The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Decision Support Core System for Cancer Therapies Using ASP-HEX

Andre Thevapalan, Gabriele Kern-Isberner, Diana Howey
Dept. of Computer Science, TU Dortmund
44227 Dortmund, Germany

Christoph Beierle
Dept. of Computer Science, University of Hagen
58084 Hagen, Germany

Ralf Georg Meyer, Mathias Nietzke
St.-Johannes-Hospital
44137 Dortmund, Germany

Abstract

MAMMA-DSCS (mammary carcinoma decision support core system) is a prototype implementation designed for the support of decision processes for breast cancer (mammary carcinoma) treatment plans: Given a set of patient values, the system suggests different applicable treatment plans. As additional knowledge sources, MAMMA-DSCS uses external ontologies containing further information and correlations which are not directly tied to the tumor itself (e.g. toxicities, drug interactions). As a consequence, general knowledge like the substance composition of a specific therapy and its pharmacological hierarchy, can be separated from the knowledge about the applicability of therapies for a patient. The latter is encoded in an ASP program that is able to access the external ontologies and to take the obtained information into account for determining the set of all therapy plans that may be applied in a given situation. The ASP program models medical knowledge combining general guidelines and up-to-date expert knowledge for treating breast cancer on a very fine-grained level, originating from a hospital in Germany.

Introduction

Fast progress in the field of cancer research leads to a growing variety of substances, therapies and therapy types for the treatment of cancer. Today medicamentous cancer therapies entail combinations of drugs, which are applied to the patient based on temporal schemes. For the determination of a suitable treatment plan, several important attributes have to be taken into account. Beyond the recommendations given in *clinical practice guidelines (CPGs)*, medical experts may also deviate from these suggestions and define individual therapies based on their own expert experience and on the patient's specific medical situation.

Over the past 16 years the pharmaceutical department together with the clinic of Hematology and Medical Oncology of the St.-Johannes-Hospital in Dortmund, Germany, has developed a structured collection of treatment plans containing more than 45,000 therapeutical cycles for over 2,300 individual treatment situations. These plans include important information about application, dose modifications and tumor board decisions. Furthermore each treatment plan is extended with notes about behavioural rules and explanations for medical experts and patients.

Besides the reference treatment plans, also the *OCTA-Ontology* (Beierle et al. 2016) has been built up as an additional knowledge source. This ontology provides general knowledge about substances of cancer therapies and their interrelationships. The ontology classifies the different substances and drugs and contains information about the substance composition (regimen) of a therapy, rules about the administration of the containing drugs and their toxicity. The availability of sources with such large and complex knowledge about the disease and its treatment calls for the usage of computer-based support systems.

In this paper we introduce the core implementation of a decision support system for the mammary carcinoma (MAMMA-DSCS) with *Answer Set Programming* (ASP). ASP with external atoms (HEX) provides an excellent formalism for a transparent and modular representation of therapy decisions which leads to an extendable and maintainable core implementation for this type of systems. So we used ASP-HEX to implement the knowledge base of this system; first work in this regard has been done in (Thevapalan 2017). Given the medical information about a patient, MAMMA-DSCS suggests possibly applicable treatment plans for this patient according to both the information given in the reference treatment plans and the ontology. The ontology can be accessed through external atoms and allows us to separate basic medical knowledge from the expert knowledge about the cancer therapies. The patient information can be defined in a simple text file, which the system takes as an input to compute all possible treatment plans. As a specific feature of ASP, default rules involving default negation allow for modelling rules with exceptions as well as incomplete and uncertain information. This leads to a flexible and practical supporting tool for the treating doctors. To guarantee the maintainability and extendability of the system, we created a model of the decision processes in cancer therapies by considering the relationships between entities, like therapies and patient attributes, and the classification of these entities. This leads to a modular design which was accepted by researchers from the computer science field as well as the medical field.

After outlining some related work and the preliminaries needed for this paper, we describe the different types of cancer therapies, the design of a treatment plan and the relation of therapies to the tumor with its characteristics. After that we describe our modeling approach by presenting the structure of MAMMA-DSCS and the encoding of the different entities like patient values, therapies and treatment plans. Furthermore we show how external atoms allow the system to handle the aspect of contraindications independent of the actual treatment plan determination in the ASP program. After that we describe how the implementation was verified by outlining the test procedure. This procedure ensures that the program computes the treatment plans according to the given reference treatment plans. Finally we conclude our findings and present ideas for future work.

