Decision Making Support in the Scheduling of Chemotherapy Coping with Quality of Care, Resources and Ethical Constraints

Christophe Ponsard[1], Renaud De Landtsheer[1], Yoann Guyot[1],
François Roucoux[2] and Bernard Lambeau[3]
[1]*CETIC Research Centre, Charleroi, Belgium*
[2]*Grand Hôpital de Charleroi, Charleroi, Belgium*
[3]*Enspirit, Sombreffe, Belgium*

Keywords: Clinical Pathway, On-line Scheduling, Oncology, Workflow Enactment, Constraint-Based Local Search, Agent System.

Abstract: The scheduling of clinical pathways such as oncological treatments involves a tricky decision process because the therapeutic regimens require to respect strict timing constraints with possibly limited resources such as beds and caregivers availability with an increasing number of patients. Such constraints must be met simultaneously for every patient treated at the same time, by making the best use of limited hospital resources. The scheduling must also be robust in case of adverse events such as unexpected delays or partial treatment deliveries due to their toxicity. In this paper, we show how such a decision process can be driven by care quality indicators to ensure all the dimensions. We demonstrate how constraint-based local search techniques can cope with real-world size chemotherapy pathways and efficiently adapt to changes. We also share some ethical concerns about the way the objective function is expressed and more generally about how the tool integrates in the medical decision process.

1 INTRODUCTION

In Western countries, due to progress in medical care and ageing of the population, hospitals have to manage increasingly complex and multidisciplinary medical procedures over a growing pool of patients. In the worst case, this results in a decrease in the quality of care received by patients, which does not always match the recommended care process yet prescribed. A survey of 30 pathologies ranging from osteoarthritis to breast cancer, observed that, on average, half of the patients received the recommended medical care (McGlynn et al., 2003).

In order to reduce the variability in clinical processes and improve the care quality, a level of standardisation was proposed through clinical pathways (or care pathways). A clinical pathway is defined as a multi-disciplinary specification of the treatment process required by a group of patients presenting the same medical condition with a predictable clinical course (Campbell et al., 1998). It describes concrete treatment activities for patients having identical diagnoses or receiving the same therapy.

This standardization results usually in less delays, higher quality assurance and in reduced costs. As they are strongly oriented on the process description, clinical pathways also maintain a global view on the patients overall journey, instead of individual doctors having a view exclusively limited to their medical speciality (van Dam et al., 2013).

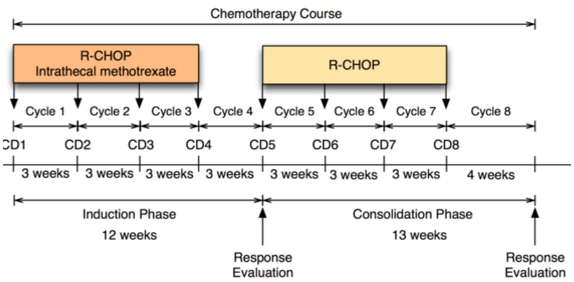

Figure 1: A typical chemotherapy workflow.

The use of clinical pathways have been reported as successful in many therapies, such as arthroplasty (Walter et al., 2007) and breast cancer (van Dam et al., 2013). Clinical pathways in oncology involve a precise description of the therapeutic workflow and all its ancillary activities. Such a partial workflow about

Ponsard, C., Landtsheer, R., Guyot, Y., Roucoux, F. and Lambeau, B.
Decision Making Support in the Scheduling of Chemotherapy Coping with Quality of Care, Resources and Ethical Constraints.
DOI: 10.5220/0006360504600470
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 1*, pages 460-470
ISBN: 978-989-758-247-9

the chemotherapeutic aspect of a treatment is illustrated in Figure 1. Of course, implementing clinical pathways requires involvement. A number of success factors have been reported, like continued clinician acceptance, top management support and a dedicated team of case managers, nurses and paramedical professionals (Choo, 2001).

On the IT side, the computerization of workflows, guidelines, and care pathways is also reported as a key step for process-oriented health information systems (Gooch and Roudsari, 2011). This allows these processes to be managed by hospital information systems or in dedicated workflow management systems. Such autonomous workflow management systems can indeed use clinical workflows as a process model description (Mauro et al., 2010). A key component in this evolution is to provide efficient tools to support the scheduling of these workflows. While scheduling the pathway of a single patient or scheduling the activities of a specific medical department are not that difficult, scheduling a large pool of patients in an hospital with limited resources raises a lot of trade-off concerns (Marynissen and Demeulemeester, 2016). Ideally, such concerns should not impact the quality of care of individual patient. More, basic ethical principles state that every patient deserves optimal care regardless of his medical condition or prognosis. Given that the patients flow is continuous and that a number of unforeseen events require postponing or adaptation of treatment sessions, schedules need to be adjusted on the go. These adjustments should of course comply with already confirmed appointments. The observed practice is that treatment scheduling still usually rely on human-operated manual tools such as spreadsheets or scheduling templates (Ahmed et al., 2011).

