Medical Workflow Design and Planning Using Node-RED Data Fusion

by

Lisa Ewen
Master's of Computer Science, Lakehead University, 2021

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTER'S OF COMPUTER SCIENCE

in the Department of Computer Science of Lakehead University

Supervisory Committee

Medical Workflow Design and Planning Using Node-RED Data Fusion

by

Lisa Ewen
Master's of Computer Science, Lakehead University, 2021

Supervisory Committee

Dr. Sabah Mohammed, Department of Computer Science
Supervisor

Dr. Arnold Kim
Co-Supervisor

Dr. Jinan Fiaidhi, Department of Computer Science
Departmental Member

Dr. Carlos Zerpa, Department of Kinesiology
Outside Member

Abstract

The space of clinical planning requires a complex arrangement of information, often not capable of being captured in a singular dataset. As a result, data fusion techniques can be used to combine multiple data sources as a method of enriching data to mimic and compliment the nature of clinical planning. These techniques are capable of aiding healthcare providers to produce higher quality clinical plans and better progression monitoring techniques. Clinical planning and monitoring are important facets of healthcare which are essential to improving the prognosis and quality of life of patients with chronic and debilitating conditions such as COPD. To exemplify this concept, we utilize a Node-Red-based clinical planning and monitoring too that combines data fusion techniques using the JDL Model for data fusion and a domain specific language which features a self-organizing abstract syntax tree.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to extend my sincere gratitude to my supervisor, Dr. Sabah Mohammed, who has continued to believe in, support, and provide patience with me over the years he has known me. I am also extremely grateful for my co-supervisor, Dr. Arnold Kim, who has been a positive influence in both my work and my daily life, and without his guidance and support I would not be where I am today.

Chapter 1: An Introduction to Clinical Planning, COPD and Data Fusion-based Design

1.1: Clinical Planning

All physicians and healthcare providers are required to participate in clinical planning either collaboratively or on their own. Clinical planning involves using diagnostic and other medically relevant information about a patient to discern a set of tests or medications to order, or instructions the patient must follow to aid in a diagnosis or improve their overall health. As a result, the process of clinical planning is a very mentally strenuous task that involves a large number of decisions to be made based on the information at hand. In addition to the mere step of planning a course of action for a patient, there are many administrative and procedural tasks to enact this plan required by a hospital or clinic to ensure the safety of patients and physicians in this process. These extra steps can add more strain on the healthcare provider, which can eventually result in poorer decision making as a result of physician burnout [36]. While many technological systems have attempted to counteract this, most have greatly contributed to this phenomenon of physician burnout [36]. While research has been done to try to explain why these systems are ill-fitted for clinical practice, little progress has been made to improve or revitalize these systems as a result.

1.2: COPD

Among clinical planning duties, chronic diseases often weigh the most heavily on healthcare providers due to their complexity and their long-term effects [35]. One such chronic disease, COPD, can be more or less severe depending on how well it is managed by both the patient and the physician. As a result of this, we can deduce that a higher quality clinical plan for a patient with COPD will most likely result in a patient facing a higher quality of life, and potentially a longer life-span than they would with a poorly designed clinical plan. It is obvious that in terms of COPD, high quality clinical plans are essential to providing a patient with the best care and quality of life possible.

1.3: Data Fusion

Data fusion is a series of techniques used to combine a multitude of datasets for further processing. This data is not required to be of the same type, or in the same format, and can be used for big data or machine learning processes after undergoing data fusion. In many ways, the concept of data fusion mimics the way that humans make decisions. We do not often decide what to wear based on one single data point, such as the presence (or lack thereof) of rain. Most commonly we make our clothing decisions based on the temperature, the weather, where we are going, whether or not we will be indoors, etc. These are all different types of data that we are “fusing” in our brain in attempt to choose the best thing to wear for our specific circumstances. Similarly, data fusion combines data of different types from different sources in order to provide a more well-rounded data set capable of providing more accurate or powerful outcomes when combined with other techniques.

1.4: A Data-Driven Solution for Complex COPD Clinical Planning and Monitoring

In the following chapters, we will discuss a tool driven by data fusion that may aid physicians in the clinical planning and monitoring process for patients with COPD. This tool features the ability to create workflows that may be able to model physicians’ personal workflows when creating a clinical plan for their COPD patients. These workflows are equipped with the ability to connect with a variety of data sources, perform data fusion and machine learning to produce clinical plans, and alerts that can be customized to monitor different symptoms or vital signs.

This tool relies on a Domain Specific Language (DSL) specific to the domain of clinical planning using data fusion to ensure the workflows maintain a structure possible of creating clinical plans and monitoring patients without a significant need for the physician to have an understanding of the system to be able to do so. The overall aim of the DSL is to provide an infrastructure that is flexible, unlike many popular technology systems currently used in a clinical setting, without giving the user too much freedom which requires the user to invest a significant amount of time learning how to use the system.

The tool also heavily features data fusion as a way to provide suggestions for a physician to utilize in their clinical plan based on information available to them, such as a patient’s health history, lab results, and more generic sources such as a Canadian medications database. The data fusion process is intended to provide an improvement over previous machine learning or big data techniques that are incapable of amassing all relevant data for use when formulating a clinical plan. As a result, the accuracy of a clinical plan produced is often dependent on the physician’s individual workflows, but has the capacity to provide meaningful plans to aid the physician in their daily clinical tasks.

Chapter 2: A Review of Data Fusion in Clinical Planning

2.1 Introduction

Healthcare planning is an extraordinarily difficult aspect of healthcare, and healthcare professionals are often required to use a vast amount of knowledge to make appropriate decisions for their patients. With the large amount of data available to healthcare professionals, it is clear that much of that data may be able to support them in their planning processes to potentially improve their speed an accuracy.

This is particularly the case in situations when critical decisions must be made within small timeframes. In these circumstances, healthcare professionals must take the information about the patient in question, information they have learned from previous patients, and information pertaining to guidelines relevant to the patient they are caring for in order to make informed decisions for their patient's care. The process of acquiring all the relevant information needed to formulate a plan can often be time consuming and difficult due, in part, by the consideration of multiple sources of information. Additionally, not all the required information is readily available or can be accessed quickly, thus, quick workflows and high workloads make it ever more difficult to consider all the necessary sources in a timely manner when preparing a clinical plan. It makes sense, then, to incorporate a tool capable of supporting healthcare professionals in their decision-making efforts.

Previous efforts to aid physicians and healthcare professionals during their strenuous decision-making processes, most notably in the sector of big data and predictive analytics, have been moderately successful [10]. Despite this success, it is important to note that big data does not completely encompass the complex nature of decision making as it is limited in the number of data sources it is capable of referencing during processing. Because of this, the decisions made may only focus on one area of the knowledge that is essential for making decisions. These decisions are often extremely important, such as considering what treatments to recommend for a given condition. Should we only consider the treatments used successfully for all past patients with the same condition, we are discounting other important information such as allergies, other conditions, or any special circumstances unique to the current patient. While this system may succeed for many patients with typical presentations, we can see many areas of weakness of a system that is not capable of considering more than one source of information. In this way we either fail to successfully produce plans useful for a given patient, or the physician is required to incorporate their own knowledge in combination with what is provided by the system, which is not a significant change from the existing workflow in healthcare planning.

This leads to the discussion of data fusion: the set of techniques that integrates multiple data sources with an overall goal to improve data quality, reduce uncertainty, and provide statistics. In the context of healthcare, every single data source present is important as it has the capability to reveal information about a patient's health at different levels of granularity. We can also acknowledge that these data sources are heterogeneous in nature, due to the variation in size, formatting, and noise levels, making other methods of analysis on this data exceedingly difficult. Due to this heterogeneity, however, we can see how data fusion lends itself very useful in the area of medical planning. Incorporating more heterogeneous sources of information more closely imitates the way information is gathered on behalf of a typical healthcare practitioner to make decisions. As per the definition of data fusion, we can mimic the way the human brain assembles information from multiple sources to make a clinical plan for a given patient [1]. If we were to use only homogeneous data sources, we limit the types of data that may usually be considered in a standard workflow when making decisions regarding a particular patient. In the following section we will give a more concrete definition of data fusion, discuss the current literature surrounding the use of data fusion in healthcare planning, and provide a problem definition.

2.2 A Definition of Data Fusion

As mentioned in the previous section, it was noted that data fusion is a set of techniques used to combine data sources for the purposes of obtaining improved information. Further, we can define data fusion upon the different standards present in the field. Among these standards we see classifications according to the relationships between data sources: the Dasarathy Model, the Waterfall Model, the Omnibus Model, the Boyd Control Loop, and the Joint Directors of Laboratories (JDL) Model.

One of the earliest, and most popularly utilized, models is the JDL model [13] which can be seen in Figure 1. This model operates using five different levels of data fusion:

- Level 0: the task of source preprocessing including fusion at the lowest forms (i.e. signal and pixel levels).
- Level 1: the task of object refinement where data from the previous level is employed for estimation and prediction.
- Level 2: the task of situation assessment where a higher level of inferences are made than in the previous level and relationships between objects are established

- Level 3: the task of impact assessment where the impact of the detected activities from the previous level are evaluated to obtain an assessment of possible risks, vulnerabilities, and predictions of possible outcomes.
- Level 4: the task of process refinement in which the process from levels 0 to 3 are improved.

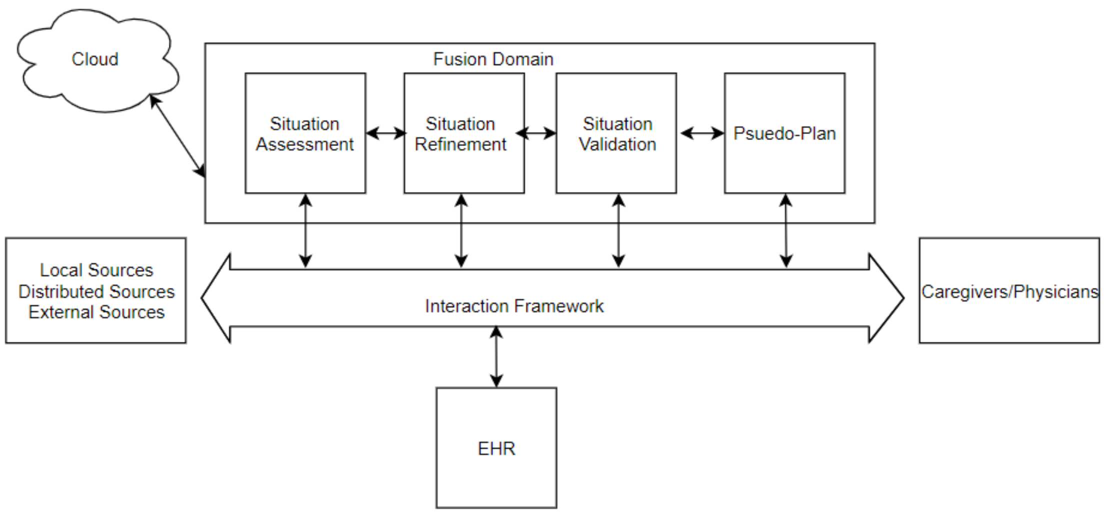

Figure 1: A clinical perspective of the JDL fusion levels

While the JDL model is the most popular model, the most well known is the Dasarathy Classification [15]. Similarly, this model also features five categories:

- Data in-Data out: raw data are the inputs and outputs with the results being typically being more accurate.
- Data in-Feature out: raw data is the input and features or characteristics about the data are extracted.

- Feature in-Feature out: features are the inputs and outputs with the goal of refining, improving, or obtaining new features.
- Feature in-Decision out: features are input with the goal of providing a set of decisions as the output.
- Decision in-Decision out: decisions are the inputs and outputs where the decisions are fused to obtain new or better decisions.

Similar to the Dasarathy Classification, instead of focusing on levels, Durrant-Whyte [16] proposed classification criteria for the relations of data sources. The three criteria include:

- Complementary fusion: input sources representing different pieces of information that can be combined to create a more complete set of information.
- Redundant fusion: features two or more inputs providing the same information which can be fused to increase confidence levels.
- Cooperative fusion: the combination of provided information to create new information that is more complex.

Proposed in a similar timeline, the Waterfall model proposed by Harris [14] describes the flow of data operates from data level to decision making level making use of information that arrives via the decision-making module. This model features 3 levels:

- Level 1: handles the transformation of raw data to acquire the necessary information from the environment itself via sensors.
- Level 2: where feature extraction and fusion of these features takes place, outputting a list of estimates and their associated probabilities.

- Level 3: utilizes the information gathered, human interaction, and any other available data sources to produce possible routes of action.

Among some of the previously discussed models we have seen a relatively linear progression of the data as it moves through each stage. The Boyd Control loop [12] features a more circular flow of the data as it moves through its four phases: Observer, Orient, Decide, and Act. The information following the Act phase is output to the environment, sensors, and actuators in which the loop is capable of starting over again.

The most modern of these models, the Omnibus Model, was introduced by Bedworth and O'Brien [11] built around the Boyd Control Loop. The original four phases of the Boyd Control Loop are still present within the Omnibus Model with modifications. A flow chart is given by Figure 2 to describe the Model.

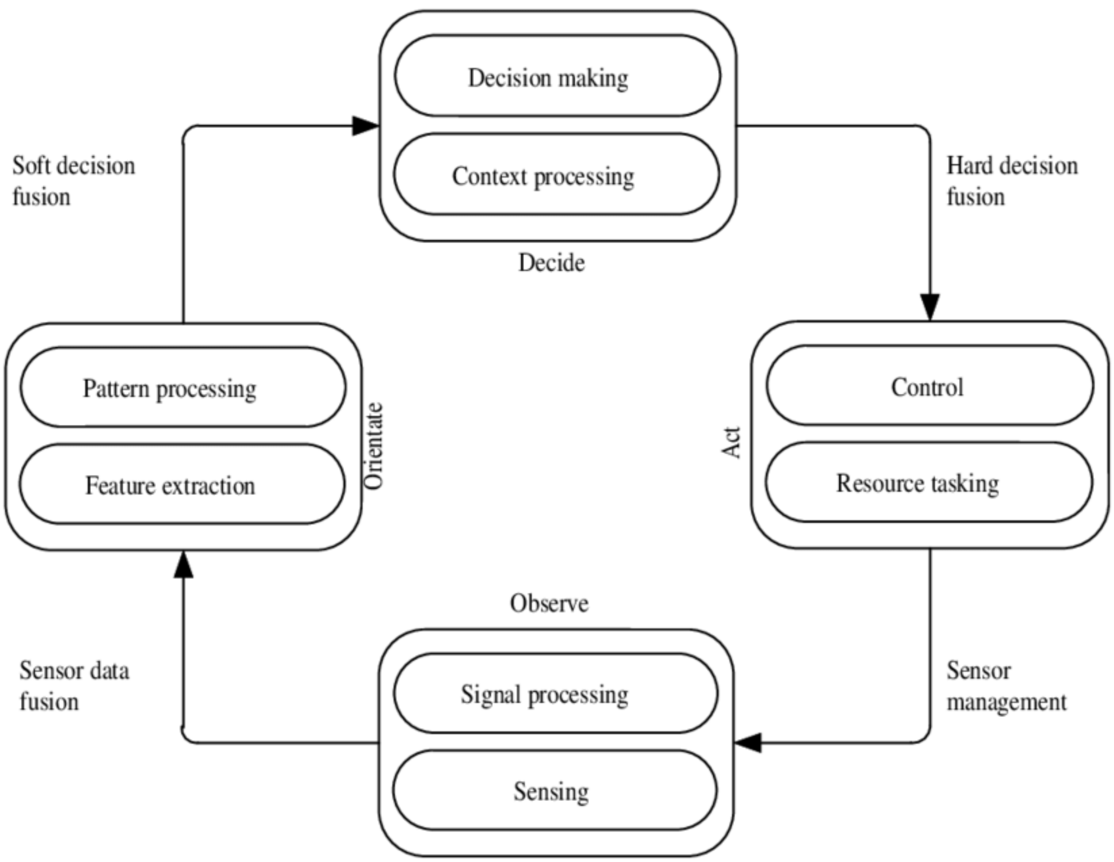

Figure 2: Omnibus Model

While each of these models feature robust capabilities, it is important to consider these models in the context of healthcare. While the Boyd Control Loop is a closed loop capable of acting in meaningful ways, there is little feedback from later stages of the loop on earlier stages to guide decision making. This results in little refinement capabilities utilizing information learned at a later stage that may strengthen or improve the outcome when applied to earlier stages. This is especially troubling in the context of clinical planning as refinement is essential to ensuring that the output decisions properly consider all the information needed and a meaningful plan is resulted. The Waterfall Model does improve this issue by allowing some feedback among each of the levels, but we can see that the JDL model has its refinement stage completely outside of the fusion domain. This provides the output plans with a much higher level of interconnectivity between each of the stages and a more accurate outcome based on the input data. As a result, we will be using the JDL model as a basis for our data fusion processes.

