendations* to clinical users, or *commitments* if the decision making process operates autonomously.

Figure 1 shows a simple PROforma application for making a decision about whether a patient with certain symptoms of breast cancer ought to be referred to a cancer specialist for urgent investigation. The top left panel is a view of a simple task network design for a referral decision in Tallis, with the PROforma representation of this process shown in the right panel. The enacted decision support system is presented at the bottom left panel. It shows alternative decision options ordered in a way that is governed by the overall confidence justified by the set of arguments for or against each candidate. This is an expandable view and may be linked to relevant guidelines, published evidence etc. Currently, an engine centrally orchestrates all tasks and control enactment using a simple state transition algorithm at a single location. In reality, patient data and messages are passed in geographically different locations and it is almost impossible to synchronise them all from a single central point. This leads us to investigating a distributed MAS solution.

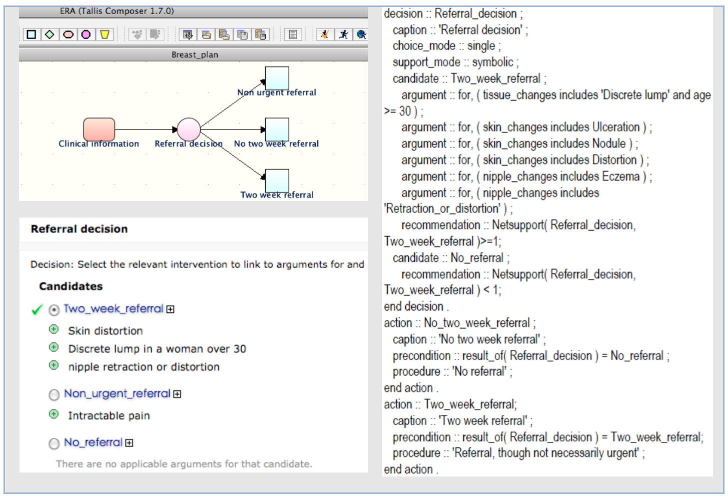


Figure 1. A PROforma application for advising on urgent referral for suspected cancer, from design modelling to web enactment

3 MULTI-AGENT SYSTEMS: METHODS, MODELS, AND APPLICATIONS

3.1 Agents and Multi-Agent Systems in medicine

Agents are computational entities with featurs of autonomy, concurrency, decentralisation, and pro-activeness. These are attractive characteristics for modelling distributed clinical services. *Multi-Agent Systems* (MASs) already see a wide range of applications [14] and as far as complex care pathways are concerned and when a multidisciplinary clinical team is involved, they are more appropriate than alternative technical solutions such as Web Services, for the following reasons.

- Being *knowledge-driven*, agents could be designed to dynamically reconfigure themselves to address a group of related but diverse health problems, just as human specialists do in their day to day routine, rather than engineered to accomplish predefined tasks. *This helps to address Challenge 1 raised in Section 1.1.*
- Being *goal-oriented*, agents could group together opportunistically and divide up tasks between them with respect to constituent clinical roles. *This helps to address Challenge 2.*
- Being imitative to human *mind*, agents may be well regarded as representatives of clinicians and assist tasks such as: receive events, enquire information, draw up plans, make decisions, carry out actions, cognitively and computationally. In addition, recurrent agent collaboration can model multidisciplinary practice and decision-making. *This helps to address Challenge 3.*
- Being *pro-active*, agents could, for example, monitor the blood pressure of the eldly and respond to adverse events by choosing a pre-defined plan from a library, if one is available or otherwise sending alerts to responsible parties. This is a useful feature for general clinical planning.

3.2 The Domino model: a decision support theory and its potential extension to support MASs

Underlying the PROforma language is the *Domino* model [6] [9] [19], which is capable of modeling medical expertise and cognitive processes across a wide range of clinical and other settings. The model embodies a complete state transition framework for "cognitive states": monitoring the environment and updating beliefs; setting up goals; solving problems; choosing or recommending decision options; planning and carrying out actions. The Domino model has been formalised in a set of 12 computational "signatures", abstracted to define the main cognitive functions and their subsuming pre-conditions and post-conditions for the above processes. The Domino model may be extended to support collaborative decision making, if multiple cognitive processes are linked by communication among agents with the same goals. In the MAS context, agents' information processing and decision-making can be triggered by incoming messages and as a result send outgoing messages: a uniform message-driven communication pattern will facilitate this extension.

3.3 Agent-oriented methodologies and AMDA

Our goal is to design an agent architectural model that meets the requirements of supporting a cognitive decision making process such as Domino, and also easy adaptation to guidelines and multidisciplinary coordination. Existing agent methodologies can be drawn upon, e.g. BDI [21] is useful for modelling beliefs and goals; Gaia [22] and FIPA AUML for modelling roles and protocols in multidisciplinary team collaboration; and Tropos [23] and *i** [24] for modelling goals and establishing interdependency among actors. Nevertheless, whatever agent-oriented notions are being proposed, though useful in analysis and design, must be later transformed into object-oriented constructs if a running system is finally to be developed. Many original features are lost or compromised in the process. Even worse is the insufficient support to maintenance or adaptation in later implementation.

The problem is largely caused by the fact that even agents are supposed to behave dynamically in their situated environment, when it comes to engineering they are often implemented (and maintained) as static objects running on platforms such as JADE (Java Agent DEvelopment Framework http://jade.tilab.com/). The direct object-oriented coding of agents limits their capabilities in reactive and pro-active planning and decision-making when environmental events (e.g. receiving a new kind of patient symptom or imaging report) have not been predicted and programmed into agents. Therefore, new methods are required to make the ever-growing knowledge and datasets immediately available and their meanings precisely interpretable by the MASs. Agents are expected to make sensible decisions, reflecting the changing environment, current needs, and increasing human understanding of the world, even after they have been built.

