EXPERT SYSTEMS FOR DISASTER FORECASTING WARNING RECOVERY AND RESPONSE IN WATER RESOURCES MANAGEMENT

by

XIAOYIN ZHANG

GARY P. MOYNIHAN， COMMITTEE CHAIR
ANDREW N. S. ERNEST, COMMITTEE CO-CHAIR

GLENN A. TOOTLE
MARK ELLIOTT
ABDOUL A. OUBEIDILLAH

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Civil, Construction, and Environmental Engineering
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2017

ABSTRACT

Disaster forecasting, warning, recovery, and response in water resources management require the application of knowledge from a diverse range of domains. Identifying the appropriate approach necessitates integrating rules and requirements from these knowledge domains in such a way that the operational goals are achieved with minimally available situational information. Disaster forecasting, warning, recovery, and response must be able to adapt and evolve as new information becomes available. To date, there has been a limited amount of work developing expert systems in this area. In order to fill the knowledge gap, this study 1) identifies and assimilates the knowledge necessary for Water Distribution Network (WDN) decontamination, local flood forecasting and warning, and local flood response coordination and training; 2) determines the relative utility of architectures of expert systems and conventional codes; 3) evaluates the relative benefits of forward and backward chaining inferential logic in these scenarios. Based on the outcome of the conceptual systems, we develop three complete backward chaining expert systems, respectively. With extensible knowledge bases combined with the information provided by the users, the expert systems successfully provide reasoning routines, recommendations, and guidance on disaster forecasting, warning, recovery, and response in water resources management.

DEDICATION

I dedicate this research to my family: my father, Mingxi Zhang (张明西), my mother Naizhen Ye (叶乃珍), my husband, Zongtang Fang (方宗堂), my daughter, Lucy Zhang Fang (方张露兮), and my son, Lucas Zhang Fang (方张禄开) for their unconditional love, support and encouragement.

LIST OF ABBREVIATIONS AND SYMBOLS

AI	Artificial Intelligence
ARC	American Red Cross
CIPAC	Critical Infrastructure Partnership Advisory Council
DHS	U.S. Department of Homeland Security
DOL	U.S. Department of Labor
DSS	Decision Support System
EPA	U.S. Environmental Protection Agency
ES	Expert System
FEMA	Federal Emergency Management Agency
G_E	Goal Exceedances
G_{flow}	Goal Flow
G_{fore}	Goal of making the forecast and warning
G_I	Goal Interaction
G_{local}	Goal Local
G_m	Goal of warning stage
G_{rain}	Goal Rainfall
G_{set}	Goal of training the system
G_{stage}	Goal of warning message
G_T	Goal Treatment Technologies

GUI	Graphic User Interface
G_W	Goal Warnings
ICS	Incident Command System
IS	Information System
IWRM	Integrated Water Resources Management
LFFWS	Local Flood Forecasting and Warning Systems
LFRS	Local Flood Response Coordination and Training System
MTF	Management and Transition Framework
NACS	Multiagency Coordination Systems
NGO	Nongovernmental Organization
NIC	National Integration Center
NIMS	National Incident Management System
NPS	National Preparedness System
NRF	National Response Framework
PI	Public Information
PPD	Presidential Policy Directive
PyKE	Python Knowledge Engine
R_E	Rule Exceedances
R_I	Rule Interaction
ROS	Response Operational Structure
RRA	Roles, Responses, and Actions
R_T	Rule Treatment Technologies
R_W	Rule Warnings
V&V	Verification and Validation

VD	Volunteers and Donations
VOAD	National Voluntary Organizations Active in Disaster
WDN	Water Distribution Network
WMO	World Meteorological Organization

ACKNOWLEDGMENTS

This research and dissertation efforts were completed with the combined efforts and support of many incredible people: family members, classmates, professors, friends and everyone contributed to this achievement. First, I would like to thank two of my advisors, Dr. Gary Moynihan and Dr. Andrew Ernest. Their guidance and understanding have been instrumental in the development of my research skills and interests and in the completion of this work. I would also like to express my gratitude to my committee members Dr. Glenn Tootle, Dr. Mark Elliott, and Dr. Abdoul Oubeidillah. Their technical assistance was invaluable in developing and polishing this research. I would also like to thank all my colleagues that I have had the privilege to work with: Joseph Gutenson, Lian Zhu, and Sahar T. Sadeghi. Without their assistance, this dissertation could have never been completed. Special acknowledgement must be extended to the Science & Technology Directorate of the US Department of Homeland Security (DHS) and the Department of Civil, Construction, and Environmental Engineering of the University of Alabama. Without their funding, the cost of this research would have been prohibitive. On a personal note, I would like to express my gratitude to my family and friends for their continuous support over the years that have led to this effort.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

This chapter introduces the motivation, needs, and current studies of expert systems for disaster forecasting, warning, and response in water resources management. After an overview of the background information of the decision-making process, this chapter lists a variety of issues in the water resources management area. Then, this chapter expresses the proposed solutions to specific problems (such as drinking water contamination and flooding) addressed in this paper. Entailed is a summary of our research.

Problem Statement

Decision-making is the thought process of identifying and selecting a choice from the available alternatives based on the values and preferences of the decision-maker by his or her reasoning and critical thinking analysis (Decision Making, 2015). Oftentimes, fuzzy or insufficient data, diverse spatial information, variable characteristics, and significantly short period of time challenge the decision maker’s ability, expertise, and reliability to effectively optimize the solution pathway. This is true during both drinking water contamination scenarios and flooding events. The decision maker in both instances must take into account various information streams and produce recommendations promptly.

Nature of the Problem

Water is essential for life to survive, society to develop, and economy to boom. On Earth, only 2.5% of water resources are fresh (United States Geological Survey, 2017). Climate change, pollution, and increasing demands for drinking, sanitation, irrigating, manufacturing, etc. have made clean water even scarcer (Seckler, et al., 2010). Nowadays, more than 40% of the global population is suffering from water scarcity. To ensure the sustainability of water use and the resilience to climate change, integrated water resources management (IWRM) has become a progressively urgent topic in several locations worldwide (The World Bank, 2017).

Water resources are inherently difficult to be managed by conventional approaches because of the uncertainty of the availability and usage. Water can cross various boundaries. The uses of upper stream users affect the water quality and quantity available for the lower stream users. Climate change, aberrant weather (such as stubborn droughts and major floods), and other unpredictable conditions fluctuate the planned demands and usages. Ideally, water resources management should address all demands and distribute water on an equitable basis to satisfy all usages. However, this is rarely possible since the financial value, or significance, of the diverse uses or competing demands is hard to evaluate in the commonly acceptable ways (McNabb, 2017; Cardwell, et al., 2009).

Numerous researchers have directed their effort at optimizing the use of fresh water. On the one hand, some scientists focused on water governace and water policy issues. Examples are as follows.

Pahl-Wostl and Knieper (2014) analyzed the underlying feature of effective polycentric governance by using the method of Fuzzy Set Qualitative Comparative Analysis and made a distinction between polycentric, fragmented, and centralized governance regimes. The authors suggested that combination of different governance modes was promising (Pahl-Wostl & Knieper, 2014).

Kolinjivadi and Kosoy (2014) illustrated that payments for ecosystem services (PES) were useful in achieving integrated and adaptive water resource management, but it was impossible to effectuate sheer market-based trades for regulating, cultural and supporting ecosystem services. To reflect the economic characteristics of services by governance arrangements, the authors offered a nested institutional framework for watershed governance (Kolinjivadi & Kosoy, 2014).

Benson et al. (2015) reviewed the burgeoning nexus literature and compared two divergent national contexts, the UK and Bangladesh. The evidence suggested that the nexus had not usurped IWRM. Larger scope for merging of nexus thinking within IWRM was needed because integrating between water, energy, climate, and agricultural policy objectives was generally limited (Benson, et al., 2015).

Halbe et al. (2013) proposed a new methodology for the integrated analysis of water resources management and governance systems to elicit and analyze case-specific management paradigms. The new system combined participatory modeling and analysis of the governance system by using the Management and Transition Framework (MTF) to investigate case-specific management paradigms. Finally, the author provided a case study on flood management in the Tisza River Basin in Hungary (Halbe, et al., 2013).

On the other hand, a number of engineers and experts were intrested in the methods and techiques for water resource management. Below are several prosperous applications in this area.

Fang et al. (2014) developed a prototype Integrated Information System (IIS) called Water Resource Management Enterprise Information System (WRMEIS) based on geoinformatics including technologies such as Remote Sensing (RS), Geographical Information Systems (GIS), Global Positioning Systems (GPS), Enterprise Information Systems (EIS), and cloud services. WRMEIS integrated functions such as data acquisition, data management, sharing, and modeling. Based on the WRMEIS structure, a system called SFFEIS (Snowmelt Flood Forecasting Enterprise Information System) has been implemented to water resource management (Fang, et al., 2014)

Geissen et al. (2015) presented a concept that illustrated the current state of the art and challenges for monitoring programs, fate and risk assessment tools and requirements for policies on emerging pollutants for sustainable water resource management (Geissen, et al., 2015).

Alterburger et al. (2015) described the vision of the international, EU-funded project SOLUTIONS. Three routes (multi-residue target and non-target screening techniques; bioanalytical tools; and effect-directed analysis) were explored to link the occurrence of chemical mixtures at specific sites to the assessment of adverse biological combination effects for water resource management. The proposed approach can provide guidance for future solution-oriented environmental monitoring and explore more systematic ways to assess mixture exposures and combination effects in future water quality monitoring (Altenburger, et al., 2015).

Specific Problems to be Addressed

In the United States (U.S.), the tap water is generally considered of high quality if it comes from a public water system. This maybe publicly or privately owned (U.S. EPA, 2015). However, drinking water is not always safe, even in the U.S. Contamination emergencies caused by natural disasters and/or human activities have affected entire water distribution networks (WDN) or isolated partitions (Fulton, 2015; Wei, 2014; Chiketo, 2012; Sharma & Kansal, 2011; CBC News, 2010). In the aftermath of drinking water contamination events, experts and water utility managers must evaluate various emergencies, and varying severities of damage to water systems and related infrastructures by quickly processing specific knowledge about a diversity of related disciplines and situational data from a variety of information streams of the impacted WDN. Frequently, the decision makers rely on experience, simple personal heuristics and intuition to "guess" applicable solutions. Therefore, false alarms and delay response to drinking water contaminations have been the main problems for years (Xin, et al., 2017; Sankary & Ostfeld, 2017).

Floods are the most common and most destructive natural hazards on Earth. Floods have endangered lives, damaged properties, and threatened the environment worldwide. In recent decades, floods have caused one-third of economic losses. And almost half of the people killed in natural disasters have been victims of floods. To minimize the undesired impacts of floods, numerous measures have been devised. Among those approaches, flood forecasting and warning is widely accepted as a necessary component for flood risk management strategies. Flood forecasting and warning is efficient in protecting lives by facilitating timely evacuation, reducing the losses, and optimizing the use of floodwater. (U.N. ESCAP, 2016; APFM, 2013).

Flood response is an essential step after flood forecasting and warning. An effective response requires accurate forecasting, timely warning of floods, efficiency plans and qualified responders who can deliver the core capabilities of flood response promptly in numerous cases, especially in emergency cases. Possible flood responders include authorities, transport and communications operators, emergency service providers, and family individuals. To ensure every successful response, all responders should have adequate training on structural roles, responsibilities, and actions to deliver the core capabilities (Flin, et al., 2008).

We are living in an era of “big data”. Collecting and utilizing large amounts of data has been a trend in water resources management for the past few decades. The rapid growth of computer science enables us to develop and apply novel approaches to increase the efficiency of current network based water resources management and extend functionalities of today’s systems (Eggimann, et al., 2017). For example, decontaminating water distribution networks requires the application of knowledge from a diverse range of domains including health impact, utility operations, chemical characteristics, and hydraulics. Flood forecasting and warning needs a huge amount of hydraulic, hydrologic, and meteorological data to run models. Flood response engages responders from a board range of levels and each corresponds to distinct role, responsibilities and actions. In addition, the drinking water decontamination scenarios and flood events are both disasters. Quick decision-making and rapid actions are desired in such emergencies. Therefore, computer-based tools or Information Systems (IS) are preferred to collect, process, store, analyze, and disseminate information for our specific water management problems (Zwass, 2017).

Proposed Solution

Information systems have long been applied to solve managerial problems. To adapt a variety of functions for particular purposes, information systems consist of numerous types such as Data Processing Systems, Management Information Systems, Decision Support Systems (DSS), and Artificial Intelligence Systems (AI) (Panetto & Cecil, 2013; Fang, et al., 2014). Decision support systems are one of the popular types of IS applied in water resource management. As the name implied, decision support systems are designed to assist both analysts and managers in the process of decision-making through sophisticated software technologies, operations research and management science models (Giupponi & Sgobbi, 2013).

An expert system is a form of Artificial Intelligence that can emulate a human problem solver by applying knowledge and reasoning normally known and used by experts in that specific field (Jackson, 1998). Expert systems solve problems that conventional programming techniques cannot address. This technology works especially well with applications that deal with an ambiguous, uncertain, or complex subject where human intuition or heuristic knowledge is scarce and highly valued (Velasquez & Hester, 2013; Afshar, et al., 2015).

An expert system has an inference engine and a knowledge base. To simplify and expedite the programming process, IT specialists shape an inference engine as a general-purpose shell. Repository for the problem-specific heuristics. These heuristics are normally obtained from a human domain expert, then structured and input by the knowledge engineer through the system's interface and support environment (Jackson, 1998; Medsker & Liebowitz, 1994). Edward Feigenbaum, the father of expert systems, asserted a knowledge base is the power source of an expert system (Feigenbaum, 1977). Typically, an inference engine works either in a forward chaining or backward chaining mode. Forward chaining, also referred as to data-driven, works top-down to assert conclusions or new facts. Backward chaining, also referred as to goal-driven, works bottom-up to determine what facts must be asserted (Velasquez & Hester, 2013; Heckerman, et al., 1992).

The most popular computer languages for programming expert systems include Visual Basic (Spyridakos, et al., 2005), Java or JESS (Java Expert System Shell) (Robindro & Sarma, 2013), CLIPS (Ooshaksaraie & Basri, 2011), MATLAB or NETLAB (Mounce, et al., 2010), Visual Rule Studio (Chau & Phil, 2004), ART*Enterprise (Leon, et al., 2000), and PyKE (Python Knowledge Engine) (PyKE, 2015).

Recently, a considerable amount of research utilizing AI or expert system technology has been applied to water management. The examples include a variety of different water management issues.

Altunkaynak (2014) combined wavelet transform, fuzzy logic and multilayer perceptron techniques to obtain new approaches for forecasting lake level fluctuation based on the historic water level data for the period between 1855 and 2006 (Altunkaynak, 2014).

Cretescu et al. (2013) discussed some aspects of the development of an expert system for surface water monitoring in the setting of sustainable management of water resources in Romania and its neighborhood. Technical considerations were addressed focusing on the water quality monitoring. Mathematical models from the Mike software package were used for the dispersion of chemical pollutants, and the evolution of water quality indicators. Finally, a monitoring expert system for Bahlui River, especially the urban area of Iasi City, was presented (Certescu, et al., 2013).

Mavrommati et al. (2013) proposed a system dynamics approach for Ecologically Sustainable Development (ESD) in urban coastal systems. The systematic analysis based on theoretical considerations, policy analysis and experts' knowledge. The principles underlying ESD feed the development of a System Dynamics Model (SDM) (Mavrommati, et al., 2013).

Marti et al. (2012) proposed the methodology for a simple and fast expert system (ES) applied to the Hardhof well field. The study compared its performance to the historical management method. The ES is quite simplistic but considerably improves the water quality in the drinking water wells. The knowledge base is crucial for successful management application (Marti, et al., 2012).

Wang et al. (2015) developed a modeling framework and a mixed integer optimization model (MIOM). This model combined Quantitative Qater Rights (QWR) objectives and Duration-based Water Rights (DWR) tools. Results illustrated that the reliability of DWR allocation was enhanced, and Water Intakes Closing (WIC) duration was reduced significantly compared with current regulation (Wang, et al., 2015).

Kenov and Ramos (2012) established the applicability and limitations of two commercial software products to simulate the operation of a water system based on the Sorraia water project in Portugal. Two products AQUATOOL and WEAP were analyzed focusing on the capabilities to reproduce the operation of a water system; estimate the system's reliability to meet water demands; easiness of the modeling process and usefulness to support decisions of water authorities (Kenov & Ramos, 2012).

Ladopoulos (2013) designed a non-linear real-time expert water management telematics system to minimize the waste of 15% to 40% of the water supplies in the ground pipe networks because of leakage. Using the step testing method, this expert system determined the approximate location of leakage for every big urban area. ARC/INFO Geographic Information Systems (GIS) are used to deal with data (Ladopoulos, 2013).

Francis et al. (2014) used Bayesian Belief Networks (BBNs) to construct a knowledge model for pipe breaks in a water zone with the hope of contributing to the "hypothesis-generating" class of models. The model was learned from pipe breaks and covariate data from a mid-Atlantic United States drinking water distribution system network (Francis, et al., 2014).

Although expert systems have been applied to a number of areas of water resource management, there is a lack of research focused specifically on WDN decontamination, flood forecasting and warning, and flood response coordination and training. To fill the knowledge gap, our study aims to identify and assimilate the necessary knowledge;

determine the relative utility of architectures of expert systems and conventional codes; and evaluate the relative benefits of forward and backward chaining inferential logic in the three issues.

Current Studies

Due to the unstructured or semi-structured nature of decision making in WDN decontamination scenarios and flooding events, effective decision making during both cases relies upon historical human expertise. Further, the decision making procedures usually start with obscure and limited information at hand and must be able to adapt and evolve as new information becomes available. Decision makers need expert systems that can 1) learn from human experience; 2) mimic the logic procedure of thinking and reasoning of human experts; 3) recognize and collect relative data with progressing goals; 4) work out and optimize the possible solutions in a stable and fast way; 5) explain the reasoning routines; 6) respond to the users in a friendly and human way; 7) can easily adapt or change to meet new standards or methods of decision making.

In this regard, expert system with a backward chaining inference engine can support decision makers with recommendations in both procedures. The research presented in this dissertation will utilize these technologies. For drinking water distribution systems, the expert system Decon is employed to provide recommendations following contamination events. For local flooding events, expert systems LFFWS and LFRS is used to provide intuitive warning message and operational response recommendations by taking advantages of historical experiences and heuristic expertise.