2.3 Related Research

Previously, discussion regarding medical planning using data fusion was merely conceptual [2]. Hall and Llinas appear to be a well-known early source which briefly conceptualizes how data fusion can be applied to diagnostic and medical applications, however, they do not provide a specific approach on how to perform such a task. Much earlier than this discussion, we see a primitive real-time data fusion tool, ICM [3]. ICM features basic data fusion that combines 4 different data: diastolic pulmonary artery pressure, central venous pressure, blood pressure, and heart rate. With these data points, both filters and a decision making algorithm are applied to provide physicians with real-time diagnoses, suggestions, and alerts for patients who are being monitored in an intensive care unit. The authors discuss preliminary testing that shows evidence that diagnoses were made comparable to expert clinicians within three to five minutes. Despite the impressive work done by the authors, no modern improvements have been made to this approach to include web or cloud data, EHR data, or any of the more recent additions to technology since the publication of this paper that may improve overall accuracy.

In more recent research, we see an emergence of data fusion in the field of medicine for surgical-based plans [4, 5, 6, 7], Alzheimer's detection [8], and orthodontic treatment planning [9]. These approaches provide excellent insight into approaches for image data fusion that make use of the more modern techniques currently available and provide great results.

An interesting application of data fusion can be seen regarding computer-assisted navigated surgery. Nemec et. al. describe a technique of MRI and MDCT image data fusion for surgical planning to be used during computer-assisted navigated surgery of orbital tumors [4]. The authors discuss a fusion process on the imaging to create CT-MR fusion images which then aid in creating a plan that is later used for intraoperative image guidance. As a result of this study, the authors conclude that the accuracy of the navigation unit due to the preoperative plan was 1.35 mm, with at least some improvements shown in 90% of the 10 patients the technique was tested on.

Similar to the previously discussed research, Reynier et. al. propose an approach of data fusion on MRI and transrectal ultrasound (TRUS) imaging to improve the process of prostate brachytherapy [7]. The authors describe that using TRUS imaging alone is difficult for prostate visualization, and that MRI provided more sensitive information.

Because open MR is prohibitive for most centers, fusion of MRI and TRUS imaging is suggested in their work. The quantitative results from this study are close to 1mm in average and provide overall positive experiences for the surgeons utilizing the fused images for their surgical plans.

When discussing Alzheimer's disease, experts in the field express that early diagnosis and subsequent early treatment can help slow down the progression of the disease. It goes without saying that due to these findings, it is important to clinicians to be able to diagnose Alzheimer's as early as possible. Polikar et. al. provide a data fusion approach which attempts to provide a method for early diagnosis of Alzheimer's using EEG data [8]. The authors use an incremental learning algorithm, Learn++, to perform the data fusion on the EEG data which seems to perform at the same level, if not better than, current non-invasive evaluations of Alzheimer's detection and diagnostic. The author's also note that the approach is extremely cost effective in comparison to other clinical evaluations currently present for early diagnosis of Alzheimer's.

In the field of orthodontic treatment planning, Hong-Tzong Yau, Tsan-Jui Yang, and Yi-Chen Chen discuss the shortcomings of traditional plastic models which provide only surface level information of the teeth and may result in unexpected outcomes during treatment [9]. They describe an approach that uses data fusion to integrate scanned data of the teeth and CT imaging by gathering slope-based detection, the level set method, the Marching Cubes Algorithm, and a contour detection algorithm to reconstruct the teeth into a 3D model. The results provide a very accurate 3D model of the teeth that can significantly aid orthodontists in predicting a patient's outcomes during treatment and help them formulate a more informed treatment plan.

2.4 Data Fusion for Clinical Planning

It is important to note that, despite the impressive work by the authors described in the previous section, many of the surgical-based approaches do not incorporate data fusion directly into an automated planning process. Much of this research demonstrates the use of data fusion to provide information that is subsequently useful for surgeons or other healthcare providers in creating their surgical plan, but does not directly provide or attempt to directly construct that plan. This is also the case present in the research of Yau, Yang, and Chen [9] where data fusion is used to create an in-depth model of the patient's teeth to provide the orthodontist with the information needed to create a treatment plan for the patient. As we have seen in the early work on ICM, it is certainly possible to create a tool that can take the data after fusion and make decisions at a similar rate as expert clinicians very quickly. We can see that despite the sophistication of the fusion that is being done, little research has explored how to use that data for a more automatic planning process.

Another major point to focus on is that the applications of many of these approaches are for very specific cases and, as a result, are not multi-purposeful. While this work performs excellently for the sole task it is designed for, there is a significant lack of a more adaptable tool that is capable of producing plans and diagnostic processes for a wider or more general range of ailments. This is also seen in the case of ICM, which is only capable of making diagnostic decisions regarding heart conditions. Should a healthcare provider be presented with an unknown, uncommon, or an unusual presentation of a condition, it can be extremely difficult to create a meaningful plan to diagnose and subsequently treat the patient in their care. That is an area in which all of the previous research lacks.

In addition to the strictness of the applications, we see that there is also a lack of variability in the data sources being considered. There is a significant focus on image data fusion within these applications, with only two of the discussed works performing fusion on other data types. Of course, the tools which utilize image data alone were designed for applications where image data was the primary source of data required for decision making and guiding the healthcare providers during treatment. That being said, other data is often very important regardless of the application, such as the patient's EHR data, vitals and other physiological data gathered by sensors, data from other patients with similar conditions, etc. These are things physicians may be aware of at the time of planning and treatment, however, it may make a planning tool more effective when including these data points in the fusion process. This is especially true when discussing more general applications, in which considering data from patients with similar symptoms, a patient's previous health history, as well as their current vitals, imaging and other tests may lead to a meaningful diagnosis and more informed, robust treatment plan.

Given what has been presented in current research regarding data fusion used in clinical planning, we can derive key factors important for effective clinical plans. The first requirement is to incorporate decision making mechanisms within the clinical planning tool. Of course, the final decisions and determinations on the usefulness of a produced plan is left with the healthcare provider. That being said, there is plenty of information within all of the combined data sources to produce clinical plans that should be utilized to maximize the utility of such a clinical planning tool.

The second important inclusion for a tool using data fusion for clinical planning is to allow more flexibility to be able to apply the tool to a wider range of situations. With a broader scope of capabilities, this type of tool can be used for a larger number of patients, increasing the likelihood that a healthcare provider can use it during clinical practice. An effective clinical planning tool using data fusion must be able to create plans regarding a relatively wide range of ailments, much like many of the healthcare providers treating a large number of patients are required to.

Finally, the third consideration is of the data sources themselves. There is a large number of knowledge sources required for any healthcare provider to make a meaningful plan, and there are very few instances when one specific type of information (i.e. vitals, imaging, previous cases, etc.) is used to make an entire plan. Some pieces of information may hold more weight than others under certain circumstances, but all information available to a healthcare provider is typically useful in their clinical planning. As a result, it is important to include a variety of data sources in the process of data fusion to ensure that the plan produced considers all relevant information during its formation.

2.5 Problem Definition

From the discussion in this chapter of existing research, we can see that there are many positive aspects to maintain throughout any data fusion endeavour. Most notably it is important to include the feature of real-time feedback and have high agreement between the plans of experts in the field and those generated by the clinical tool.

With that being said, as has been demonstrated in previous parts of the chapter, the capabilities of data fusion extend beyond what current research has shown. In order to properly execute this extension, we will have to address the issue of a narrow scope of both data types and potential use-cases of the clinical planning tool, as well as the lack of automaticity present in the planning aspects of data fusion tools.

In order to address expansion of data types, it is required to expand the data sources to include EHR data, cloud and web data, clinical care pathways, and sensor data to be included for use in the fusion domain. Adding more data sources will also aim to expand the number of use-cases that can be applied, as with a robust level of data sources we have more variables to consider that will make the tool more flexible and able to be used under a wider range of clinical circumstances.

When discussing automaticity, it is important to note that we are not aiming to entirely replace manual decision making from clinical workflows when making clinical plans. The goal is to remove the lower level processes required of physicians to make decisions and formulate plans by using data fusion in combination with decision making algorithms that a healthcare provider can then use as an effective starting point for more higher-level decisions that may need to be made for any given patient. Using these simple adjustments to existing data fusion research while still maintaining the rapid feedback and high agreement between the tool and expert opinions it is possible to create a very powerful and versatile tool.

Chapter 3: Methodology for Clinical Planning of COPD Cases Based on Cross-Domain Data Fusion

3.1: Introduction

To provide an effective data fusion-based medical planning platform, it is important to incorporate a few enabling techniques to allow the platform to provide clinical inferences including prognosis. In this chapter, we are describing a new methodology that captures the semantic context of clinical cases through the use of a domain specific language (DSL) that describes these cases and is used to guide, monitor, and infer the progression of the clinical cases through the linkage to dynamically evolving patient data that are updated from different sources including repositories over the cloud or sensors that are hooked to the patient(s). Additionally, we are featuring cross-domain data fusion, which does not feature the traditional schema mapping and data merging that is present in conventional data fusion. Instead, cross-domain data fusion features datasets from different domains, knowledge extraction of each of these datasets, which is followed by knowledge fusion – the process of combining multiple sets of information to form new information. Moreover, our method, and later, the platform, need to be designed to be useful and meaningful to physicians and clinicians following the progress of these clinical cases. To show the effectiveness of our methodology, we decided to focus on Chronic Obstructive Pulmonary Disease (COPD) as it is a progressive type of chronic disease which can worsen over time. However, COPD is treatable with proper management and planning, as most patients with COPD can achieve good symptom control and quality of life, as well as reduced risk of other associated conditions (e.g. heart disease, lung cancer).

3.1.1: The Importance of Clinical Planning for COPD:

It was reported in a study on the global burden of COPD that in 2015 that there were 210 million cases of COPD globally [26]. This is significant because, while it is treatable, COPD is currently an incurable disease. Because of this fact, patients diagnosed with COPD require long-term care to prevent symptoms from worsening over time. It is obvious, then, that the higher quality of care those with COPD receive, the better quality of life those patients will have by being able to slow down the progression of the disease.

As a result, it makes monitoring and proper management of COPD extremely vital for these patients. Without adequate monitoring or treatment of patients with COPD, it has been demonstrated that not only can the disease progress significantly faster, but untreated COPD can also result in heart problems and significant respiratory infections [24]. We can also see that even with proper management, the rate of progression of COPD is relatively difficult to predict and is measured in a 5-year survival rate of 40% to 70%[25]. This results in needing to monitor COPD as closely as possible using the best techniques available.

In order to strengthen COPD monitoring and clinical planning, it seems important to begin involving technology to more adequately monitor the disease by incorporating ideas such as data standardization and increased access to patient data and history [23]. More than that, we propose that by using data fusion techniques, the available data will be more robust, and, with the help of a DSL, is capable of improving the process of monitoring and creation of clinical plans for patients with COPD with regards to both time and quality.

3.1.2: Surveying Literature for Clinical Planning Methods

In Chapter 2, we discussed the literature regarding data fusion in clinical planning. This critical assessment provided important research done in the area of data fusion, and how it can be used to aid clinical planning. In this section we will discuss literature in the domain of data fusion that is not directly applied to clinical planning, and how these concepts can be applied to clinical planning methods.

The first of these resources to discuss is Yu Zheng's review methodology of cross-domain data fusion. Zheng outlines three main categories of data fusion techniques: stage-based data fusion, feature-level-based data fusion, and semantic meaning-based data fusion [27]. Stage-based data fusion involves the loose coupling of datasets using different datasets at different stages of mining tasks without any relation to the content of the data. Feature-level based data fusion involves the concatenation of each dataset which can be used in clustering or classification tasks. Semantic meaning-based data fusion is more focused on the meaning of each feature, something ignored in feature-based data fusion, and attempts to relate the datasets with more meaning than simple concatenation. This type of data fusion is made up of multi-view-based, similarity-based, probabilistic dependency-based, and transfer learning-based data fusion. Overall, Zheng outlines that the most powerful forms of data fusion are the four types of semantic meaning-based data fusion, as there are often strong correlations between features, but often suffers from performance issues. We can also see problems arise when attempting to fuse dynamic and static datasets as static features often are ignored when using certain types of data fusion techniques. As a result, feature-based data fusion will be used to address these concerns.

Additionally, another resource important to the discussion of clinical planning methods using data fusion is Sarvesh Rawat and Surabhi Rawat's hybrid methodology for multi-sensor data fusion. Most notably, Rawat describes rough sets as a way to discover ambiguity and remove redundancy from datasets [28]. These rough sets act as feature reduction and pre-processing layer allowing for a higher accuracy when used for backpropagation neural networks in comparison to using a backpropagation neural network on its own.

This research helps outline that in order to create an effective clinical planning tool that makes use of data fusion, utilizing a pre-processing layer combined with a feature-based data fusion method provides a strong basis for the methodology. In our case, instead of using rough sets as our pre-processing layer, we have opted to use a DSL. As we are utilizing the JDL method as our base, we will also require this DSL to be dynamic in order to provide updates to the end user as the workflows defined in Node-Red progress. This will require us to use a self-organizing abstract syntax tree to best optimize the interactions between our DSL and the data fusion workflows [29].

3.1.3: Providing Healthcare Providers with Meaningful Clinical Planning Methods

While we have discussed creating clinical plans, and how it is best to do so with data fusion, it is important to discuss what a meaningful clinical plan is. It is also important to describe how best to formulate a clinical plan so that it is meaningful.