In this paper, we address the problem of scheduling treatment appointments in clinical pathways. In order to ensure care quality over a large pool of patients within available resources, we put care quality indicators at the heart of the scheduling algorithms. The latter are implemented using Constraint-Based Local Search (CBLS), a technique known for its ability to scale on large scheduling problems (Van Hentenryck and Michel, 2009). Our algorithms can also efficiently provide new schedules in reaction to changes in some patient constraints (on-line problems). In order to provide a concrete validation, we have focused our efforts on the scheduling of the chemotherapy part of oncological pathways and more precisely on cancer chemotherapy (breast cancers, brain cancers, lymphoma ...) for which a key treatment quality indicator called RDI (Relative Dose Intensity) has been defined (Lyman, 2009). This case study is relevant because proper enforcement of time constraint is critical to maximize the chances of survival of the patients. The availability of the RDI for these treatments, and its proven correlation with survival rates, enables us to quantify the enforcement of time constraints, so that corrective actions might be considered in case of a deviation.

This paper is organised as follows. In section 2, we present domain background about chemotherapy pathways and technical background on constraint-based local search. Section 3 gives a clear statement of the problem tackled in the paper. Section 4 discusses key design aspects of the solution while Section 5 details its implementation. The validation is carried out in Section 6 and relies on an environment simulator. Related work is discussed in Section 7. Finally Section 8 concludes and highlights future work.

2 BACKGROUND

This section first presents the problem domain of chemotherapy pathways before introducing local search frameworks, and then focusing on the necessary vocabulary of CBLS framework used in the remaining of the paper, based on the OscaR.cbls engine.

2.1 Chemotherapy Pathways

The typical workflow for a chemotherapy is a sequence of drugs deliveries or cures, typically administered in day hospital. Each cure is followed by a resting period at home that lasts for a few days to a few weeks. A minimal interval between cures is required because chemotherapy drugs are toxic agents and the body needs some time to recover between two drugs deliveries. When following the ideal treatment protocol, the number of cancerous cells are progressively reduced, hopefully to reach a full healing or cancer remission as shown in Figure 2.

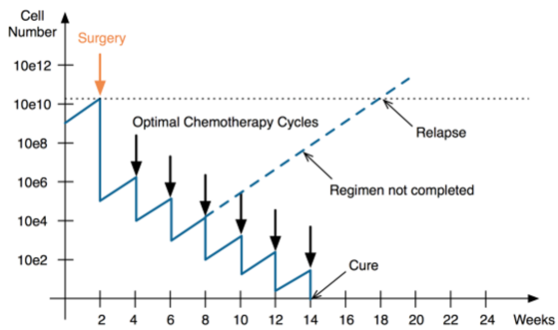

Figure 2: Optimal chemotherapy cycles.

If for some reason, chemotherapy cures do not closely follow the intended periodicity or if doses are significantly reduced, the treatment efficiency may be suboptimal. In such conditions, cancerous cells may multiply again, which can result in a cancer relapse as shown in Figure 3

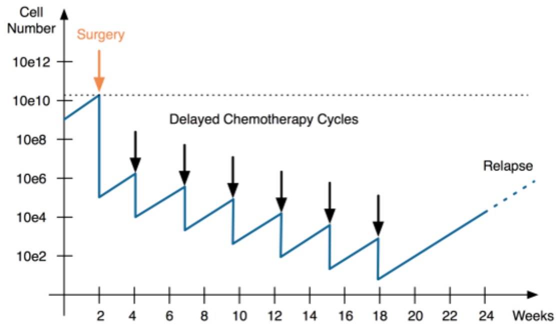

Figure 3: Delayed chemotherapy cycles.

In order to measure the quality of chemotherapeutic cares, a quantifiable indicator called the "Relative Dose Intensity" (RDI) (Lyman, 2009) was defined. It captures both the the fact that the required dose is administered and the timing of the delivery, on a scale from 0% (no treatment) to 100 % (total conformance).

$$RDI = \frac{planned\_dose}{delivered\_dose} x \frac{real\_duration}{planned\_duration}$$

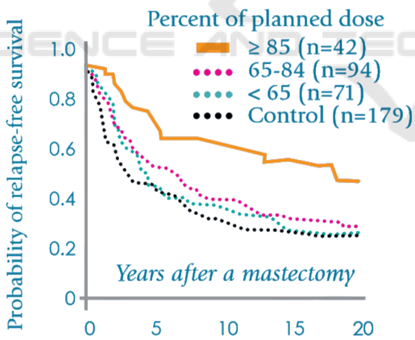

Figure 4: Probability of relapse-free survival vs. RDI.

Medical literature has shown, for a number of cancers, that the relapse-free survival is strongly correlated with the RDI. For instance, for breast cancer, a key threshold value is 85 % as illustrated in Figure 4 (Piccart et al., 2000). Hence this indicator can be seen has a gauge that should be carefully managed across the whole clinical pathway.

2.2 Local Search Frameworks

Local search frameworks aim at making the development of algorithmic local search solutions much simpler than traditional coding. To this end, they provide different degrees of implementation support, from problem modelling to the elaboration of a search procedure. We use such a framework to develop a schedule optimizer for clinical pathways.

Among the general local search frameworks, EasyLocal++ is a well known and representative one that requires a dedicated model to be developed from scratch using ad-hoc algorithms. It mainly provides support for declaring the search procedure (Di Gaspero and Schaerf, 2003). It does not provide as much assistance in the development of a model as a CBLS framework would. Notably it does not allow the developer to package efficient global constraints that can be instantiated on demand.