The *Agent-oriented Model Driven Architecture* (AMDA) proposed in [15] [16] encapsulates changing requirements in executable models and rules. These are organised according to a goal structure, under constant (re-) configuration by domain experts, and interpreted by agents dynamically. In any given business context, agents always retrieve the most up-to-date and appropriate rules for advising on participation in business processes, making business decisions, and invoking business components. This makes existing components fully reusable and at the same time new knowledge immediately available, agent behaviour always align with the evolving needs and implementation and maintenance efforts minimised.

We believe that it is possible to reuse this pattern in a clinical context, if we view *clinical guidelines* as a special kind of requirement statements under constant maintenance; clinicians as a special kind of domain experts; and clinical protocols, decisions and events as special kinds of business processes, decisions and events, respectively. In this way, new clinical guidelines can be interpreted and new interaction protocols formed by agents to solve new clinical problems, dynamically. Aiming at a complete agent-oriented modelling and development process, we introduce in the following sections a generic paradigm towards agent-coordinated decision-making, supporting design models with platform-independency, and finally illustrate with a case study the transformation of adaptive agent systems executable upon existing platforms. The central notion of an agent serves various purposes, from capturing clinical role behaviour as required by guidelines and team coordination, to representing runtime software functions such as collecting information, making recommendations, or performing actions.

4 GNAPB: A GENERIC PARADIGM FOR COORDINATED AGENT CONDUCT AND DECISION MAKING IN SOCIETY

An agent society resembles a human society in having similar notions of inter-relating norms which govern the conducts of individuals or groups, as in collaborative decision-making. An agent is seen here as being composed of five parts, in which **Norms** apply across the entire society, **Agreements** between groups or organisations, and **Goals**, **Plans** and **Beliefs** are relevant to individuals.

Agent (Role): {Goals, Norms, Agreements, Plans, Beliefs}

Definitions

Goals are the states that an agent wants to bring about in the environment.

Norms are statements that must hold for all agents in the society at any given time.

Agreements are protocols that govern the behaviour of agents working together to achieve goals.

Plans are collections of tasks that an individual intends to carry out to achieve its **Goals**, such as **Enquiries**, **Actions**, **Decisions** and including sub-plans.

Beliefs are states which an agent holds to be true, particularly with respect to the environment and other agents.

When instantiated in healthcare this generic agent paradigm provides the following more specific schema:

Agent (Role): {

Clinical Problems to Solve *(the diagnosis or treatment of a health problem, etc.)***,**

Clinical Guidelines *(published references or strategies for GP, etc.)***,**

Clinical Interaction Protocols *(the standard ways that sequences of clinical tasks unfold over time, which are carried out by cooperating clinicians)***,**

Clinical Plans *(intended clinical tasks in a logical order, consultation and intervention, etc.)*,

Clinical Opinions *(the interpretation of clinical situations, such as patient conditions)* **}**

Firstly, an agent may want to collect certain data, solve a particular diagnosis problem, or achieve other kinds of clinical target. Such are the **Goal** in focus appropriate to the agent's **Roles**.

Secondly, **Roles** are associated with addressing particular types of problems, rather than specific tasks. For this reason, a versatile agent behavioural model will be employed, appropriate to fit in any actual **Norms** which are continuously improving in qualty and incrasing in quantity.

Thirdly, agents with different capabilities may commit to solve complex clinical problems together, under mutually agreed protocols or **Agreements**. The relevant set of clinical guidelines, or the **Norms** respected in the society, can then be distributed via the enactment of **Agreements**. The agents representing a multidisciplinary team, will base their behaviour upon the guidelines distributed to them, and in a coordinated manner set out by the protocols.

In line with the schematic definition given above for the general **GNAPB** paradigm and that fits the specific need of healthcare, it is illustrated in Figure 2 that agents take a set of *agent society acts* and follow a cyclical process, through which collaborative care service is delivered and decisions made: The driving force of the cycle is always that an agent observes some environmental change which is inconsistent with its current goals and requires some actions: **Goals** provide a bridge from **Beliefs** to **Plans**. The occurrence of an **Event** (reported symptoms, etc.) triggers an agent to *establish* a new **Goal** (diagnose the problem, etc.) or *revise* an existing one. To accomplish this, the agent needs to *initialise* an interaction protocol (as it cannot achieve it alone). Later other participants will be invited to *join* this protocol, under a binding **Agreement** for the group to solve the problem together. The enactment of the protocol will include a process which *distributes* relevant guidelines for participant agents to execute. Under the Agreement, each agent is required to *respect* the relevant parts of the guidelines as **Norms** that govern their behaviour. To bring that kind of agreed duties and responsibilities into effect, agents follow a process to *construct* their individual **Plans** which themselves include a cycle of **Enquiry**, **Decision**, and **Action**.

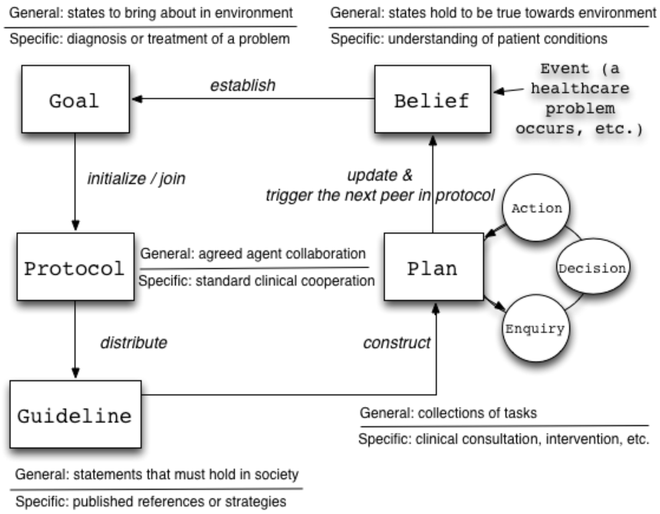


Figure 2. An agent-oriented model for joint care delivery and decision-making, from a generic scheme to a specific derivation

Upon completion of a Plan, agents *update* their **Beliefs** (symptoms have been diagnosed, etc.), and may trigger the next agents to continue within the current protocol or trigger an entirely new cycle (start a further treatment process, etc.). Throughout these processes society members accumulate and share experience and evidence, and reach consensus in order to *classify* Norms with respect to quality or even *propose* new **Norms** or *obsolete* invalid ones. A guideline ranking and evaluation mechanism as such may help self-optimisation. We may call the above a distributed version of the Domino model.