To begin, a meaningful clinical plan is one that describes how best to treat and monitor a patient using all appropriate diagnostic tools, medications and treatments, and monitoring methods. A meaningful clinical plan must involve elements that are individual to the patient, and should encompass the severity of their disease as well as properly meets their respective needs. This type of clinical plan is essential in the context of COPD, as without a plan capable of identifying the unique presentation of the disease for each individual, a patient is more likely to worsen over time.

To be able to produce such a plan, it is important to consider the Advance Care Planning model utilized by patients when describing their wishes or instructions to be given to a proxy [22]. Following this type of model, a healthcare provider should first consider what the patient wants or needs. It is important to acknowledge their personal beliefs and desires to allow them to be a part of their own clinical plan. A patient that is more involved in their own treatment plan is more likely to follow it, which results in a better quality outcome [21].

The second facet of a meaningful clinical plan is considering the unique needs of a patient before focusing on more general aspects of their care plan, such as giving the patient pure oxygen for a period of time to improve their O_2 saturation. By beginning on focusing what is specific to the patient, a more meaningful plan can be presented that addresses what may be unusual about their care plan in comparison to another patient's, such as allergies to typical medications, severe presentations of symptoms, symptoms not typically associated with COPD, diagnoses of other diseases or illnesses, etc. This not only prevents these things from being forgotten over the course of the clinical plan, but ensures that the plan is the best fit for the patient.

Once the special considerations of the patient have been acknowledged, then it is appropriate to begin considering the more basic and typical aspects of a patient's plan. These are things that are common, and thus, are easy to add into the clinical plan during the final stages of planning. Alternatively, if these things were considered first, and only later were the patient's wants and special requirements factored in, this would most definitely result in either a poor clinical plan, or require the healthcare provider to backtrack in order to provide a proper plan.

With these steps, healthcare providers are capable of creating meaningful plans. Much in this way, we aim to emulate this process using the DSL and data fusion techniques which will be discussed more thoroughly in a later section.

3.1.4: Searching for an Emerging Framework for Clinical Planning

While programming environments used to generate typical code for general use languages like Java and Python are equipped with many libraries capable of gathering data sources, performing data fusion, and completing machine learning tasks, it is important to acknowledge that these types of programming environments are not always optimal for more specific domains, such as development of a clinical planning tool. In the process of clinical planning, there are complex workflows occurring that, while can be mimicked using a general use language, are best implemented using a platform that deals with workflows themselves. Examples of these types of workflow-based platforms are n8n.io, Verj.io, Digital Business Transformation Suite, TACTIC, and Node-Red.

Many workflow platforms are meant for facilitating business workflows, such as Digital Business Transformation Suite. However, many workflow platforms have enhanced their capabilities to include more software-driven capabilities like data source and IOT integrations as is the case with n8n.io, Verj.io, TACTIC, and Node-Red. All of these platforms feature flow-based programming, which is essential to be able to model the types of workflows that exist in a clinical environment when planning and monitoring patients. While all of these platforms include the type of programming required, Node-

Red is the strongest tool available for use in the clinical planning domain. This is due to its robust integrations with other platforms like MySQL, AWS, and Google, but it is also can makes use of JavaScript and Python readily as needed. All of the different techniques required for clinical planning that are part of the aforementioned JDL model [13] work in the infrastructure as a series of interconnected “flows” [17] as defined by Node-Red. These flows provide the following infrastructure: the user (in this case, a healthcare provider) can organize a series of nodes that are define a patient’s information, clinical presentation, and data sources to be referenced. The user may then define their own Domain Specific Language equipped with rules to be used in the data fusion process, or they may select a previously defined Domain Specific Language appropriate for their use-case. These nodes may be connected with a series of other flows that may perform alerts or machine learning. Data and sensor information will be procured based on any provided data sources, as well as specific symptoms recorded may result in different pathways of the Domain Specific Language being considered as part of the data gathered. Following the collection of data, this data should be referred to another flow that will perform the fusion techniques. Upon undergoing data fusion, this data may be used for monitoring, planning, or further processing by other flows. If desired, the user can connect this data to a flow capable of performing machine learning to provide further insight of the resulting plan and data. Alerts or updates to patient status (i.e. changes in symptoms based on data gathered in earlier stages) have the capacity to cause the process of data fusion to be repeated. After the flow responsible for creating the clinical plan is complete, the clinical plan should be output to the healthcare provider via an alert. At each stage, important information may be returned to the user as part of their ability to properly manage their patient's needs, such as potential drug interactions or concerning lab tests. The user has the ability to modify their workflow by making adjustments to the nodes and flows they have present within their workspace, or by adding additional nodes and flows to customize their experience.

3.2: Describing Clinical Context Based on a DSL

As has been made clear in earlier discussion, monitoring and planning for a chronic disease such as COPD is a long-term and complicated process. To best provide a tool capable of completing monitoring and planning tasks for COPD, a series of robust techniques are required. The first of these, the DSL, is used as an essential foundation for healthcare providers to create plans and properly monitor their patients as research

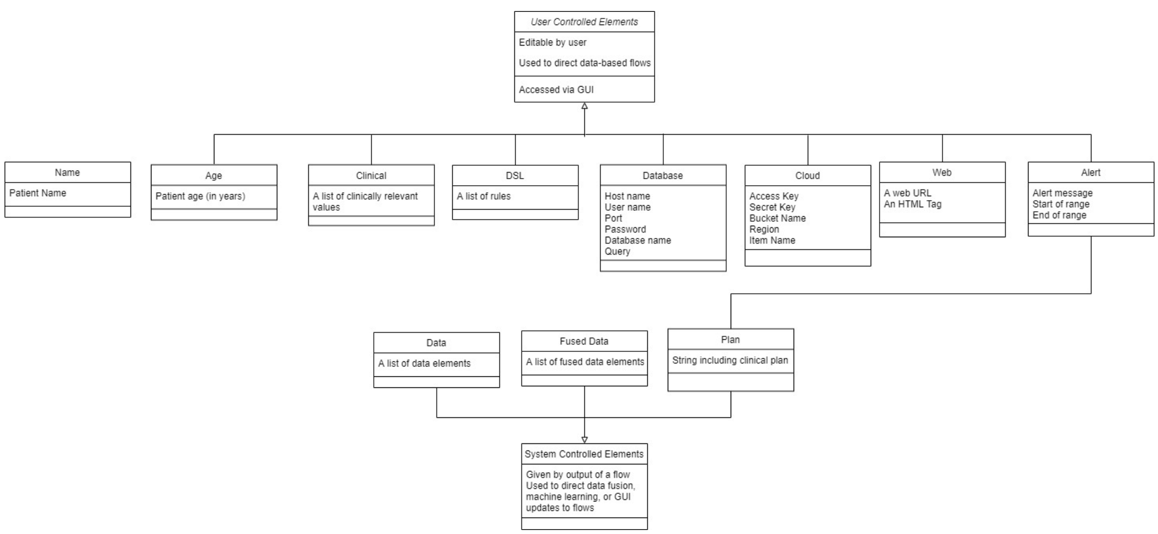

Figure 3: UML Diagram of the DSL grammar elements

suggests that most existing technology in healthcare based on a data entry paradigm is often frustrating or difficult to use [20]. In order to combat this frustration, the DSL provides users with different node-types within the environment that enables more proper clinical planning and monitoring techniques. To facilitate this, the DSL makes use of simple syntax to identify a patient's basic information (such as name and age), their current symptoms, past diagnoses of other illnesses, any lab results associated with the

Element Type	Description	Usage Example
Patient Name	Patient's name given in Last Name, First name format	Name: Ewen, Lisa
Patient Age	Patient's age given in years	Age: 23
Clinical Keywords	A list separated by commas of important clinical information relevant to a patient's clinical presentation (symptoms, medications, etc.)	Clinical: Hypotensive, Fluticasone, Wheezing, Smoker
Domain Specific Language	A set of rules specific to a given domain (COPD, Asthma, etc.) that aids in the data fusion process	(Fever, High WBC): Amoxicillin
Custom Databases	An optional list of custom database names separated by commas, credentials to these databases are provided by input boxes	Database: MHOP_patient_database, MHOP_labs
Custom Cloud Resources	An optional list of custom cloud resource names separated by commas, credentials to these cloud resources are provided by input boxes	Cloud: TBRHSC_Care_Pathways, TBRHSC_Emergency_Patient_History
Custom Web Resources	An optional list of web resources given in links that are separated by commas	Web: https://www.healthline.com/health/copd/drugs, https://www.nhs.uk/conditions/chronic-obstructive-pulmonary-disease-copd/symptoms/
Alarms	Not user controlled, alerts are triggered based on input from collected data or sensors for specific situations	Heart monitor indicates high blood pressure, and an alarm is displayed to the healthcare provider
Data for Fusion	Not user controlled, this is the set of datasets to be input for data fusion	Each data gathering flow will return a message containing the data obtained from the operation, these are then given to the DSL to combine all data into a list and pass to the data fusion flow
Fused Data	Not user controlled, the data after data fusion takes place, to be used as input for the machine learning model	The output from the data fusion flow given to the DSL to be passed to the decision making flow
Plan	A location for the plan output by the model which can also be used as input for future iterations	Plan: Administer prednisone for breathing difficulties, monitor blood pressure closely, enrol patient in smoking cessation program

Table 1: Description of DSL Grammar Elements

patient, and their current vitals. There also exists syntax capable of defining a set of rules that the clinical plan will be based on. This syntax is equipped with the capacity to trigger alerts during situations when the patient reaches a more critical status, such as low O_2 saturation, blood pressure outside of normal range, lab results that are concerning, fevers, or abnormal heart rates or rhythms. These alerts allow for healthcare providers to properly monitor the status of the patient. A description of the grammar elements part of the DSL and their associated relationships are provided in Table 1 and Figure 3.

The DSL makes use of basic patient information as part of the syntax with keywords to identify each aspect, such as "Name" or "Age". While a patient's name, insurance number, or home address provide little importance to their diagnostic and clinical planning, this provides a method for healthcare providers to be able to properly identify which patient the clinical plan and monitoring is associated with. Age and sex are also included as part of the basic information, but this does hold more relevance to a patient's clinical plan when it comes to addressing nutritional needs, selecting the appropriate medication and doses, or providing preventative measures [18, 19]. This information can have a direct impact on selecting appropriate clinical pathways during data collection, how severe abnormal vital or lab presentation can be regarded, or decisions made during the clinical planning stages. However, in some cases this information is not always relevant to a patient's care monitoring and planning, so these fields are considered optional. The DSL also features types that allow for healthcare providers to provide their own data sources as they see fit. These can be local hospital patient databases or web resources that they find useful for diagnostic or managing purposes for their patients.

Another source of important information from the healthcare provider is a set of keywords used to discuss the patient's current status (which can be updated automatically as necessary with data that is gathered during later stages) including their current symptoms, vitals, previous diagnoses, medications they are taking, etc. This allows the healthcare provider to shape the plan based on the patient's current status, however, changes may occur based on any acquired data in the process of forming the plan.

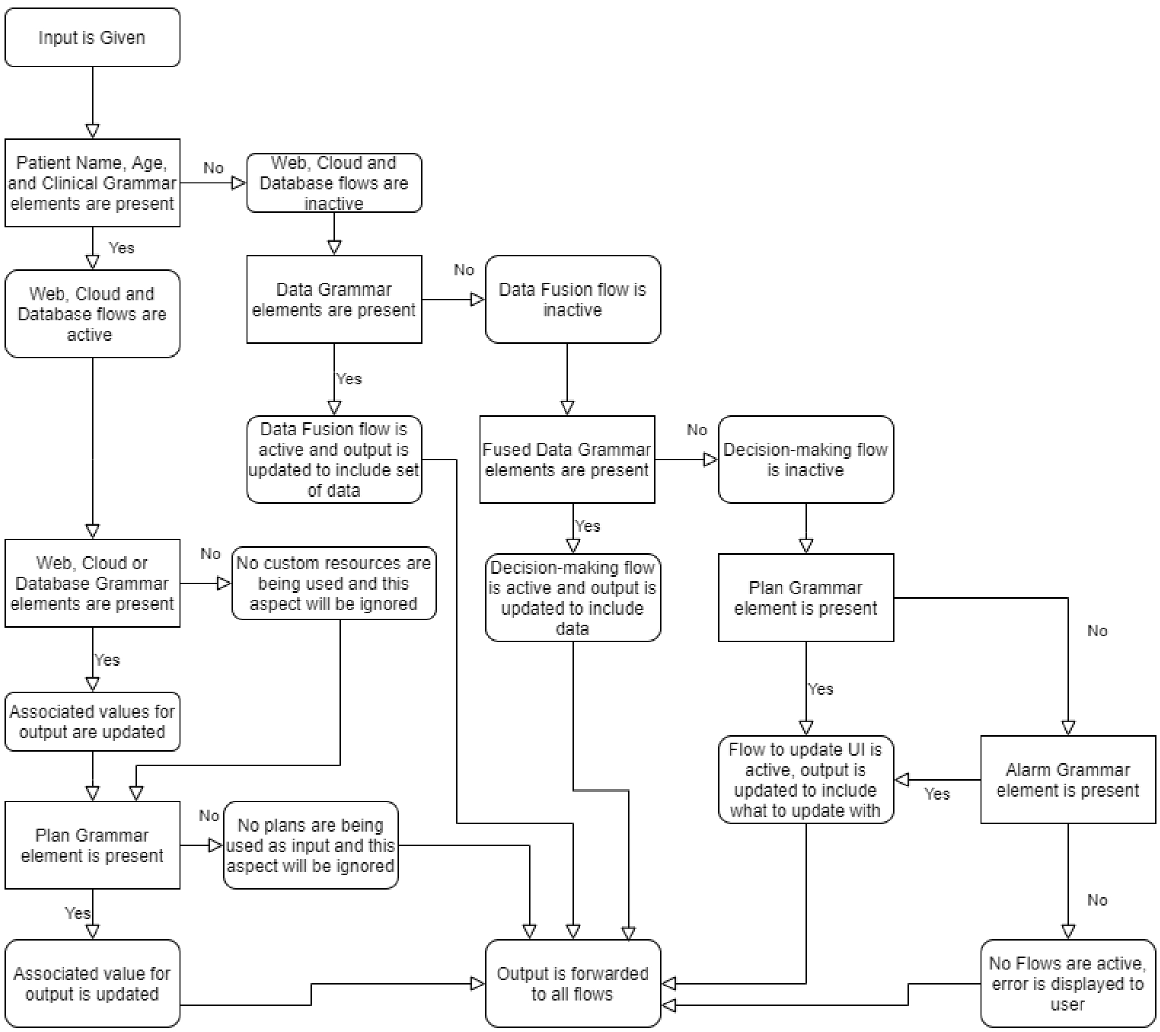

Figure 4: The DSL Context Workflow Parser Logic System

The DSL also features a set of syntax not accessible to the end users that is used to help direct the workflows resulting from the outputs of other flows. This includes data returned from the flows used to gather data from various resources, the data after undergoing data fusion, and the clinical plan that acts as the output. The data returned from any one of the data procuring flows will be one or more JSON objects given the type “Data” which allows the DSL to identify it. Similarly, the data returned after data fusion is complete will be a JSON object given the title “Fused Data” so that the DSL is able to distinguish it from the raw data that is received earlier on. Lastly, clinical plan output will be a string describing the clinical plan, and is marked with the title “Plan.” This plan is made visible to the user to make necessary edits to their workflows, and if necessary, results may be used in later processing