This dissertation is a collection of three articles with introduction and conclusion sections that summarize the research. Each article is a standalone effort supported with its own literature review, methodology, and discussion sections.

Article One-Evaluation of the Benefits of Using a Backward Chaining Expert System for Water Distribution Networks

This study 1) identifies and assimilates the knowledge necessary for Water Distribution Network (WDN) decontamination; 2) determines the relative utility of architectures of expert systems and conventional codes; and 3) evaluates the relative benefits of forward and backward chaining inferential logic in WDN decontamination. Based on the outcome of the conceptual systems, we develop a complete backward chaining expert system for WDN decontamination, called Decon. With its extensible knowledge base combined with the information provided by the users, Decon provides reasoning routines and recommendations on the type of event and consequences on the water operator's clients, the public in general, the environment, and the potential hazard from the different interactions with the network pipe material. It also gives the users guidance on the currently available technologies and their effectiveness along the best way to apply them for a thorough and expedited solution.

Article Two-Evaluation of the Benefits of Using a Backward Chaining Expert System for Local Flood Forecasting and Warning

This paper discusses the development and implementation of an expert system, LFFWS, for local flood forecasting and warning. LFFWS has a wide-ranging static knowledge base. With the dynamic user input, this expert system provides reasoning procedures and quick forecasting on the flood warning stages, possible consequences, and recommendations for community managers, landowners, or the public in general without enormous computational resources.

Article Three- Evaluation of the Benefits of Using a Backward Chaining Expert System for Local Flood Response Coordination and Training

An expert system for local flood response coordination and training (LFRS) was developed and introduced in this paper. LFRS can help emergency managers construct scalable, flexible, and adaptable coordination structures and support educating flood response entities, such as individuals, communities, nongovernmental organizations, private sector entities, and local governments. The output of the prototype expert system contains two CSV formatted reports as well as prompt screens. The operational structure report hierarchically depicts the crisscross linkages among all responders, their primary functions, and contact information. Another report summarizes the responsibilities and actions of a certain role of flood responders from commanders to individuals.

2. EVALUATION OF THE BENEFITS OF USING A BACKWARD CHAINING EXPERT SYSTEM FOR WATER DISTRIBUTION NETWORKS DECONTAMINATION

2.1. Introduction

The safety of drinking water is important to the economic growth and social development all over the world. Water safety in water distribution network (WDN) has long been an issue as water can be polluted by natural or anthropogenic activities and decontamination of WDNs is difficult due to the complexity and/or emergency. After a contamination event takes place, to find an efficient and effective solution in a short time, human experts and decision makers must process knowledge from different fields as well as a large number of information streams about the impacted WDN. Issues of amalgamating knowledge to determine optimum solutions vary: 1) lack of a well-defined policy framework; 2) difficulty in information collection; 3) variability of WDN topology; 4) aging of WDN; 5) shortage of human experts with many years of empirical knowledge; 6) easy loss of non-user friendly access to the tacit and undocumented expertise once the specialists leave or retire; 7) unreliability of heuristic experience from different human experts; 8) limitation of expertise in the distinct areas (Kulshrestha & Khosa, 2010). As a result, in any new event, the expert only relies on his/her simple personal heuristic and intuitive feelings to seek the “best” applicable solution.

In short, decision making in WDN decontamination episodes, which are oftentimes unstructured or semi-structured is a potentially complex problem where expertise is highly valued. Furthermore, the decision making procedures usually start with obscure and limited information at hand and must be able to adapt and evolve as new information becomes available. Decision makers in WDN decontamination need an expert system that can 1) learn from human experience; 2) mimic the logic procedure of thinking and reasoning of human experts; 3) recognize and collect relative data with progressing goals; 4) work out and optimize the possible solutions in a stable and fast way; 5) explain the reasoning routines; 6) respond to the users in a friendly and human way; 7) can easily adapt or change to meet new standards or methods of decision making.

2.2. Background

To develop a strategy or plan that supports priorities for Water Sector decontamination and recovery for the purpose of water security, Critical Infrastructure Partnership Advisory Council (CIPAC) Water Sector Decontamination Working Group identified and prioritized 16 key decontamination issues and 35 corresponding recommendations in their 2008 report "Recommendations and Proposed Strategic Plan: Water Sector Decontamination Priorities" (referred to as the CIPAC 2008 report in the rest of this paper). Influenced by the report, we aim to develop an expert system for WDN decontamination to address the most necessary precursors of the issues: especially, Priority Issue 4: Decision-making frameworks for decontamination, Priority Issue 7: Utility communications to public officials, responders, the public and others on decontamination, and Priority Issue 9: Treatment procedures of contaminated drinking water and wastewater (Critical Infrastructure Partenership Advisory Council Water Sector Decontamination Working Group, 2008).

This system can be programmed as an expert system or a conventional procedural code.

Expert systems as one successful form of Artificial Intelligence (AI) technology emulates the decision-making ability of a human expert by reasoning about knowledge represented primarily as “if-then” rules (Jackson, 1998). Edward Feigenbaum, also referred to as the “father of expert systems,” asserted that expert systems gain their power from the specific knowledge they process, rather than from any one particular scheme or formalism (Feigenbaum, 1977). Any expert system consists of two parts: 1) a general-purpose shell or a packaged inference engine created by IT specialists to simplify and expedite the whole programming process and 2) knowledge base in any specific area deduced and compiled by domain experts. Inference engines work primarily as forward chaining or backward chaining. Forward chaining starts with the known facts and then works top-down to assert new facts. Backward chaining begins with goals and then works bottom-up to determine what facts must be asserted so that the goals can be achieved (Hayer-Roth, et al., 1983). Once the first part, a general-purpose shell or an inference engine, is packaged, domain experts in various fields can work independently to develop their specific expert systems. In contrast, writing a program like the expert system for WDN decontamination in a traditional procedural language, such as an assembly language or a high-level compiler language (C, Pascal, COBOL, FORTRAN, etc.), requires that the IT specialists and domain experts work together. We refer to this process as conventional procedural codes in the rest of this paper. Thus, expert systems have the advantages of rapid development, easy maintenance, and flexible running with evolving goals (Wong & Monaco, 1995).

Expert system programming languages include Visual Basic (Spyridakos, et al., 2005), Java or JESS (Java Expert System Shell) (Robindro & Sarma, 2013), CLIPS (Ooshaksaraie & Basri, 2011), MATLAB or NETLAB (Mounce, et al., 2010), Visual Rule Studio (Chau & Phil, 2004), ART*Enterprise (Leon, et al., 2000), and PyKE (Python Knowledge

Engine) (PyKE, 2015).

However, it is a challenge to encode knowledge in structured computer programs. One bottleneck is the acquisition of heuristic knowledge. To collect and compile the necessary knowledge in even a small area requires notable effort (Comas, et al., 2003; Kaewboonma, et al., 2013). Recently, a considerable amount of research utilizing AI or expert system technology has been done on water management. The examples include a variety of different water management issues: 1) water quality and quantity monitoring and control of river basin (Certescu, et al., 2013), lake basin (Altunkaynak, 2014), coastal system (Mavrommati, et al., 2013), and well field (Marti, et al., 2012); 2) water rights management (Wang, et al., 2015); 3) irrigation system network management (Kenov & Ramos, 2012); 4) drinking water network management: minimization of water leakage in the pipe networks (Ladopoulos, 2013) and prediction of drinking water distribution system pipe breaks (Francis, et al., 2014). In the water management research community, there is a lack of expert system scholarship on WDN decontamination.

2.3. Objectives

In order to fill the knowledge gap, this study has three major objectives: 1) to identify and assimilate the knowledge necessary to WDN decontamination; 2) to determine the relative utility of architectures of expert systems and conventional codes; 3) to evaluate the relative benefits of forward and backward chaining inferential logic in WDN decontamination.

2.4. Materials and Methods

After identifying the knowledge necessary for WDN decontamination from an extensive literature review, we pursued a process of assimilation into machine readable formats in order to complete the remaining project elements. We compared inferential logic embedded in expert system shells with the procedural logic of conventional codes. We selected an expert system shell that was capable of both backward and forward chaining inferential logic. We developed three conceptual systems: a conventional procedural pseudo-code, a forward chaining expert system framework, and a backward chaining expert system framework. Based on the outcome of the conceptual systems, we chose backward chaining framework and continued to develop a complete backward chaining expert system for WDN decontamination, called Decon. In this section, we introduce the materials and methods we use to develop these three systems.

2.4.1. Goals

In order to easily demonstrate our study, we name our goals as: 1) Goal Exceedances (G_E), to know whether the concentration of contaminant exceeds the permissible limit; 2) Goal Warnings (G_W), to find out whether public health and/or environment is in danger; 3) Goal Interaction (G_I), to work out whether the contaminant harms the WDN infrastructure like pipe degradation; 4) Goal Treatment Technologies (G_T), to suggest the potential treatment technologies for decontamination. These four goals are specific components of the general goal, to respond to a WDN contamination event. These component goals are not comprehensive and do not address every issue that users might face. New demands or goals can be shaped and added to accommodate an expanding set of issues.

2.4.2. Knowledge

In our systems, the knowledge, covering the health impact, utility operations, chemical characteristics and hydraulics of the WDN decontamination area, can be broken down into two primary categories: facts and rules. Facts are simple statements containing data values that represent, and show relationships among entities; Rules are declarative knowledge linking sets of premises and conclusions (Chen, et al., 1985). How the knowledge primitives are incorporated into the systems is dependent on the expert system shell or data repository used to support procedural logic.

With the intention of stability, ease of maintenance, and flexibility of our system, we classify facts as static global facts and dynamic case facts. Like a dictionary, global facts represent those general and common facts applicable to all scenarios. On the other hand, case facts, like a single word or only a few words in the dictionary, record the specific information of each particular case. Case facts can be represented in the knowledge base by a series of questions that function as placeholders for dynamic provided information. Currently, four types of questions would be obtained: 1) Contaminant and its associated concentration; 2) Taxonomies of each unknown contaminant; 3) Materials found in the System; 4) Expected effectiveness of the treatment technologies, if needed. Besides the four essential series of questions mentioned, a unique case ID will be requested to specify different scenarios after the introductory screen as well. The reasoning rules are named correspondingly to the goals they prove. For example, R_E, R_W, R_I, and R_T are four sets of rules to prove G_E, G_W, G_I, and G_T, respectively.

The global facts and rules are extracted from literature and other authoritative information sources. These include possible contaminants and taxonomies (Agency for Toxic Substances & Disease Registry, 2012), EPA drinking water regulations (EPA, 2013), pipe materials (American Chemistry Council, 2004; Knuuttila, et al., 2004), pipe material – contaminant interaction (Ong, et al., 2008), and treatment technologies (U.S. EPA, 2004).

2.4.3. System Architectures

The architectures of the three systems discussed in this study are shown in Figure 2.1, Figure 2.2, and Figure 2.3, respectively. In the procedural code, questions, case facts, and reasoning rules (R_E, R_W, R_I, and R_T) are incorporated with the operation structure (e.g. from querying information to asserting case facts to proving goals in a fixed sequence: $G_E \rightarrow G_W \rightarrow G_I \rightarrow G_T$), but are detached from global facts (referred to as database). In the forward chaining and backward chaining expert systems, operation structures are compiled in inference engines separated from the knowledge base containing questions, case facts, global facts, and rules. Unlike the procedural code, goals in expert systems are parallel proved. The forward chaining inference engine begins with the collection of all available information. However, the backward chaining inference engine starts from the goal selection. Similar in structure to other reasoning rules, the strategy for asking questions is controlled by certain information query rules in the backward chaining system. According to the query rules for each selected goal, the backward chaining inference engine collects necessary information from the existing case facts or previous analyses first, then conducts conversations between users and machines to collect the rest of the essential data (if there is any). After new case facts are asserted, the engine proves a certain goal with all related case facts, global facts, and reasoning rules.

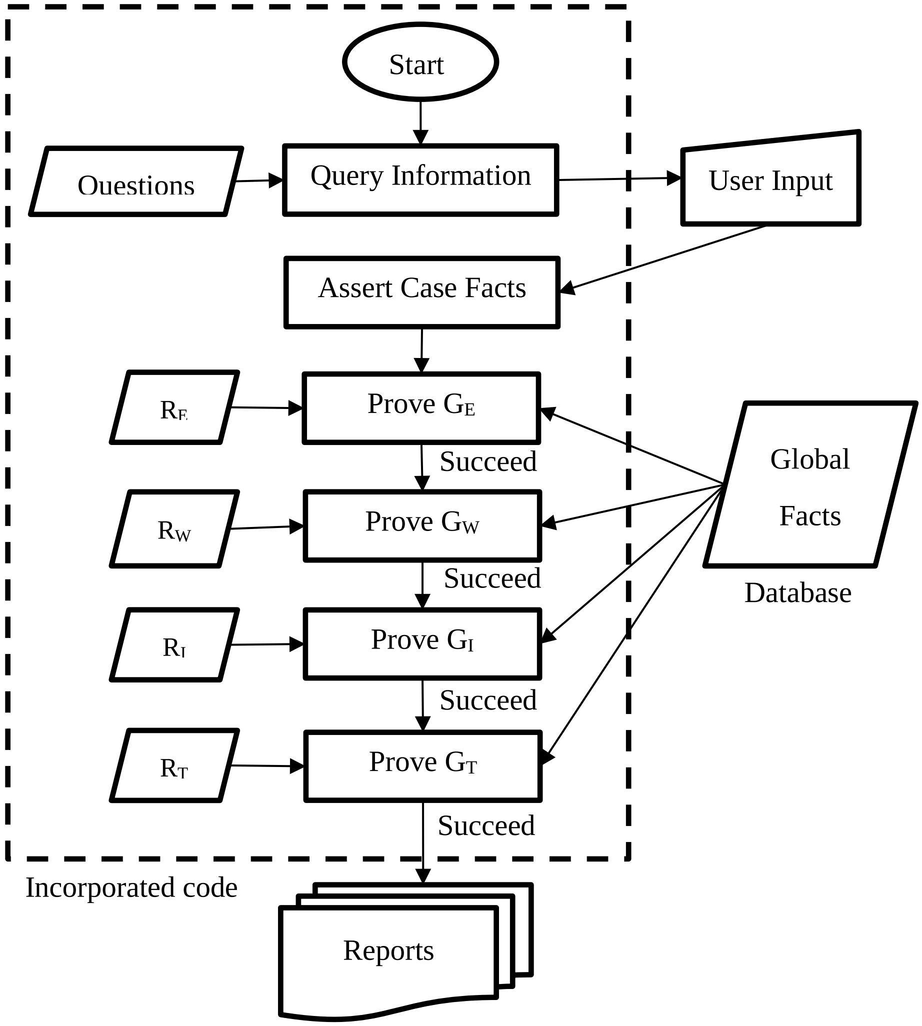

Figure 2.1. Architecture of the conventional procedural code

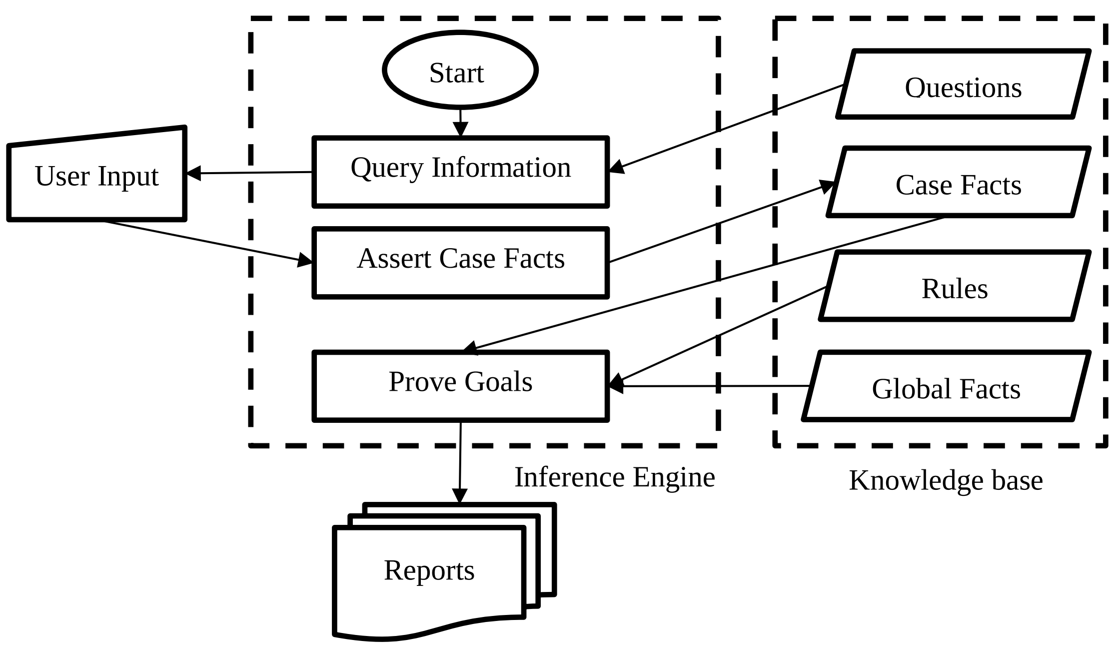

Figure 2.2. Architecture of the forward chaining expert system

2.4.4. Language

We completely developed our backward chaining expert system (Decon), with a combination of Python (Python, 2015) and its Knowledge Engine PyKE (PyKE, 2015) because: 1) Python is an interpreted language which allows quick “ad-hoc” development even once the code is published and deployed; 2) Python is open source and any software based on Python can freely run on almost all platforms including Windows, Mac, Linux, iOS, and Android; 3) PyKE provides a way to directly "program in the large" (PyKE, 2015). This advantage accelerates programming efficiency by reusing code and reducing loops, and it is quite helpful to develop an expert system with a large knowledge base like our system. 4) In addition to forward-chaining logic, PyKE includes backward-chaining logic which enables interactive data acquisition. 5) Extensive libraries are available. For instance, the GUI (Graphic User Interface) of our systems is built on its library Kivy to allow quick and easy interaction design and rapid prototyping (Kivy, 2015). The contents of our systems are similar. For instance, Decon’s contents are summarized in Figure 2.4.