Related Work

Currently there are many approaches for implementing support systems with logic programming in the medical field. Besides cancer therapies, the handling of comorbidities with ASP is a current topic. This is addressed in (Zhang and Zhang 2014) and (Merhej et al. 2016). Here the authors define a mathematical model to represent clinical practice guidelines. CPGs contain systematically developed statements to assist the treating doctors. One goal of CPGs is to enable the delivery of high quality, evidence-based health care (Lohr and Field 1990). The approach of Zhang et al. is based on activity graphs with the goal to automatically compute treatments for patients with multiple diseases based on the corresponding CPGs. Furthermore redundancies and inconsistencies between therapies are removed by considering all possible therapies. The implementation of this model is an ASP program which takes patient information as the input data and computes a possible treatment based on the CPGs encoded as ASP rules. In contrast to our approach, general medical knowledge is encoded in the program itself. Therapies are computed by mainly considering the available information and *guessing* other patient values which are not available but required for the computation. Our approach deals with unknown information by making default assumptions which can be overridden due to the nonmonotonic properties of ASP. Furthermore their ASP program encodes every single possible treatment plan of a CPG while our approach condenses therapies and unitizes the treatment plans according to the underlying treatment type and other categories to remove redundant rules and definitions.

In (Chen 2016) Chen proposes a decision support system for the management of chronic heart failure (CHF), which contains the set of clinical guidelines encoded as ASP rules. Given a patient's medical information, the system generates recommendations for treatment plans. The system uses *s(ASP)*[1], a query-driven ASP system, to compute its models. The knowledge in Chen's system is based on the entire set of clinical practice guidelines for CHF and can deal with incomplete patient information. In contrast to our approach, aspects like contraindications are encoded directly into the ASP rules and due to the usage of the s(ASP) system, the answer sets contain default negation.

Another approach towards formalization of knowledge for clinical decision support systems is the development of *computer-interpretable clinical guidelines (CIGs)*. CIGs are based on *task model networks* which enable the specification of complex, multistep guidelines with temporal information and dependencies (Peleg et al. 2003). *GLIF* (Patel et al. 1998) and *PROforma* (Fox, Johns, and Rahmanzadeh 1998) are two frameworks which provide the creation of CIGs via a custom language. GLIF also uses knowledge in taxonomic hierachies like SNOMED to reason in. PROForma provides, like our approach, an interface for accessing external sources and implements the modeling of components with logic programming and object-oriented modeling. In our approach the treatment plans were already developed in a decision-tree-like structure. In contrast to CIG-oriented systems our system does not introduce a new format or language for the definition of CPGs. For now this kind of meta layer is redundant. The knowledge in the given mindmap was modelled exactly by the ASP, but in a more abstract and compact way than what was possible with the mindmap.

ASP and HEX Programs

Answer Set Programming (ASP) is a well developed form of declarative programming. An extended logic program is a finite set of rules of the form

$$H \leftarrow A_1, \ldots, A_n,\ not\ B_1, \ldots, not\ B_m.$$

where $H, A_1, \ldots, A_n, B_1, \ldots, B_m$ are literals, i.e. logical atoms in positive or negated form, *not* is called default negation. Each such rule may be read as: *If* $A_1, \ldots, A_n$ *all hold, and it can be assumed that none of* $B_1, \ldots, B_m$ *holds, then conclude* H. H is called *head* of the rule and $A_1, \ldots, A_n,\ not\ B_1, \ldots, not\ B_m$ its *body*. A rule without a body is called a *fact* and is identified with its head H. A rule without a head is called a *constraint*. A set of literals $X \subseteq \mathcal{A}$ of ground atoms is a model of a logical program $\mathcal{P}$, if $H \in X$ whenever $A_1, \ldots, A_n \subseteq X$ and $B_1, \ldots, B_n \cap X = \emptyset$ for every rule in $\mathcal{P}$. The concept of a stable model, or answer set, is used to define a declarative semantics for logic programs with default negation. If AS is a state, then AS $\in \mathcal{AS}(\mathcal{P})$ is the *answer set* of $\mathcal{P}$ if AS is a minimal set closed under $\mathcal{P}$ in case of a program $\mathcal{P}$ without default negation and AS is an answer set of $\mathcal{P}$ if AS is answer set of the reduct $\mathcal{P}^{\mathcal{AS}}$ in case of a program $\mathcal{P}$ with default negation. For further details of answer set programming please see e.g. (Baral and Gelfond 1994).