5 AGENT-ORIENTED DESIGN MODELS FOR JOINT CLINICAL DECISION SUPPORT

Applying the **GNAPB** paradigm, as shown in Figure 2, to a multidisciplinary team suggests the sharing, distribution, execution, and coordination of clinical guidelines in an agent society, in that order: 1) the decomposition and distribution of guidelines to a group of peers which later commit to achieving their shared **Goals**; 2) the interpretation of distributed parts of guidelines by agents as **Norms**; 3) the execution of role-specific parts of the guidelines including individual decision making processes by constructing **Plans**; 4) the coordination of agents and the aggregation of results under **Agreements** and 5) the completion of guidelines adding to peers' internal **Beliefs**. Several agent-oriented design models support such a paradigm for later implementation.

Agent-Protocol Subscription Model (Society Agreements)

An interaction protocol specifies the collective behaviour that a group of peers work together, each playing a corresponding **Role**, to accomplish a shared goal, e.g. solving a comprehensive health problem. Roles are characteristics that distinguish one agent from another. At runtime, when an agent subscribes to an interaction protocol and assumes one of the roles required in the protocol, that agent gains the capabilities and commits to the responsibilities associated with that role. Agents can dynamically *subscribe* to new interaction protocols and assume the required roles. As members of different teams and in different settings, they have different problem solving capabilities. The design of agents as flexible protocol subscribers enables grouping and re-grouping of clinical expertise towards emerging multidisciplinary healthcare and provides a diverse range of services.

Agent-Guideline Interpretation Model (Society Norms)

When an agent participates in an interaction protocol, a portion of a clinical guideline(s) related to its role is assigned to this agent for execution – and at a different time, a different guideline may come into effect. In order to enable agents to understand guidelines and behave upon them without predefined constraints or limitation, **Behavioural Rules** are employed as a uniform container to which knowledge can be filled in and from which the required behaviour translated. Therefore, guidelines are transformed into standard rule formats, distributed to agents, and agents always follow the same pattern to *interpret* their required behaviour, dynamically. These rules have the same structural scheme but runtime instances have unique contents encoded and matched with the exact role behaviour. The design of agents as versatile problem-solver and coupled with a uniform rule scheme that advises on runtime behaviour ensures the emerging clinical guidelines will be taken into effect immediately, with minimum system re-development overhead.

Agent-Agent Coordination Model & Agent-Component Binding Model

Agents exchange data and knowledge by **Messaging**, using a common set of performative acts, data dictionaries, and message encoding tags. This allows agents to "speak the same language" in the agent society, though in their independent domains they may use private datasets, components, and applications for local computation and decision-making. Lower level data interoperability between partner sites will be of less concern, since agents exchange knowledge and *coordinate* actions via message passing, and interoperate in such higher levels as achieving shared goals. Issues about data sharing between Primary Care and Secondary Care might be alleviated in this way.

Agents can also make private arrangements with their local components, web services, or event agents wherever their computational capabilities best match the actual needs. Such are the contracts that *bind* agents with their local components. An agent may swap an old component with a new one or choose from alternatives opportunistically, based on their capabilities, performance, cost-efficiency, and other attributes. This might be achieved via human experts re-configuring the binding contracts in **Behavioural Rules** and agents interpreting these rules and binding with the desired components dynamically at runtime.

Agent-Decision Making Model

Situated in a society, agents may receive notifications of emergency or routine requests and respond with suggestion for resolving these requests. The key function is a *decision-*

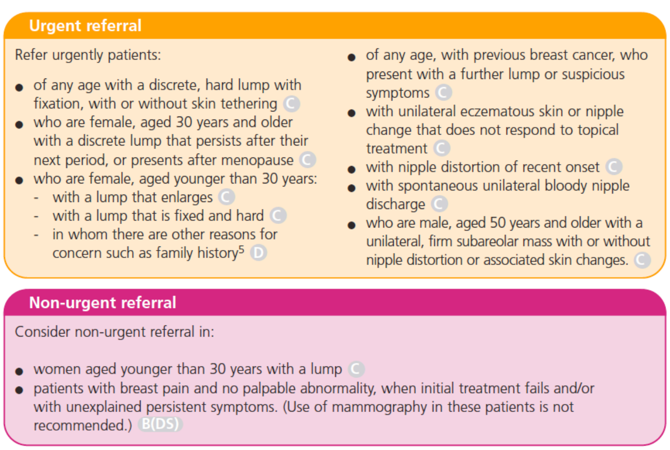
making or planning process. In the Domino model, **Plans may** include **Enquiries** to collect information prior to reaching **Decisions**, and committing to **Actions** once those Decisions are made. In addition to such procedural knowledge as structured in Behavioural Rules, declarative statements are captured in **Production Rules**. They specify the logical relationships embedded in guidelines and may fire in a forward-chaining control structure to deduce extra knowledge. Such knowledge may supplement the enactment of Behavioural Rules, where enquired information needs to be processed, conditions checked, and actions enacted. The schemes and examples of both types of rules will be illustrated in the next section.