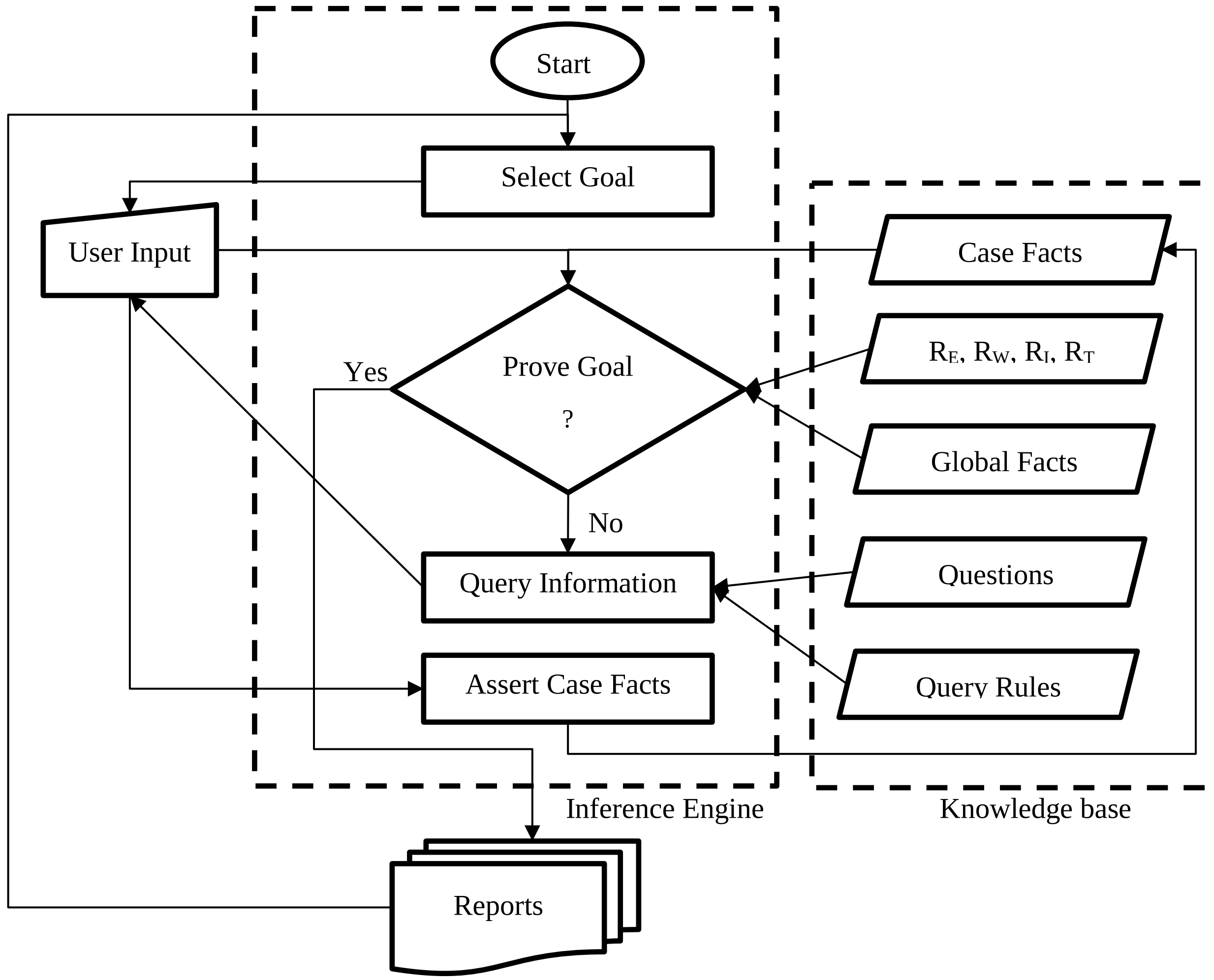

Figure 2.3. Architecture of backward chaining expert system

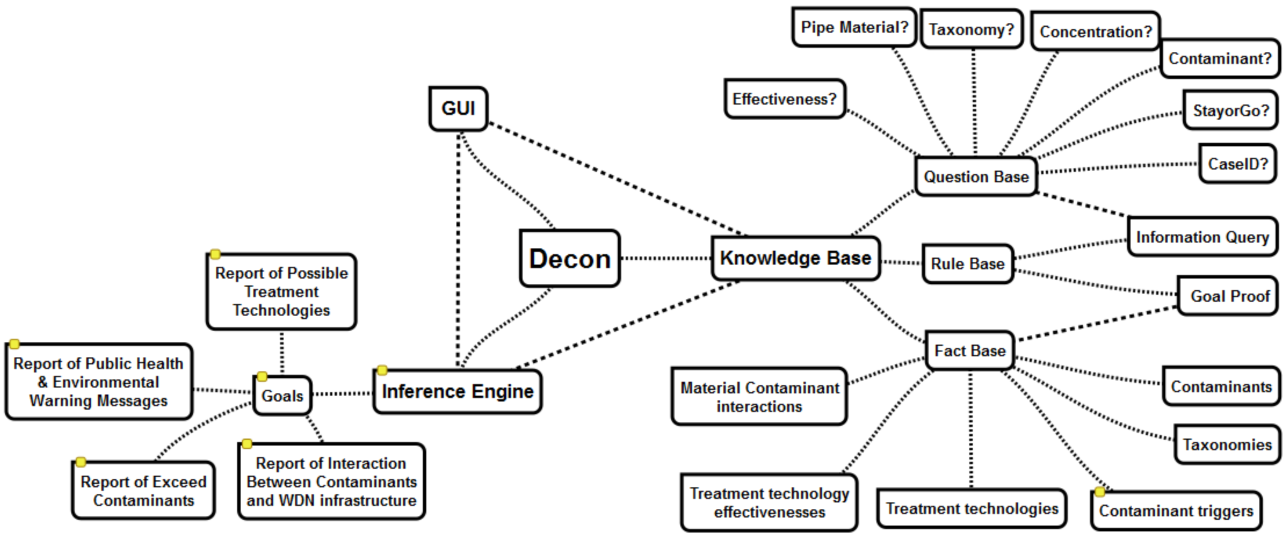

Figure 2.4. Contents of Decon

2.5. Results and Discussion

In the following results and discussion session, we illustrate and analyze the effectiveness of development and advantages of maintenance of expert systems and procedural codes first and then the feasibilities and flexibilities of forward and backward chaining inferential logic.

2.5.1. Expert System vs. Procedural Code

To develop an expert system rather than procedural code for WDN decontamination are proved to have four primary advantages: 1) extensible and explicit knowledge base: Without impact on other components of the system, new knowledge corresponding to existing goals and/or new goals can be easily added; 2) updatable workflow: Without impact on other goals, existing goals can be changed, modified or deleted, and fresh goals related to different decontamination issues can join into the workflow; 3) flexible workflow sequence: Goals can be proved in flexible sequences without changing the original scripts of our system; 4) explanatory capability: Knowledge primitives applied validly to prove a goal can be listed to show the reasoning routines. Details of the benefits are illustrated and discussed below:

2.5.1.1. Benefits of extensible and explicit knowledge base

Consider this scenario: a new contaminant, benzene, is found in a WDN, but not in the knowledge base or database. Adding knowledge primitives about benzene is then required.

In a well-organized procedural code, adding a couple of knowledge primitives in its database is not difficult. However, from its architecture shown in Figure 2.1, the rules of applying certain knowledge to prove goals are incorporated with implicit operation structure. Therefore, besides updating the database, the implicit code has to be modified too. In contrast, putting knowledge about a new contaminant into fact base has no requirement of changing the scripts in inference engines. To demonstrate the details to solve the problem stated in this scenario, sample codes of expert systems in PyKE syntax are given below.

```
IDENTIFIER($ARGUMENT1, $ARGUMENT2, …)
```

Where,

"IDENTIFIER" represents a certain category of facts;

"$ARGUMENT1" and "$ARGUMENT2" represent different facts corresponding to the identifier, respectively.

Figure 2.5. General PyKE syntax of adding facts

In the manner of a general syntax of adding facts shown in Figure 2.5, one category of facts, called "contaminant_is" with four essential information entities can be coded in the following way shown in Figure 2.6:

```
contaminant_is($species, $health, $environment, $source)


where,
 "$species" represents the species of the contaminant;
"$health" represents the warning message on the risk of public health;
"$environment" represents the warning message on the risk of environment;
"$source" represents the possible source of the contaminant.
```

Figure 2.6. General format of one fact category

The last step is to replace arguments shown above with specific facts of benzene in the same sequence as follows (see Figure 2.7):

```
contaminant_is("Benzene","Anemia…","Benzene can …","Discharge from …")
```

Figure 2.7. Specific facts of benzene in the category of "contaminant_is"

In this sample code, text in the first pair of quotation marks is the first argument, "$species", which records the species of the contaminant; text in the second pair of quotation marks is the second argument, "$health", which records the warning message on the risk of environment of benzene; the third is "$environment"; and the fourth is "$source". Unlike procedural code, inference engine of an expert system can automatically search all facts under the same category (referred to as identifier) in every loop.

2.5.1.2. Benefits of Updatable Workflow

Assume the users want to know "whether the contaminant exceeds some standard," then, G_E is required to be added into the workflow of our expert systems.

In procedural code, the workflow is coded in a fixed order and incorporated with questions and rules correspondingly proving the goals. Therefore, inserting one novel goal into the workflow has to change the original scripts of the operation structure; and all existing goals, rules, and questions are potentially impacted. If the new goal fails for some reason, all goals after it will stop. On the contrary, because of the architectures of expert systems, all goals can be parallel proved. After simply inserting a new goal into the workflow, the inference engines can automatically modify the compiled code later. Even if the new goal fails, other existing goals can also be functional. To demonstrate the details to solve the problem stated in this scenario, sample codes of backward chaining expert system in PyKE syntax are given below.

```
with engine.prove_goal('rulebase.RULE_IDENTIFIER($ARGUMENT1, …)') as gen:
    for vars, plan in gen:
    …
```

where,
"RULE_IDENTIFIER" represents a certain rule;
"$ARGUMENT1" represents a certain knowledge primitive corresponding to the rule.

Figure 2.8. General PyKE syntax of adding a goal

In the same manner of the general syntax of adding a goal shown in Figure 2.8, G_E is added by replacing those capitalized parameters with the specific information entities corresponding to G_E shown in Figure 2.9:

```
with engine.prove_goal('rulebase.exceed($cont,$conc,$trigger,$level)') as gen:
for vars, plan in gen:
   …
```

Figure 2.9. Adding a specific goal

Based on our study, we identify that the essential information entities, corresponding to the R_E are contaminant ($cont), actual concentration of contaminant ($conc), drinking water standard ($trigger), and the permissible limit ($level), so four arguments are included in the inner parentheses. In the same fashion, to put an original goal into the workflow, we simply need two steps: 1) copy and paste one old goal; 2) “plug and chug” the rule and facts to prove the original goal. The newly joined knowledge and goal will not affect the existing goals at all. Therefore, we can develop our code by each goal and later combine the tested goals together.

In this study, we only addressed partial issues listed in the CIPAC 2008 report. However, as research goes, new facts corresponding to other contaminants and new rules to prove novel goals will definitely be needed in decision making in WDN decontamination. Therefore, the benefits from the extensible and explicit knowledge base and from the updatable workflow of an expert system will be more and more attractive while more issues are taken into consideration. Partially developed expert systems can also be functional.

2.5.1.3. Benefits of flexible workflow sequence

As the architectures of expert systems shown in Figure 2.2 and Figure 2.3, goals can be parallel proved in various sequences, according to the availability of information at hand. In addition, backward chaining inferential logic also enables the workflow to adapt to the demands of users. However, the procedure code can only work in a fixed workflow, for instance, $G_E \rightarrow G_W \rightarrow G_I \rightarrow G_T$). If G_E fails, then all goals after G_E stop. Therefore, the procedure code cannot cope with the following scenario. Usually, after a pollution event, the first question water sector managers have is, "Does the concentration of the contaminant exceed some standards?", so G_E comes first. However, the species of contaminant might be unknown at the very beginning. Assume the managers just know that the taxonomies of this pollutant are VOC (Volatile Organic Chemical) and hydrocarbon. Although G_E fails without the species of contaminant, known taxonomies still can help fulfilling the other goals of water sector managers, such as "whether the contaminant threatens the WDN infrastructure?" and "what are the potential technologies for decontamination?" In contrast to the incapability of procedural code in such situation, without making any changes to the original scripts, expert systems can skip G_E automatically and prove the G_T or G_I first. Simply speaking, our expert systems can prove the goals in any of the following workflows: $G_T \rightarrow G_I \rightarrow G_E \rightarrow G_W$, $G_I \rightarrow G_T \rightarrow G_E \rightarrow G_W$, $G_E \rightarrow G_W \rightarrow G_I \rightarrow G_T$, and etc.

When more issues of WDN decontamination as fresh goals have been considered, a couple of goals may fail due to the absence of some information at hand or may be skipped because of lack of user interest. The WDN decontamination expert system should enable the users to select some goals to be skipped and to prove some other goals first. At the same time, the system should automatically skip those failed goals and move ahead to other goals. In this perspective, an expert system has striking benefits over a procedural code.

2.5.1.4. Benefits of explanatory capability

On the one hand, conventional procedural codes do not incorporate facts with rules, conventional procedural systems habitually lack the capability of explaining why a fact is deduced or inferred in a particular way. In other words, procedural codes cannot tell the users which facts and rules are applied to create a reasonable conclusion.

On the other hand, expert systems can list valid facts and rules, which are applied to solve problems. From the list, developers can easily track down all reasoning routines, diagnose the logic or syntax errors, and lock the problematic scripts. This explanatory capability is especially helpful when the developers are coding complex courses, such as the procedure of decision making in WDN decontamination. The explanatory capability not only can simplify and accelerate development of the computer system, but also can train those water sector managers with the routines for reasoning in particular scenarios. For example, our expert system can create a new fact in the following way shown in Figure 2.10:

User input: Contaminate is benzene.

Knowledge primitive 1: Benzene is a hydrocarbon.

Knowledge primitive 2: Hydrocarbons can degrade PVC (polyvinyl chloride).

➔ Infer new knowledge primitive: Benzene can degrade PVC.

Figure 2.10. Example of fact assertion

While the developers or users read the final conclusion, they can understand the cause-effective routines from the list of knowledge primitives that validly applied to the goal. Assume that knowledge primitive 1 is accidentally coded wrong, for example, saying: “Benzene is a heavy metal.” Then, the developers will not get the expected conclusion. Instead of searching all facts, the developers search the knowledge primitives on the reasoning list only. Obviously, diagnosing and correction processes are expedited in this way. Therefore, the larger the knowledge base and the more complicated the reasoning routines are, the more appealing the explanatory capability of an expert system is.

In summary, developing an expert system is more proficient than procedural code for decision making in WDN decontamination.

2.5.2. Backward Chaining vs. Forward Chaining

To illustrate how backward chaining mechanism is applied to, and profits our expert system, simplified forward and backward chaining logic to prove goals is shown in Figure 2.11 and Figure 2.12, respectively. Different dash types and arrow types indicate diverse information flows. Take the proof of Goal Exceedance as an example. In this case, only the information about the species and concentration of the contaminant is essential, so the backward chaining inference engine only collect those two facts; while the forward chaining inference engine also blindly collect other information such as pipe material, target removal rate, and taxonomies of the contaminant. The performance of a forward chaining expert system (Gutenson, et al., 2015) that relies on an earlier version of knowledge base shows that users have to provide a complete set of queries for all incorporated goals at the beginning in order for the expert system to complete processing. To allow for response questions only as inferential logic and thus simplify the user experience, the knowledge base in this study is entirely rewritten to backward chaining inferencing.

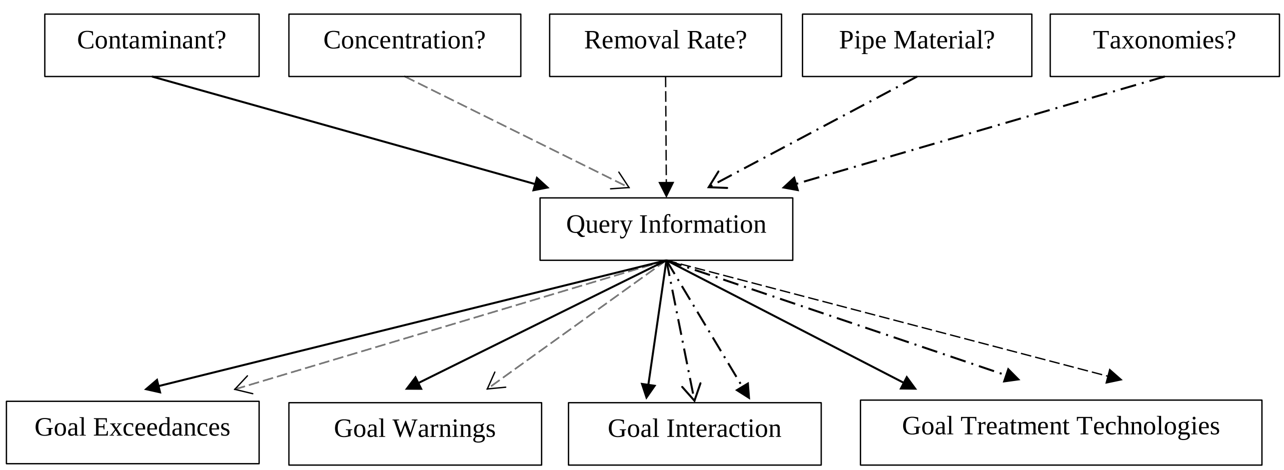

Figure 2.11. Simplified forward chaining logic to prove goals

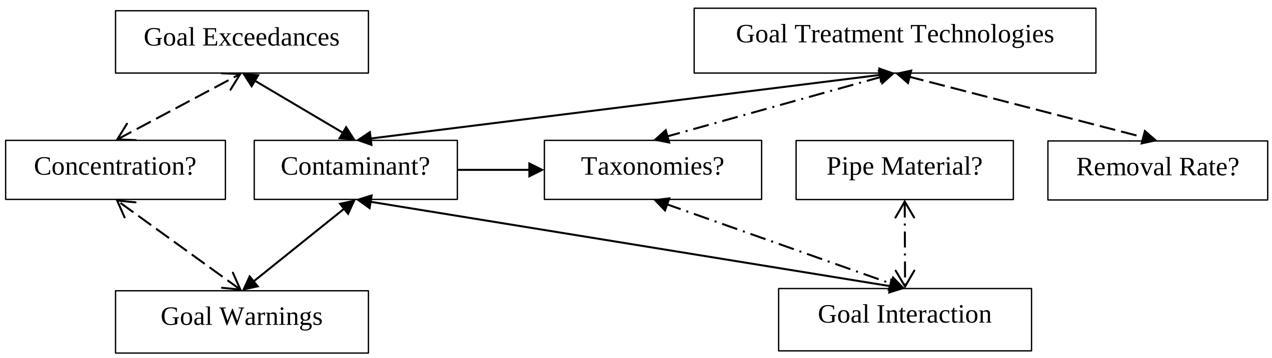

Figure 2.12. Simplified backward chaining logic to prove goals

In fact, pipe material, target removal rate, and taxonomies of the contaminant are useless if no permission level has been exceeded. Even when all goals are desired and all information is vital, user input is not always obligated, because an expert system can infer much information from the analyses of previous results and/or from other user input. As the example demonstrated in "benefits of explanatory capability", the expert system deduces new fact to prove G_I without requesting user input on taxonomies of contaminant. Similarly, the expert system can create another new fact, "Benzene is a VOC", to prove G_T if needed. Actually, it is only when the species of contaminant is unknown and the users need to prove G_I or G_T, then, user input on taxonomies is required. Consequently, despite the necessity of direct user input, using this way to collect all data is a waste of time and computational resources. Besides, the direct user input could be wrong.