HEX Programs are extensions of answer set programs, admitting *higher-order atoms* and *external atoms*. The semantics of a HEX program is given by generalizing the answer set semantics (Eiter et al. 2005).

An *external atom* is of the form $\&g[Y_1, \ldots, Y_n](X_1, \ldots, X_m)$, where $Y_1, \ldots, Y_n$ is an input list, $X_1, \ldots, X_m$ is an output list and $\&g$ is an *external* predicate name. *External atoms* allow the communication of ASP programs with external resources. Thus, the program may also include external knowledge, e.g. in the form of an ontology. Therefore a plugin can be implemented in C++ or, as in our case, in Python for accessing external resources. A *higher-order atom* is a tuple $Y_0(Y_1, \ldots, Y_n)$, where $Y_0, \ldots, Y_n$ are terms.

DLVHEX System is a logic-programming reasoner for

[1] https://sourceforge.net/projects/sasp-system/

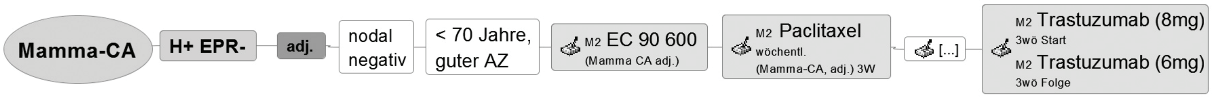

Figure 1: An exemplary path of the mindmap

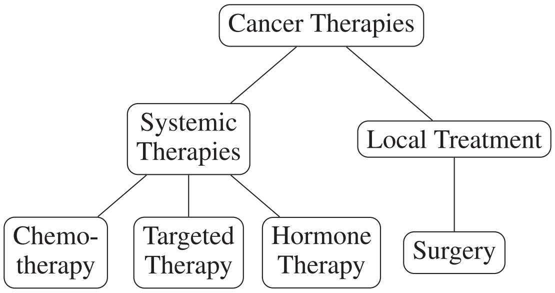

Figure 2: Therapy types

computing the models of HEX-programs. It is an extension of DLV (Eiter et al. 2000), a system implementing answer set programming which has been developed at the universities of Vienna, Austria, and Calabria, Italy. As a special feature, it also allows disjunctive information in the head of logic program rules. For our application the system computed answer sets in a couple of seconds.

Cancer therapies and treatment plans

The collection of treatment plans at the St.-Johannes-Hospital is represented in a mindmap-based structure. The nodes of the mindmap are either medical attributes or therapy names. A path from the root to a leaf represents a full treatment plan (*reference treatment plan*), containing the information about the therapies, the requirements for the applicability of these therapies and the temporal scheme, stating when and how each therapy has to be applied to the patient. In this paper, we only consider treatment plans for early breast cancer tumors which are treatable and not metastasized or locally advanced. Figure 1 shows a path which represents the following treatment plan: A patient, whose biological age is under 70, who has a good general medical condition and a *HER2*-positive, *ER/PR*-negative and node-negative tumor, can get a chemotherapy with the internal name *EC_T* followed by a targeted therapy with the internal name *H8_6*. After a patient is diagnosed with breast cancer, it is the task of the treating doctors to tailor a combination of therapies specifically for this particular patient. The planned cancer treatment together with the information about the patient's tumor is called a *treatment plan*. The determination of a suitable treatment plan is complex and requires a lot of expert knowledge and information. Depending on the tumor's biological characteristics, the stage of the breast cancer and general patient values, like the patient's biological age, general medical condition and the existence of additional diseases or disorders (*comorbidities*), different therapies and therapy types are possible (see figure 2). During systemic therapies the cancer is treated by using drugs, which can be given intravenously or orally. There are several types of systemic therapies. In this application we look at chemotherapies, targeted therapies and hormone therapies.

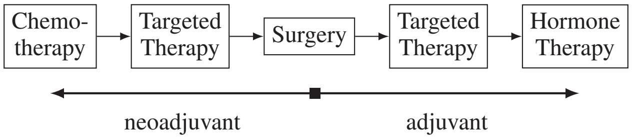

Figure 3: General structure of a treatment plan

In *chemotherapies* cytotoxic substances are used to disturb the proliferation of tumor cells. In *targeted therapies* substances attack specific cell structures of the tumor. If a patient has a HER2-positive tumor, there is an increased occurrence of the protein HER2/neu in the cell membrane of the tumor cells. The monoclonal antibody trastuzumab is a targeted therapy that interferes with the HER2/neu receptor. *Hormonal therapies* include substances which alter the activity and production of specific hormones in the body (Beierle et al. 2016).