Besides OscaR, the framework we have used and described in the next section, there are a few tools supporting constraint-based local search specifically:

- Comet is the seminal system for constraint-based local search (Van Hentenryck and Michel, 2009). It features a differentiation facility that is not implemented in OscaR.cbls. OscaR.cbls relies rather on partial propagation to provide a comparable efficiency. Besides, differentiation as provided by Comet cannot handle intricate models where constraints are posted on variables controlled by invariants. Comet is a proprietary system available under a commercial licence.
- LocalSolver is a commercial solver implementing CBLS. It supports boolean and floating point variables. It does not require the user to specify neighbourhoods or meta-heuristics (Benoist et al., 2011).
- Kangaroo provides a partial propagation feature that is more selective than OscaR.cbls (Newton et al., 2011).

2.3 CBLS, the OscaR Way

Among the different solvers, OscaR.cbls was selected. Since this contribution has been done in the context of the OscaR.cbls tool, we further introduce the basic concepts of CBLS using the vocabulary of OscaR.cbls.

As usual in local search, solving a problem involves specifying a *model* and a *search procedure*.

The *model* is composed of *incremental variables* (integers and sets of integers at this point), and *invariants* which are incremental directed constraints maintaining one or more output variables according to the atomic expressions they implement (e.g. Sum: the sum of inputs). *Constraints* are special invariants that maintain their violation as an output variable.

They are Lagrangian relaxations of their specification. Besides, they also maintain some information about which variable causes the violation.

The *search procedure* is expressed using *neighbourhoods*, which can be queried for a *move*, given the current state of the model, an acceptance criterion, and an objective function. *Combinators* are a set of operators on neighbourhoods that combine them and incorporate several metaheuristics, so that a complex search strategy can be represented by a composite neighbourhood totally expressed in a declarative way (De Landtsheer et al., 2015). A library of combinators is available for specifying standard metaheuristics (e.g. simulated annealing, restart, hill climbing), for managing solutions (e.g. when to save the current state, or restore a saved state), and for expressing stop criteria.

In order to set up the floor for designing a scheduling solution, we give details on how the model is represented and updated during the search.

The *data structure* behind a model is a graph, called the *propagation graph*, which we can approximate to a directed acyclic graph, where the nodes are *variables* and *invariants*. Variables have an associated type and implement specific algorithms related to their type. Invariants have specific definitions, and implement this definition mostly through incremental algorithms. Edges in the graph represent data flows from variables to listening invariants and from invariants to controlled variables. The directed acyclic graph starts with input (a.k.a. decision) variables, and typically ends at a variable whose value is maintained to be the one of the objective function. Figure 5 illustrates a propagation graph for a simple warehouse location problem.

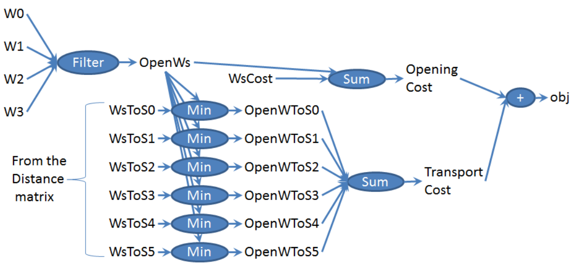

Figure 5: Propagation graph on a warehouse location problem.

In such engine, *propagation* is about propagating updates along the propagation graph in such a way that a node is reached at most once by the update wave, and only if one of its inputs has changed and if needed by the model update. OscaR.cbls manages this wave by sorting the nodes based on the distance from the decision variables. The propagation is coordinated through a dedicated heap that aggregates nodes at the same distance in a list. This offers a slightly better time complexity than the classical approach based on topological sort initially presented in (Van Hentenryck and Michel, 2009).

The search starts from an initial solution and explores the specified neighbourhood. Each neighbour solution is examined by modifying the input variables, and querying the objective function of the model which is updated through propagation.

During propagation, variables *notify* each invariant listening to them about their value change. For integer variables, a notification carries a reference to the variable, and the old and new value of the variable. For set variables, it carries a reference to the variable, the old value of the variable, the new value of the variable, and both the set of values that have been added and removed from the variable. All values transmitted by variables, through notification or through queries to the variables are immutable, to make the implementation of algorithms in invariants easier.

3 PROBLEM STATEMENT

The problem considered here is to continuously optimize the scheduling on an evolving set of patients following a specific chemotherapy process as described in Section 2. The goals of the scheduling optimization are the following:

- maintain the best quality of care (i.e. achieve the best RDI) by avoiding delay for all the patients in the pool
- meet the resources constraints: available treatment rooms and nurses.
- respect service opening days (weekends, holidays) and hours.
- take into account strong unavailabilities of patients, when known.
- when possible, distribute the workload evenly over time to avoid work peaks.

When entering his chemotherapy pathway, a patient can be given an indicative optimal schedule based on what is known at that time. However the global scheduling can be impacted by many events such as:

- the cancellation of treatment delivery, because of patient no shows or medical no-go (e.g. too low white blood cells detected in last blood test),
- the delivery of a partial dose, due to degraded condition induced by chemotherapy toxicity,
- other patients entering and leaving the pathway.

These events are communicated by different actors to the system (e.g. nurses monitoring the drugs delivery, doctors checking the patient condition, administrative staff registering the arrival or non-attendance of a patient).

To maintain optimality, the occurrence of such events will trigger a re-scheduling. Consequently, the considered scheduling is an *on-line problem* which should meet the following additional constraints:

- the recorded past is of course irreversible: this makes any deviation to the ideal care delivery schedule hard to reverse.
- confirmed appointments for other patients should preferably not be changed because it requires administrative work and can induce a cascading effect.