TABLE I. AGENT-ORIENTED DESIGN MODELS FOR JOINT DECISION SUPPORT IN SOCIETY

	Resolving Component	**Model**	**Society Notion**	**Problem Solved**
Agent-Protocol Subscription	Role	Subscription	Society Agreement	Distribution of Guideline
Agent-Guideline Interpretation	Behavioural Rule & Production Rule	Interpretation	Society Norm	Interpretation of Guideline Portion
Agent-Agent Coordination	Speech Act	Messaging	Society Performative Act	Adaptive Interaction
Agent-Component Binding	Behavioural Rule	Runtime Invocation	Local Agreement	Adaptive Computation
Agent-Decision Making	Production Rule	Argumentation	Local Plan	Reasoning

6 A CASE STUDY: BREAST CANCER REFERRAL

6.1 Introduction and Computation Independent Model


Urgent referral

Refer urgently patients:

- of any age with a discrete, hard lump with fixation, with or without skin tethering C
- who are female, aged 30 years and older with a discrete lump that persists after their next period, or presents after menopause C
- who are female, aged younger than 30 years:
  - with a lump that enlarges C
  - with a lump that is fixed and hard C
  - in whom there are other reasons for concern such as family history[5] D
- of any age, with previous breast cancer, who present with a further lump or suspicious symptoms C
- with unilateral eczematous skin or nipple change that does not respond to topical treatment C
- with nipple distortion of recent onset C
- with spontaneous unilateral bloody nipple discharge C
- who are male, aged 50 years and older with a unilateral, firm subareolar mass with or without nipple distortion or associated skin changes. C

Non-urgent referral

Consider non-urgent referral in:

- women aged younger than 30 years with a lump C
- patients with breast pain and no palpable abnormality, when initial treatment fails and/or with unexplained persistent symptoms. (Use of mammography in these patients is not recommended.) B(DS)

Figure 3. Clinical guideline for breast cancer referral in natural language [17], previously described within a centralised process in PROforma (Figure 1)

A joint decision support model in Figure 2 suggests a cycle starting when Events *occur*, Beliefs are *updated*, Goals *establishd*, Protocols *initialised*, Guidelines *distributed*, Plans *constructed*, and with multiple occurrences of the above cycle ending up with a patient being successfully diagnosed, treated, etc. Also an agent architecture and its supporting design models are summarised in Table 1. We illustrate below how the decision support model and the agent architectural model may be integrated into a unified framework using the breast cancer referral example. Our principle methodology follows that of Model-Driven Development [15][16].

We start from the guidelines of this example, presented in Figure 3. Guidelines from a nationally or internationally recognised repository of evidence-based recommendations (e.g. *NICE pathways* or *BMJ Clinical Evidence*) are usually formalised using a standardised representation, though most likely in plain text. The distribution of formalised guidelines among the members of a multidisciplinary team reflects the taking of different clinical roles. In the present case study, urgent and non-urgent referral criteria (from NICE CG27) are structured and later distributed to a "GP Agent", which can be seens as part of the Computation Independent Model.

6.2 Platform-Independent Models

1) **Agent-Protocol Subscription**. A breast cancer referral protocol initialises when an **Event** occurs such as a "Patient Agent" reporting an abnormal lump (shown diagrammatically in the top layer of Figure 4 and later, textually in Figure 5). The protocol enacts in a way that three **Roles** (a Patient Agent, a GP Agent, and a Specialist Agent) group together, join the protocol, and commit to solve the problem together. All agents except the initialising one subscribe to the protocol when they receive invitation messages from their coordinators.

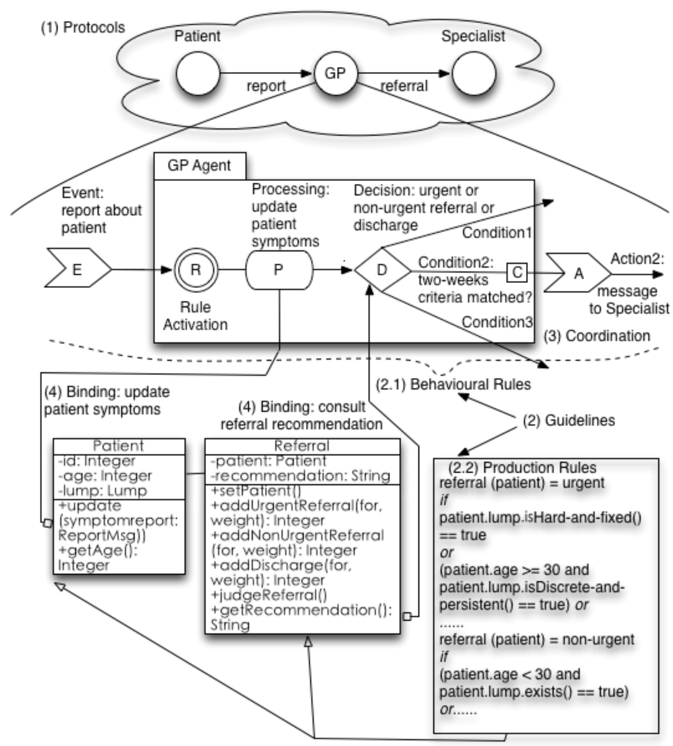


Figure 4. Agents use Behavioural Rules and Production Rules to execute guidelines, under the agreements of Interaction Protocols and over the support of local Components

2) **Agent-Guideline Interpretation**. The GP Agent uses a **Behavioural Rule** to interpret and execute its assigned part of the guideline (shown diagrammatically in the middle layer of Figure 4 and later, textually in Figure 6). On receipt of a

patient report event, it updates its knowledge about this patient using a "Patient" type of component, evaluates its referral criteria using a "Referral" type of component, and finally either sends a discharge message to the Patient Agent or an urgent referral message to a "Specialist Agent", or holds it in abeyance.