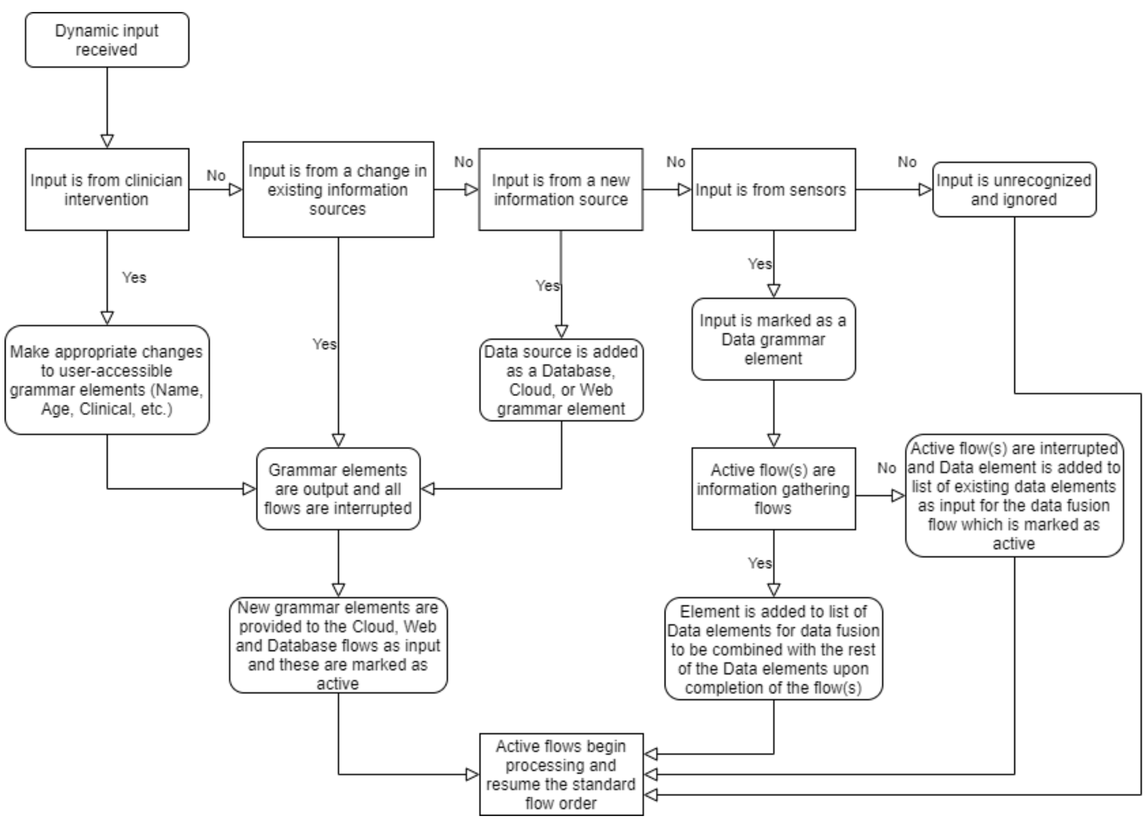

Figure 5: The Data Fusion Interpreter Logic System

All of the grammar elements and syntaxes are incorporated into the DSL Context Workflow Parser, shown in Figure 4, which directs the activation of different workflows based on the input given. This input includes direct input from the healthcare provider in the text environment provided and output from different flows, as all flows are directly connected back to the DSL upon completion. Based on the type assigned to each of the provided input elements, the DSL Context Workflow Parser will generate a list of outputs to be directed to each of the flows (information gathering, data fusion, etc.). This set of outputs includes the clinical keywords provided by the healthcare provider, the patient's name and age, any custom resources that may be associated with the flow, and whether or not the flow is active. A flow's status is dependent on the type of input received by the DSL exemplified in Figure 4. Each flow is also equipped with an internal switch which will ensure that the flow only functions when the DSL Context Workflow Parser returns that the flow is active.

3.3: DSL for COPD Monitoring and Progression Planning

The monitoring and planning of a patient is the most important aspect of this tool. As such, a specific set of syntax devoted to those ideas is vital. While gathering of patient information, specific data sources, and general information regarding their clinical status is important, there is much more that needs to be done utilizing the DSL to create clinical plans and monitor a patient. This requires the DSL to feature a specific set of syntaxes to address both of these aspects.

In order to achieve the planning aspects, the set of "Data" grammar elements exist to take any acquired data and properly relay that data to a set of machine learning functions capable of, in essence, making decisions. These decisions feature what medications or treatments to administer to the patient, investigative tests that must be performed, recommendations on treating a patient in-hospital or as an out-patient, etc.

These decisions are formalized in the DSL grammar element as rules that shape the data fusion process. Once all of the data has undergone data fusion, and the appropriate rules are triggered, this output is marked as a "Plan" grammar element to be able to relay this information properly to the healthcare provider so that they are able to view and edit the clinical plan as needed.

As part of the monitoring aspects of the tool, it is required to be able to update information present as the situation changes, as well as send updates to the healthcare provider as necessary. This is represented by changes to the clinical keywords made by the healthcare provider to add or remove symptoms, medications, etc. as the patient's status changes. This can also happen through changes via sensors such as heart or blood pressure monitors, or updates in lab results associated with the patient. These changes are handled by the Data Fusion Interpreter, shown in Figure 5, which ensures that the necessary flows can be activated by other means than initial user input or flow outputs (i.e. sensor input, changes in data such as returned lab results, clinician intervention, etc.). The Data Fusion Interpreter accepts dynamic inputs that have the capacity to interrupt workflows to provide important information. Upon receiving input, the Data Fusion Interpreter notifies all flows of the new input. If the flow is currently inactive, nothing will happen. Should the flow be active, it will receive the new information and add it to its existing inputs, or remove/override certain previously determined inputs as necessary, and restart its processing. There may be some occurrences where the Data Fusion Interpreter may require the new input be relayed to a previous flow (alongside the rest of the input), such as a healthcare provider updating the clinical keywords during data fusion which requires the process of data gathering to be redone using the new or removed keywords before data fusion can proceed.

3.4 Data and Sensor Collection

In order to provide the DSL with relevant patient data capable of providing the syntax with functionality, web scraping, database/cloud integration, and sensor data collection is utilized. Healthcare providers are able to provide connections to local databases or web/cloud resources that may be tailored to their patients and clinical setting, such as their local EHR database. This allows for a variety of specific patient information (as in the case with direct access to patient EHRs), as well as more general information available from web that may include vital information about a patient's presentation. These custom resource nodes are intended for use by healthcare providers who may have more direct knowledge about data sources that may be beneficial for their endeavours. These data sources can be pre-determined by users more knowledgeable about these data sources so that they can be provided to users who require this data for their clinical planning processes, but don't have acute knowledge of how to access the specific data sources themselves. In addition to these more static resources, dynamic sensor information may also be gathered to provide information about a patient's vitals.

3.5: Clinical Data Fusion

In order for the DSL to make proper use of all of this data, data fusion techniques must also be utilized. As has been discussed in earlier sections, the use of data fusion is what makes the data used by the tool more adequate than treating the data as separate entities to produce a meaningful clinical plan. The purpose of data fusion in this context is also to allow the DSL to be able to make use of the syntaxes for monitoring, and to produce updates and alerts to healthcare providers based on all available data. This is in contrast to having only one available data source, such as vitals. While vitals are important to monitoring a patient's condition, other information may describe other aspects of a patient's condition which may also be considered critical, such as lab or other test results.

The task of data fusion is completed using feature-based data fusion. As previously discussed, this involves a process of concatenation on each of the datasets received. Before this process takes place, each dataset is to be one-hot encoded. This is done for two reasons: 1) to allow for concatenation to take place much more easily, so we aren't concerned about size, and 2) to properly represent the data should it be used for machine learning at a later time. After each dataset is one-hot encoded, a new dataset is created featuring the column names of each of the previously one-hot encoded datasets. The dataset is then populated with each row from each of the available datasets, adding zeroes to pad all areas that are empty. The final step is to modify the data to display relationships between the data. For example, if a patient presents with symptoms of a respiratory infection (which are a common result of COPD), a white blood count test should be ordered to check for an elevated white blood cell count which would indicate infection. This would result in the row describing the symptoms associated with respiratory infection also having a "1" in the column for the white blood count test. We also are able to define negative relationships, such as a patient having a penicillin allergy, and being unable to take certain medications such as Amoxicillin. These negative relationships have the power to "negate" any existing relationships by re-setting the appropriate rows and columns to “0”. Once all relationships are properly defined, all rows that only have a single “1” will be considered orphaned data points, and may not contribute to the overall outcome of the model.

Upon the completion of data fusion and associated modification to display proper relationships among the data points, the complete dataset is then tagged with the “FusedData” grammar element, and the associated plan that is generated is tagged with the “Plan” grammar element. Both elements are then returned to the DSL Context Workflow Parser for further processing, or to be displayed to the user as necessary.

Chapter 4: A Flexible Prototype to Support COPD Clinical Planning and Monitoring

4.1: Introduction

The prototype features implementation for the nodes and flows described in Chapter 3. This implementation features JavaScript as a core language due to the Node-Red infrastructure being built on JavaScript and Node.js. However, there are some features implemented in Python due to the nature of the data fusion and machine learning aspects of the prototype. All nodes regardless of type utilize the DSL Context Workflow Parser, and the Data Fusion Interpreter to process the input to the node and ensure that nodes and flows are only activated under the correct conditions.

4.2: JavaScript Nodes

JavaScript nodes are defined by three main requirements: a JSON object describing its contents, a JavaScript function that instantiates the node and performs all of the necessary node computations, and an HTML file that provides the node definition, the edit template, and the help text. Figures 6-8 represents an example of each of the three requirements [30].

The JSON object can be generated using the command `npm init`, which asks the user questions to help define the starting point of the JSON object. After the object is initialized, a section titled “node-red” must be added to define the nodes files included.

The workhorse of the node definition, the JavaScript function, is wrapped in a Node.js module which exports the defined computation function to be called by the runtime when a node is utilized. The defined function must always first create a copy of the node before performing any computations. At the end of the module, the node must be registered to the runtime.

Lastly, the HTML file helps to define the node definition which was previously registered in the JavaScript function. Additionally, the HTML also defines which aspects of the node may be edited, such as the node's name, the input type, etc. The help text which describes the function of the node is also defined within the HTML file.

```
<script type="text/javascript">
    RED.nodes.registerType('lower-case',{
        category: 'function',
        color: '#a6bbcf',
        defaults: {
            name: {value:""}
        },
        inputs:1,
        outputs:1,
        icon: "file.png",
        label: function() {
            return this.name||"lower-case";
        }
    });
</script>

<script type="text/html" data-template-name="lower-case">
    <div class="form-row">
        <label for="node-input-name"><i class="fa fa-tag"></i> Name</label>
        <input type="text" id="node-input-name" placeholder="Name">
    </div>
</script>

<script type="text/html" data-help-name="lower-case">
    <p>A simple node that converts the message payloads into all lower-case characters</p>
</script>
```

Figure 6: An example of the HTML requirement

```json
{
    "name" : "node-red-contrib-example-lower-case",
    ...
    "node-red" : {
        "nodes": {
            "lower-case": "lower-case.js"
        }
    }
}
```

Figure 7: An example of the JSON package requirement

```javascript
module.exports = function(RED) {
    function LowerCaseNode(config) {
        RED.nodes.createNode(this,config);
        var node = this;
        node.on('input', function(msg) {
            msg.payload = msg.payload.toLowerCase();
            node.send(msg);
        });
    }
    RED.nodes.registerType("lower-case",LowerCaseNode);
}
```

Figure 8: An example of the Javascript requirement

```python
from pynodered import node_red

@node_red(category="pyfuncs")
def lower_case(node, msg):

    msg['payload'] = str(msg['payload']).lower()
    return msg
```

Figure 9: A Python implementation of the lower-case node

4.3: Python Nodes

There are a number of ways to define Python nodes in Node-Red, however, the simplest of these options utilizes a Python library called “Pynodered.” This library performs most of the actions required by the JavaScript nodes for automatically, thus, saving time. All node definitions may be defined in a single Python script, with each node being defined by a single Python function prefaced with the text `@node_red` above the function. These nodes may also be categorized under different titles. Much like the JavaScript function, the Python function that defines a single node provides all of the node’s calculations. Figure 9 represents a Python version of the same node exemplified in Figures 6-8 [31].

4.4: Hidden Nodes and Explicit Nodes

Hidden nodes are nodes that are not defined within the Node-Red palette, and as a result, the user may not directly access them. These nodes are only referred to within the computations of a node which is not hidden, also known as an explicit node. Hidden nodes provide important computational support to each of the explicit nodes, but are not meant to be used independently by the user.

Explicit nodes have at least one of two main presentations: generic and specific. A generic presentation of an explicit node is a version of that node type with no customizations. These nodes are not required for use, but give flexibility to users who want greater customization within their workflows. Specific presentations are nodes that have their settings partially or completely pre-determined for quicker use.

4.5: Parser and Interpreter Nodes

Two important types of hidden nodes are the Parser and Interpreter nodes that feature the DSL Context Workflow Parser and Data Fusion Interpreter. These nodes, as mentioned in section 4.1, are referenced at the first step in every explicit node to ensure that the computations of the node may be completed under the proper circumstances to avoid errors.

The Parser node utilizes the parse function featured to categorize the input and refer it to the Interpreter node. Then, the Interpreter node assesses the available input using the algorithm defined in Figure 11 to determine which nodes and flows may be activated to prevent improper input being applied.

```
If element is type clinical:
        Gather clinical elements
        Add each element to a Clinical node type
Else if element is type age:
        Add age to an Age node type
Else if element is type name:
        Add name to a Name node type
Else if element is type DSL:
        Split test into individual lines
        Collect each observation between "(" and ")"
        Collect each result following ":"
        Add observations and results to a DSL node typ
Else if element is type web:
        Add URL to a Web node type
Else if element is type database:
        Gather database properties
        Add properties to a Database node type
Else if element is type cloud:
        Gather cloud resource properties
        Add properties to a Cloud node type
Else if element is type data:
        Gather datapoints
        Add to a Data node type
Else if element is type fused data:
        Add to a Fused Data node type
Else, element is type plan:
        Add plan string to a Plan node type
```

Figure 10: The Parser algorithm

4.6: Grammar Element Nodes

Most grammar elements discussed in Chapter 3 are represented by explicit nodes within the Node-Red palette. That being said, a few of the grammar elements often serve as the output from another node, and as such, are defined as hidden nodes within the prototype. These grammar elements include the Data, FusedData, and Plan elements.

The Clinical, Age, and Name nodes are the only nodes that are have exclusively a generic presentation due to the fact that they are used to define a patient's unique information.