Besides the systemic, non-invasive therapies, doctors can recommend invasive surgery as a local treatment to remove the tumor cells directly. Whether the patient is qualified for a surgery depends on various attributes including the type and size of the tumor. If possible, surgery is additionally preceded by one or more systemic therapies to shrink the tumor and ease the surgery. Likewise systemic therapies can follow after a surgery to minimize the risk of recurrence. A treatment plan for the patient usually comprises some combination of applicable systemic therapies (generally one therapy of each type) and surgery (see figure 3).

Modeling approach

The knowledge model of MAMMA-DSCS was designed by considering the medical coherence between therapy types, therapies and treatment plans. This allows a generic modeling approach and ensures that this system can be used as a prototype in other applications and domains. Figure 4 shows an overview of the model: The input of MAMMA-DSCS is the patient file and the OCTA-ontology. The Python plugin allows the usage of this input data in the ASP program, which is extended with corresponding external atoms. Furthermore the content of the extended ASP-program (ASP-HEX program) can be separated into two groups: The information about a patient p is encoded in the problem instance $\mathcal{P}_I^p$. The problem class $\mathcal{P}_C$ consists of the rules which encode the decision process. It contains the knowledge about the applicability of therapies and further

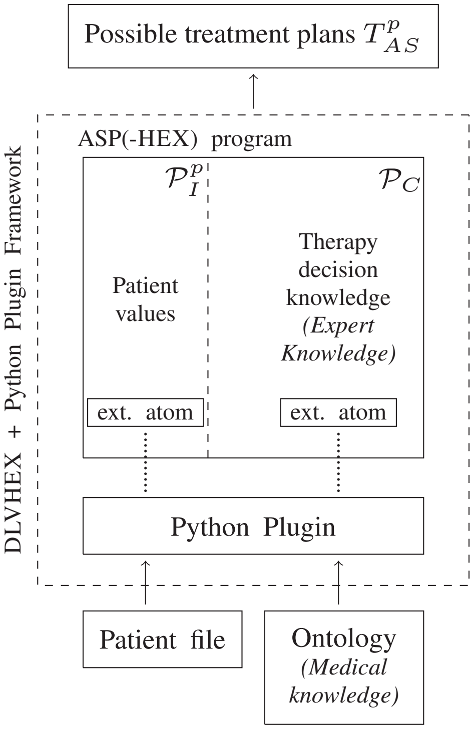
Possible treatment plans T^p_{AS}
DLVHEX + Python Plugin Framework
ASP(-HEX) program
$\mathcal{P}^p_I$
$\mathcal{P}_C$
Patient values
Therapy decision knowledge (Expert Knowledge)
ext. atom
ext. atom
Python Plugin
Patient file
Ontology (Medical knowledge)

Figure 4: Main components of MAMMA-DSCS

decision rules. For a program $\mathcal{P}^p = \mathcal{P}^p_I \cup \mathcal{P}_C$, we compute the set of answer sets $\mathcal{AS}(\mathcal{P}^p)$ as the output. An answer set $AS \in \mathcal{AS}(\mathcal{P}^p)$ represents a possible treatment plan T^p_{AS} for patient p. The treating doctors and physicans can subsequently pick the treatment plan which is most suitable for the patient or use the output for the development of a new, adjusted treatment plan.

Knowledge types: As mentioned above, we distinguish between general, domain-specific knowledge (*basic medical knowledge*) and the knowledge about the applicability of specific therapies and relations among them (*expert knowledge*). Medical knowledge contains classic logical, certain information, expert knowledge on the other hand often contains generic rules with exceptions and uncertain knowledge. The latter is encoded in the ASP program with rules like:

```
ctApplicable:-her(pos),epr(neg).
ctCandidate("EC_T"):-ctApplicable,               (1)
nodal(neg),bioAgeLower70,not -gmc(good).
```

The second rule encodes the reference treatment plan shown in figure 1 and states, that if a patient can get a chemotherapy (`ctApplicable`), does not have metastasized axillary lymph nodes (`nodal(neg)`), has a biological age under 70 (`bioAgeLower70`) and has a good general medical condition (`not -gmc(good)`), then the chemotherapy scheme *EC-T* is a possible treatment candidate (`ctCandidate("EC_T")`). The usage of default and classical negation, like in the *gmc*-atom, allows the consideration of default knowledge about the patient and the tumor.