3) **Agent-Agent Coordination**. An agent enacts its role behaviour by carrying out **Plans** and making decisions, and in our approach these are uniformly structured as Behavioural Rules. Their structure defines a very simple and uniform coordination pattern: an agent joins coordination when it receives an Event message(s) and invites other agents to join when it produces an Action message(s). The recurrent **Message Passing** pattern between pairs of agents determines the way that multiple roles are coordinated in interaction protocols, and decisions are made jointly, where the outcome of one decision becomes the input of another. In our example, when a Patient Agent *reports* a problem to a GP Agent (*receiving a report message*), the latter may *refer* the patient to a Specialist Agent in appropriate situations (*sending a referral request message*). In this way guidelines previously distributed across different locations are jointly enacted with regards to the coordinative multi-agent behaviour.

4) **Agent-Component Binding**. A Patient and a Referral component are bound to the GP Agent as contracts configured in the **Behavioural Rule** (shown diagrammatically in the bottom layer of Figure 4 and later, textually in Figure 6). These have computational capabilities and invoked by the GP Agent for event processing, decision-making, and outcome production. A Patient instance is updated with the reported symptoms and conditions (result of event processing). A Referral instance, in an association relationship with the Patient instance, will have its attribute values deduced with the assistance of Production Rules. This will be useful to judge the patient referral criteria (condition checking). The interplay of the two types of rules and the components is essential to decision-making.

5) **Agent-Decision Making**. The original source guidelines, shown in Figure 3, guide both the establishment of procedural interrelationship between agents such as patients consult GP or referral to specialists, but also the declarative relationships between key concepts and their logical connection. The former type of knowledge is modelled in Behavioural Rules and the latter in **Production Rules**, which support the deduction of extra facts based on existing ones. In our case study, the GP Agent uses the Behavioural Rules, shown later in Figure 6, to carry out the main body of its plans, and the Production Rules to support the evaluation of Referral upon Patient. That leads to the final judgment and of most importance to the decision making process as part of the plans.

6.3 Platform-Specific Models

Our agent architectural model is generic enough to provide a coordinated decision making framework. No technology commitment has been made thus far but abstract notions must be made concrete and so models be executable. Hereby we introduce XML-based protocol and rule specification.

1) **Interaction Protocol Specification**

```
- <CancerReferral>
  - <protocol-initialiser>
      <name>Patient</name>
      <event>report a lump</event>
    </protocol-initialiser>
  - <roles>
    - <role>
        <name>GP</name>
        <port>RequestPatientInfo</port>
        <port>SendReferralDetails</port>
      </role>
    - <role>
        <name>Specialist</name>
        <port>ReceiveReferralDetails</port>
        <port>NA</port>
      - <property>
          <specialisation>breast cancer</specialisation>
        </property>
      </role>
    </roles>
  - <connector>
      <name>BreastCancerReferral</name>
      <role>referral</role>
      <role>referee</role>
    </connector>
  - <contracts>
    - <binding>
        GP.SendReferralDetails to BreastCancerReferral.referral
      </binding>
    - <binding>
        Specialist.ReceiveReferralDetails to BreastCancerReferral.referee
      </binding>
    </contracts>
  </CancerReferral>
```

Figure 5. A partial specification of a protocol "CancerReferral"

Specification of Interaction Protocols lays down the runtime *agent coordination architecture* required to achieve agents' goals and the distribution of guidelines to specific roles. Some Architecture Description Languages (ADLs) such as ACME [18], C2, and UniCon are well recognised for describing Component-Connector based architectures and may be adapted for specifying role-based agent interactions. We show such an example in Figure 5, an XML-based interaction protocol for the breast cancer referral case.

2) **Behavioural Rule Formalism**

Behavioural Rules have a common behavioural pattern that needs to be recognised by all agents when they carry out their plans. The rules are formulated as below.

Agent (Behavioural Rule):

{Event, Processing, Decision (Condition, Action)n, Belief}

The agents' role-playing behaviour and their interaction is via runtime interpretation of Behavioural Rules. These rules guide agents in a manner similar to the enactment of a Plan in PROforma: *process* **Events** (loosely matching PROforma Enquiry but with explicit incoming messages), *make* **Decisions** (branches loosely matching PROforma candidates and argumentation), and *carry on* further **Actions** (loosely matching PROforma Actions but with explicit outgoing messages) when given **Conditions** are satisfied, and finally *update* its own **Beliefs** towards the environment. Plans are distributed across different sites and then coordinated or "choreographed". An XML-based specification of a Behavioural Rule is shown in Figure 5. It says when a patient reports symptoms (Event), the GP Agent updates its knowledge about this patient (Processing), considers the criteria for urgent or non-urgent referral (Decision), sends the patient to a specialist or keeps the patient on a waiting list or simply discharges her (three matching Condition-Action pairs), and finally updates himself for this occasion (Belief).

```
- <BehaviouralRule>
    <role>GP</role>
    <protocol>CancerReferral</protocol>
  - <component>
      <instance>patient</instance>
      <type>Patient</type>
    </component>
  - <component>
      <instance>referral</instance>
      <type>Referral</type>
    </component>
  - <event>
      <sender>Patient</sender>
    - <message>
        <type>patient.symptomreport();</type>
        <content>patient.symptomreportinXML();</content>
      </message>
    </event>
  - <processings>
      <processing>patient.update(thisMessage);</processing>
      <processing>referral.setPatient(thisPatient);</processing>
      <processing>referral.judgeReferral();</processing>
    </processings>
  - <decision-tree>
    - <branch>
      - <condition>
          referral.getRecommendation().equals("urgent");
        </condition>
      - <action>
          <receiver>Specialist</receiver>
          <message-content>referral.getReferralDetails();</message-content>
        </action>
      </branch>
    - <branch>
      - <condition>
          referral.getRecommendation().equals("non-urgent");
        </condition>
        <action> referral.hold(); </action>
      </branch>
    - <branch>
      - <condition>
          referral.getRecommendation().equals("discharge");
        </condition>
      - <action>
          <receiver>Patient</receiver>
          <message-content>referral.discharge();</message-content>
        </action>
      </branch>
    </decision-tree>
  </BehaviouralRule>
```