A key actor in charge of activity re-planning is the administrative nurse. She is frequently in contact with the patient and acts as a relay between the patient and the system. She is also in charge of negotiating and confirming the future treatment delivery dates between the patient and the system:

- in the ideal case, all dates initially computed are respected.
- in case of problems in the patient's pathway, the patient schedule is adapted. This can impact the schedules of other patients. This is not important if the schedule shift is minor and concerns dates that are not yet confirmed.
- if the patient wants to delay a treatment delivery, the system shall estimate the impact of such delay in terms of degradation of the chances of healing. This degradation shall be reported to the patient, e.g. by strongly insisting on the importance to comply with the proposed date if a good RDI is compromised.

Figure 6 gives a complete contextual view of the information flow between the patient and the scheduling system and the information the system need an access to.

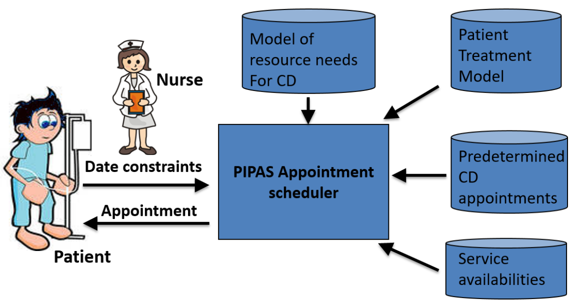

Figure 6: Problem context diagram.

4 SOLUTION DESIGN

In this section, we first propose a global architecture for the scheduling solution and then consider the more tricky problem of specifying a fair objective function.

4.1 Global Scheduling vs. First Come First Serve

Our approach is about scheduling the care of all patients together in such a way that some global time constraints are enforced. The actual situation in most day hospitals is that patients are scheduled on a first-come first-serve basis. With such a policy, in case of resource shortage (beds, nurses), the treatment of a patient might be postponed by some days. For some patients, such a delay can result in great harm in terms of chance of healing.

In contrast, our solution avoids resource shortage by smartly spreading over time the start date of the chemotherapy pathways. However, if resources were still limited, the system will smartly select patients to postpone by limiting the impact on their time constraints and thus their RDI.

4.2 Proposed Architecture

An agile prototype-based approach was applied to design our scheduling solution. The architectural design of our solution quickly evolved towards the agent-based architecture depicted in Figure 7 and composed of the following agents:

- the *Orchestrator* is the central agent. It ensures that the system behaviour is consistent with the input received and that the information generated by the system is dispatched to the end user.
- the *User interface* captures relevant patient information and gives comprehensive views over the pathways at different levels of detail.
- the *Simulator* is used for validation purpose (see Section 5).
- the *Persister* is is charge of managing the state information about the patients involved in the clinical pathways. It provides a domain representation to the orchestrator and relies on a relational database for persistent storage.
- the *Oncoplanner* is responsible for proposing solutions matching the domain constraints sent by the Orchestrator when a change requires to compute an updated solution. It relies on the OscaR.cbls framework.

This architecture has the following benefits:

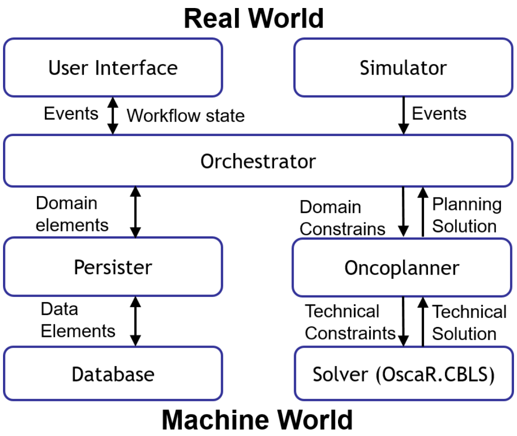

Figure 7: Global architecture.

- it ensures a clear separation between the real world (user side) and the computer world.
- it allows one to plug different kinds of user interfaces easily: first, a basic command line user interface, and later a web-based one using the same communication protocol.
- it also enables to easily integrate a full environment simulator which can play complex scenarios that are able to test the system under high stress.
- it provides good integration capabilities with hospital systems, e.g. to retrieve information from available databases through specific agents.

In this architecture, the scheduler can also work as a background service constantly trying to improve the solution in the open future (i.e. beyond all confirmed appointments), while the orchestrator can take care of reporting when some change occurs in the real world. In case a change makes the current computation irrelevant, the orchestrator can ask the scheduler to stop his work and launch a new computation based on the updated constraints.

We started with a simplified model combining the chemotherapy workflows models resulting from a rigorous analysis process (Damas et al., 2014), resource constraints and possible interfering events. An appointment scheduler was developed along with key companion tools such as a scenario repository, a graphical interface to manage appointments and a simulator of patient-related events. This greatly made easier the validation described in Section 6. The following features were progressively addressed to reach a model that is now realistic enough to consider a validation at day hospital :

- simple resource model, expressed in bed/nurse hours evolving to a finer grained model where each nurse/bed is explicitly allocated.
- service opening days and hours.
- treatment plans, modelled as sequences of steps (day of cure, resting periods) parameterized with doses, durations and involved resources.
- constraints on treatment plan instances: earliest/latest start date, patient unavailability, set appointment (past or confirmed).

4.3 Modelling the Objective Function

The objective function to maximize is the global RDI over the pool of patients. We have developed two global criteria:

A first criterion was to maximize the minimal RDI among all patients. It is implemented by minimizing the schedules makespan among all patients using *iFlatRelax* (Michel and Hentenryck, 2004). The schedule of a patient is an interleaving of appointments and resting periods, followed by a "stub" activity at the end. This stub is needed because all patients do not start their treatment at the same time. That stub activity enables us to consider their treatment duration instead of reasoning on their ending date. This criteria may look fair but patients with the highest "healing chances at start" (e.g. with no dose reduction) could be considered as "neglected".