Medical knowledge is encoded in ontologies. The ASP program can access this knowledge through external atoms. This allows the use of knowledge in external sources independently of the expert knowledge in the ASP program.

Mandatory attribute	Possible values
HER2 status	`her(pos),her(neg)`
Hormone receptor status	`epr(pos),epr(neg)`
Biological age	`bioAge([integer])`

Treatment type	`schedule(adj),`
	`schedule(neoAdj)`

Lymph node status	`nodal(pos),`
	`nodal(neg)`

Contraindication anthracyclines	`ci(ant),-ci(ant)`

Figure 5: Mandatory information in patient file

Patient model: The information about a patient p can be modeled by stating relevant attributes like tumor size or risk (e.g. *tumor size:80mm*, *risk:high*). In the ASP program these *patient values* are encoded as the set of facts $\mathcal{P}^p_I$.

In MAMMA-DSCS one can model a patient in form of a *patient file* F^p, which is a text file containing key-value pairs for the different patient values, e.g.: `her=pos`, `epr=pos` or `schedule=adj`. The Python plugin framework of DLVHEX enables the import of those key-value pairs into the ASP program as the set of atoms $\mathcal{P}^p_I$. We distinguish between mandatory attributes (see figure 5) and optional attributes. The specification of mandatory attributes ensures the proper computation of treatment plans with minimal information. The specification of all mandatory attributes is required. The definition of the optional attributes are handled as defaults, if necessary, but those additional information can lead to more refined treatment plan suggestions. For instance the example in (1) shows that in the absence of a value for the attribute *general medical condition* (`gmc`) a good medical condition is taken as the default value.

Therapies and therapy types: The ASP program encodes a set of reference treatment plans R given by the St.-Johannes-Hospital. The hospital provided these treatment plans in form of a mindmap, where each decision path encodes a unique treatment plan. We identified the three types of systemic therapies chemotherapy, targeted therapy and hormone therapy and also surgery as a non-systemic treatment. Systemic therapies are unique and can be identified by an acronym which indicates the substances used in the therapy, e.g.: The atom `ctScheme("EC_T")` encodes a chemotherapy with the substances **e**pirubucin, **c**yclophosphamide and pacli**t**axel. In the following, a set of reference treatment plans $R^p \subseteq R$ is the set of treatment plans which are possibly applicable for a patient p.

Handling contraindications: The usage of an ontology enables the outsourcing of medical knowledge out of the ASP program into an external source. This type of knowledge can then be handled independently of the ASP program. Furthermore this ontology can be utilized in other applications.

The knowledge in the ontology connected to this program contains information about the content of several chemotherapies specified in the program. This knowledge can be used to determine contraindicated therapies by comparing the substances of an applicable therapy with the contraindicated substances of the patient.

Determination of treatment plans: The computation of treatment plans consists of the following steps: (1) determination of applicable therapy types, (2) gathering of specific, applicable therapies for every applicable therapy type and (3) calculation of all possible treatment plans by assembling the possible therapies according to the treatment plans given by the St.-Johannes-Hospital. Figure 6 shows the calculated values of each step. In every step the knowledge from the patient values and from the steps before are used. The general applicability of a therapy type usually depends on the *HER2* and *ER/PR* status:

```
ctApplicable :- not ci(ct).
abtApplicable :- her(pos), ctApplicable.
ahtApplicable :- epr(pos), not gender(male).
```

In the next step we gather all therapy candidates for each applicable therapy type. The applicability of a specific therapy depends on various patient values (and in some cases on the other therapies of the treatment plan):

```
ctCandidate("EC_T") :- ctApplicable,
nodal(neg), ageLower70, not -gmc(good).
ctCandidate("ETC") :- ctApplicable,
adjctApplicable, nodal(pos), ageLower50, not
 -gmc(good), not -lndCount4to9. % [...]
```