If present nodes are of type Clinical, Age, and Name:

Data may be gathered from specified sources

If present node is of type Data:

Data fusion may be performed

If present nodes are of type Data and Clinical:

Data fusion may be performed

If present nodes are of type Fused Data, Data, and Clinical:

Machine learning may be performed

If the present node is of type Plan:

No action necessary

Figure 11: The Algorithm for Node and Flow Activation

These nodes can be used in a number of flows to help guide them with their patient's specific needs. A change to one of these nodes may cause a workflow to restart its functions as it is seen as updated input

The Cloud, Web, and Database nodes are data-based nodes used to gather data from various information sources of their respective type. Both generic and specific data-based nodes will return a Data node as its output regardless of which data-based node type is defined.

Generic data-based nodes give the infrastructure to extract data from a user-defined data source of any type which requires some configuration on behalf of the user. The required specifications for a generic node type includes the credentials for the data source, and any necessary queries.

Unlike generic data-based nodes, specific data-based nodes feature pre-determined data sources that have already configured the possible credentials, queries, and data types required to produce output. These pre-determined nodes may be defined by administrators, or users with more acute knowledge of the infrastructure they are obtaining data from.

4.7: Data Fusion Nodes

Unlike the grammar element nodes, the Data Fusion node type is defined using Python to make use of the powerful libraries that exist within Python to process a large number of data points. The Data Fusion node type utilizes the numpy and sci-kit learn packages to perform its computations.

As discussed in section 3.5, the process of data fusion employed by the prototype involves vectorizing and binary one-hot encoding the input, and modifying the data to describe relationships. It is important to note that the input of a Data Fusion node is a series of one or more Data grammar elements, and a DSL grammar element. The DSL grammar element is used to define the relationships between the Data grammar elements, and aids in producing the clinical plan as output.

To one-hot encode the data points, an NxM matrix of zeroes is generated where N is the number of rules in the DSL and M is the number of observations. Following the one-hot encoding, an algorithm exemplified in Figure 15 is utilized to modify any relationships between the data points with rules from the DSL. This function simply cross-references each endpoint of the relationship defined within the DSL rules with the original set of data points (before one-hot encoding) and changes a “0” to a “1” in the row for each value, or a “1” to “0” in the case of negative relationships. An example of this process can be seen in Figures 13 and 14. Upon completion of the relationship modification process, the Data Fusion node returns both a FusedData node, featuring the data after undergoing the fusion process, a Data node that contains the original data vector for later processing, and a Plan grammar element to be displayed to the user.

It is also important to discuss the ability to use only a single data source as input to a Data Fusion node. While data fusion itself will not actually be taking place, it will prepare the data by binary one-hot encoding the data as a pre-processing step for use in later applications. This is to allow the user flexibility to work with a single data source to complete simple tasks as desired by the user. Despite this feature being included, it is recommended to use data that has undergone data fusion for the most appropriate and meaningful plan to be produced.

Data = [Hypoxiema, Anemia, Oxygen]

Binary One-hot Encoded Matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 12: A binary one-hot encoded matrix to represent three data points

Data = [Hypoxiema, Anemia, Oxygen]

Relationship = [Hypoxemia, Oxygen]

Binary One-hot Encoded Matrix:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Figure 13: The binary one-hot encoded matrix from Figure 12 with relationships applied

For all rules in the DSL:

If an observation is present in the set of data points:

Modify the location of that observation in the matrix to "1"

Add the results of the observation to the output plan

If a result is defined as a negative relationship:

Negate a "1" in the location of the observation in the matrix

Add the result with the leading string "Do not include "

Figure 14: The modifyRelationships function

4.9: Alert Nodes

The Alert node type is a simple node that features a generic presentation that allows you to define what the alert is monitoring (such as blood pressure or heart rate), and define a value that specifies the range in which the alert will be triggered if necessary. For some circumstances, alerts may act as a modal that provide feedback to the user upon the completion of another node's computation.

Specific types of this node, as seen in Figures 16 - 19, include an alert for hypertension (a blood pressure over 140 mm Hg systolic and 90 mm Hg diastolic), fever (a temperature over 37.5°C), or tachycardia (A heart rate over 100 beats per minute). When used in conjunction with other node types, an alert node can cause a workflow to restart as it is interpreted as new input.

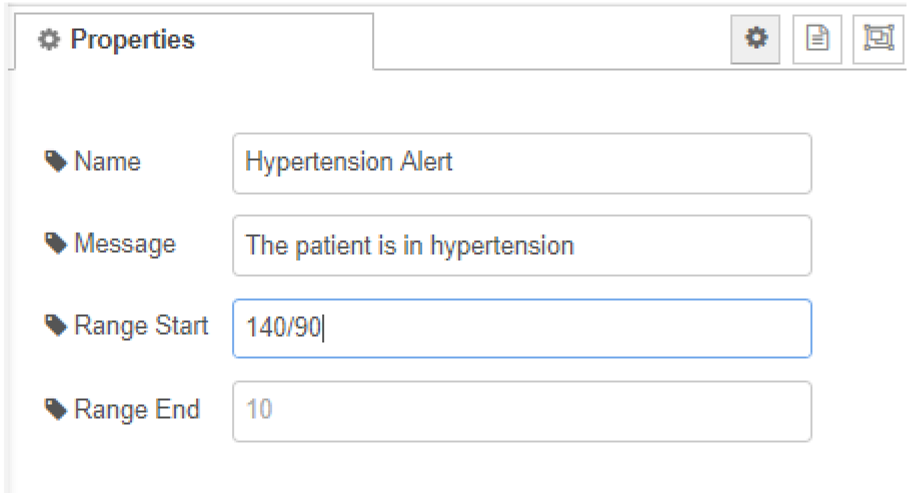

Figure 15: A hypertension alert node

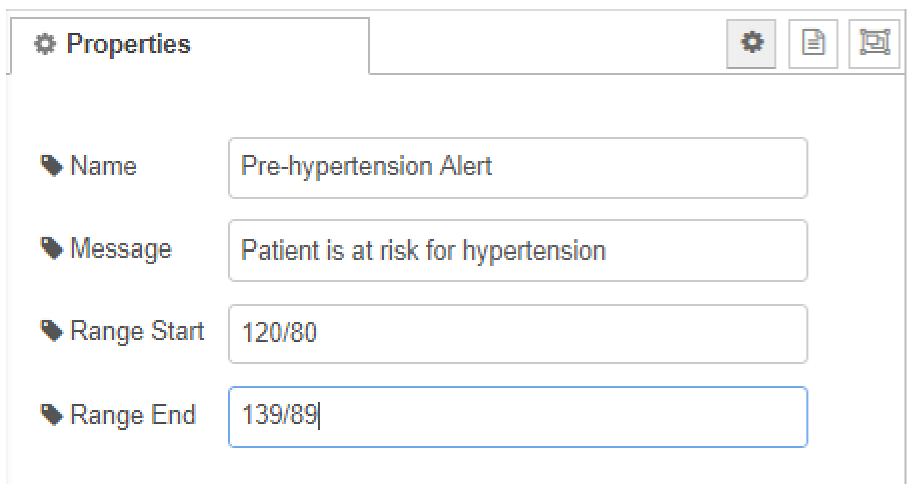

Figure 16: A pre-hypertension alert node

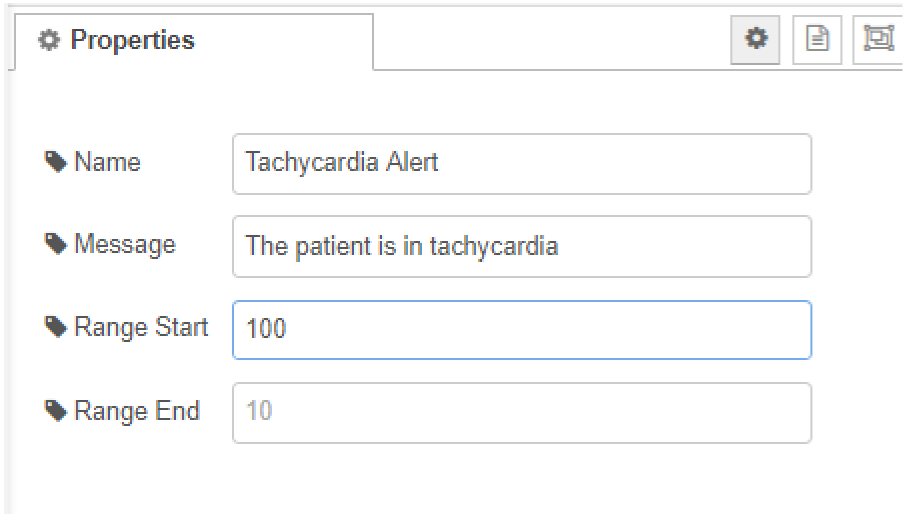

Figure 17: A tachycardia alert node

Figure 18: A fever alert node

4.10: Flows

With each of the available nodes, the end user has the ability to create their own workflows, much like they have the customization available to them to define different preferences within generic node types. Much of this can be time consuming or difficult depending on the desired actions. For this reason, certain flows have been defined for use in the same way as specific node types. Each of these flows feature a Name, Age, and Clinical node, which will be referred to as Patient nodes in this section.

The first defined flow is a flow for clinical planning of COPD patients. This will provide the user with a set of Patient nodes each of which connect to a set of four Database nodes, and DSL node. The DSL node features a representation of a COPD pathway [32] described in the appropriate syntax used by the prototype. Each of the data-based nodes are connected to a Data Fusion node, which is also connected to an alert node to allow for the clinical plan to be displayed. The COPD flow featured in Figure 20, will collect data from all four specific data-based node types, perform data fusion on the data based on the pre-set DSL shown in Figure 21, and provide a potential clinical plan for the given patient. This flow may easily be modified to include a customized Machine Learning node pointing to an existing decision tree as desired.

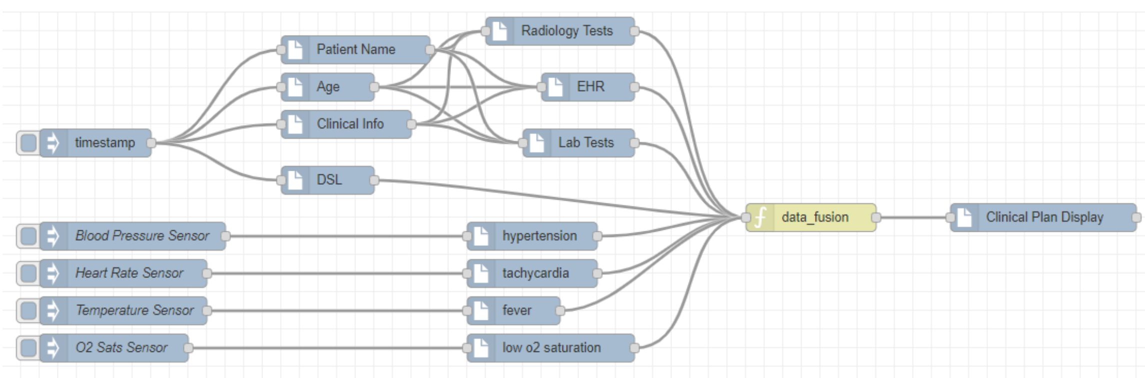

Figure 19: The COPD Workflow

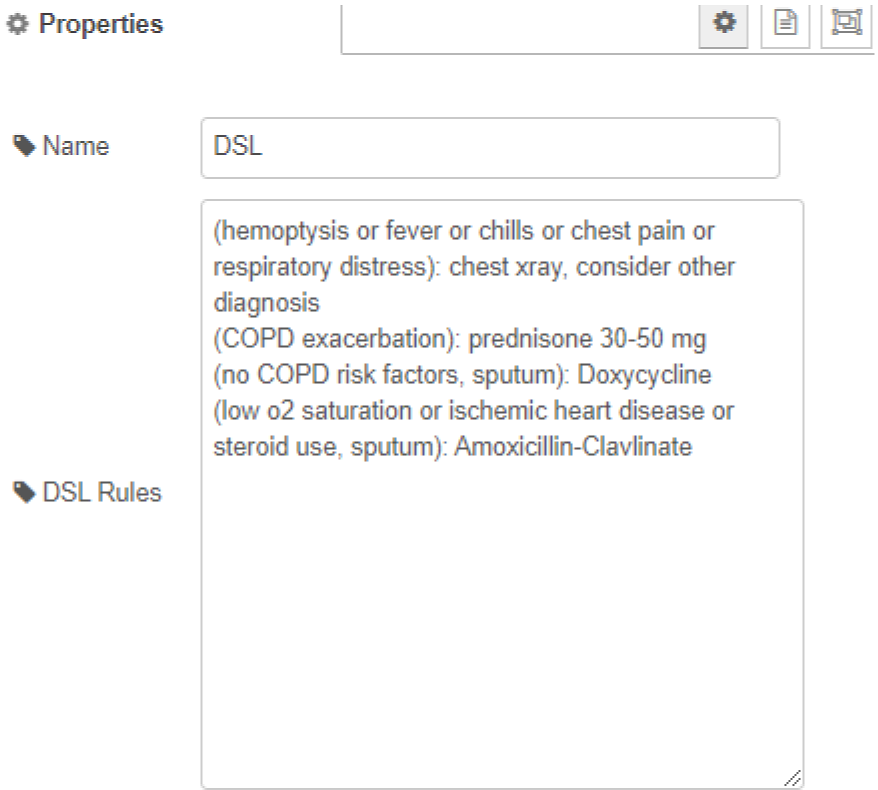

Figure 20: The COPD DSL

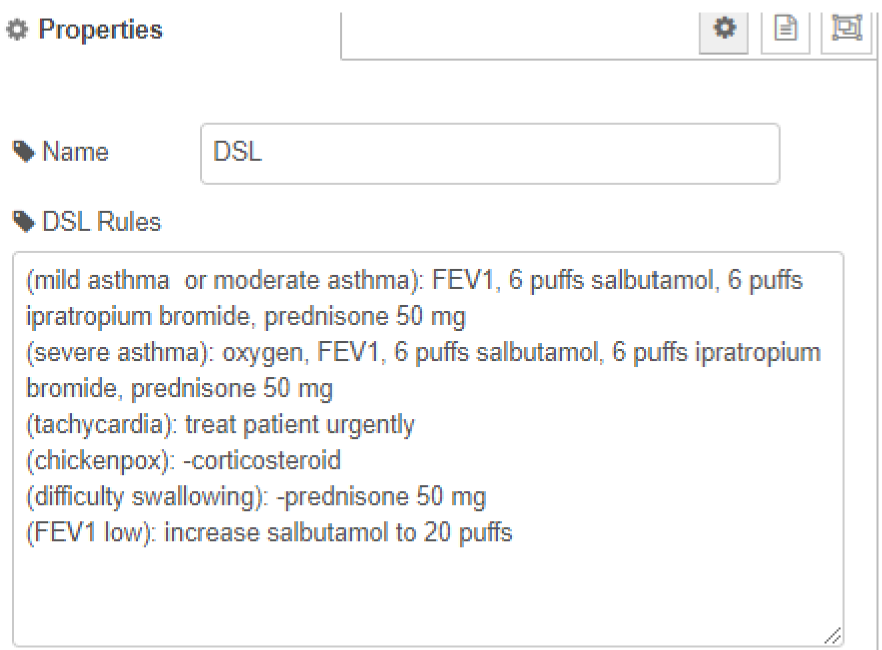

Figure 21: The Asthma DSL

The second major flow features all of the nodes seen in the first flow, with a different DSL present. This DSL follows an asthma care path [34], as there is often interaction between asthma and COPD in clinical settings. This DSL is shown in Figure 22.