Figure 6. Specification of a Behavioural Rule for the GP Agent

3) **Production Rule: Fact Deduction Facilities**

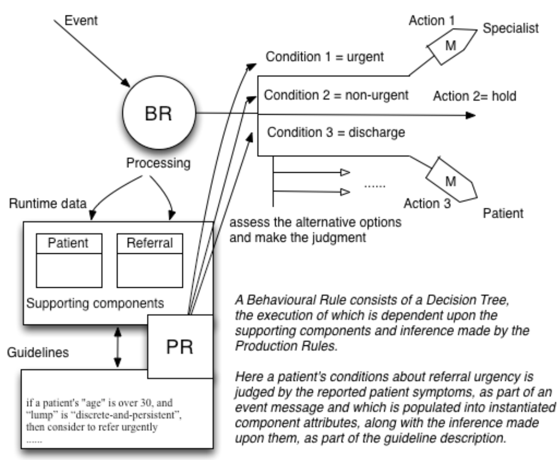


Figure 7. Interaction between a Behavioural Rule (BR), Production Rule (PR), and the Binding Components of an Agent towards Decision Making

Behavioural Rules specify the procedural plans for agents including message processing, decision-making and collaboration, as well as the binding of components that can assist in such tasks. Declarative logic statements concerned with reasoning about candidate decision options and other matters are captured in **Production Rules**. They deduce additional component attributes based on the known set, establish connections between component instances, and support reasoning and argumentation. For example, a fragment of descriptions in Figure 3 is represented as a Production Rule in the bottom right of Figure 4. It establishes the relationship between two component instances: if the attribute "age" of a Patient has a value over 30, and the attribute "lump" has the value "discrete-and-persistent", then a method of Referral will fire: *addUrgentReferral(for, weight)*. This will affect the aggregation of arguments, the assessment of the overall preferences over decision options (urgent, non-urgent, or discharge), and the final ordering of decision options for recommendation ("*judgeReferral*" and "*getRecommendation*"), the overall deision making process illustrated in Figure 7.

6.4 Implementation Model: Adaptive Interpretation

Encapsulating such knowledge of multidisciplinary collaboration and clinical guidelines in executable formats as interaction protocols and behavioural rules makes adaptive architecture and behaviour possible. The system can adapt under continuous maintenance of the knowledgebase by clinical experts, when at the same time a versatile agent engine interprets the up-to-date knowledge as collaborative agent behaviour.

```
thisAgent.addBehaviour(Rule thisRule) {
  // Instantiate all binding components
  Patient patient;
  Referral referral;
  Message m = thisAgent.receiveMessage();
  while(m != null)
  {
     Agent fromAgent = m.getSenderAgent();
     // Check if the rule is defined to handle this event
     if (fromAgent.equals(behaviourRule.getEvent().
       getMessage().getFromAgent()))
     {
     XMLSchema schemaIn =
       thisRule.getEvent().getMessage().getSchema();
     for (int i=0;i<thisRule.getDecisionTree.size();i++)
     {
       XMLSchema schemaOut[i] = thisRule.getDecisionTree.
         getAction(i).getMessage().getSchema();
     }
     ObjMsg symptomReportMsg =
       m.getContentObject().unmarshal(schemaIn);
     // Process the event, etc.
     patient.update(symptomReportMsg);
     referral.setPatient(patient);
     referral.judgeReferral();
     /* Take an action if its corresponding condition is
       satisfied in the decision tree
       if (referral.getRecommendation().equals("urgent"))
       then send referral details' XML serialization form
       to Specialist
       if (referral.getRecommendation().equals("non-
       urgent")) then hold
       if(referral.getRecommendation().equals
       ("discharge")) then discharge Patient */
     for (int i=0;i<thisRule.getDecisionTree.size();i++)
     {
       if (thisRule.getDecisionTree.getCondition(i))
       {
         XMLMsg xmlMsg = thisRule.getDecisionTree.
           getActionObj(i).marshal(schemaOut[i]);
         Message m2 = new Message ();
         m2.setContentObject (xmlMsg);
         Agent toAgent =
          thisRule.getAction().getMessage().getToAgent();
         m2.addReceiverAgent(toAgent);
         thisAgent.send(m2);
        }
      }
    }
    // Start the next cycle
    m = thisAgent.receiveMessage();
  }
}
```

Figure 8. Pseudo code of executing a behavioural rule by the agent engine

We demonstrate in Figure 8 the pseudo code that an agent engine executes a behavioural rule as specified in Figure 6, upon the JADE platform. Unlike a regular code fragment of agent behaviour which is often dedicated to a pre-defined purpose and designed for a single task, a runtime rule interpretation process pulls together the emerging knowledge, services, and components to the current clinical needs and problem-solving requirements, as soon as the rule content is re-configured. Nevertheless, the rule formalism remains uniform in configuration and interpretation: First, when an event message is received by an agent, a rule's eligibility is checked against this event prior to its execution; Then, the content of the message is un-marshaled and the relevant binding components instantiated. As in our case study, a patient's record is updated with the informed symptoms to the GP; Later, an internal processing carries on for establishing the relationships among all relevant components and if necessary, triggering an interaction with Production Rules which may deduce additional facts as a result of inference. As in our case study, a referral object is bound to the patient, and with the assistance of inference on the informed symptoms, a referral judgment can be made; Finally, dependent upon this judgment, the decision tree is executed: either a specialist is contacted due to the nature of an urgent referral, or the GP holds for a while due to the nature of a non-urgent referral, or the patient is discharged. Since then the partner agents of the GP Agent, upon receiving a message as a result, will carry on the interaction protocol as described in Figure 5.