2.5.3. Case Study

To demonstrate how collected knowledge works in the packaged expert system shell with backward chaining inference engine for decision making in WDN decontamination, a large fuel tank is assumed to be broken in an accident. Benzene, the most common component of spilled gasoline, permeates into the soil, comes into the pipe through cracks, and finally pollutes the WDN. Such accidents have been reported several times in the US (Atkin, 2014; U.S. Department of Commerce, 2015) .

In this hypothetical disaster, the water utility managers may want to know if they need to worry about something. Beginning with this general and fuzzy question, our expert system quickly recognizes that the next step is to find out whether the contaminant exceeds a certain drinking water standard. In other words: G_E should be proved first. Instead of asking users to input contaminant and its concentration directly and immediately, Decon searches the essential data from previous analysis first, then, asks for user input.

In this scenario, assume the concentration of benzene is 0.05mg/L. From the exceedances report shown in Table 2.1, we can see three permissible limits are exceeded: the MCL of 0.005mg/L, MCLG of 0mg/L, and the MRL COI of 0.0005mg/kg/day (U.S. EPA, 2009).

Table 2.1. Exceedances report as a CSV file

Contaminant	Concentration	Unit	Trigger	Limit	Unit
Benzene	0.05	mg/L	MCL	0.005	mg/L
Benzene	0.05	mg/L	MCLG	0	mg/L
Benzene	0.05	mg/kg/day	MRL COI	0.0005	mg/kg/day

Then, the water sector managers may want to know "How does the contaminant impact the public health and/or environment?" In other words, the general goal evolves to G_w. From the warnings report shown in Table 2.2, we see how the pollution threatens public health and what actions are suggested for water sector manager.

Table 2.2. Warnings report as a CSV file

Contaminant	Concentration	Unit	Alert Type	Action Needed	Health or Environment
Benzene	0.05	mg/L	Public Health	Concentration is sufficiently high to cause a public health concern. Please notify your consumers and your public health agency. Potential health	Anemia; decrease in blood platelets; increased risk of cancer

			impacts include:

After the managers read the warnings report, they may want to know: "Which technologies can be used?" In other words, G_T is required. Assume the target removal rate is 80%. Based on the fact: Benzene is a VOC, all possible technologies with the efficiencies greater than or equal to 80% are listed in the technologies report. In this scenario, nine potentially technologies (activated carbon, activated alumina, air stripping, chlorination, chlorine dioxide, direct filtration, ozonation, ultraviolet, and advanced oxidation processes) and their brief introductions are shown in the report.

Another possible new request could be G_I: "Does the contaminant damage the pipe?" Assume the pipe material of this WDN is PVC. From the interaction report shown in Table 2.3, we can see that our expert system marks benzene as a hydrocarbon and warns the managers on the interaction between contaminant and pipes. From the list of keywords, we can see the reasoning routines as well.

Table 2.3. Interaction report as a CSV file

Contaminant	Keyword	Material	Interaction
Benzene	Hydrocarbon	PVC	Prolonged exposure to hydrocarbons causes PVC to degrade

2.6. References

Agency for Toxic Substances & Disease Registry . (2012, October 16). ATSDR Toxic Substances Portal. (Agency for Toxic Substances & Disease Registry) Retrieved December 3, 2013, from Agency for Toxic Substances & Disease Registry: http://www.atsdr.cdc.gov/substances/index.asp

Altunkaynak, A. (2014). Predicting Water Level Fluctuation in Lake Michigan-Huroon Using Wavelet-Expert System Methods. Water Resource Manage, 28(8), 2293-2314.

American Chemistry Council. (2004, February). Polyvinyl Chloride (PVC): It's Hard to Imagine Life Without It. (American Chemistry Council) Retrieved December 3, 2013, from Chlorine Chemistry.

Atkin, E. (2014, June 13). Company That Caused Historic Chemical Spill Leaks More Waste Into West Virginia Waters. Retrieved August 2014, from Climate Progress: http://thinkprogress.org/climate/2014/06/13/3448678/freedom-industries-spills-again/

Certescu, I., Craciun, I., Benchea, R. E., Kovács, Z., Iavorschi, A., Sontea, V., & Macoveanu, M. (2013). Development of An Expert System for Surface Water Quality Monitoring in The Context of Sustainable Mangement of Water Resources. Environmental Engineering and Management Journal, 12(8), 1721-1734.

Chau, K. W., & Phil, M. (2004). Knowledge-Based System on water resources management in Coastal Waters. Water and Environmental Journal, 18(1), 25-28.

Chen, M.-K. S., Chau, C.-f. C., & Kabat, W. C. (1985). Decision Support Systems: A Rule-Based Approach. ACM '85 Proceedings of the 1985 ACM annual conference on The range of computing : mid-80's perspective: mid-80's perspective (pp. 511-515). New York: Association for Computing Machinery.

Comas, J., Llorens, E., Marit, E., Puig, M., Riera, J., Sabater, F., & Poch, M. (2003). Knowledge acquistion in the STREAMES project: the key process in the Environmental Desicion Support System development. AI Communications, 16, 253-265.

Critical Infrastructure Partenership Advisory Council Water Sector Decontamination Working Group. (2008). Recommendations and Proposed Strategic Plan: Water Sector Decontamination Priorites. Retrieved from http://www.nawc.org/uploads/documents-and-publications/documents/document_ca7f0ed5-0dfe-40ed-afc1-a92a8beb3988.pdf

Feigenbaum, E. (1977). The art of artificial intelligence: themes and case studies of knowledge engineering. School of Humanities and Sxiences, Stanford University, Computer Science Department, Stanford.

Francis, R., Guikema, S., & Henneman, L. (2014). Bayesian Belief Networks for predicting drinking water distribution system pipe breaks. Reliability Engineering and System Safety, 130, 1-11.

Gutenson, J. L., Ernest, A. N., Fattic, J. R., Ormsbee, L. E., Oubeidillah, A. A., & Zhang, X. (2015). Water Expert: a conceptualized framework for development of a rule-based decision support system for distribution system decontamination. Drinking Water Engineering and Science, 8, 9-24.

Hayer-Roth, F., Waterman, D., & Lenat, D. (1983). Building Expert Systems. Addison-Wesley.

Jackson, P. (1998). Introduction To Expert Systems. In P. Jackson, Introduction To Expert Systems (3ed ed., p. 2). Addison Wesley.

Kaewboonma, N., Tuamsuk, K., & Kanarkard, W. (2013). Knowledge Acquisition for the Design of Flood Management Information System: Chi River Basin, Thailand. Social and Behavioral Sciences. 73, pp. 109-114. Budapest: Elsevier Ltd.

Kenov, K. N., & Ramos, H. M. (2012). Water and energy sustainable management in irrigation systems network. International Journal of Energy and Environment, 3(6), 833-860.

Kivy. (2015). Retrieved from http://kivy.org/#home

Knuuttila, H., Lehtinen, A., & Nummila-Pakarinen, A. (2004). Advanced Polyethylene Technologies—Controlled Material Properties. In Long Term Properties of Polyolefins. Berlin/Heidelberg: Springer.

Kulshrestha, S., & Khosa, R. (2010). Expert System for Management of Water Distribution Network (WDN). International Journal of Engineering Science and Technology, 2(12), 7401-7412.

Ladopoulos, E. G. (2013). Non-linear Pipe Networks Water Management Real-Time Expert Telematics System. Journal of Water Resource and Hydraulic Engineering, 2(1), 13-20.

Leon, C., Martin, S., Elena, J. M., & Luque, J. (2000). Explore: Hybrid Expert System For Water Networks Management. Journal of Water Resources Planning and Management, 126(2), 65-74.

List of Contaminants and their (MCLs). (2009, May). Retrieved from United States Environmental Protection Agency: http://water.epa.gov/drink/contaminants/index.cfm

Marti, B., Bauser, G., Stauffer, F., Kuhlmann, U., Kaiser, H.-P., & Kinzelbach, W. (2012). An Expert System for Real-time Well Field Management. Water Science & Technology: Water Supply, 12(5), 699-706.

Mavrommati, G., Bithas, K., & Panayiotidis, P. (2013). Operationalizing Sustainability in Urban Coastal Systems: A System Dynamics Analysis. Water Research, 47(20), 7235-7250.

Mounce, S., Boxall, J., & Machell, J. (2010). Development and Verification of an Online Artificial Intelligence System for Detection of Bursts and Other Abnormal Flows. Journal of Water Resources Planning and Management, 136(3), 309-318.

Oil and Chemical Spills. (2015). Retrieved from U.S. Department of Commerce| National Oceanic and Atmospheric Administration| National Ocean Service| Office of Response and Restoration: http://response.restoration.noaa.gov/oil-and-chemical-spills

Ong, S., Gaunt, K. J., Mao, F., Cheng, C. L., Esteve-Agelet, L., & Hurburgh, C. (2008). Impact of Hydrocarbons on PE/PVC Pipes and Pipe Gaskets. American Water Works Association Research Foundation.

Ooshaksaraie, L., & Basri, N. E. (2011). An Expert System Applied in Constuction Water Quality Monitoring. American Journal of Environmental Sciences, 7(11), 75-81.

PyKE. (2015). Retrieved from Welcome to Pyke: http://pyke.sourceforge.net/

Python. (2015). Retrieved from Python: https://www.python.org/

Robindro, K., & Sarma, S. K. (2013). JESS Based Expert System Architecture For Diagnosis Of Rice Plant Diseases: Design And Prototype Development. 2013 4th International Conference on Intelligent Systems, Modelling and Simulation, (pp. 674-676).

Spyridakos, T., Pierakos, G., Metaxas, V., & Logotheti, S. (2005). Supporting the Management of Measurement Network with an Expert System: The NeMO System. Operational Research An International Journeal, 5(2), 273-288.

U.S. Environmental Protection Agency. (2004). Module 6: Remediation and Recovery Guide. Washington, D.C.: U.S. Environmental Protection Agency.

U.S. Environmental Protection Agency. (2013, June 03). National Primary Drinking Water Regulations. (U.S. Environmental Protection Agency) Retrieved December 3, 2013, from Water: Drinking Water Contaminants.

Wang, Z., Zhu, J., & Zheng, H. (2015). Improvement of Duration-Based Water Rights Management with Optimal Water Intake On/Off Events. water resources management, 29(8), 2927-2945.

3. EVALUATION OF THE BENEFITS OF USING A BACKWARD CHAINING EXPERT SYSTEM FOR LOCAL FLOOD FORECASTING AND WARNING

3.1. Introduction

Flood incidents can endanger human life, cause extensive property damage and result in significant harm to the environment. To attenuate the risk and reduce the loss caused by flood accidents, flood forecasting has been studied and developed throughout human history. Although global or nationwide flood forecasts and warnings are available through mass media, the comparatively low accuracy of prediction for a certain region causes false alarms, improper responses, and therefore unnecessary loss of property and/or life. One conventional method to improve the accuracy is to increase the resolution or decrease the based cluster size. Either way, the occupancy of computational resources must be increased enormously. The higher the resolution and the smaller the cluster size, the more computing power is needed. Another alternative method is to develop standalone systems only for small regions. Recent examples include using ensemble numerical weather prediction systems for medium-range flood forecasting (Cloke & Pappenberger, 2009); applying data-driven approaches, such as traditional artificial neural networks (ANN), adaptive neuro-fuzzy inference systems (ANFIS), wavelet neural networks (WNN), and hybrid ANFIS with multi-resolution analysis using wavelets (WNF) to develop models for hourly runoff forecasting at Casino station on the Richmond River in Australia (Badrzadeh, et al., 2015); coupling meteorological observations and forecasts with a distributed hydrological model to advance flood forecasting in Alpine watersheds (Jasper, et al., 2002); coupling HEC-HMS with atmospheric models for predicting watershed runoff in California (Anderson, et al., 2002); and combining multi-models for operational forecasting for river basins in the Western United States (Najafi & Moradkhani, 2015). Although the models or systems listed above provided overall better performance for the whole river basins or catchments examined in the cited studies, the accuracy of forecasting for a small place such as a small town, a little community, and a specific house was not mentioned or was completely ignored. The reason is the same: to obtain accurate forecasting for a comparatively small place, the resolution of the entire studied region of the river basin or catchment must be higher. More detailed local situations must be collected and considered, more memory space must be allocated and a heavier computational burden must be loaded onto the models that already have vast amounts of meteorological, hydrologic, and hydraulic data to analyze through complicated calculations (Cloke & Pappenberger, 2009). In fact, most incidents begin and end locally and are managed at the local level (DHS, 2013). The most useful data is locally collected, although it is correlated with the data from outside the specific region.

In the meantime, the complicated numerical models or systems employed in recent studies considerably deplete available computing power. Broadly, most numerical models can be categorized as physics-based models or data-driven models. The physics-based models represent the intricate hydrological cycle and transform precipitation into channel flow through hydrologic and hydraulic routing. Oftentimes, these classical rainfall-runoff models with complex mathematical formulations require high computation times (Garcia-Pintado, et al., 2015). In contrast, data-driven models, for example, time series models, focus on the variation of hydrological variables with time and input-output stochastic processes instead of the mechanism of the rainfall-runoff transformation (Badrzadeh, et al, 2015). Generally, the stochastic processes entail the transformation and analysis of big data and thus consume dramatic computing power.

Understandably, decreasing the scales of these catchment-wide or river-wide models and systems of flood forecasting to even smaller local sizes and limiting the usage of numerical models can decrease the engagement of computational resources. Therefore, local flood forecasting systems without sophisticated numerical models are more cost-effective for smaller places, especially for those small communities with limited budgets.

3.2. Background

With the purpose of improving forecasting and forecasting based services, the World Meteorological Organization (WMO) has initiated various programs and projects. The “Manual on Flood Forecasting and Warning” (referred to as the WMO manual hereafter) published in 2011 is one of the crucial outcomes. This manual provides the fundamental knowledge and guidance to develop or to set up applicable and tailored systems for flood forecasting and warning. In addition, the manual offers extensive references to further sources of information in both paper and online formats (WMO, 2011).

A local flood forecasting and warning system can be programmed as an expert system with inferential logic. Expert systems are one successful form of Artificial Intelligence (AI) technology that emulates the decision-making ability of a human expert by utilizing knowledge represented primarily as “if-then” rules (Jackson, 1998). Typically, an expert system consists of an inference engine and a knowledge base. An inference engine, also referred to as a general-purpose shell, is created by IT specialists to simplify and expedite the programming process. A knowledge base is deduced and compiled by knowledge engineers and domain experts in a certain domain to store pertinent facts and rules. A knowledge base is where an expert system gains power, asserted by Edward Feigenbaum, the father of expert systems (Feigenbaum, 1977). An inference engine, working primarily in either a forward chaining or backward chaining mode, applies logic rules to generate new knowledge. Forward chaining, driven by known data, works top-down to assert conclusions or new facts. Backward chaining, driven by goals, works bottom-up to determine what facts must be asserted (Hayer-Roth, et al., 1983). Conventionally, a local flood forecasting and warning system can also be procedurally coded in a traditional procedural language, such as an assembly language or a high-level compiler language (C, Pascal, COBOL, FORTRAN, etc.). In this coding process, IT specialists are required from beginning to end. However, in the coding process of an expert system, IT specialists are not necessary once an inference engine is packaged. Domain experts can work independently or cooperate with knowledge engineers to develop their specific expert systems in various fields. Thus, expert systems can be more rapidly and easily developed and maintained. Expert systems have greater flexibility to run with evolving goals (Wong & Monaco, 1995).

The most popular computer languages for programming expert systems include Visual Basic (Spyridakos, et al., 2005), Java or JESS (Java Expert System Shell) (Robindro & Sarma, 2013), CLIPS (Ooshaksaraie & Basri, 2011), MATLAB or NETLAB (Mounce, et al., 2010), Visual Rule Studio (Chau & Phil, 2004), ART*Enterprise (Leon, et al., 2000), and PyKE (Python Knowledge Engine) (PyKE, 2015).

The bottleneck to encoding knowledge in structured computer programs is knowledge acquisition. Collecting and compiling the necessary knowledge in even a small area involves the effort of countless individuals (Comas, et al., 2003; Kaewboonma, et al., 2013). Recently, a considerable amount of research utilizing AI or expert system technology has been done on flood forecasting and warning (Emerton, et al., 2016; Mabrouk, et al., 2015; Pinto, et al., 2015; Fang, et al., 2015; Ghalkhani, et al., 2012; Mahabir, et al., 2007; Todini, 1999). However, in the local flood forecasting and warning research community, there is a lack of scholarship on expert systems without complex numerical models.

To take advantage of logic programming and the concept of facts and rules, we collect and assimilate local hydraulic and hydrological situations, both local and global historical flooding records, and distilled the wisdom of experts into our systems as the knowledge base or database. By matching the case facts and global facts with rules, the inferential logic determines the flooding forecasting and warnings directly. Simply speaking, the whole process is similar to using weather lore, e.g., “Red sky at night, sailors' delight. Red sky at morning, sailors take warning”. When we see a red sky in the morning, we get the forecast and warning of a storm or bad weather. The following sections will provide a more scientific explanation of our system.