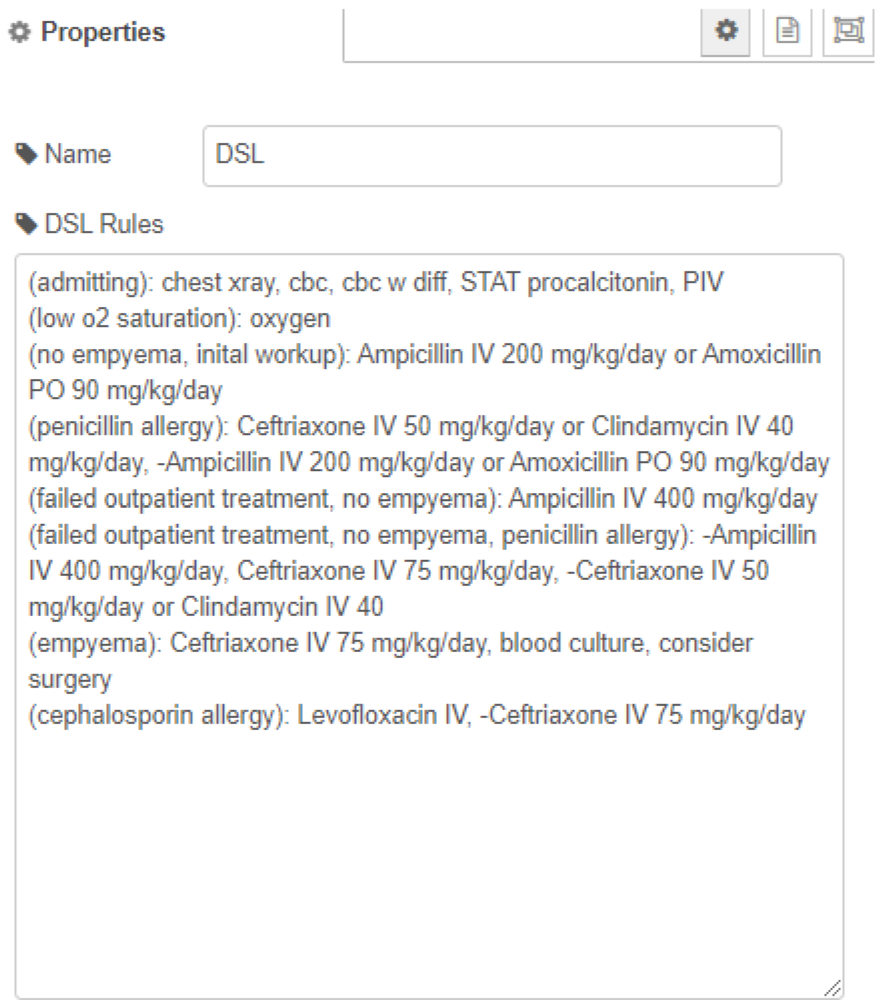

Figure 22: The Pnemonia DSL

The third major flow, much like the previously discussed flow, simply features a different DSL. The following DSL references a pneumonia care path [33], which is often another ailment seen in patients who suffer from COPD. This DSL is shown in Figure 23.

Lastly, there exists a set of flows that can be used for patient monitoring by connecting Patient nodes to one or more Alert nodes. These flows can be seen in Figures 24-27. On their own, these flows do not do much as the only time they may be triggered is upon changes to the Clinical node. In order to be truly effective, this flow should be combined with a data-based node that has the ability to provide data for the alert to be triggered by.

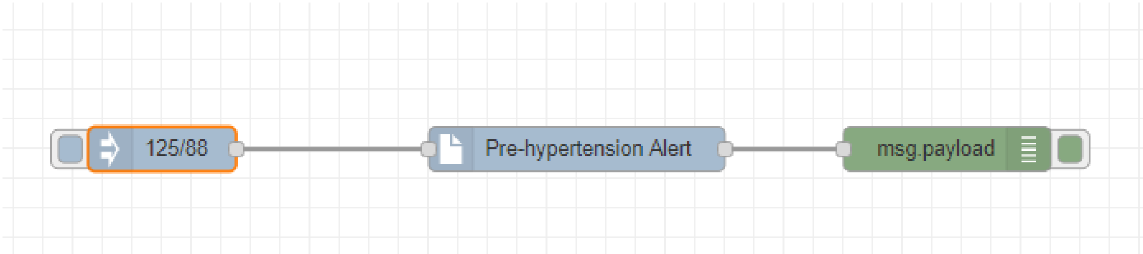

Figure 23: The pre-hypertension alert workflow

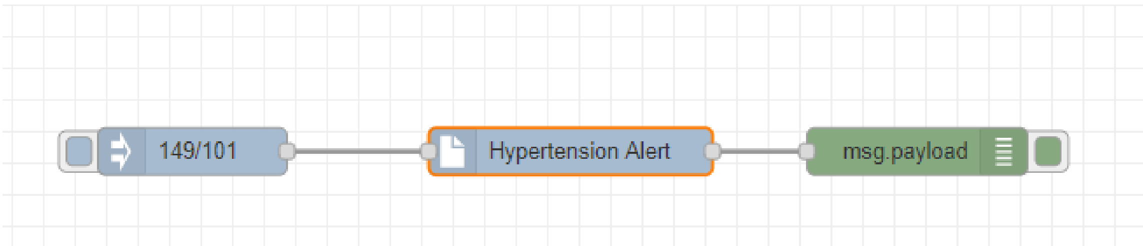

Figure 24: The hypertension alert workflow

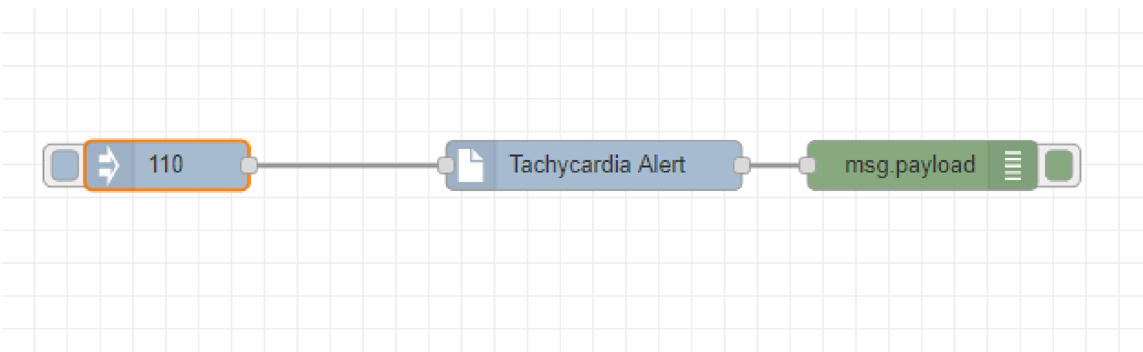

Figure 25: The tachycardia alert workflow

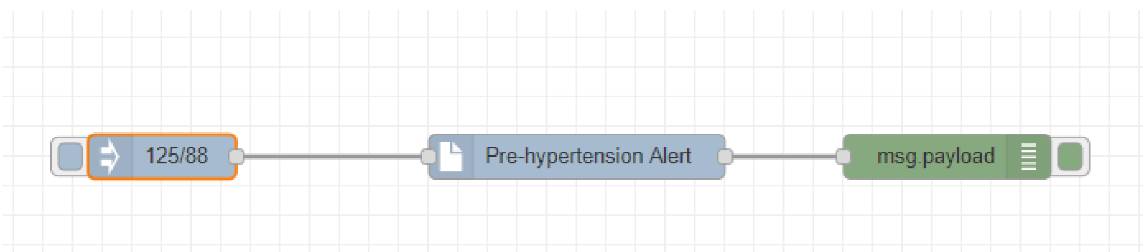

Figure 26: The pre-hypertension alert workflow

Chapter 5: A Discussion of Data Fusion-Based Design as it Applies to COPD and Clinical Planning

5.1: Discussion

The presented methodology and following prototype provide a proof of concept of the capabilities that data fusion can have in the context of clinical planning. That being said, it is important to acknowledge the current constraints and limitations of this concept.

It is important to note that at this time the tool has not undergone significant testing to verify the utility or feasibility of the software. While the prototype meets the criteria of the implemented care paths demonstrated in chapter 4, more extensive tests will need to be completed under a larger variety of scenarios and circumstances.

We also acknowledge that the software, as it presents now, would need to undergo more development to allow for integration of real patient data, as well as external sensors. At this time, we have not developed the prototype to handle noise from any connected sensors, and it will be essential to account for the potential of noise during future iterations of development.

5.2: Future Work

We have provided a few possible developmental additions in 5.1 that will need to be undertaken to provide a more useful prototype capable of being used in clinical practice. In addition to these requirements, there are also a few other areas that may be explored as an extension of this concept. The first of these options is to investigate using a backpropagation neural network as an opportunity to provide more accurate results, as the DSL is only as useful and as accurate as the user who created the rules. The DSL requires the user to provide expert knowledge that may be provided more easily and more accurately by a backpropagation neural network (or possibly other machine learning techniques).

Secondly, external validation of the software, both in concept and design, may also be performed. Often healthcare technology and software solutions are brought directly to the consumer market, and are not properly researched in the space of healthcare. As a result, this appears to be a contributing factor to the current dislike and improper usage of popular healthcare software available at the current time. Proper evaluation alongside healthcare providers better ensures the likelihood of adoption and compliance of this tool in future commercialization efforts.

Lastly, there is opportunity to investigate this tool as a useful aid in the battle against COVID-19 and future pandemics or novel illnesses. Due to the uncertainty and volatile environment surrounding an illness like COVID-19, there is a possibility that this tool may provide a more effective way of diagnosing and treating COVID-19 or other novel illnesses as a direct result of the mutability of the DSL on which the data fusion processes are based.

5.3: Conclusion

As we have discussed in the previous four chapters, data fusion is a powerful tool that has been shown to greatly benefit clinical planning in current research. We have also seen that the present state of data fusion research does have some areas that need to be more deeply investigated, such as the narrow scope and lack of automation. While there is still a significant amount of research that still needs to take place in the realm of data fusion as it pertains to healthcare and clinical planning, it is still clear that the tool presented in chapter 4 aims to address some of these concerns by adding the ability to for a user to provide any number of data sources, and apply these data sources (and the resulting fused data that results) in a variety of ways to suit their individual workflows. There is also the ability to monitor and provide clinical plans for patients automatically based on data received via sensors or other data sources to address the automaticity aspect that has been lacking in most research.

We believe that the tool provided, given more time, research, and development, could potentially be a very powerful tool for physicians to use to aid in their clinical planning duties. This tool has the potential to address significant concerns in the healthcare space, such as physician burnout and physician errors, if properly developed. This tool also has the capacity to be extended for use in more general clinical planning scenarios beyond COPD, as is currently designed within the tool's structure, and has even more capabilities in the future to aid physicians provide the best care possible.

Bibliography

[1] Girao, Pedro S., Octavian Postolache, and José MD Pereira. "Data fusion, decision-making, and risk analysis: Mathematical tools and techniques." In *Data Modeling for Metrology and Testing in Measurement Science*, pp. 1-50. Birkhäuser Boston, 2009.

[2] Hall, David L., and James Llinas. "An introduction to multisensor data fusion." *Proceedings of the IEEE* 85, no. 1 (1997): 6-23.

[3] Factor, Michael, David H. Gelernter, Craig E. Kolb, Perry L. Miller, and Dean F. Sittig. "Real-time data fusion in the intensive care unit." *Computer* 24, no. 11 (1991): 45-54.

[4] Nemec, Stefan Franz, Philipp Peloschek, Maria Theresa Schmook, Christian Robert Krestan, Wolfgang Hauff, Christian Matula, and Christian Czerny. "CT–MR image data fusion for computer-assisted navigated surgery of orbital tumors." *European journal of radiology* 73, no. 2 (2010): 224-229.

[5] Jannin, Pierre, Oliver J. Fleig, Eric Seigneuret, Christophe Grova, Xavier Morandi, and Jean-Marie Scarabin. "A data fusion environment for multimodal and multi-informational neuronavigation." *Computer Aided Surgery* 5, no. 1 (2000): 1-10.

[6] Moche, M., H. Busse, C. Dannenberg, T. Schulz, A. Schmitgen, C. Trantakis, D. Winkler, F. Schmidt, and T. Kahn. "Fusion of MRI, fMRI and intraoperative MRI data. Methods and clinical significance exemplified by neurosurgical interventions." *Der Radiologe* 41, no. 11 (2001): 993-1000.

[7] Reynier, Christophe, Jocelyne Troccaz, Philippe Fourneret, André Dusserre, Cécile Gay-Jeune, Jean-Luc Descotes, Michel Bolla, and Jean-Yves Giraud. "MRI/TRUS data fusion for prostate brachytherapy. Preliminary results." *Medical physics* 31, no. 6 (2004): 1568-1575.

[8] Polikar, Robi, Apostolos Topalis, Devi Parikh, Deborah Green, Jennifer Frymiare, John Kounios, and Christopher M. Clark. "An ensemble based data fusion approach for early diagnosis of Alzheimer's disease." *Information Fusion* 9, no. 1 (2008): 83-95.

[9] Yau, Hong-Tzong, Tsan-Jui Yang, and Yi-Chen Chen. "Tooth model reconstruction based upon data fusion for orthodontic treatment simulation." *Computers in biology and medicine* 48 (2014): 8-16.

[10] Harris, Shannon L., Jerrold H. May, and Luis G. Vargas. "Predictive analytics model for healthcare planning and scheduling." *European Journal of Operational Research* 253, no. 1 (2016): 121-131.

[11] Bedworth, Mark, and Jane O'Brien. "The Omnibus model: a new model of data fusion?." *IEEE Aerospace and Electronic Systems Magazine* 15, no. 4 (2000): 30-36.

[12] Boyd, John. *A discourse on winning and losing*. Air University Press, Curtis E. LeMay Center for Doctrine Development and Education, 2018.

[13] White, Franklin E. "A model for data fusion." In *Proc. 1st National Symposium on Sensor Fusion*, vol. 2, pp. 149-158. 1988.

[14] Harris, C. J., A. Bailey, and T. J. Dodd. "Multi-sensor data fusion in defence and aerospace." *The Aeronautical Journal* 102, no. 1015 (1998): 229-244.

[15] Dasarathy, Belur V. "Sensor fusion potential exploitation-innovative architectures and illustrative applications." *Proceedings of the IEEE* 85, no. 1 (1997): 24-38.

[16] Durrant-Whyte, Hugh F. "Sensor models and multisensor integration." In *Autonomous robot vehicles*, pp. 73-89. Springer, New York, NY, 1990.

[17] Heath, Nick. "How IBM's Node-RED Is Hacking Together the Internet of Things." TechRepublic. TechRepublic, March 14, 2014. https://www.techrepublic.com/article/node-red/.

[18] Tannenbaum, Cara, Colleen M. Norris, and M. Sean McMurtry. "Sex-specific considerations in guidelines generation and application." *Canadian Journal of Cardiology* 35, no. 5 (2019): 598-605.