A second criterion was to maximize the summed RDI. This can be modelled as a tardiness problem, i.e. overshot of a given point in time (patient dependent) multiplied by a constant. This problem is widely studied and was solved using a task swapping neighbourhood starting from a solution provided by *iFlatRelax* because it was tightly packed and computed very quickly.

5 IMPLEMENTATION OF AGENTS

The implementation of the architecture detailed in the previous section relies on web services technologies: our agents communicate through a RESTful protocol relying on the JSON format for exchanging the required information. This section highlight key implementation issues of the agents, except for the simulator which is detailed in Section 6.

5.1 Scheduler Agent

A strong requirement was to cope with large patient sets, typically involving hundreds of patients simultaneously at various stages of their own clinical pathway. In order to scale to such size, we used local search-based approaches, mainly *iFlatRelax* for scheduling, and in a later phase, *BinPacking* for day-level reasoning. Both algorithms were implemented using the CBLS engine of OscaR (De Landtsheer and Ponsard, 2013)(OscaR Team, 2012). They are further described in the next subsections.

Based on this techniques, our prototype is able to schedule chemotherapy appointments over roughly five hundred of patients in a few seconds and supports interactive adjustments.

5.1.1 Iterative Flattening-Relaxation Search

The algorithm implemented in our prototype is the one already presented in our previous report, namely: the iFlatRelax. This algorithm has been improved with the following new features:

- The possibility to define non-moveable tasks, that is: tasks that already have a given date. We need to represent these, because some appointments of treatments might already be fixed, and we do not want them to be moved, since they were communicated to the patient.
- The possibility to define forbidden zones for some activities, that is: a set of points in time where the activity cannot take place. The activity is henceforth moved forward in time until a proper position is found. We will use this feature to represent patient unavailability.
- A more flexible model of resources, that will enable us to represent bin-packing resources, as mentioned in the next section.

5.1.2 Bin Packing

Beds and nurses need to be modelled as they are in the real world: a patient needs to occupy a bed for a certain amount of time, and more beds allocated for a patient will not reduce his time spent in a specific bed. The consequence is that we cannot represent the bed resources as a single integer value in terms of bed hours available in a certain day. Instead, we need to model beds as done in a bin packing problem: each bed is a bin whose size is the duration of the day, and each patient of the day must be put in a bed selected among the available ones. This allocation shall comply to the fact that the sum of duration of each patient occupying the same bed is lower or equal to the duration of the day. Patients will occupy the same bed in sequence, of course.

We therefore needed to represent a so-called “bin packing resources” exhibiting this behaviour. A bin packing resource in a scheduling problem is a resource divisible into bins of given size. Each activity using this resource declares a certain amount of usage. The activities using the same resource at the same time must be scheduled in proper bins, so that the sum of each activity using the same bin do not exceed the capacity of that bin.

So far, we have developed a model of the bin-packing problem, and a solver for bin-packing problems. We still need to integrate this solver into our scheduling engine.

5.2 Persister Agent

The persister agent provides a service-based implementation of the normalised conceptual model depicted in Figure 8.

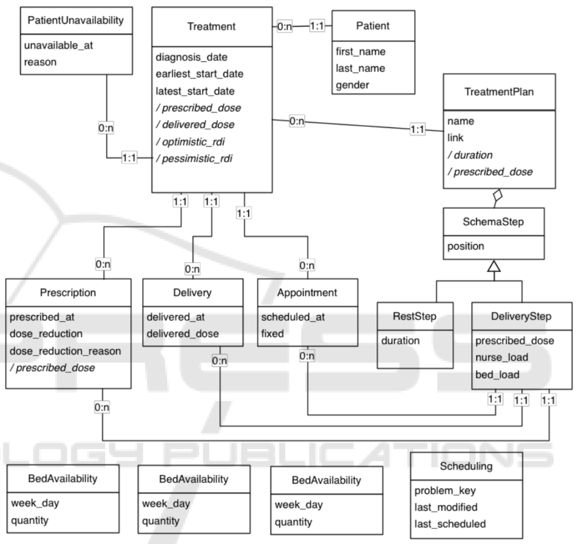

Figure 8: Data model for a chemotherapy pathway.

It can also be considered as a domain specific language for clinical pathways and is composed of the following concepts:

- *Patient*: models the information about patients treated and monitored by the tool.
- *TreatmentPlan*: captures a reusable treatment template which is composed of one or more *SchemaSteps*, either *RestStep* or *DeliveryStep*.
- *Treatment*: captures a processing instance, for a given patient. It involves a number of monitored events such as *Prescription*, *Delivery* and *Appointment*. These are linked to drug injection steps through *DeliveryStep*.
- *PatientUnavailability*, *NurseAvailability*, and *BedAvailability* respectively capture the availability of beds, nurses and patients for the scheduling of appointments.