In addition, to benefit from other existing systems, the new systems should be able to read the output data from those large systems as well as the user input data directly. Moreover, the new systems should adapt to other small places in the same region or in other regions if needed. Most importantly, the development and operation of the new systems must occupy less computational resources for a much shorter time and be economically feasible for smaller districts.

3.3. Materials and Methods

Knowledge identified as necessary for local flood forecasting and warning was obtained from an extensive literature review and assimilated into machine-readable formats. To compare the benefits of inferential logic embedded in expert system shells with the procedural logic of conventional codes and the utility of backward chaining with forward chaining, we selected an expert system shell-PyKE that was capable of both forward and backward chaining inferential logic and developed three conceptual systems: a conventional procedural pseudo-code, a forward chaining expert system framework, and a backward chaining expert system framework. Based on the analysis of the conceptual systems, we decided to turn the backward chaining framework into a complete backward chaining expert system for local flood forecasting and warning. In this section, we introduce the materials and methods used to develop these three Local Flood Forecasting and Warning Systems (LFFWS).

3.3.1. Goals

LFFWS consist of two phases: the training phase and the determining phase. In the training phase, LFFWS collect the local hydraulic and hydrologic conditions, historical records, and heuristic expertise to realize the goal (G_{set}): training the system by setting up the variables and parameters for the next determining phase. In the determining phase, the well-trained LFFWS learn the current or proposed situations by interviewing the users to achieve the goal (G_{fore}): making the forecast and warning by matching and comparing current situations with the stored variables and parameters.

In the training phase, LFFWS gather data such as 1) the depth of past severe floods in the local area; 2) the causes of flooding in the local area; 3) the speed at which the stream flow might rise; 4) the length of time floodwater might remain in the locality; and 5) the direction of the flood flow. We categorize these data into three types of triggers: triggers related to rainfall, triggers related to stream flows, and triggers related to local conditions. Correspondingly, we name the goals to set up those variables and parameters regarding the three types of triggers as Goal Rainfall (G_{rain}), Goal Flow (G_{flow}), and Goal Local (G_{local}), respectively.

In the determining phrase, after LFFWS collect sufficient data about the current situation, LFFWS determine a CSV formatted report on the warning stage (G_{stage}) and warning messages (G_{m1}, G_{m2}, and $G_{m...}$ corresponding to various triggers). Figure 3.1 demonstrates the relations between these goals.

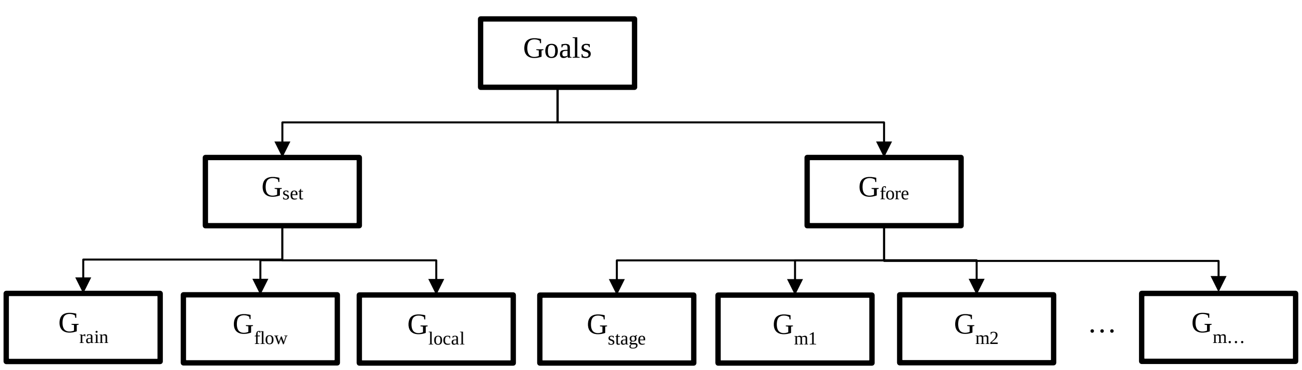

Figure 3.1. Goals of LFFWS

3.3.2. Knowledge

In LFFWS, the knowledge, covering local hydraulic and hydrologic conditions, historical records, and heuristic expertise of local flood forecasting and warning, was classified into two primary categories: facts and rules. Facts are simple statements containing data values that represent and show relationships among entities; rules are declarative knowledge linking sets of premises and conclusions (Chen, et al., 1985). The expert system shell or data repository used to support procedural logic decides on the format needed to assimilate knowledge primitives.

To make LFFWS stable, flexible, and sustainable, facts are categorized as static global facts and dynamic case facts. Those general and common facts applicable to all scenarios are symbolized as global facts, while other specific information about each particular case is denoted as case facts. A chain of questions performed as placeholders represent those dynamically provided case facts in the knowledge base. Currently, three types of questions would be obtained: 1) current or proposed accumulation and intensity of rainfall; 2) current or proposed water depth, velocity, and rise rate of streams; and 3) historical or recorded thresholds of rainfall and stream flow at different stages. LFFWS ask the third type of questions when the system must be reset. In addition to the three essential series of questions mentioned, a unique case ID, rain gauge ID, and stream gauge ID will also be requested to specify different scenarios and locations after the introductory screen. The reasoning rules are named corresponding to the goals they prove. For example, R_{set} and R_{fore} are two main sets of rules to prove G_{reset} and G_{fore}, respectively. Specifically, R_{rain}, R_{flow}, and R_{local} are sets of rules to prove G_{rain}, G_{flow}, and G_{local}, respectively. R_{stage}, R_{m1}, R_{m2}, and $R_{m\ldots}$ are sets of rules to prove G_{stage}, G_{m1}, G_{m2}, and $G_{m\ldots}$. Figure 3.2 illustrates the relations between these rules. The global facts and rules are extracted from the literature and other authoritative information sources. (WMO, 2011).

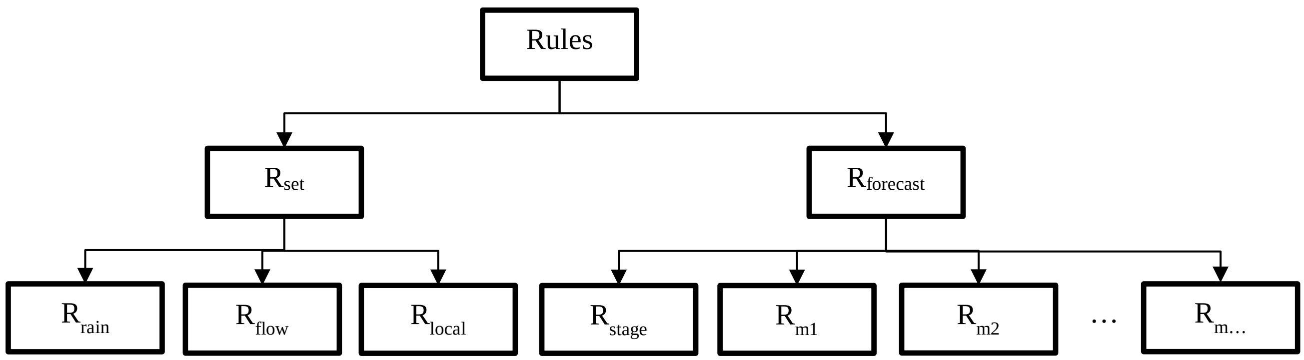

Figure 3.2. Rules of LFFWS

3.3.3. System Architectures

In the system architecture of the procedural code shown in Figure 3.3, questions, case facts, and rules are incorporated into the operation structure (e.g., from querying information to asserting case facts to proving goals in a fixed sequence: $G_{rain} \rightarrow G_{flow} \rightarrow G_{local} \rightarrow G_{stage} \rightarrow G_{m1} \rightarrow G_{m2} \rightarrow \ldots \rightarrow G_{m\ldots}$) but are detached from global facts (referred to as the database). The system architectures of forward chaining and backward chaining expert systems are, respectively, shown in Figure 3.4 and Figure 3.5. Unlike the procedural code, inference engines assemble operation structures separated from the knowledge base containing questions, case facts, global facts, and rules. Goals in expert systems are proved in parallel. The forward chaining inference engine begins by gathering all available information; however, backward chaining starts from the goal selection and collects necessary information for the certain goals. The backward chaining inference engine searches for the needed data from the existing case facts or previous analyses first, then interviews users for the remainder (if there is any) according to the query rules. After new case facts are asserted, the engine proves particular goals with all related case facts, global facts, and reasoning rules.