[19] DeVoe, Jennifer E et al. "Patient age influences perceptions about health care communication." *Family medicine* vol. 41,2 (2009): 126-33.

[20] Walji, Muhammad F., Elsbeth Kalenderian, Duong Tran, Krishna K. Kookal, Vickie Nguyen, Oluwabunmi Tokede, Joel M. White et al. "Detection and characterization of usability problems in structured data entry interfaces in dentistry." *International journal of medical informatics* 82, no. 2 (2013): 128-138.

[21] Vahdat, Shaghayegh, Leila Hamzehgardeshi, Somayeh Hessam, and Zeinab Hamzehgardeshi. "Patient involvement in health care decision making: a review." *Iranian Red Crescent Medical Journal* 16, no. 1 (2014).

[22] Health, Ministry of. "Advance Care Planning." Province of British Columbia. Province of British Columbia, July 31, 2020. https://www2.gov.bc.ca/gov/content/family-social-supports/seniors/health-safety/advance-care-planning.

[23] Lougheed, M. Diane, Ann Taite, Julia ten Hove, Alison Morra, Anne Van Dam, Francine M. Ducharme, Madonna Ferrone et al. "Pan-Canadian asthma and COPD standards for electronic health records: A Canadian Thoracic Society Expert Working Group Report." *Canadian Journal of Respiratory, Critical Care, and Sleep Medicine* 2, no. 4 (2018): 244-250.

[24] "Leaving COPD Untreated: Risks: Lung Center of Nevada." Comprehensive Cancer Centers, September 10, 2019. https://cccnevada.com/leaving-copd-untreated/.

[25] Chung, Li Ping et al. "Five-year outcome in COPD patients after their first episode of acute exacerbation treated with non-invasive ventilation." *Respirology (Carlton, Vic.)* vol. 15,7 (2010): 1084-91. doi:10.1111/j.1440-1843.2010.01795.x

[26] López-Campos, José Luis et al. "Global burden of COPD." *Respirology (Carlton, Vic.)* vol. 21,1 (2016): 14-23. doi:10.1111/resp.12660

[27] Y. Zheng, "Methodologies for Cross-Domain Data Fusion: An Overview," in *IEEE Transactions on Big Data*, vol. 1, no. 1, pp. 16-34, 1 March 2015, doi: 10.1109/TBDATA.2015.2465959.

[28] Rawat, Sarvesh, and Surabhi Rawat. "Multi-sensor data fusion by a hybrid methodology–A comparative study." *Computers in Industry* 75 (2016): 27-34.

[29] Würthinger, Thomas, Andreas Wöß, Lukas Stadler, Gilles Duboscq, Doug Simon, and Christian Wimmer. "Self-optimizing AST interpreters." In *Proceedings of the 8th symposium on Dynamic languages*, pp. 73-82. 2012.

[30] Creating your first node. (n.d.). Retrieved February 09, 2021, from https://nodered.org/docs/creating-nodes/first-node

[31] Pynodered. (n.d.). Retrieved February 09, 2021, from https://pypi.org/project/pynodered/

[32] Alberta Health Services. (n.d.). Heart failure & COPD Clinical Pathways. Retrieved February 09, 2021, from https://www.albertahealthservices.ca/scns/Page13643.aspx

[33] ?

[34] Adult Emergency Department Asthma Care Pathway (EDACP) . lung health foundation. Accessed February 15, 2021. http://hcp.lunghealth.ca/wp-content/uploads/2020/02/Adult_ED_Asthma_Care_Pathway.pdf.

[35] Schmidt, Harald. "Chronic disease and prevention and health promotion." *Public health ethics: Cases spanning the globe* (2016): 137-176.

[36] West, Colin P., Liselotte N. Dyrbye, and Tait D. Shanafelt. "Physician burnout: contributors, consequences and solutions." *Journal of internal medicine* 283, no. 6 (2018): 516-529.

Appendix A

Key Terms:

Asthma: A chronic disease of the respiratory system that causes narrowing of the airways resulting in shortness of breath

Chronic Pulmonary Obstructive Disease (COPD): A chronic inflammatory lung disease that causes obstructed airflow from the lungs

Cross-Domain Data Fusion: The process of performing data fusion of multiple datasets across more than one domain

Clinical Planning: Identifying the priorities and strategic directions for clinical services or plans to guide delivery of services

Data Fusion: The process of integrating multiple data sources to produce more consistent, accurate, and useful information than can be provided by an individual data source

Domain Specific Language (DSL): A computer language specialized to a particular application domain

Electronic Health Record (EHR): Digital records of a patient's health information and history

Electronic Medical Record (EMR): A digital version of a patient's paper chart

Explicit Nodes: Nodes that are available for users to access

Generic Nodes: Nodes that have are not pre-configured by another user

Implicit Nodes: Nodes that are unavailable for users to access

Joint Directors of Laboratories (JDL) Model: A model of data fusion that divides the process into six different levels

Node-Red: A workflow-based programming platform

Pneumonia: An infection that inflames your lungs' air sacs

Specific Nodes: Nodes that have been pre-configured by another user

Traditional Data Fusion: Data fusion that occurs in a single domain, and is performed using schema mapping and data merging

Workflows: The sequence through which a piece of work passes from initiation to completion

Workflow-Based Programming: Programming using interconnecting workflows instead of traditional functions and methods

Appendix B

Dear Ms. Ewen

Your submission was successfully uploaded and has got the following data:
Medical Workflow Design and Planning Using Node-RED Data Fusion (L Ewen, S Mohammed, A Kim)

If there are any questions left, please contact imia@imia-services.org.

Yours sincerely

Conference Office

MedInfo 2021
One World, One Health: Global Partnership for Digital Innovation
Virtual conference 02 - 04 October 2021
https://access.online-registry.net/medinfo2021/

online Registry

Figure 15 - Submission to the MedInfo Conference

Appendix C

Medical Workflow Design and Planning Using Node-RED Data Fusion

Lisa Ewen[a], Sabah Mohammed[a], Arnold Kim[a]

[a] *Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada*

Abstract

The space of clinical planning requires a complex arrangement of information, often not capable of being captured in a singular dataset. As a result, data fusion techniques can be used to combine multiple data sources as a method of enriching data to mimic and compliment the nature of clinical planning. These techniques are capable of aiding healthcare providers to produce higher quality clinical plans and better progression monitoring techniques. Clinical planning and monitoring are important facets of healthcare which are essential to improving the prognosis and quality of life of patients with chronic and debilitating conditions such as COPD. To exemplify this concept, we utilize a Node-Red-based clinical planning and monitoring too that combines data fusion techniques using the JDL Model for data fusion and a domain specific language which features a self-organizing abstract syntax tree.

Keywords:

Medical Informatics, Biomedical Technology, Information Technology.

Introduction

All physicians and healthcare providers are required to participate in clinical planning either collaboratively or on their own. Clinical planning involves using diagnostic and other medically relevant information about a patient to discern a set of tests or medications to order, or instructions the patient must follow to aid in a diagnosis or improve their overall health. As a result, the process of clinical planning is a very mentally strenuous task that involves a large number of decisions to be made based on the information at hand. In addition to the mere step of planning a course of action for a patient, there are many administrative and procedural tasks to enact this plan required by a hospital or clinic to ensure the safety of patients and physicians in this process. These extra steps can add more strain on the healthcare provider, which can eventually result in poorer decision making as a result of physician burnout [1]. While many technological systems have attempted to counteract this, most have greatly contributed to this phenomenon of physician burnout [1]. While research has been done to try to explain why these systems are ill-fitted for clinical is the section where the authors introduce their work.

It is important to discuss that a contributing factor of frustration surrounding existing healthcare software often comes from a discrepancy between how physicians process information and perform their clinical workflows, and how the software systems function. Data fusion, a series of techniques utilised to combine a multitude of datasets for processing, however, more closely mimics how human's make decision due to the integration of multiple sources of information [13]. As a result, data fusion techniques has the capacity to provide a more well-rounded set of data which, in combination with the appropriate infrastructure, has the potential to address the existing software frustrations in healthcare.

Methods

Introduction

To provide an effective data fusion-based medical planning platform, it is important to incorporate a few enabling techniques to allow the platform to provide clinical inferences including prognosis. We are proposing a new methodology that captures the semantic context of clinical cases through the use of a domain specific language (DSL) that describes these cases and is used to guide, monitor, and infer the progression of the clinical cases through the linkage to dynamically evolving patient data that are updated from different sources including repositories over the cloud or sensors that are hooked to the patient(s). Additionally, our method, and later, the platform, need to be designed to be useful and meaningful to physicians and clinicians following the progress of these clinical cases. To show the effectiveness of our methodology, we decided to focus on Chronic Obstructive Pulmonary Disease (COPD) as it is a progressive type of chronic disease which can get worse over time. However, COPD is treatable with proper management and planning, as most patients with COPD can achieve good symptom control and quality of life, as well as reduced risk of other associated conditions (e.g. heart disease, lung cancer).

A Survey of Data Fusion Methodologies

To develop our methodology, we surveyed data fusion models, as well as other methodologies related to data fusion and DSLs. Firstly, we needed to select an appropriate data fusion model with which to base our methodology. The main data fusion models we can utilize to create our methodology are the Dasarathy Classification, the Waterfall Model, the Omnibus Model, the Boyd Control Loop, and the JDL Model. To describe each model in short, the Waterfall Model [9], the Boyd Control Loop [6], and the Omnibus Model [5] are each concerned with the flow of data during the data fusion process. The Waterfall Model follows data in a linear fashion through 3 levels, starting with raw data, moving through to feature extraction and feature fusion, and finishing with incorporating the data with human interaction to produce possible results. The Boyd Control Loop processes the data more circularly through four separate phases that may start again based on the outcome of the fourth phase. A derivation of the Boyd Control Loop, the Omnibus Model, works similarly but involves some modifications to each of the processes followed by the four stages.

In contrast to these three models, the Dasarathy Classification [8] and the JDL Model [7] work by performing refinement and fusion tasks based on different levels of data. An important aspect of the JDL Model is that each stage has the ability to refer back to previous stages for further refinement before producing output, something not present in the other fusion models. This is significant due to the nature of healthcare and clinical planning, where data is often dynamic and new data may affect the interpretation of previous data, resulting in a need for each stage of refinement and fusion capable of being interrupted or revisited at any time.

When discussing existing methodologies, we can refer to Yu Zheng's review methodology of cross-domain data fusion which discusses the different types of data fusion, as well as their strengths and weaknesses. Zheng discusses stage-based data fusion, which involves the loose coupling of datasets, feature-level-based data fusion, which involves the concatenation of each dataset, and semantic meaning-based data fusion, which involves focusing on the meaning of each feature while relating each dataset to each other [2]. The general conclusion that Zheng makes among the discussion of these types of data fusion is that semantic meaning-based data fusion often provides the most powerful form of data fusion, it often suffers from performance issues or discrepancies among dynamic and static datasets. As a result, despite the semantic meaning-based data fusion providing the most powerful relationships between datasets, we will use feature-level-based data fusion to avoid the negative aspects of semantic meaning-based data fusion.

The next resource utilized in the foundation of our methodology is Sarvesh Rawat and Surabhi Rawat's hybrid methodology for multi-sensor data fusion. It is described in this methodology that rough sets act to discover ambiguity and remove redundancy from datasets while performing data fusion [3]. These rough sets act as feature reduction and a pre-processing layer which allows for higher accuracy when used in conjunction with backpropagation neural networks. As a result of these findings, it is clear that a pre-processing layer act to strengthen the data fusion process, and provide better results. In place of rough sets, however, we are opted to utilize a DSL.

Due to the dynamic nature of the JDL model, our DSL must also be dynamic and capable of providing updates to the end user as their workflows progress. In order to provide a DSL capable of performing these tasks, we are using a self-organizing abstract syntax tree [4]. By utilizing a self-organizing abstract syntax tree we are able to best optimize the data fusion workflows and interactions with the users via the DSL.

Lastly, to be able to adequately support a methodology that impacts workflow-based procedures and tasks, the implementation of the methodology is reliant on a workflow-based platform. Examples of these types of platforms include n8n.io, Verj.io, TACTIC, and Node-Red. While all of these platforms feature flow-based programming, which is essential for a workflow-based software, Node-Red provides the most robust integrations with other platforms, such as MySQL, Aws, and Google. Node-Red can also make use of JavaScript and Python, which is not nearly as seamlessly integrated among other flow-based programming platforms.

The DSL Implementation

It has been discussed that a pre-processing step is necessary for improving the outcomes of data fusion techniques. Our pre-processing step, the DSL, incorporates syntax that allows users to identify important patient information directly (such as symptoms described by the patient during an appointment or intake in an emergency room), incorporate data sources such as integration with lab reports or their EMR that may be located in a database, the cloud, or on a website, as well configure as custom alerts based on a patient's status for monitoring. Most notably, users are able to define a set of rules to guide any data fusion tasks by using a series of observations and results. Observations act as potential data points that may be observed before undergoing fusion, such as tachycardia. Healthcare providers may define a list of observations that must be present in order for that rule to be triggered. Results are the following actions that are noted in the clinical plan, such as ordering a medication or a lab test. Similar to observations, any number of results may be defined for each rule.

The DSL makes use of two systems: the DSL Context Workflow Parser, exemplified in Figure 1, and the Data Fusion Interpreter shown in Figure 2. The first system, the DSL Context Workflow Parser, ensures that the appropriate actions are being undertaken based on the syntax that is present and the current moment. For example, if a dataset, or multiple datasets are received, the system is able to proceed with data fusion tasks, however, if these elements are not present, data fusion will not occur.

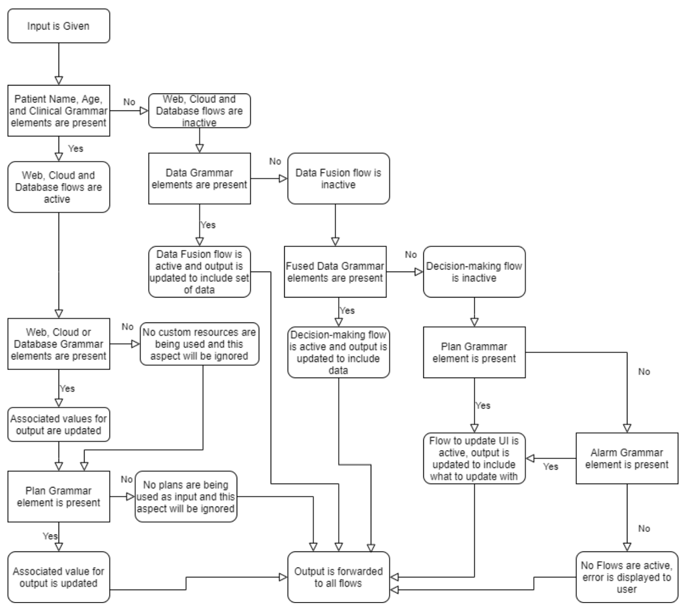

Figure 16- The DSL Context Workflow Parser Logic System

The second of these systems, the Data Fusion Interpreter, ensures that the different tasks and procedures performed by the system are capable of being enacted or revisited based on different system inputs, whether that be input directly from the user or coming from within the system itself. This is a direct implementation of the interaction data processing and fusion tasks are required to undergo while utilizing the JDL model.