5.3 Agenda-based User Interface Agent

The user interface is a browser-based HTML/-JavaScript application interacting with the orchestrator. It is implemented using AngularJS. The interface is depicted in Figure 9. It provides:

- the visualisation of the clinical pathways of the whole set of patients, allowing to spot the past chemotherapy deliveries, the future deliveries with a distinction between the confirmed ones (the scheduler will not alter them but the nurse could move them provided the patient is in the loop) and unconfirmed ones (these can be moved by the scheduler until they become confirmed). Service and patient unavailability are also displayed.
- control over the pool of patient, e.g. adding a new patient and encoding specific characteristics of his workflow instance (earliest start date, regimen periodicity and duration, target dose, etc.).
- encoding of delivery related information, e.g. partial dose delivery, cancellation, no-show, ...

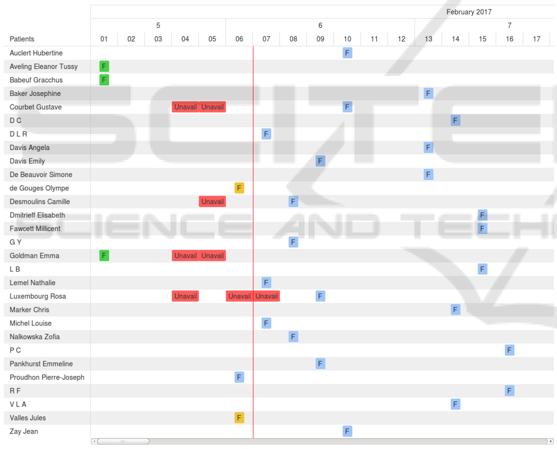

Figure 9: Agenda user interface.

6 VALIDATION

6.1 Validation Approach

Validation of care pathways in a real day hospital environment will be considered in advanced research phases. Until now we used a simulation-based approach because it helps to understand the system behaviour over long periods and under stressed conditions that are difficult to experience in the real world. It also has the ability to step inside processes and provide a good understanding of problematic scenarios.

In order to enable an earlier validation, we designed the simulator to:

- Provide a reactive and complete user interface, clearly illustrating the key characteristics of the algorithms: quality of the solutions, speed of recalculation, taking into account complex events (no-show, partial dose, report ...)
- Integrate an environment simulator able to generate flows of planned and unforeseen events that are experienced by the targeted hospital services and that can be used to consolidate the required indicators proving the value of our tool, especially in relation to the quality of care in terms of compliance with RDI and load management.

6.2 Simulation Framework

As shown by Figure 7, the simulator agent is directly connected to the orchestrator through the same protocol as the user interface. The simulator has two main components:

- a "driver" responsible for simulating the interactions a user would normally have with the orchestrator using the UI. It relies on exactly the same communication protocol (JSON via HTTP) as the UI.
- a "control panel" to have the control over the running simulation, depicted in Figure 10. Two modes are available: (1) step-by-step mode. This mode enables to have a deep understanding of a specific run. It can also be used to introduce specific events manually using the usual user interface. (2) fully automated mode, running at machine speed. This mode relies on the generation of events based on probability distributions. It can be used to spot specific problems and also to assess the performance over a large number of runs using a Monte-Carlo process.
- a component collecting and displaying the evolution of relevant indicators, such as the RDIs and the service load.

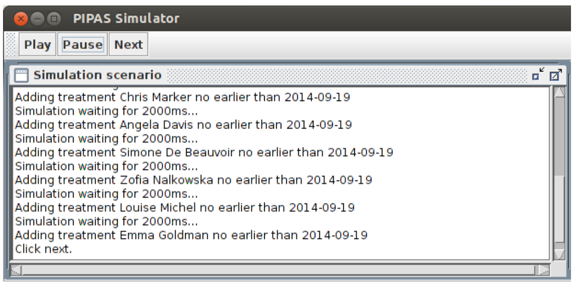

Figure 10: Simulator control panel.

The state machine for the patient is illustrated in Figure 11: between treatment deliveries, it should be in an “appointment fixed” state and during delivery, it will be in “Delivery” state (if correctly reported). The “waiting state” should only be transitory: either just after a delivery or in case of no show. Those states should be tracked for minimal duration. The simulator generated events covering those transitions with given probabilities, e.g. for no show events.

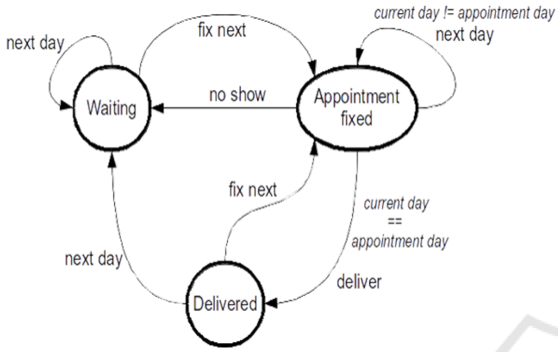

Figure 11: Possible patient states and transitions.

6.3 Results Validation

Several simulations sessions were organised both with the technical team and in sessions with oncology practitioners involving three hospitals (UCL/Cancer Institute, Grand Hospital of Charleroi and UZ Leuven). The feedback from doctors was quite positive about our contribution to ensure both quality of care and the smarter use of resources. Figure 12 shows that the RDI is kept above the 0.85 and above 0.90 in many cases. It is decreasing over time as the result of minor delays or partial doses delivered due to chemotherapy toxicity. The day hospital load is also exhibiting a smooth curve meaning that appointments can be evenly dispatched over time.

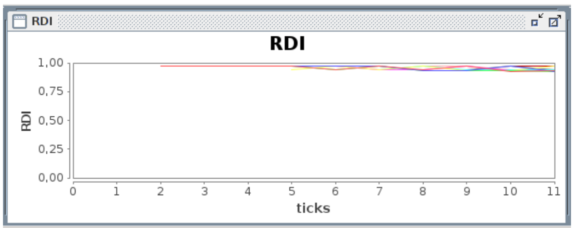

Figure 12: RDI reached in a typical oncology unit.