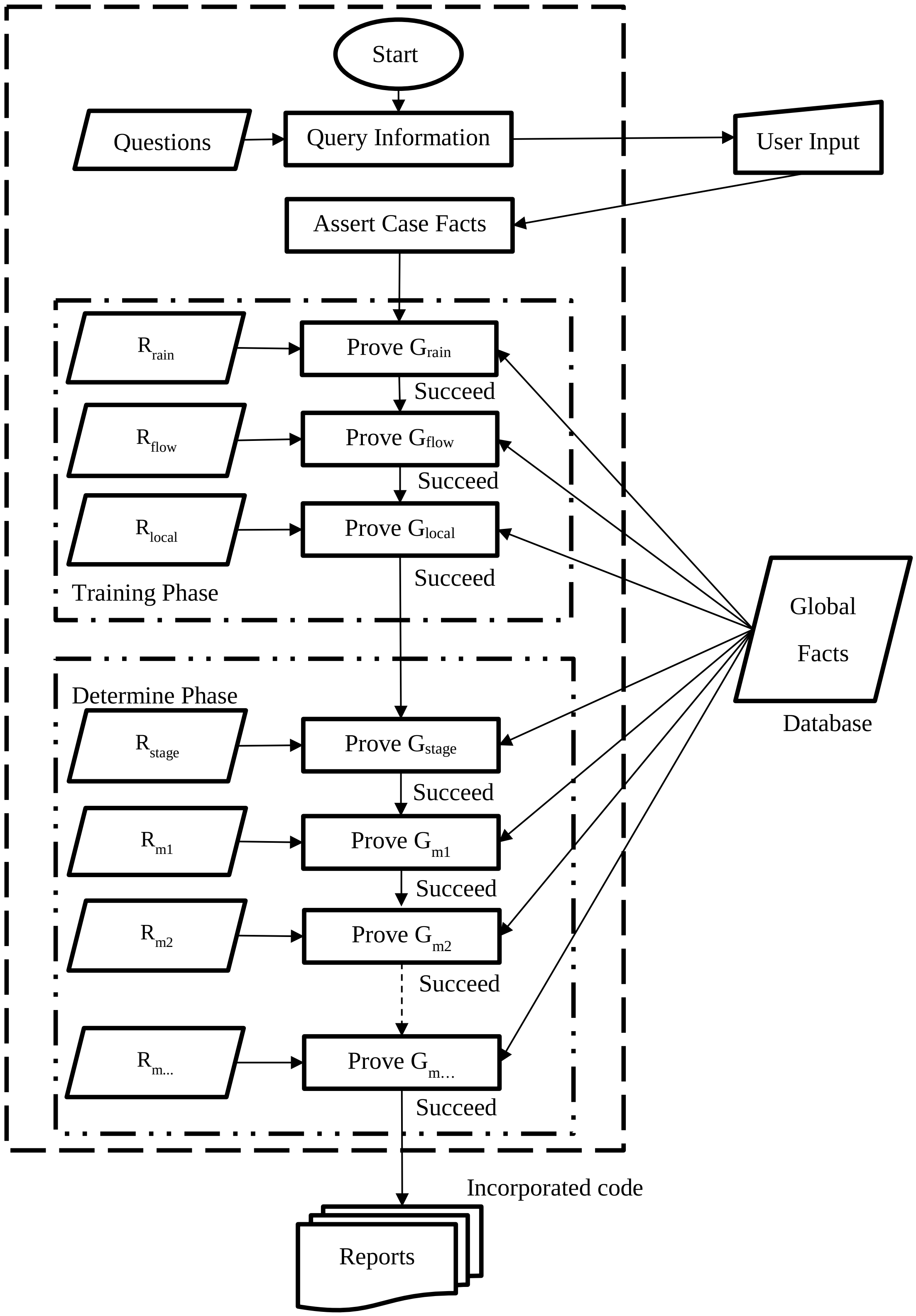

Figure 3.3. Architecture of the conventional procedural code of LFFWS

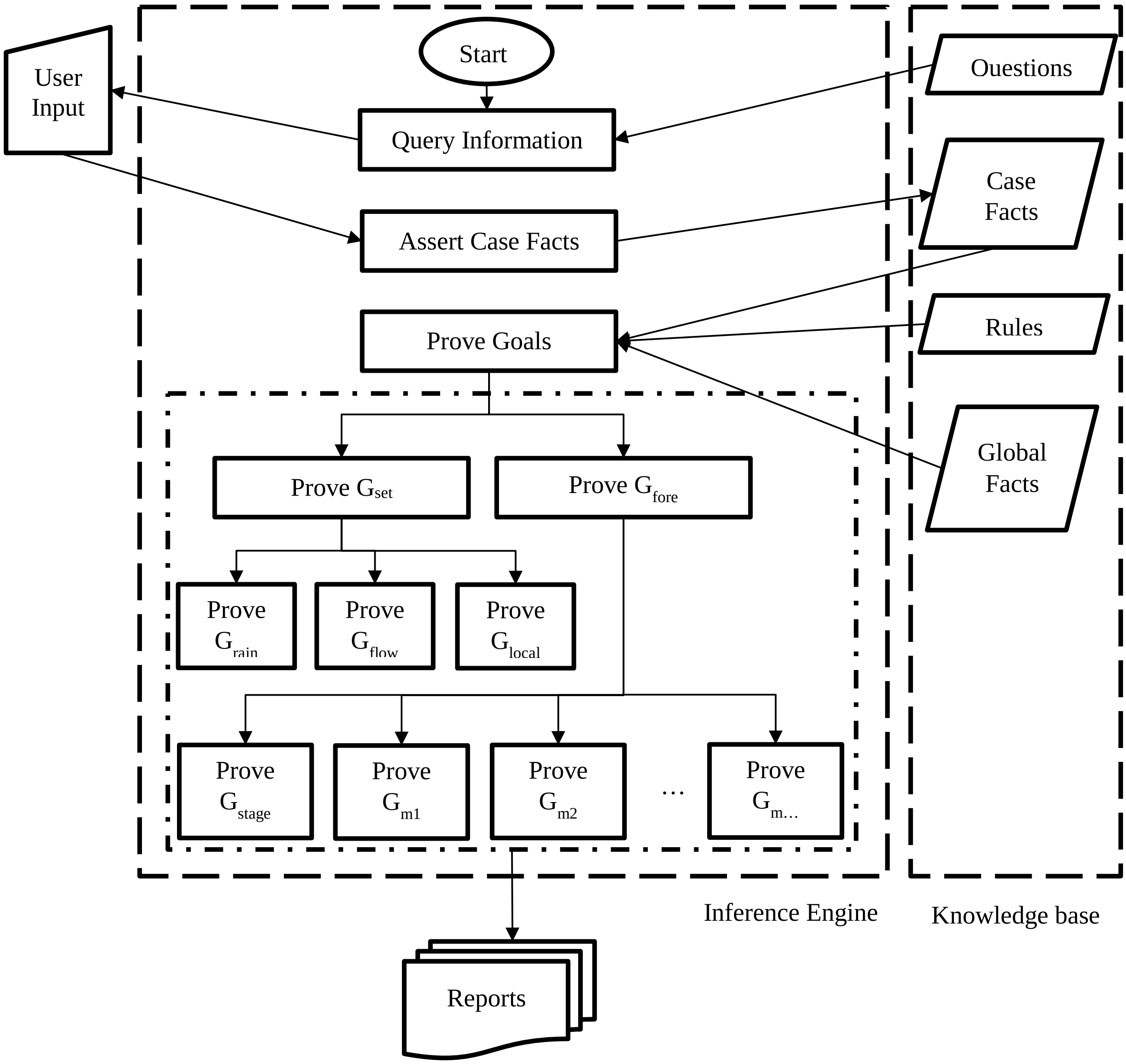

Figure 3.4. Architecture of the forward chaining expert system of LFFWS

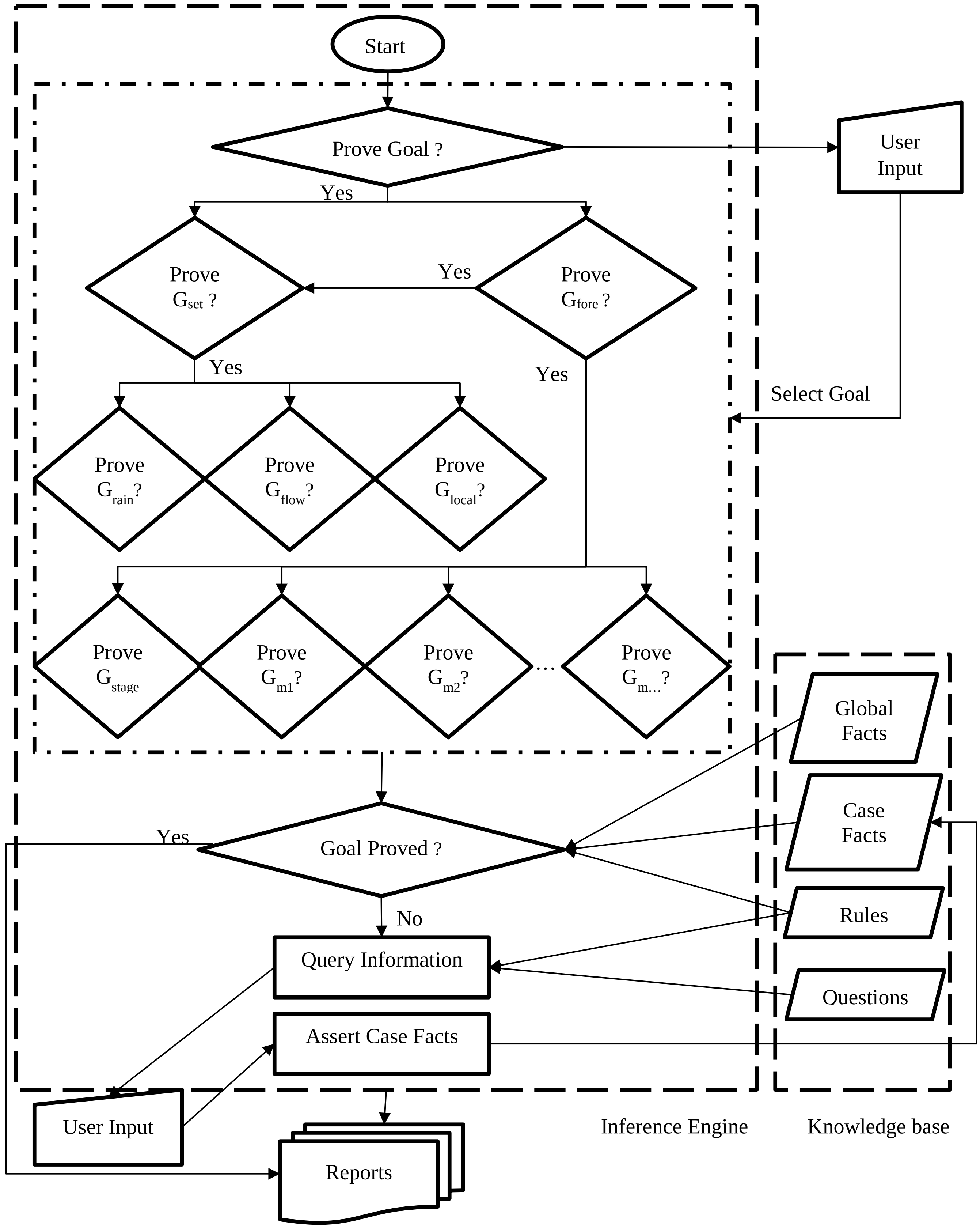

Figure 3.5. Architecture of backward chaining expert system of LFFWS

3.4. Language

The backward chaining expert system for local flood forecasting and warning is developed with a combination of Python and its Knowledge Engine PyKE because 1) unlike a compiled language, Python, as an interpreted language, allows quick “ad-hoc” development once the code is published and deployed; and 2) PyKE provides a way to directly "program in the large". The first two advantages speed up programming an expert system with a vast knowledge base such as LFFWS by reusing code and reducing loop. 3) In addition to forward-chaining logic, PyKE includes backward-chaining logic, which enables interactive data acquisition. This capability is the key to the comparison of the expert systems with forward and backward chaining inference engines, respectively. 4) Python is open source and free. Any software based on Python is friendly to almost all platforms including Windows, Mac, Linux, iOS, and Android. This flexibility helps create a large market for LFFWS (Python, 2015; PyKE, 2015). 5) Extensive libraries are available. For instance, the GUI (Graphic User Interface) of our systems can be built on its library Kivy to allow quick and easy interaction design and rapid prototyping (Kivy, 2015). Better performance for user practice can take advantage of this benefit.

3.5. Use of the System

Figure 3.6 depicts the banner screen displayed upon entry to the system. Inputting “Y” loads the system reset screens, including resetting triggers of rainfall, water level, and local conditions shown in Figure 3.7. Once the user selects any one, two, or all three types of triggers to reset the characteristics and parameters, a corresponding parameter-resetting screen appears. For example, if the user selects stream water level triggers to reset, the system then conducts an interview about the stream gauge and the threshold of flood warning stages corresponding to the stream gauge. Figure 3.8 illustrates the conversation between the system and user about resetting water level triggers. Once the user confirms all the triggers that s/he resets (shown in Figure 3.9), or decides to use existing default values by selecting no triggers to reset, the system moves to the second phase to collect necessary and available data and write a CSV report on the forecasting and warnings shown in Table 3.1.

```
Flood warning Wizard provides warnings and recommendations for local
community in the event of inundations.

Through the following interview, you will be asked a series of questions
regarding your river system and the flood event, from which you'll be
provided recommendations for flood rescue.

Do you wish to proceed? (y/n) y
```

Figure 3.6. Banner screen

```
----------------------------------------------------------------------
Reset water level triggers based on local conditions can provide more
precise warnings.

Do you wish to reset water level triggers? (y/n) y
```

```
-----------------------------------------------------------------------------
What is the water level of All Clear stage of SG1 in meters? [0-999999] 0.5
-----------------------------------------------------------------------------
What is the water level of Flood Watch stage of SG1 in meters? [0-999999] 1
-----------------------------------------------------------------------------
What is the water level of Flood Warning stage of SG1 in meters? [0-999999] 2
-----------------------------------------------------------------------------
What is the water level of Severe Flood Warning stage of SG1 in meters?
[0-999999] 3.5
```

Figure 3.7. System resetting screen

Figure 3.8. Water level triggers resetting screen

```
-----------------------------------------------------------------------------
The water level of All Clear stage of SG1 is 0.5 m.
The water level of Flood Watch stage of SG1 is 1.0 m.
The water level of Flood Warning stage of SG1 is 2.0 m.
The water level of Severe Flood Warning stage of SG1 is 3.5 m.

Save new water level triggers? (y/n) y
```

Figure 3.9. Water level triggers confirming screen

Table 3.1. Report on flood forecasting and warning presented to users as a CSV file

CaseID	StreamgaugeID	Depth(m)	Warning_Stage	Note
Case 1	SG1	4	Severe Flood Warning	This is the warning issued when serious flooding is expected and there is an imminent danger to life and property. If your warning is upgraded to this, you should be prepared for your gas, electricity, water and telephone supplies being lost. You're advised to keep calm and reassure others and cooperate with the emergency services.

3.6. Verification and Validation

Verification and validation (V&V) processes are critical components to guarantee the quality of developed expert systems. V&V processes include the analysis, evaluation, review, inspection, assessment, and testing of products. The technical aim of expert systems' V&V is determining whether the expert systems conform to the requirements and satisfy customers (IEEE, 2012; O'Keefe & O'Leary, 1993).

To assure an expert system is built correctly, developers typically verify their software by using a set of test cases either collected from real life situations or designed by domain experts to represent the possible problems in implementation (Adrion, et al., 1982; O'Leary, et al., 1990). With the assistance of debugging tools built in Python, we periodically verified our system throughout the development stage by conducting a complete set of pre-defined tests. Based on the results of the tests of V&V, we redesigned and reprogrammed the necessary heuristic knowledge and inferential logic.

To assure that we built the correct expert system, a paradigm for prototype validation combines face validation (the process by which the experts assess the prototype "at face value") with component testing and system validation through cases or Turing tests (O'Leary, et al., 1990). According to this method, experts from a water resources management area reviewed the system's operation, output, and documentation. In addition, the experts tested our system using selected cases from their experience. The accuracy of the system is evaluated by comparing the forecasting provided by the system with the documented test cases.

3.7. Results and Discussion

In the following results and discussion section, the advantages of development and maintenance of expert systems over procedural codes are illuminated and analyzed; then, the practicalities of forward and backward chaining inferential logic are studied and explained.

3.7.1. Expert System vs. Procedural Code

An expert system for local flood forecasting and warning has three primary advantages over procedural code: 1) an updatable knowledge base. The knowledge base of an expert system is explicit and separated from the inference engine. With no impact on other system components, errors and obsolete data can simply be corrected and replaced, while new knowledge corresponding to existing goals and/or new goals can be easily added. 2) Flexible workflow: End users' goals can be proved in parallel by either a forward or backward chaining inference engine of an expert system. Without changing the original scripts of an expert system, all goals can be proved in various sequences. With minor modifications to the scripts, existing goals can be modified or deleted, and fresh goals related to different local flood forecasting and warning issues can enter the workflow. 3) Explanatory capability. Developers and end users can track the knowledge primitives' applied validly to prove a goal and understand the reason routine. These advantages enable rapid development, simple maintenance, and quick diagnoses.

3.7.1.1. Benefits of an Extensible Knowledge Base

The system architecture (shown in Figure 3.3) of a procedural code predetermines the difficulty of upgrading its database. The rubrics of applying particular knowledge to prove goals are merged into implicit operation structure. As a result, every modification of the database, such as inserting additional knowledge entities about a new stream gauge, requires adjusting the implicit operation code. In contrast, an enlarging knowledge base of an expert system has no requirement to change the scripts of inference engines. To demonstrate the details, we assume the following scenario: a new stream gauge, called Gauge 1, is installed or considered. Adding knowledge primitives such as gauge ID and warning stage thresholds about this gauge is then required. To solve this problem, sample codes of expert systems in PyKE syntax are given below.

```
IDENTIFIER($ARGUMENT1, $ARGUMENT2, …)

Where,
"IDENTIFIER" represents a certain category of facts;
"$ARGUMENT1" and "$ARGUMENT2" represent different facts corresponding to the identifier, respectively.
```

Figure 3.10. General PyKE syntax of adding facts

```
warning_stage_triggers($gaugeID, $clear, $watch, $warning, $severe_warning)

where,
 "$gaugeID" represents the ID of the stream gauge;
"$clear" represents the threshold of the warning stage "All Clear";
"$watch" represents the threshold of the warning stage "Flood Watch";
"$warning" represents the threshold of the warning stage "Flood Warning";
"$severe_warning" represents the threshold of the warning stage "Flood Warning".
```

Figure 3.11. General format of adding warning stage thresholds

```
warning_stage_triggers ("Gauge 1","1.5","2","2.5","3")
```

Figure 3.12. Specific facts of Gauge 1 in the warning stage category

In the manner of the general syntax for adding facts shown in Figure 3.10, one category of facts, called “warning_stage_triggers” with five essential information entities can be coded in the following way, as shown in Figure 3.11.

The last step is to replace arguments shown above with specific thresholds of stream flows in meters of Gauge 1 in the same sequence as follows (see Figure 3.12):

In this sample code, the text in the first pair of quotation marks is the first argument, “$gaugeID”, which records the name or ID of the gauge; the number “1.5” in the second pair of quotation marks is the second argument. Then, “$clear” records the threshold of flood warning stage “All Clear” in meters, followed by “$watch”, “$warning”, and “$severe_warning”. Unlike procedural code, the inference engine of an expert system can automatically search all facts under the same category (referred to as identifiers) in every loop.

3.7.1.2. Benefits of a flexible workflow

In procedural code, the workflow is packaged in a fixed order and combined with questions and rules that correspondingly prove the goals. Therefore, the original scripts of the operation structure require amendments after any change in the workflow. For instance, inserting one novel goal into the workflow requires one to edit the original scripts of the operation structure, and all existing goals, rules, and questions are potentially impacted. If the new goal fails for some reason, all goals after it will stop proving. In contrast, all the goals of an expert system can be proved in parallel. After writing a short line of code to simply insert a new goal into the workflow, the inference engines automatically modify the compiled code. Even if the new goal fails, any other existing goals remain functional. To demonstrate the details, we assume that users want to know the warning stage; then, G_{stage} is required to be added into the workflows. To solve the problem, sample codes of the backward chaining expert system in PyKE syntax are given in Figure 3.13 and Figure 3.14. In the same manner of the general syntax for adding a goal shown in Figure 3.13. G_{stage} is added by replacing those capitalized parameters with the specific information entities corresponding to G_{stage}, shown in Figure 3.14.

Based on our study, we identify that there are essential information entities corresponding to the R_{stage}, such as gauge ID ($gaugeID), the actual water level in meters ($depth), and the threshold of flood warning stage ($trigger), so three arguments are included in the inner parentheses. In the same fashion, to insert an original goal into the workflow, we simply need two steps: 1) copy and paste one old goal; 2) "plug and chug" the rules and facts to prove the original goal. Within the architectures of expert systems demonstrated in Figure 3.4 and Figure 3.5, goals execute in various parallel sequences according to the availability of information at hand. The newly joined knowledge and goals will not affect the existing goals. In addition, backward chaining inferential logic also enables the workflow to adapt to the demands of users. However, the procedure code can only work in a fixed workflow, e.g., $G_{rain} \rightarrow G_{flow} \rightarrow G_{local} \rightarrow G_{stage} \rightarrow G_{m1} \rightarrow G_{m2} \rightarrow \ldots \rightarrow G_{m\ldots}$. If G_{rain} fails, then all goals after G_{rain} do not process. One common cause of the failure of G_{rain} is the lack of rainfall data, which often occurs for the following reasons: 1) rain gauge malfunction because of poor maintenance or other technical problems; 2) no local precipitation, e.g., it rains heavily upstream but outside the local boundary, or at least beyond the rain gauge; and 3) no present precipitation, such as in the case of a snowmelt flood. Therefore, the procedure code cannot cope with the scenarios stated above. In contrast to the incapability of procedural code in such situations, without making any changes to the original scripts, expert systems can skip G_{rain} automatically and prove the rest of the goals that are unrelated to G_{rain}. Although G_{rain} fails, the known rising rate, depth, and/or velocity of stream flow can still fulfill the other goals such as the flood warning stage and other warning messages related to stream flows.

Simply speaking, our expert systems can prove the goals in any sequence of workflows.

Therefore, we can develop our system by each goal and later pool the tested goals together. In this study, we only address partial issues listed in the 2011 WMO manual. However, as research continues, new knowledge, including facts and rules corresponding to other triggers and novel goals, will definitely be needed in LFFWS. When more issues of local flood forecasting and warning have been considered as fresh goals, one or two of these goals may fail due to the absence of information at hand or may be skipped because of lack of user interest. The LFFWS should enable the users to select some goals to be skipped or prove other goals first. At the same time, the system should automatically skip failed goals and move ahead to other goals. From this perspective, expert systems have striking benefits over procedural codes. The benefits from the extensible knowledge base and the flexible workflow of an expert system will be more and more attractive as more issues are considered.

```
with engine.prove_goal('rulebase.RULE_IDENTIFIER($ARGUMENT1, …)') as gen:
    for vars, plan in gen:
        …
```

where,

"RULE_IDENTIFIER" represents a certain rule;

"$ARGUMENT1" represents a certain knowledge primitive corresponding to the rule.

Figure 3.13. General PyKE syntax of adding a goal

```
with engine.prove_goal('rulebase.warning_stage($gaugeID,$depth,$trigger)') as gen:
    for vars, plan in gen:
        …
```

Figure 3.14. Syntax of adding G_{stage}

3.7.1.3. Benefits of explanatory capability

On the one hand, since facts are isolated from rules in conventional procedural codes, conventional procedural systems habitually lack the capability to explain why a fact is deduced or inferred in a particular way. In other words, procedural codes cannot tell users which facts and rules lead to creating the reasonable conclusions.

On the other hand, facts and rules are stored together in the knowledge base of expert systems. Developers can detect all reasoning routines for logic or syntax errors by tracing the list of valid facts and rules applied to solve a certain problem. As a result, problematic scripts can be locked down quickly. This explanatory capability is especially helpful when the developers are coding complex courses, such as the procedure of local flood forecasting and warning. The explanatory capability simplifies and accelerates the development of the computer systems. In addition, the explanatory capability can train those local flood managers with routines for reasoning in particular scenarios. For example, our expert systems can create a new fact in the following way, shown in Figure 3.15:

User input: Water depth is 3.2 meters.

Knowledge primitive 1: The threshold of Severe Flood Warning is 3 meters.

Knowledge primitive 2: Severe Flood Warning: "This is the warning issued when serious flooding is expected…"

➔ Infer new knowledge primitive: Issue Severe Flood Warning: "This is the warning issued when serious…"

Figure 3.15. Example of fact assertion

While developers or users read the analysis and conclusion, they can understand the cause-effective routines from the list of knowledge primitives validly applied to the goal. For example, assume that knowledge primitive 1 is accidentally coded incorrectly as "The threshold of Severe Flood Warning is 3000 meters." Then, the developers will not obtain the expected Severe Flood Warning. Instead of searching for all facts, the developers search the knowledge primitives on the reasoning list only.

Obviously, diagnosing and correction processes are expedited in this way. Therefore, the larger the knowledge base and the more complicated the reasoning routines are, the more appealing the explanatory capability of an expert system is. Thus, developing an expert system is more proficient than procedural code for LFFWS.

3.7.2. Backward Chaining vs. Forward Chaining

To illustrate how the backward chaining mechanism is applied to and enhances our expert system, simplified forward and backward chaining logic is shown in Figure 3.16 and Figure 3.17, respectively. Different dash types and arrow types indicate diverse information flows. Generally, thresholds of rainfall, flow, and local conditions are only required during system initialization to prove G_{rain}, G_{flow}, and G_{local}, respectively. Once the system is well trained, these parameters and variables are saved for repeated use until the system resets. Other current or proposed data on rainfall, stream flows, and local conditions are only needed during the forecasting phase. Take the proof of G_{flow} as an example. In this case, only the thresholds of stream flow, such as depth, velocity, and rising rate, are essential, so the backward chaining inference engine only collects the facts on these triggers. On the other hand, the forward chaining inference engine also blindly collects other information such as thresholds of rainfall, thresholds of local conditions, and current or proposed data on rainfall, stream flows, and local conditions. With the forward chaining expert system, users must provide the complete query for all incorporated goals at the beginning in order for the expert system to complete processing, even when some of this information is not available or is not of interest to the users. To allow for response questions only as inferential logic and thus simplify the user experience, the knowledge base in this study is entirely written for backward chaining inferencing.

The advantages of backward chaining do not present significant benefits in the determinant phase.

The combinations of current data of rainfall, stream flows, and local conditions are required multiple times to determine distinct forecasting and warning messages. In addition, common users do not usually have any clear desire for specific goals. To enter all known information and expect every potential forecasting and warning recommendation is less challenging than selecting “useful” goals. Therefore, setting up an ultimate goal (G_{fore}) to discover complete information on the current or proposed situation and prove the entire goal set of the determinant phase, which is to some extent equivalent to forward chaining logic, is desired to shorten the practice of common users. In the meantime, the convenience of goal selection is there for skilled users and system developers.