The Data Fusion Implementation

The most important aspect of this tool, data fusion, is what allows the tool to provide feedback and clinical plans to the users. The data fusion process makes use of all provided sources of data, as well as the user-defined DSL rules to be able to create the resulting clinical plan.

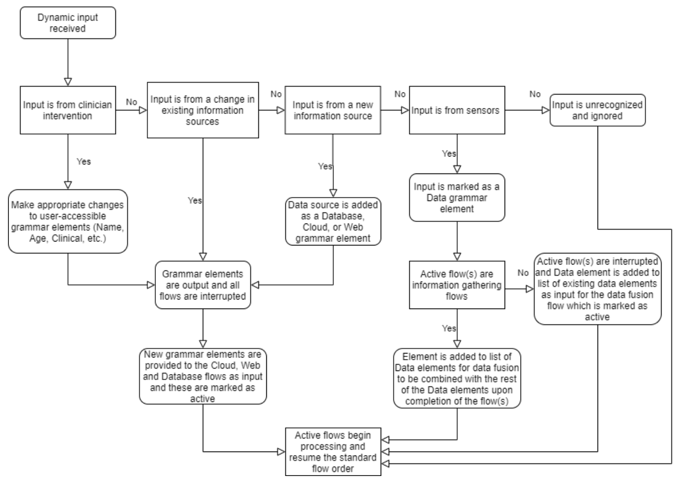

Figure 17- The Data Fusion Interpreter Logic System

Due to the fact that we are using feature-level-based data fusion, we are taking each dataset and concatenating them into one cohesive dataset. Despite having one singular dataset at this stage, we still must utilize our DSL rules to properly refine the data. This is done by creating a NxM binary one-hot encoded matrix where N is the number of rules, and M is the number of results given by the DSL. For each rule, if all the necessary observations are present in the dataset obtained by concatenation, then the associated results are represented by a 1 in the appropriate row and column. For example, if a rule requires the observation of low blood oxygenation, and the result is an order of pure oxygen, the column representative of pure oxygen will have a row for that rule. This will proceed for each rule. This process is shown in Figure 3.

There is also the presence of negative results that have the ability to negate a relationship between a given rule and its results. An example of such a negative relationship exists between a penicillin allergy and an order for Amoxicillin. This rule would inlcude the negative result for Amoxicillin, and when encountered in the processing stage, any instance of a 1 under the amoxicillin column will become a 0. After all rules have been appropriately applied to the data, the remaining results will be displayed to the user as the clinical plan.

Data = [Hypoxiema, Anemia, Oxygen]

Relationship = [Hypoxemia, Oxygen]

Binary One-hot Encoded Matrix:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Figure 18- Result Definition via DSL Rules

Results

To demonstrate the efficacy of the proposed data fusion-based clinical planning and monitoring tool, we have implemented three care pathways as DSL rules to provide examples of different use-cases. As we have focused our domain on COPD, the first implementation followed the Alberta Health Services COPD Pathway [10]. The pathway is described using the DSL syntax in Figure 4. In this scenario we are describing a patient who is presenting with tachycardia, hypertension, anemia, fluid in the lungs, increased coughing, and sputum. As a result of these symptoms gathered via a variety of data sources, the following recommended clinical plan output by the system can be seen in Figure 5.

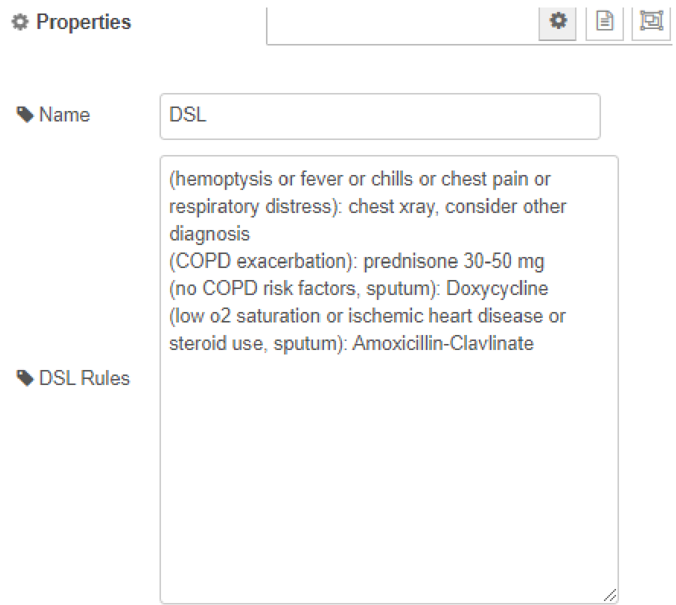

Figure 20- The COPD Pathway Rule Syntax

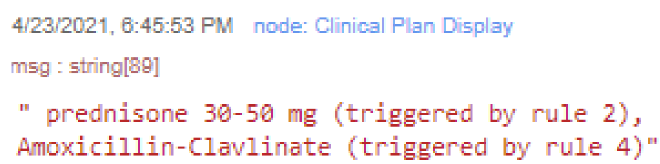

Figure 19 - The COPD Patient's Clinical Plan

Similarly, we have utilized the Lung Health Foundation's Adult Emergency Department Asthma Care Pathway [11], due to the fact that COPD and Asthma are often interrelated illnesses. A representation of this pathway utilizing the system's DSL syntax can be found in Figure 6. The resulting clinical plan of a patient who presents with moderate asthma, low blood oxygenation, and tachycardia is shown in Figure 7.

The last example we will discuss is the Connecticuit Children's Community Acquired Pneumonia pathway [12] seen in Figure 8. Figure 9 displays the clinical plan for a patient's initial admission workup who is presenting with empyema and penicillin allergy.

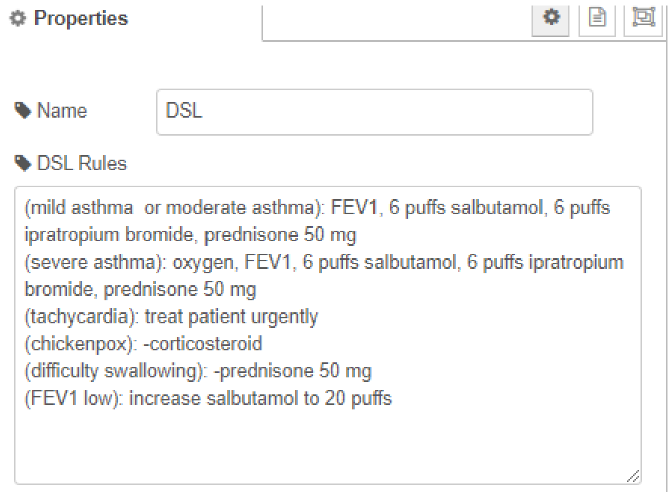

Figure 22- The Asthma Pathway Rule Syntax

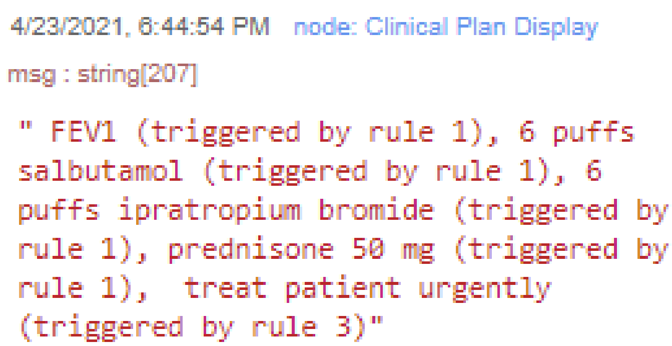

Figure 21- The Asthma Patient's Clinical Plan

Discussion

While the discussed tool is primarily being applied to the care of patients diagnosed with COPD, it is possible to expand this methodology to adequately provide clinical planning capabilities for other diagnoses. This can be done by extrapolating related care pathways available for these diagnoses, or consulting with multiple physicians and hospital administrators capable of the important rules the DSL will enact within the tool.

It is also important to acknowledge future work that will be required to support the integration of hospital or clinic-based data sources to be utilized as part of the fusion process. Interoperability between sensors, EMR/EHR records, lab reports, and other important data sources is a challenging task that has not been discussed as part of this methodology, however, is essential to the process of development of the tool for use

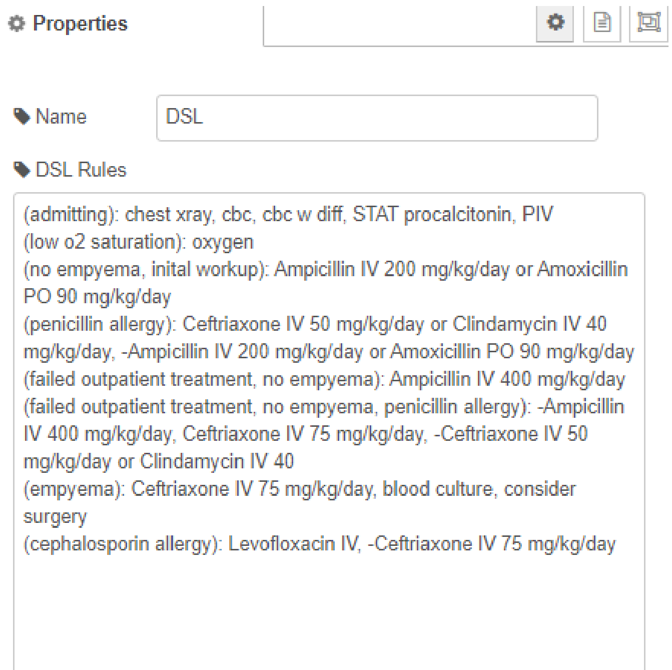

Figure 24- The Pneumonia Pathway Rule Syntax

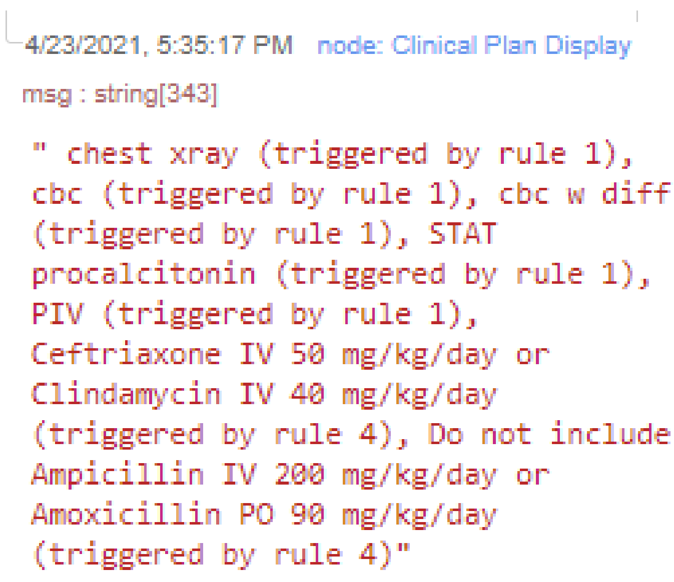

Figure 23- The Pneumonia Patient's Clinical Plan

among healthcare providers.

We also acknowledge that, while feature-level-based fusion provides an adequate method of fusion for the presented purposes, there is the potential for exploration into other fusion types. Most notably, there may be important discussion into the effects semantic meaning-based fusion may have on this methodology, and what improvements may be made by doing so.

Conclusions

During the clinical planning and monitoring processes, healthcare providers are required to make use of a variety of vital information sources to adequately make informed decisions about their patients. A failure to have access to all of these sources in a reasonable manner, either directly as the healthcare provider or via software, can result in poorer clinical plans. To overcome these difficulties, we have proposed a data fusion-based tool that has the capacity to incorporate all relevant data sources in order to allow for fully-informed decisions to be made in the process of providing a clinical plan or monitoring a patient. This proposed tool makes use of feature-level-based data fusion and a DSL with an self-organizing abstract syntax tree for preprocessing.

Implemented in Node-Red with a combination of Python and Javascript prgramming, the data fusion tool has the capacity to combine a variety of data sources that are essential to clinical planning and monitoring (patient vital sensors, EMR/EHR, lab reports, etc.). These data sources are pre-processed by the user-defined DSL rules, and, based on these rules, have the ability to provide a suggested clinical plan based on all of the provided information. The effectiveness of this tool is demonstrated by implementing COPD, Asthma, and Pneumonia care pathways as DSL rules which are adequately able to provide clinical plans as a result of the data fusion process.

References

[1] West, Colin P., Dyrbye , Liselotte N., Shanafelt, Tait D., Physician burnout: contributors, consequences and solutions., Journal of internal medicine 283, 13 (2018) 516-529.

[2] Y. Zheng, Methodologies for Cross-Domain Data Fusion: An Overview, IEEE Transactions on Big Data, vol. 1, no. 1, 18 (2015) 16-34, doi: 10.1109/TBDATA.2015.2465959.

[3] Rawat, Sarvesh, Rawat, Surabhi, Multi-sensor data fusion by a hybrid methodology–A comparative study, Computers in Industry 75, 7 (2016) 27-34.

[4] Würthinger, Thomas, Wöß, Andreas, Stadler, Lukas, Duboscq, Gilles, Simon, Doug, Wimmer, Christian, Self-optimizing AST interpreters., Proceedings of the 8th symposium on Dynamic languages, 9 (2012) 73-82.

[5] Bedworth, Mark, O'Brien, Jane, The Omnibus model: a new model of data fusion?, IEEE Aerospace and Electronic Systems Magazine 15, no. 4, 6 (2000) 30-36.

[6] Boyd, John, A discourse on winning and losing, Air University Press, Curtis E. LeMay Center for Doctrine Development and Education, (2018).

[7] White, Franklin E. "A model for data fusion." In Proc. 1st National Symposium on Sensor Fusion, vol. 2, 9 (1988) 149-158.

[8] Dasarathy, Belur V., Sensor fusion potential exploitation-innovative architectures and illustrative applications., Proceedings of the IEEE 85, no. 1, 14 (1997) 24-38.

[9] Harris, C. J., A. Bailey, and T. J. Dodd. "Multi-sensor data fusion in defence and aerospace." The Aeronautical Journal 102, no. 1015, 15 (1998) 229-244.

[10] Heart failure & COPD Clinical Pathways, Alberta Health Services, https://www.albertahealthservices.ca/scns/Page13643.aspx (accessed February 9, 2021)

[11] Adult Emergency Department Asthma Care Pathway (EDACP), Lung Health Foundation, http://hcp.lunghealth.ca/wp-content/uploads/2020/02/Adult_ED_Asthma_Care_Pathway.pdf (accessed February 15, 2021).

[12] Community-Acquired Pnemonia, Connecticuit Children’s, https://www.connecticutchildrens.org/clinical-pathways/community-acquired-pneumonia/ (accessed February 9, 2021).

[13] Klein, Lawrence A., Sensor and data fusion: A tool for information assessment and decision making., SPIE Press, 1 (2004) 51.

Address for correspondence

Lisa Ewen, lewen@lakeheadu.ca

Dr. Sabah Mohammed, sabah.mohammed@lakeheadu.ca

Dr. Arnold Kim, docere@gmail.com.