The prototype also raised ethical concerns, such as the capacity of the tool to choose to favor some patients rather than others in case of resource shortage. Our conclusion is that the system should report such situations ahead of time to allow the day hospital team to take corrective measures, like a transient increase of staffing. In order to keep the medical team in control, the developed graphical display was also a huge practical improvement. Some interesting feature such as the visualization of allocation windows ensuring a good RDI level definitely helped oncologists and nurses in charge of appointments updates.

7 RELATED WORK

A complete literature review on integrated hospital scheduling problems, including pathways, was recently published by (Marynissen and Demeulemeester, 2016). Although concepts such as clinical pathways or diagnosis related groups have been around for more than 20 years, the study reveals that most of the relevant work is quite recent. Besides progress in methods and tooling, the main trigger factor is that hospitals are facing the necessity to break barriers across services for dealing with the performance and capacity challenges they have to face. Currently, off-line scheduling approaches are more often reported than on-line methods because hospitals want to provide the best possible solutions which are related to the largest possible exploration of the state space. This rules outs methods which are mostly based on (meta-) heuristics. However very good (and in some cases optimal) results of using local search for outpatient scheduling has been reported in the literature (Kaandorp and Koole, 2007) and are confirmed by the quality of our simulation results. A key point is that our CBLS engine ensures fast exploration of the search space, resulting in a good coverage. Moreover our design also allows the system to continue optimizing the time horizon for appointments not yet confirmed to patients and thus without strong time constraints.

A methodology to design appointment systems for outpatient clinics and diagnostic facilities that offer both walk-in and scheduled service is presented in (Kortbeek et al., 2011). The proposed schedule has two levels: a global level stating the number of appointments per day and a day level detailing when each appointment should be scheduled on a given day. Each level is managed by a specific model and the two models are connected by an algorithm. Our approach also proposes two levels with a coarser granularity at the global level (e.g. we consider global bed-hours availabilities) while at the day level a fine grained model is used (i.e. we assign patient to available beds). Our current validation is however currently limited to the global level.

Scheduling has also been successfully applied in other hospital areas. In radiotherapy, several approaches of operation research ranging from strategic capacity management to operational scheduling levels are reported in (Vieira et al., 2016). It highlights that many improvements regarding the waiting times and resource utilization can be achieved. A substantive attention is also devoted to the scheduling of the operation room because it accounts for up 40% of resource costs in a hospital. However it differs from oncology pathways by a far more important level of unpredictability. Reported results shows that about 30% more patients can be scheduled than in actual practice and the operating room utilization rate is increased by 20% (Barbagallo et al., 2015).

Multiple algorithms and software tools to generate qualitative surgery schedules on the tactical and operational level are also reported in (Demeulemeester et al., 2007). This work actually also shows a wider impact on the whole hospital, since these operation rooms are interrelated to many other departments or organizational problems like nurse scheduling or bed levelling. It also points out the necessity of a good visualization capabilities because they help health managers to have good insights and they also guide them in testing different scenarios.

Regarding the computer tools used, the observed practice is that tool support is often still relying on manual or basic tools such as spreadsheets or scheduling templates (Ahmed et al., 2011).

8 CONCLUSION

This paper presented an approach for assisting the scheduling of drug deliveries for treatments where such deliveries must comply to time constraints under relatively limited resources, namely chemotherapy regimens. A chemotherapy is delivered in several deliveries or cures; some timing constraints must be enforced between these cures. If the cures are spaced too much, the efficiency of the treatment gradually decreases. If the cures are too close in time, there is a high risk of side effect. These side effects should be avoided as much as possible.

When it comes to mass medicine, we must consider not only individual patients, but pools of patients. Every patient that is treated by chemotherapy depends on the resource of the hospital for the delivery of his cures. These resources consist mostly in bed time, nurse time, and doctor time. Since they are shared resource, there exists a form of competition - in the mathematical sense - to access these resources.

In practice, such competition is usually solved in a first come first serve way: every patient sets its appointments for deliveries, according to its constraints, and to the available resources. The level of criticality of time enforcement is patient-dependent as it depends on their past history, the status of their disease, their actual chemotherapy, etc. An indicator has been proposed, called the RDI to measure the timeliness of a chemotherapy regimen, and it has been shown to be correlated with the survival rate of patients for some cancer types.

Our approach suggests setting the dates of treatment deliveries based on this RDI indicator, by maximizing a global measure of the RDI among the considered pool of patients.

By combining the use of a scalable open source CBLS scheduling technology with visualization and simulation components, we were able to show the feasibility of quality indicator-driven scheduling of a large pool of patients.

At some point, the following provocative question for operations research practitioners was raised: ***is it really a good or a fair idea to install an optimizing engine in such a critical setting?*** As usual the answer is not in the technology but in the way it is used and controlled. For instance, the global formula of RDI is very critical in the sense that it might set global policies for deciding some life-or-death trade-offs in the pool of patients. If a patient has a very poor RDI for its past treatment deliveries, should he get a very high priority for his coming deliveries or should the hospital resources be allocated to patients that have higher probabilities of survival, thus abandoning the ones that have lower chances?

Clinical pathways involve an intricate decision making process and our experience shows that the scheduler can definitely support the medical actors in their work.