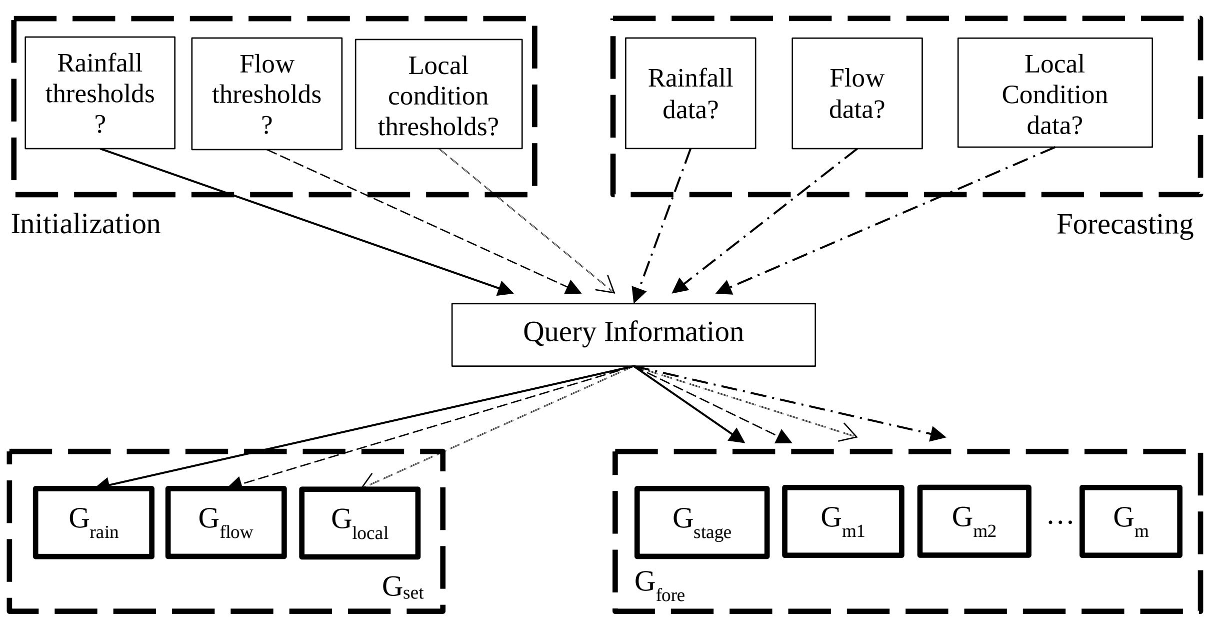

Figure 3.16. Simplified forward chaining logic to prove goals

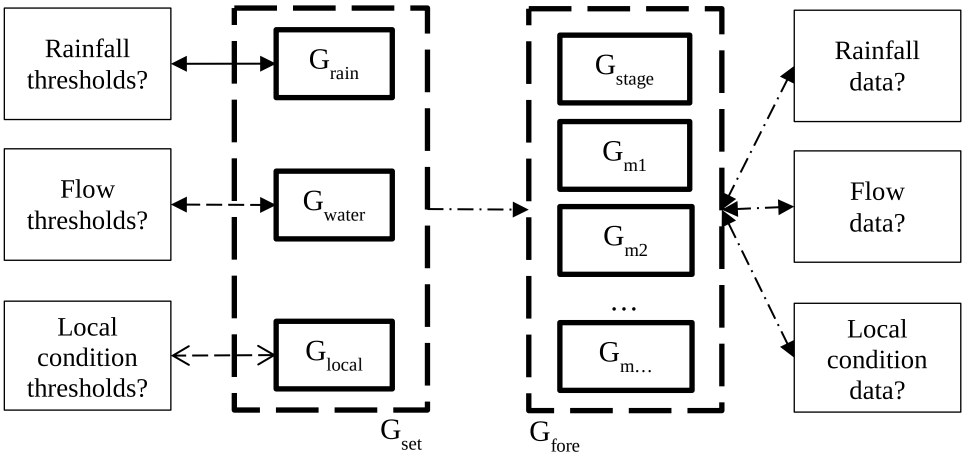

Figure 3.17. Simplified backward chaining logic to prove goals

3.8. Conclusions

To break through the bottleneck of knowledge acquisition in local flood forecasting and warning, we identified the knowledge necessary to LFFWS and assimilated these dynamic and static knowledge primitives into the knowledge base. The case study illustrates that the collected knowledge works successfully to initialize the system and provide flood forecasting and warning messages on current or proposed data of rainfall, stream flows, and local conditions.

In addition, this study shows that, to provide local flood forecasting and warning, developing an expert system is more effective than procedural code. The advantages of rapid development and easy maintenance stem from the system architecture of expert systems. The explicit knowledge base and packaged expert system shell ensure that 1) new facts, rules, questions, and goals can be easily added to the extensible knowledge base by domain experts, such as environmental engineers, or even skilled users to adapt to more scenarios; 2) all goals can be skipped or proved in various sequences automatically (forward chaining) or according to users' demands (backward chaining); 3) partially developed expert systems can be functional; 4) logic or syntax errors and outdated data can be rapidly identified and corrected; and 5) the users can understand and learn the reasoning simultaneously when they obtain the reports.

Furthermore, the backward chaining method is shown to work more effectively than forward chaining to satisfy local flood managers' evolving demands (referred to as goals) and growing new information on LFFWS. Backward chaining enables the inference engine of expert systems to work with incomplete information at the beginning and to keep running as more and more data become available. With a backward chaining inference engine, our expert system can quickly figure out and optimally collect the necessary data from previous analysis results or user interviews based on the evolving goals and efficiently develop reports and recommendations with the growing information at hand. In the meantime, without sacrificing convenience for skilled users, to ease the difficulty of goal selection and shorten the practice for common users, an ultimate goal (G_{fore}) is set up to acquire a complete picture of the current or proposed situation and issue all likely forecasting and warning messages.

Although the contemporary LFFWS currently work with limited goals for local flood forecasting and warning, additional goals and their corresponding knowledge related to other key issues can be easily updated in LFFWS. This prototype of a backward chaining expert system of local flood forecasting and warning, giving reasonably accurate predictions and recommendations, can decrease the engagement of computational resources by minimizing the boundary of the interested area and decreasing the usage of numerical models. This makes flood forecasting more cost-effective and therefore feasible for small communities, especially for those with tight flood management budgets. This research represents an advance in the applicability of expert systems to solve flood prediction and management problems. Further, the framework of our expert systems can easily be duplicated to create other expert systems. Domain experts in other areas can make use of our framework to record their valuable expertise and undocumented "rules of thumb" in computer-readable language and create more expert systems to perform repeated work efficiently.

3.9. References

Adrion, W. R., Branstad, M. A., & Cherniavsky, J. C. (1982). Validation, Verification, and Testing of Computer Software. *ACM Computing Surveys, 14*(2), 159-192.

Anderson, M. L., Chen, Z.-Q., Kavvas, M. L., & Feldmand, A. (2002). Coupling HEC-HMS with Atmospheric Models for Prediction of Watershed Runoff. *Journal of Hydrologic Engineering, 7*(4), 312-318.

Badrzadeh, H., Sarukkalige, R., & Jayawardena, A. W. (2015). Hourly runoff forecasting for flood risk management: Application of various computational intelligence models. *Journal of Hydrology, 529*, 1633-1643.

Chau, K. W., & Phil, M. (2004). Knowledge-Based System on water resources management in Coastal Waters. *Water and Environmental Journal, 18*(1), 25-28.

Chen, M.-K. S., Chau, C.-f. C., & Kabat, W. C. (1985). Decision Support Systems: A Rule-Based Approach. *ACM '85 Proceedings of the 1985 ACM annual conference on The range of computing : mid-80's perspective: mid-80's perspective* (pp. 511-515). New York: Association for Computing Machinery.

Cloke, H., & Pappenberger, F. (2009). Ensemble flood forecasting: A review. *Journal of Hydrology, 375*, 613-626.

Comas, J., Llorens, E., Marit, E., Puig, M., Riera, J., Sabater, F., & Poch, M. (2003). Knowledge acquisition in the STREAMES project: the key process in the Environmental Decision Support System development. *AI Communications, 16*, 253-265.

DHS. (2013). *National Response Framework.*

Emerton, R. E., Stephens, E. M., Pappenberger, F., Pagano, T. C., Weerts, A. H., Wood, A. W., Cloke, H. L. (2016). Continental and global scale flood forecasting systems. *Wiley Interdisciplinary Reviews: Water, 3*(3), 391-481.

Fang, S., Xu, L., Zhu, Y., Liu, Y., Liu, Z., Pei, H., Zhang, H. (2015). An integrated information system for snowmelt flood early-warning based on internet of things. *Information Systems Frontiers, 17*(2), 321-335. doi:10.1007/s10796-013-9466-1

Feigenbaum, E. (1977). *The art of artificial intelligence: themes and case studies of knowledge engineering.* School of Humanities and Sciences, Stanford University, Computer Science Department, Stanford.

Garcia-Pintado, J., Mason, D. C., Dance, S. L., Cloke, H. L., Neal, J. C., Freer, J., & Bates, P. D. (2015). Satellite-supported flood forecasting in river networks: A real case study. *Journal of Hydrology, 523*, 706-724.

Ghalkhani, H., Golian, S., Saghafian, B., Farokhnia, A., & Shamseldin, A. (2012, August 16).

Application of surrogate artificial intelligent models for real-time flood routing. *Water and Environment Journal, 27*(4), 535-548.

Hayer-Roth, F., Waterman, D., & Lenat, D. (1983). *Building Expert Systems.* Addison-Wesley.

IEEE. (2012, 10 12). *IEEE Standard for System and Software Verification and Validation.* Retrieved from IEEE Standards Association: https://standards.ieee.org/findstds/standard/1012-2012.html

Jackson, P. (1998). Introduction To Expert Systems. In P. Jackson, *Introduction To Expert Systems* (3ed ed., p. 2). Addison Wesley.

Jasper, K., Gurtz, J., & Lang, H. (2002). Advanced flood forecasting in Alpine watersheds by coupling meteorological observations and forecasts with a distributed hydrological model. *Journal of Hydrology, 267*, 40-52.

Kaewboonma, N., Tuamsuk, K., & Kanarkard, W. (2013). Knowledge Acquisition for the Design of Flood Management Information System: Chi River Basin, Thailand. *Social and Behavioral Sciences. 73*, pp. 109-114. Budapest: Elsevier Ltd.

Kivy. (2015). Retrieved from http://kivy.org/#home

Leon, C., Martin, S., Elena, J. M., & Luque, J. (2000). Explore: Hybrid Expert System For Water Networks Management. *Journal of Water Resources Planning and Management, 126*(2), 65-74.

Mabrouk, M. E., Ezziyyani, M., Sadouq, Z. A., & Essaaidi, M. (2015, August). New Expert System for Short, Medium and Long term Flood Forecasting and Warning. *Journal of Theoretical and Applied Information Technology*, 286-302.

Mahabir, C., Hicks, F., & Fayek, A. R. (2007). Transferability of a neuro-fuzzy river ice jam flood forecasting model. *Cold Regions Science and Technology, 48*(3), 188-201.

Mounce, S., Boxall, J., & Machell, J. (2010). Development and Verification of an Online Artificial Intelligence System for Detection of Bursts and Other Abnormal Flows. *Journal of Water Resources Planning and Management, 136*(3), 309-318.

Najafi, M. R., & Moradkhani, H. (2015). Ensemble Combination of Seasonal Streamflow Forecasts. *Journal of Hydrologic Engineering*, 2438-2453.

O'Keefe, R. M., & O'Leary, D. E. (1993). Expert system verification and validation: a survey and tutorial. *Artificial Intelligence Review, 7*, 3-42.

O'Leary, T. J., Goul, M., Moffitt, K. E., & Radwan, A. E. (1990). Validating expert systems. *IEEE Intelligent Systems, 5*(3), 51-58.

Ooshaksaraie, L., & Basri, N. E. (2011). An Expert System Applied in Construction Water Quality Monitoring. *American Journal of Environmental Sciences, 7*(11), 75-81.

Pinto, D., Castro, L., Cruzat, M., Barros, S., Gironas, J., Oberli, C., Cipriano, A. (2015). Decision

Support System for a Pilot Flash Flood Early Warning System in Central Chile. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering, 9*(3), 990-997.

PyKE. (2015). Retrieved from Welcome to Pyke: http://pyke.sourceforge.net/

Python. (2015). Retrieved from Python: https://www.python.org/

Robindro, K., & Sarma, S. K. (2013). JESS Based Expert System Architecture For Diagnosis Of Rice Plant Diseases: Design And Prototype Development. *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, (pp. 674-676).

Spyridakos, T., Pierakos, G., Metaxas, V., & Logotheti, S. (2005). Supporting the Management of Measurement Network with an Expert System: The NeMO System. *Operational Research An International Journal, 5*(2), 273-288.

Todini, E. (1999). An Operational Decision Support System for Flood Risk Mapping, Forecasting, and Management. *Urban Water*, 131-143.

WMO. (2011). *Manual on Flood Forecasting and Warning.* WMO. Retrieved from http://www.wmo.int/pages/prog/hwrp/publications/flood_forecasting_warning/WMO%201072_en.pdf

Wong, B. K., & Monaco, J. A. (1995, September). Expert system applications in business: a review and analysis of the literature (1977-1993). *Information and Management, 29*(3), 141-152.

4. AN EXPERT SYSTEM FOR LOCAL FLOOD RESPONSE COORDIANTION AND TRAINING

4.1. Introduction

In the aftermath of accurate forecasting and timely warning of floods, the effective response should be recommended and implemented under the supervision of community authorities, city managers, state governors, or federal officers. However, management of flood response is far more complicated than many other management problems. The response to disasters like flooding is an emergency network with non-linear dynamics, uncertainty, open boundary, and varying topology (Liu, et al., 2011). Weaknesses in incident management are often due to the following issues: 1) a shortage of clear chained of command and supervision; 2) poor communication caused by inefficient use of communications systems and non-consistent terminology; 3) inadequate and unreliable data of incidents; 4) an absence of an orderly and systematic planning process; 5) a dearth of command and control coordination structure; 6) a lack of flexibility and adaptability of response procedures and plans (DOL, 2016; Select Bipartisan Committee of the U.S. House of Representatives, 2006). In addition, inferior decisions made by inexperienced flood response managers under high pressure and unproductive actions taken by untrained and stressful personnel could lead to unsuccessful flood response (Jennex, 2007). A typical example is an ineffective preparation for and response to Hurricane Katrina, the costliest natural disaster and one of the five deadliest hurricanes in the history of the United States (National Hurricane Center, 2011). Although the National Weather Service and National Hurricane Center forecasts were accurate and timely, a failure of leadership at all levels of government resulted in preventable deaths, great suffering, and tremendous property loss (Select Bipartisan Committee of the U.S. House of Representatives, 2006).

Local government officials often have difficulties in dealing with Federal guidelines (e.g. Interpreting which guidelines supply to their roles). In order to ensure that information moves within agencies, across departments, and between jurisdictions of government seamlessly, securely and efficiently and response plans are adaptable to meet whatever flood scenarios, a sound coordination system with unified responsibilities, smooth communications, and scalable response plans is required. Additionally, to shorten the distance between theory and practice, adequate training on structural roles, responsibilities, and actions to deliver the core capabilities of flood response, is vital for all potential flood response entities, such as individuals and households, private sector, nongovernmental organizations (NGOs), communities, local government, state government, and federal government (Flin, et al., 2008). Development of a computer-based tool could aid in their flood response.

4.2. Background

With the purpose of strengthening the security and resilience of the United States through systematic preparation for the threats such as flooding, Presidential Policy Directive (PPD) 8 mandated the National Preparedness System (NPS). A number of projects were launched to develop and perfect the NPS. Among them are the National Incident Management System (NIMS) and the National Response Framework (NRF). The U.S. Department of Homeland Security (DHS) provided the NIMS and the NRF in order to build a framework of response to all disasters and emergencies regardless of size and complexity. NIMS provides the overall template, while NRF provides the structure and mechanisms for the management of incidents.

The NIMS systematically blends accepted best practices into a standard national framework for emergency management. It contains five major components: 1) preparedness, 2) communications and information management, 3) resources management, 4) command and management, and 5) ongoing management and maintenance. The command and management component is designed to offer a standardized incident management structure assisting incident coordination. This structure is based on three critical organizational constructs: the Incident Command System (ICS), Multiagency Coordination Systems (MACS), and Public Information (PI). Among those, the ICS is the most widely used. The ICS hierarchy assists activities in five major functional areas: Command, Operations, Planning, Logistics, and Finance/Administration. Intelligence and investigations is an optional sixth functional area that is activated as needed. (DHS, 2008; FEMA, 2016).

The NRF describes managerial doctrine for all types of disasters and explains common response disciplines and process developed at all levels. More specifically, in addition to a scalable, flexible, and adaptable coordination structure, NRF defines other fundamental elements. These include the key roles, responsibilities, and the steps needed to prepare for delivering the fourteen core capabilities: 1) planning, 2) public information and warning, 3) operational coordination, 4) critical transportation, 5) environmental response/health and safety, 6) fatality management services, 7) infrastructure systems, 8) mass care services, 9) mass search and rescue operations, 10) on-scene security and protection, 11) operational communications, 12) public and private services and resources, 13) public health and medical services, and 14) situational assessment (DHS, 2008; DHS, 2013).

To close the gaps in qualified responders, the Federal Emergency Management Agency's (FEMA) National Integration Center (NIC) released the NIMS Training Program in February 2008, originally. This vital program defines the process for developing training and personnel qualification requirements for emergency management (NIC, 2011).

With the aim of increasing community preparedness and resilience for floods, America's

PrepareAthon, a national community-based campaign, offers easy-to-implement resources to help individuals, organizations, and communities practice simple, effective actions. The resources include a short video: It Started Like Any Other Day (Williams, 2014), practical handouts (e.g. Prepare Your Organization for a Flood Playbook (FEMA, 2014), How to Prepare for a Flood, How to Prepare for a Flood Guide, Be Smart: Know Your Alerts and Warnings, Be Smart: Protect Your Critical Documents and Valuables, and Ready's Family Communication Plan for Parents and Kids (FEMA, 2016)), and abundant further reference linkages (e.g. Turn Around Don't Drown program at National Weather Service (NWS, 2015), National Flood Insurance Program at FEMA (FEMA, 2017), and Action Guide for Emergency Management at Institutions of Higher Education (DOE, 2009)) .

4.3. Research Objectives and Methods

Learning, understanding, digesting, and mastering the extensive official documents and all-encompassing heuristic rules on flood response requires remarkable time and energy. The learning package should be individualized to each role of the response personnel. Despite distinct roles and priorities of response knowledge, training all responders with the same reference materials steepens the learning curve and elongates the learning time. As a result, a well-directed training process adaptable for various roles is required. Moreover, the training course should provide plenty of linkages to a broader scope of knowledge such as further explanations to their own jobs and tasks of other participating entities. Commanders, section chiefs, branch heads, and group leaders also desire a centrally organized but fully distributed command and control system (Turoff, et al., 2004) with hierarchical summaries illustrating the crisscross linkages among all responders, their primary functions, and contact information, as well.