This model of dynamic agent interpretation shapes a system architecture that: a) is capable of accommodating new guidelines as soon as they become available (Agent-Guideline Interpretation), with the procedural Behavioural Rules and the declarative Productions Rules separately maintained, and thus addressing the 1st challenge raised in the Introduction section; b) supports clinicians of various disciplines to team up dynamically (Agent-Agent Coordination), and thus addressing the 2nd challenge; c) abstracts the recurrent multidisciplinary collaboration patterns in reusable interaction protocols through which clinicians can join (Agent-Protocol Subscription) and decision points distribute, thus addressing the final challenge. In addition to these, some software engineering advantages are achieved: d) computing units of higher qualities or capabilities might be swapped in later, if necessary, at runtime as a result of the loosely-coupled agents and components (Agent-Component Binding); e) decision making processes have fine-grained configurability (Agent-Decision Making).

The overall architecture is open and adaptive in that rules that capture the knowledge about clinical pathways and guidelines can be maintained by domain experts, independently from the runtime agents, to accommodate new clinical needs. As soon as these are re-configured, agents will start to use them for dynamic interpretation and execution, being adaptive in coordination, computation, and inference.

Whenever there is a need to support a new kind of multidisciplinary care pathway or decision making process in practice, our approach suggests a process as such to be followed: define a new interaction protocol, make it open to clinical experts and services, let them join it as a team, pass around the relevant evidence and guidelines, and finally they coordinate and make decisions together. The framework, being open and adaptive, will allow new participants to join and new guidelines to be disseminated, at runtime. When the necessary agents are grouped together, play roles and make decisions, the team will be complied with the current clinical guidelines automatically without the need of additional coding.

7 Discussion, Conclusions, and Future Work

The growing specialisation and complex interrelationships in medicine today imply more collaborative rather than independent decisions, a process in which one decision depends upon the result of another and cannot be reached in isolation. In such collaborations, any individual specialist cannot see all the data but may share the responsibilities, which represents a risk to patient safety. We believe it is important to make the knowledge base of clinical guidelines and dataflow in interaction protocols among care specialists or services explicit, in dedicated repositories. These can be used and reused for later discovery, customisation and adaptation. The underlying software systems also need to be capable of disseminating new knowledge and using that knowledge for coordinated decision-making in a distributed environment, as flexibly and adaptively as possible. Multi-Agent Systems are a good candidate, especially when coupled with reconfigurable rules for knowledge encapsulation and runtime interpretation.

Some previous work on applying MAS to healthcare has been reviewed extensively in [14] and more recently in [26], which selected 15 most recent and important applications. Among them three are concerned with clinical decision making, where different types of agents have been proposed, in managing datasets at local clinical sites, collecting relevant data or evidence in a distributed network, interacting with end users, and so on. Major decision making solutions include case-based reasoning (Singh), pattern recognition and data training (HealthAgents), and guideline application (HeCaSe2). In these approaches and others, a focus has been put upon modelling the organisational structures and specific workflows among various types of agents, so that an agent architecture may support the real environment. In HeCaSe2 where guidelines are central and an approach most close to our own, although a Guideline Agent is proposed, authors seem to assume it can automatically accomplish its job. The approach is rather aimed at addressing agent and service interaction processes, including the mentioned Guideline Agent, Medical Record Agent, Service Agent, and User Agent, in a networked infrastructure, especially, the Catalonia medical environment. It seems thus far, no work has been dedicated to the mechanism of representation and distribution of guidelines among a multidisciplinary team, the understanding and interpretation of guidelines by an agent, and the maintenance and dissemination of new guidelines in a MAS architecture.

Our aim is to integrate a sound theoretical model that draws up cognitive concepts in decision making and a practical engineering method that applies agent-oriented models and tools in implementation. A Model-Driven methodology is followed where abstract Platform-Indepdent Models (PIMs) are transformed to more concrete Platform-Specific Models (PSMs) and finally implementation. In the process, XML-based protocol and rule specifications are built as part of an executable knowledge model, and agents start from organisation and distribution of knowledge to later interpretion and execution. Original guidelines, written in natural languages and must be complied with, define strictly what later become the agent-executable specifications. For this reason and with high regard to patient safety, agent autonomy is constrained in our approach and instead they seek common interest and shared responsibilities. Therefore, agents are limited in reasoning for their own benefit or self-interested actions. Mechanisms such as voting, bargaining or negotiation [25], commonly seen in reseach on group decision making among fully autonomic agents of conflicting interest, are thus inapplicable. Our agent-oriented approach, however, combines the advantages of component reuse as in object-oriented approaches and knowledge re-interpretation as in knowledge engineering approaches. A behavioural rule in our case study, for example, uses two components to gain processing functionalities and several production rules to gain inference capabilities, together contributing to clinical decision making.

The CDSS development community has been called for standard service architectures and interfaces so that any EHR system can subscribe to for the needed capabilities with minimum implementation effort [10]. We believe the proposed approach with its open and adaptive agent-oriented decision support architecture can offer a reference model for researchers and developers for adaptation and fitting their own data and knowledge. It will be especially useful whereas care collaboration patterns are emerging and guidelines are improving rapidly, in a multidisciplinary care environement. The approach contributes to cost-effective IT development and maintenance and will eventually provide a shift of effort from system re-design and re-coding to knowledge reconfiguration and re-dissemination.