Our next step is to conduct on-site validation based on an extended prototype. A major request is to achieve finer tasks management, i.e. within each day. This requires to rework our algorithms to integrate the bin packing solver into the scheduling engine. On site validation should be carefully designed to ensure that at no time the global process will depend solely on non-validated tooling. As such, a first step is typically to run such a tool in parallel with the existing process and compare their behaviour and outcomes. The next step is to transmit scheduling suggestions from our tool to the existing process to check with practitioners that the suggestions proposed are applicable and contribute to improve the quality of care.

ACKNOWLEDGEMENTS

This research was partly funded by the Walloon Region by the PIPAS (nr. 1017087) and PRIMa-Q (nr. 1610088) projects. We thanks UCL/Cancer Institute, Grand Hospital of Charleroi and UZ Leuven for their valuable feedback in key elicitation, design and validation phases of our work.

REFERENCES

Ahmed, Z., T., E., and S., B. (2011). Developing an efficient scheduling template of a chemotherapy treatment unit: A case study. *Australas Med J.*, 4(10):575–88.

Barbagallo, S. et al. (2015). Optimization and planning of operating theatre activities: an original definition of pathways and process modeling. *BMC Medical Informatics and Decision Making*, 15(1):38.

Benoist, T., Estellon, B., Gardi, F., Megel, R., and Nouioua, K. (2011). Localsolver 1.x: a black-box local-search solver for 0-1 programming. *4OR*, 9(3):299 – 316.

Campbell, H., Hotchkiss, R., Bradshaw, N., and Porteous, M. (1998). Integrated care pathways. *British Medical Journal*, pages 133–137.

Choo, J. (2001). Critical success factors in implementing clinical pathways/case management. *Ann Acad Med Singapore 30 (4 Suppl)*, 7:17–21.

Damas, C., Lambeau, B., and van Lamsweerde, A. (2014). Analyzing critical decision-based processes. *IEEE Trans. Software Eng.*, 40(4):338–365.

De Landtsheer, R., Guyot, Y., Ospina, G., and Ponsard, C. (2015). Combining neighborhoods into local search strategies. In *Proceedings of MIC'2015*.

De Landtsheer, R. and Ponsard, C. (2013). Oscar.cbls : an open source framework for constraint-based local search. In *Proceedings of ORBEL'27*.

Demeulemeester, E., Sermeus, W., Belin, J., and Cardoen, B. (2007). Clinical pathways and operations management. it takes two to tango. *Review of Business and Economic Literature*, LII(3):451–470.

Di Gaspero, L. and Schaerf, A. (2003). EASYLOCAL++: an object-oriented framework for the flexible design of local-search algorithms. *Software: Practice and Experience*, 33(8):733–765.

Gooch, P. and Roudsari, A. (2011). Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems. *Journal of the American Medical Informatics Association*, 18(6):738–748.

Kaandorp, G. C. and Koole, G. (2007). Optimal outpatient appointment scheduling. *Health Care Management Science*, 10(3):217–229.

Kortbeek, N., Zonderland, M. E., Boucherie, R. J., Litvak, N., and Hans, E. W. (2011). Designing cyclic appointment schedules for outpatient clinics with scheduled and unscheduled patient arrivals.

Lyman, G. (2009). Impact of chemotherapy dose intensity on cancer patient outcomes. *J Natl Compr Canc Netw*, pages 99–108.

Marynissen, J. and Demeulemeester, E. (2016). Literature review on integrated hospital scheduling problems. Working Papers Department of Decision Sciences and Information Management 555258, KU Leuven, Faculty of Economics and Business.

Mauro, C., Happle, T., Sunyaev, A., M.Leimeister, J., and Krcmar, H. (2010). From medical processes to workflows: modeling of clinical pathways with the unified modeling language. In *Int. Conf. on Health Informatics (HealthInf)*, Valencia, Spain. 161 (9-10).

McGlynn, E. A. et al. (2003). The quality of health care delivered to adults in the United States. *New England Journal of Medicine*, 348(26):26352645.

Michel, L. and Hentenryck, P. V. (2004). Iterative relaxations for iterative flattening in cumulative scheduling. In *14th International Conference on Automated Planning & Scheduling (ICAPS)*.

Newton, M. A. H., Pham, D. N., Sattar, A., and Maher, M. (2011). Kangaroo: an efficient constraint-based local search system using lazy propagation. In *Proceedings of CP'11*, pages 645–659.

OscaR Team (2012). OscaR: Operational research in Scala. Available under the LGPL licence from https://bitbucket.org/oscarlib/oscar.

Piccart, M., Biganzoli, L., and Di Leo, A. (2000). The impact of chemotherapy dose density and dose intensity on breast cancer outcome: what have we learned? *Eur J Cancer.*, 36(Suppl 1).

van Dam, P. A. et al. (2013). A dynamic clinical pathway for the treatment of patients with early breast cancer is a tool for better cancer care: implementation and prospective analysis between 2002–2010. *World Journal of Surgical Oncology*, 11(1):70.

Van Hentenryck, P. and Michel, L. (2009). *Constraint-based Local Search*. MIT Press.

Vieira, B. et al. (2016). Operations research for resource planning and -use in radiotherapy: a literature review. *BMC Medical Informatics and Decision Making*, 16(1):149.

Walter, F., Bass, N., Bock, G., and Markel, D. (2007). Success of clinical pathways for total joint arthroplasty in a community hospital. *Clin Orthop Relat Res*, 7:457:133.