In the last step we can compute all possible treatment plans for the patient by plugging a therapy candidate of each therapy type together according to the general treatment plan structure shown in figure 3. The selection of a therapy candidate for a treatment plan is done by following rules:

```
% set contraindicated therapies
ctIsCI(X) :- ci(ant), hasANT(X).
suggestedCT(X) :- ctCandidate(X), not
skipCT, not ctSuggested, not ctIsCI(X).
suggestedCT(X) v -suggestedCT(X) :-
ctCandidate(X), ctScheme(X).
suggestedAbT("H6") :- abtSuggested,
suggestedCT(TH), hasTrastu(TH).
suggestedAbT("H8_6") :- abtSuggested,
suggestedCT(TH), not hasTrastu(TH). %[...]
suggestedAHT(AHT) :- ahtCandidate(AHT), not
ahtSuggested.
suggestedAHT(AHT) v -suggestedAHT(AHT) :-
ahtCandidate(AHT), aht(AHT).
```

The computing of treatment plans is not done by only creating the cartesian product of all therapy candidates. In some cases the reference treatment plans showed explicit deviations and exceptional treatment plans. These informations are also encoded in the ASP program.

Test procedure and verification

The program was tested against the given mindmap, i.e., the treatment plans computed by the ASP programs must be exactly the same as in the mindmap for the respective patient data. In this way we also tested the correct implementation of the ontology integration.

For the verification of these criteria, we defined a semi-automatic test procedure, where we (1) defined test cases, (2) computed the according answer sets and (3) compared the treatment plan suggestions encoded in the answer sets with the reference treatment plans in the mindmap.

In order to ensure that we take all cases covered by the mindmap into account, we defined hypothetical patients, each consisting of a unique set of patient values. We identified 475 different patient value combinations from the mindmap which were relevant in the sense that each hypothetical patient was realistic and covered a unique case. We did that by generating all possible combinations of patient values and removing the impossible cases, e.g., the cases not occurring in the mindmap for medical reasons. For example, it does not make sense to distinguish between the menopausal status if the patient's tumor is ER/PR-negative because then a hormonal therapy cannot be applied and the menopausal status is only relevant for the selection of hormonal therapies.

First we defined a set P of hypothetical patients by creating a patient file F^p for each patient $p \in P$. In the next step we ran the ASP program $\mathcal{P}^p$ for every F^p. The answer sets of these computations were transformed into a more readable format and stored in a single result file:

```
patient-001.txt
===========================================
("her(pos),epr(pos),schedule(adj),nodal(pos)
,gmc(good),age(45),ci(ant)")
===========================================
("1.CT(ETC),2.AbT(H8_6),2.AHT(TMX10)")
-------------------------------------------
("1.CT(TCH),2.AbT(H6),2.AHT(AI5_TMX5)")
-------------------------------------------
("1.CT(TCH),2.AbT(H6),2.AHT(TMX10)")
-------------------------------------------
("1.CT(ETC),2.AbT(H8_6),2.AHT(AI5_TMX5)")
===========================================

patient-002.txt
===========================================
("her(pos),epr(pos),schedule(adj),nodal(pos)
,gmc(good),age(45),-ci(ant)") %[...]
```

In the last step we identified the set of reference treatment plans R^p for every test case p. A test case p is successful, if every reference treatment plan $R^p_i \in R^p$ from the mindmap matches a computed treatment plan $T^p_j \in T^p$ and vice versa. This means that for every test case p every single computed treatment plan $T^p_j \in T^p$ can be found as a path in the mindmap and every path in the mindmap is encoded correctly in the ASP program. Intuitively the verification of a test case ensures that given a patient p, MAMMA-DSCS computes all treatment plans exactly according to the hospital's reference treatment plans. The program is correct, if every test case $p \in P$ is successful.

The results of the testing procedure show that MAMMA-DSCS computes all treatment plans exactly according to the treatment plans of the hospital. Consequently the implementation of the proposed model proved to be successful.