We used the NICE CG27 clinical guideline about early referral of suspected breast cancer in this pilot study. We will later investigate other guidelines in this field, such as familial breast cancer (CG14 & CG41), early and locally advanced breast cancer (CG80), and advanced breast cancer (CG81). The ultimate aim is to integrate them into a unified framework that covers the whole "cancer journey" and involves even more diverse disciplines in coordinated decision-making. We hope to develop a full care pathway using our methodology and a system working for real environments in our future work.

ACKNOWLEDGEMENT

This work was supported by Grant 61151001 (2012.1-2012.12) & Grant 61202101 (2013.1-2015.12) from the National Natural Science Foundation of China, and a visiting scholarship sponsored by the China Scholarship Council (2012.11-2013.10, Oxford) to Liang Xiao.

8 REFERENCES

[1] Sutton, D.R., Fox, J., The Syntax and Semantics of the PRO*forma* guideline modelling language, Journal of the American Medical Informatics Association 10(5): 433-443, 2003.

[2] Fox, J., Patkar, V., Thomson, R., Decision Support for Healthcare: the PROforma evidence base, Inform Prim Care 14(1): 49-54, 2006.

[3] Bury, J., Humber, M., Fox, J., Integrating Decision Support with Electronic Referrals, Proceedings of the Tenth World Congress on Health and Medical Informatics (Medinfo2001), UK, 2001.

[4] Walton, R.T. et al., Evaluation of computer support for prescribing (CAPSULE) using simulated cases, BMJ 315(7111): 791-795, 1997.

[5] Steele, R., Fox, J., Enhancing conventional web content with intelligent knowledge processing, Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe, pp. 142-152, Springer Verlag, 2003.

[6] Fox, J., Glasspool, D., Modgil, S., A Canonical Agent Model for Healthcare Applications, IEEE Intelligent Systems 21(6): 21-28, 2006.

[7] Patkar, V., Hurt, C., Steele, R., Love, S., Purushotham, A., Williams, M., Thomson, R., Fox, J., Evidence-based guidelines and decision support services: a discussion and evaluation in triple assessment of suspected breast cancer, British Journal of Cancer 95(11): 1490–1496, 2006.

[8] Fox, J., Glasspool, D., Patkar, V., Austin, M., Black, L., South, M., Robertson, D., Vincent, C., Delivering clinical decision support services: there is nothing as practical as a good theory, Journal of Biomedical Informatics 43(5): 831-843, 2010.

[9] Fox, J., Beveridge, M., Glasspool, D., Understanding intelligent agents: analysis and synthesis, AI Communications 16(3): 139-152, 2003.

[10] Sittig, D.F., Wright, A., Osheroff, J.A., Middletone, B., Teich, J.M., Ash, J.S., Campbell, E., Bates, D.W., Grand challenges in clinical decision support, Journal of Biomedical Informatics 41(2): 387–392, 2008.

[11] Littlejohns, P., Garner, S., Doyle, N., Macbeth, F., Barnett, D., Longson, C., 10 years of NICE: still growing and still controversial, The Lancet Oncology 10(4): 417-424, 2009.

[12] Drummond, M. and Sorenson, C., Nasty or Nice? A Perspective on the Use of Health Technology Assessment in the United Kingdom, Value in Health 12: S8–S13, 2009.

[13] Field, M.J., Lohr, K.N. (Eds), Clinical Practice Guidelines: Directions for a New Program, Institute of Medicine, Washington, DC: National Academy Press, 1990.

[14] Moreno, A., Nealon, J.L. (Eds.), Application of Software Agent Technology in the Health Care Domain, Birkhäuser, 2003.

[15] Xiao, L., Greer, D., Adaptive Agent Model: Software Adaptivity using an Agent-oriented Model Driven Architecture, Information & Software Technology 51(1): 109-137, Elsevier, 2009.

[16] Xiao, L., Greer, D., Towards Agent-oriented Model-Driven Architecture, European Journal of Information Systems 16(4): 390-406, Palgrave Macmillan, 2007.

[17] National Institute for Health and Clinical Excellence (NICE), Referral guidelines for suspected cancer (CG27), p. 12, 2011.

[18] Garlan, D., Monroe, R., Wile, D., Acme: an architecture description interchange language, Proceeding of the CASCON First Decade High Impact Papers (CASCON '10), pp. 159-173, 2010.

[19] Fox J. and Das S., Safe and Sound: Artificial Intelligence in Hazardous Applications, Jointly published by the AAAI, Menlo Park, CA, and MIT Press, Cambridge Mass., 2000.

[20] Xiao, L., Cousins, G., Fahey, T., Dimitrov, B., Hederman, L., Developing a rule-driven clinical decision support system with an extensive and adaptive architecture, Proceedings of the 14th International Conference on E-health Networking, Application & Services (HealthCom'2012), pp.250-254, 2012.

[21] Rao, A. S., Georgeff, M. P., BDI agents: From theory to practice, Proceedings of the first international conference on multi-agent systems (ICMAS'95) , pp. 312-319, 1995.

[22] Wooldridge, M., Jennings, N. R., Kinny, D., The Gaia methodology for agent-oriented analysis and design, Autonomous Agents and Multi-Agent Systems 3(3): 285-312, 2000.

[23] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J., Tropos: An agent-oriented software development methodology, Autonomous Agents and Multi-Agent Systems 8(3): 203-236, 2004.

[24] Yu, E., Modelling strategic relationships for process reengineering, Social Modeling for Requirements Engineering, 11, 2011.

[25] Wooldridge, M., An introduction to multiagent systems, Wiley, 2008.

[26] Isern, D, Sánchez, D., Moreno. A., Agents applied in health care: A review, Int'l Journal of Medical Informatics 79(3): 145-166, 2010.