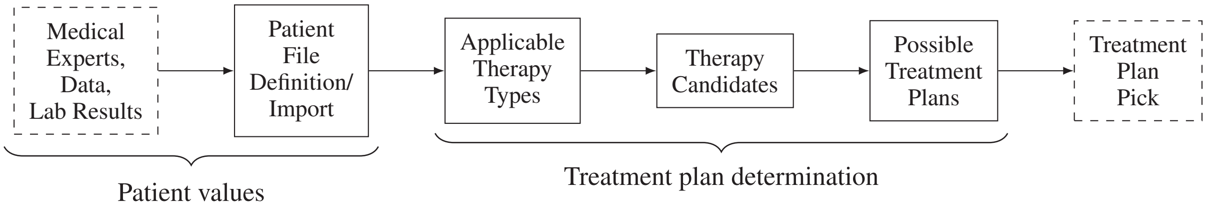

Figure 6: Complete process and determination of treatment plans

Conclusion and Future Work

MAMMA-DSCS is a prototype implementation which shows a concrete application of recent developments in answer set programming with external sources. We used current real-life data and prepared a framework for the usage of clinical decision support systems in oncological domains. The system computes all possible solutions for a patient. The treating doctors can use these suggestions to create the most suitable treatment plan for this patient. Knowledge in external sources can also be accessed and used for the decision process and for the enrichment of information in each solution. Our verification process shows that the system computes all treatment plans according to the hospital's data and ensures that no unknown (not existing) treatment plans are introduced.

In the future the matching of computed treatment plans and reference treatment plans should be extended to get a fully automated testing procedure. Also the encoded knowledge should be extended by expanding both the existing ontology and the ASP program and by adding more knowledge sources. Conceivable aspects of further functionalities could be the analysis of drug interactions and the refinement of treatment plans by adding additional statistical data. Furthermore information about current studies regarding a suggested therapy should extend the treatment plan suggestion for the treating doctors.

This paper is the first step towards setting up a system for recommending cancer therapies that works on a declarative, more abstract base than the mindmap which is currently (already very successfully) used in the clinical routines of the St.-Johannes-Hospital, and that is supposed to be maintained and updated more easily. So, next steps will involve investigations with respect to adopting new patient information, due to integrating new medical knowledge. We will compare the amount of change in the ASP program with the amount of change that would be necessary to adapt the mindmap.

References

Baral, C., and Gelfond, M. 1994. Logic programming and knowledge representation. *The Journal of Logic Programming* 19:73–148.

Beierle, C.; Eisele, L.; Kern-Isberner, G.; Meyer, R. G.; and Nietzke, M. 2016. *Using Ontological Knowledge About Active Pharmaceutical Ingredients for a Decision Support System in Medical Cancer Therapy*, volume 9904 of *Lecture Notes in Artificial Intelligence*. Springer. 119–125.

Chen, Z. 2016. Automating Disease Management Using Answer Set Programming. In Carro, M.; King, A.; Saeedloei, N.; and Vos, M. D., eds., *Technical Communications of the 32nd ICLP 2016*, volume 52 of *OASIcs*, 1–10. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Eiter, T.; Faber, W.; Leone, N.; and Pfeifer, G. 2000. Declarative problem-solving using the DLV system. In Minker, J., ed., *Logic-based Artificial Intelligence*. Norwell, MA, USA: Kluwer Academic Publishers. 79–103.

Eiter, T.; Ianni, G.; Schindlauer, R.; and Tompits, H. 2005. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 90–96.

Fox, J.; Johns, N.; and Rahmanzadeh, A. 1998. Disseminating medical knowledge: the proforma approach. *Artificial Intelligence in Medicine* 14(1):157 – 182.

Lohr, K. N., and Field, M. J. 1990. *Clinical practice guidelines: directions for a new program*, volume 90. National Academies Press.

Merhej, E.; Schockaert, S.; McKelvey, T. G.; and De Cock, M. 2016. Generating conflict-free treatments for patients with comorbidity using ASP. In *KR4HC '16 : Proceedings of the 8th International Workshop on Knowledge Representation for Health Care*, 93–100. Springer.

Patel, V. L.; Allen, V. G.; Arocha, J. F.; and Shortliffe, E. H. 1998. Representing clinical guidelines in GLIF: individual and collaborative expertise. *Journal of the American Medical Informatics Association* 5(5):467–483.

Peleg, M.; Tu, S.; Bury, J.; Ciccarese, P.; Fox, J.; Greenes, R. A.; Hall, R.; Johnson, P. D.; Jones, N.; Kumar, A.; et al. 2003. Comparing computer-interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association* 10(1):52–68.

Thevapalan, A. 2017. ASP-HEX programs for the use of ontologies in a decision support system for cancer therapies. Master's thesis, TU Dortmund, Dortmund, Germany.

Zhang, Y., and Zhang, Z. 2014. Preliminary result on finding treatments for patients with comorbidity. In Miksch, S.; Riaño, D.; and ten Teije, A., eds., *Workshop on Knowledge Representation for Health-Care Data, Processes and Guidelines*, 14–28. Springer.