The objective of this research was to investigate the development of a computer-based program to assist flood response managers in constructing coordination structures regardless of the scale, scope, and complexity of flood, it was also intended to support qualifying flood responders, such as individuals, communities, nongovernmental organizations, private sector entities, and local governments.

On the basis of NIMS, NRF, and other training guidebooks, the computer program for shaping a flood response coordination system can be conventionally coded in a traditional procedural language, such as an assembly language or a high-level compiler language (C, Pascal, COBOL, FORTRAN, etc.) Alternatively, the system can be programmed as an expert system with inferential logic.

As one successful form of Artificial Intelligence (AI) technology, expert systems are computer-based systems which emulate the decision-making ability of human experts by exploiting knowledge represented primarily as “if-then” rules (Jackson, 1998). Typically, a knowledge base and an inference engine are the major components of an expert system. Edward Feigenbaum, the father of expert systems, asserted that a knowledge base is the power source of an expert system (Feigenbaum, 1977). Knowledge primitives are usually categorized into facts and rules. Facts are simple statements containing data values that represent, and show relationships among entities; Rules are declarative knowledge linking sets of premises and conclusions. Primarily, an inference engine applies logic rules to create new facts in either a forward chaining or backward chaining mode. Forward chaining, also referred to as data-driven chaining, works top-down to assert conclusions or new facts. Backward chaining, also referred to as goal-driven chaining, works bottom-up to determine what facts must be asserted. In general, IT specialists script an inference engine as a general-purpose shell to simplify and expedite the programming process. Then, domain experts in a certain research field and/or knowledge engineers deduce and compile the necessary facts and rules into a knowledge base (Leondes, 2002).

To build a computer-based assistant system for flood response coordination and training, developing an expert system is viewed more effective than procedural code. In the procedural coding process, IT specialists are always vital from start to finish. On the other hand, in the coding process of an expert system, IT specialists are not required once an inference engine is packaged. Domain experts can either work alone or cooperate with knowledge engineers to develop their particular expert systems in the specific fields (Wong & Monaco, 1995). In other words, compared with procedural code, expert systems are more rapidly and easily developed and maintained and have more flexibilities of running with evolving goals. In addition, since rules and facts are stored together in an explicit knowledge base, an expert system has an explanatory capability. This capability enables the users (flood responders) to understand and learn the reasoning simultaneously when they get the conclusions, and therefore, promotes training. All these special characteristics make an expert system more attractive than a procedural code for this flood response application (Leondes, 2002).

Currently, the use of computer science and artificial technologies such as expert systems to support the decision makers is a widespread approach to deal with emergency problems. Some successful applications in this area are discussed below.

Lee et al. (2012) developed an unstructured information management system (UIMS) for decision-makers to achieve a better understanding of the dependence and degree of correlation between different concepts about the emergency. UIMS consists of a concept relationship model (CRM) and a dynamic knowledge flow model (DKFM). The models can organize and represent emergency knowledge (W. Lee, et al., 2012).

Liu et al. (2011) developed a framework to provide a general methodology for integrating various decision bodies. The general method was to view management as a process and control problem and then to apply system engineering technology into the emergency response management in networked safe service systems (Liu et al., 2011).

Liu (2004) developed an agent-based resource discovery framework to search for the relevant resources over the Internet. This expert system addressed two pivotal issues: resource description language (RDL) and its resource matchmaking mechanism. A possibilistic Petri net-based resource description language was proposed to provide a specification to publish and request for resources in environmental emergencies. A matchmaking, allowing a relaxed match for close semantics to find an appropriate resource for environmental emergency management was developed, as well (Liu, 2004).

Hernandez and Serrano (2001) developed knowledge-based models within the framework of ARTEMIS (a European Commission research project) for emergency management. This expert system incorporated the knowledge pieces, both from the point of view of the knowledge model calibration and the training of the emergency personnel, required to manage emergencies in different kinds of problem scenarios (Hernandez & Serrano, 2001).

However, there is limited research on developing expert systems to specifically help the incident managers establish sound coordination structures with clear and unified responsibilities of all participants. Potential responders could easily be educated by the system to identify their roles and the key concerns.

4.4. Design of the System

In order to fill the research gap, a prototype expert system for flood response coordination and training (referred to as LFRS hereafter) was designed at the local level based on the basic premise that in most cases, floods start and end locally and flood response is managed at the local level (DHS, 2013).

The most widely used computer languages for programming expert system include Visual Basic (Spyridakos, et al., 2005), Java or JESS (Java Expert System Shell) (Robindro & Sarma, 2013), CLIPS

(Ooshaksaraie & Basri, 2011), MATLAB or NETLAB (Mounce, et al., 2010), Visual Rule Studio (Chau & Phil, 2004), ART*Enterprise (Leon, et al., 2000), and PyKE (Python Knowledge Engine) (PyKE, 2015). PyKE is 100% based on Python. Python is an interpreted language which allows quick “ad-hoc” development once the code is published and deployed. PyKE offers a way to directly "program in the large". This characteristic is helpful in developing an expert system with a large knowledge base as in this application. In addition, Python is open source. Any software based on Python can freely run on almost all platforms including Windows, Mac, Linux, iOS, and Android (Python, 2015; PyKE, 2015). Based on the overview of functionality, installation, integration characteristics, and compatibility, we decided to construct our system using PyKE.

Typically, there are three phases to design and develop an expert system: design of an initial knowledge base, development of a prototype, and test and improvement (Durkin, 1994). During the knowledge base initialization phase, key concerns, connections, and heuristics about the addressed problems are identified, conceptualized, and formalized. During the development phase, the acquired knowledge was organized and coded, in often cases, module by module, function by function, or subsystem by subsystem. System verification and validation (V&V) are usually conducted simultaneously with coding and after the construction of the prototype to ensure every component of the system represent the actual knowledge and was built correctly.

4.4.1. Knowledge Acquisition

The power of an expert system derives from its knowledge base (Feigenbaum, 1977). The performance of an expert system rests on the knowledge acquisition. To complete the vital task, extensive information for construction local flood response coordination structure was gathered from literature reviews and human-expert interviews.Then, the collected information was extracted and organized in such a way that it can be easily understood. After analysis and evaluation, verified knowledge entities were finally structured into a computer-readable form (Liou, 1990; Storey, et al., 2012).

4.4.2. System Content and Architecture

The prototype of the expert system for local flood response coordination and training consists of two standalone modules (shown in Figure 4.1): RRA (Roles, Responsibilities, and Actions) module and ROS (Response Operational Structure) module. The RRA module aims at generating the report of responsibilities and actions to deliver core capabilities corresponding to certain roles. All community entities from individuals to governments play distinct roles in developing the core capabilities of flood response. Focusing on local flood response, the LFRS narrows the spectrum down to local government and community level. The RRA module emphasizes five roles (individuals and families, communities, the private sector, NGOs, and local government) and identifies their activities (planning, assessing and exercising, offering and conducting resources and capabilities, and collecting learned lessons). The ROS module targets at creating the hierarchical layered and mutually supporting operational structures based on the ICS, the most common local response operational structure. The tangible output contains formatted reports as well as prompt screens. One report hierarchically depicts the crisscross connections among all responders, their primary functions, and contact information. Another report summarizes the responsibilities and actions of a certain role of flood responders such as commanders, command officers, section chiefs, branch heads, and group leaders. Every report was named with a unique caseID and formatted in a CSV file.

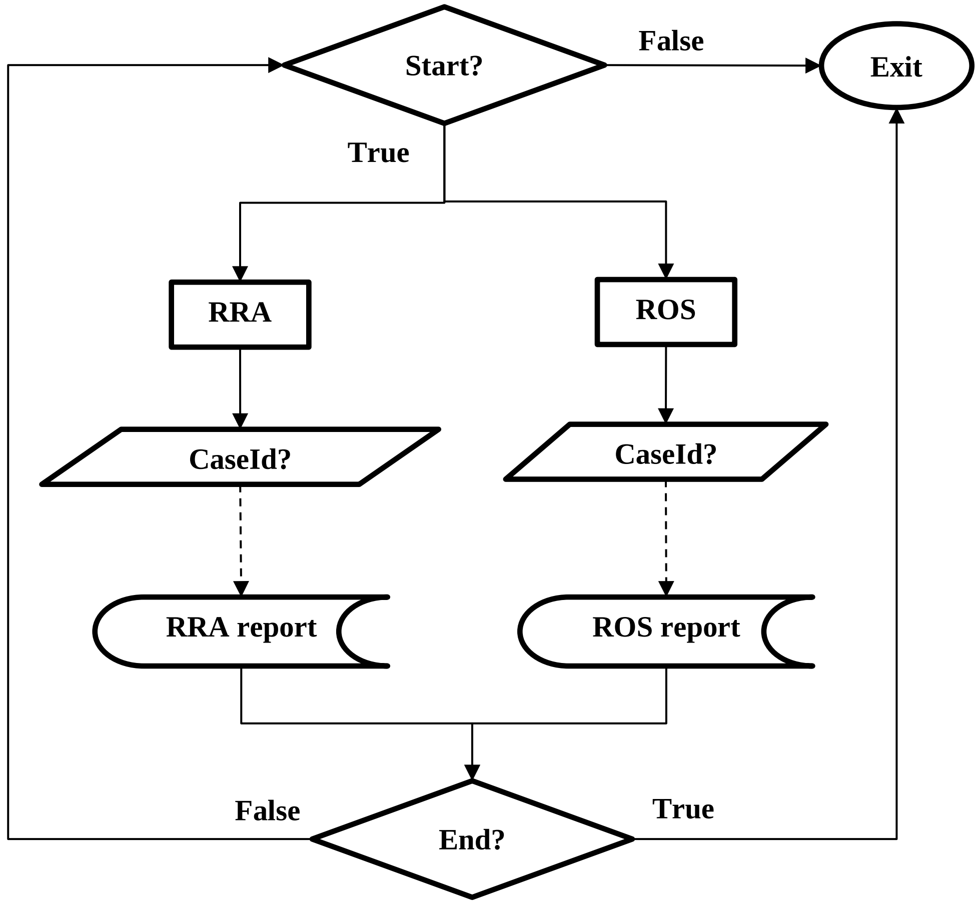

Figure 4.1. Schematic diagram of LFRS

4.1.1.1 RRA module

Currently, the RRA module offers five roles for selection: 1) individuals, families, and households, 2) communities, 3) nongovernment organizations, 4) private sectors, and 5) local government. After giving the simple definition of each selected role, the RRA module provides the guidance and instructions. For example, individuals, families, and households play an important role in emergency management operations by decreasing flood hazards in and around their homes and taking care of themselves and neighbors. Their efforts may include raising utilities above flood level, preparing emergency supply kits and plans, and volunteering with emergency organizations. Additional information is given by providing them with verified website linkages. NGOs, especially those officially assigned as support elements, play a significant role in delivering vital services associated with response core capabilities. For instance, the American Red Cross (ARC), National Voluntary Organizations Active in Disaster (VOAD), and Volunteers and Donations (VD) have their specific goals in emergency management. Private sector entities contribute to response efforts through partnerships with governments. A private sector entity could be one or multiple roles of the six categories: 1) Affected Organization/ Component of the Nations' Economy, 2) Affected Infrastructure, 3) Regulated and/or Responsible Party, 4) Response Resource, 5) Partner with Federal/State/Local Emergency Organizations, and 6) Components of the Nation's Economy. The responsibilities and actions of local government vary with specific local officials, such as chief elected or appointed officials, emergency managers, and department and agency heads (DHS, 2008; DHS, 2013). Role selection and the inferencing mechanism are discussed in the following section 4.2.2.

4.1.1.2 ROS module

ROS module adopts the major structure of the ICS. However, the ROS module offers additional information on the relationships among ICS, Multiagency Coordination System (MACS), and Public Information (PI) for the further reference of response entities. In order to education the users, the ROS module provides introductions to the fourteen proven management characteristics of the ICS: 1) command terminology, 2) modular organization, 3) management by objectives, 4) Incident Action Planning (IAP), 5) manageable span of control, 6) incident facilities and locations, 7) comprehensive resource management, 8) integrated communications, 9) establishment and transfer of command, 10) chain of command and unity of command, 11) unified command, 12) accountability, 13) dispatch/deployment, and 14) information and intelligence management (DHS, 2008; FEMA, 2016). The users can choose any one or ones to learn more. Also, users can skip those lessons to set up the operational structure directly.

Consistent with the ICS, the response operational structure created by the ROS module consists of six sections: command, operations, planning, logistics, finance/administration, and intelligence/investigations function. The sixth section is optional. In each section, there are several units, branches, or groups (shown in Figure 4.2). The construction of the flood response operational structure is a complex task. To simplify users' practice, the ROS module provides a number of breaking points throughout the process of structure developing. These points give users flexibilities to build or edit one functional unit first, save changes, and then come back after taking a nap. The ROS module will combine those pieces together to form the structures for various sections and the entire response coordination function.

4.4.3. Inference Engine

The LFRS is equipped with a backward chaining inference engine for more efficient processing. The information needed for the RRA module is different from the data necessary to the ROS module. For example, certain roles determine the responsibilities and actions in the RRA module but have little impact on the construction of the ROS; the contact information of a response chief is vital in ROS but does not affect his/her job tasks. Obviously, the information query and report production are driven by users' goals and their selection of modules. The backward chaining inference engine enables the LFRS to collect the critical data for each module respectively after the module selection.

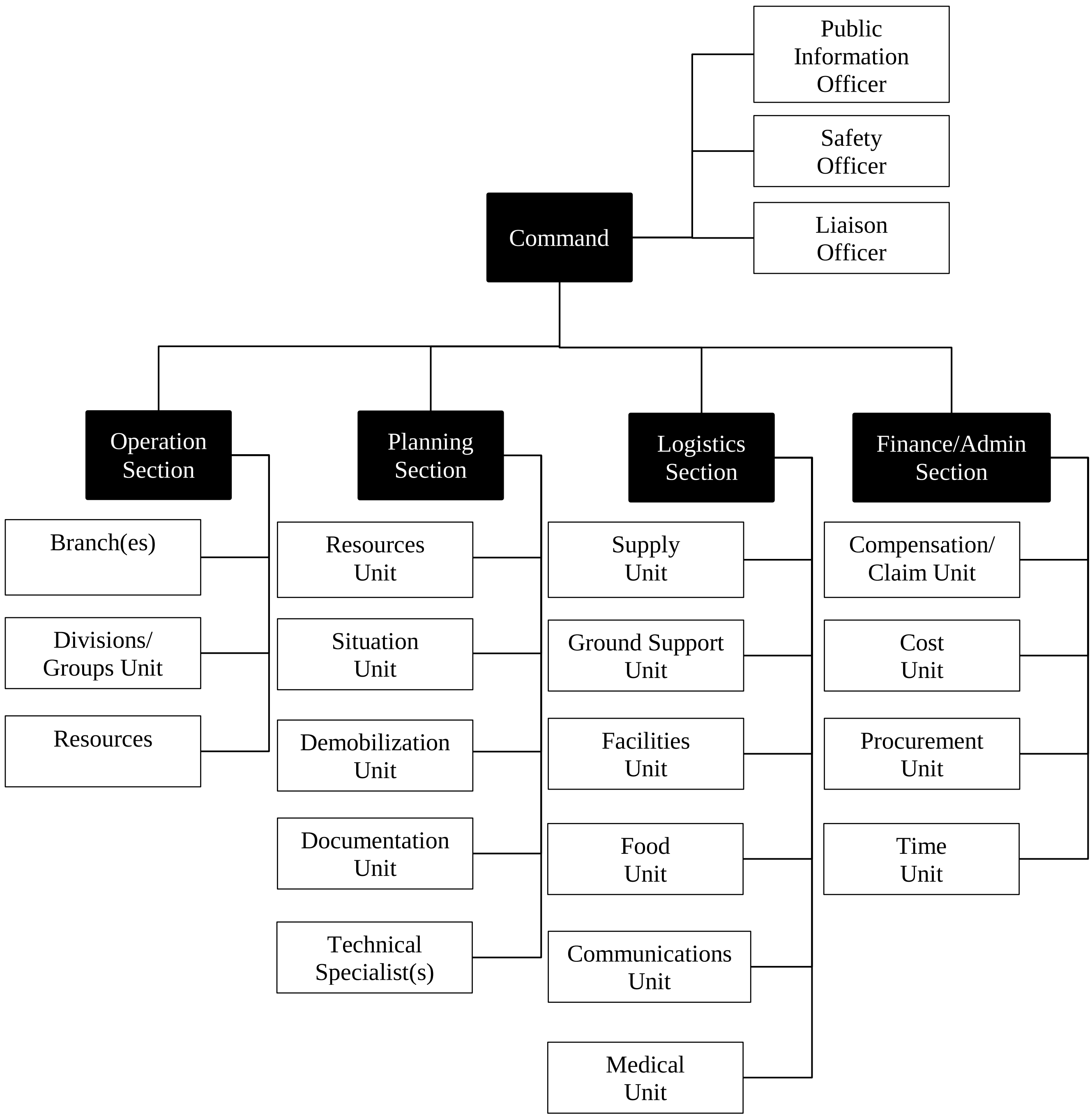

Figure 4.2. Section layers

4.5. Use of the System

Figure 4.3 depicts the banner screen displayed upon entry to the system. Inputting “Y” loads the subsequent screens of module selection screen. Once the users select either the RRA module or the ROS module, a unique caseID will be requested (shown in Figure 4.4). After a brief introduction to the selected module, an interview is conducted by the module starts. For example, Figure 4.5 to Figure 4.7 demonstrate the use of the RRA module; Figure 4.8 to Figure 4.13 illustrates the process to construct a coordination structure. Note that the user responses appear as part of the interview dialogue. Figure 4.5 and Figure 4.6 are the screen shots of the introduction to the RRA module and the interview dialogue. Once the users select a role, the screen of the corresponding responsibilities and actions matched by the RRA module displays (shown in Figure 4.7). At the same time, for the users’ future need, more detailed information is written in a CSV formatted report named with the unique caseID. Figure 4.7 is the screenshot of RRA report for the role of individual, families, and households.

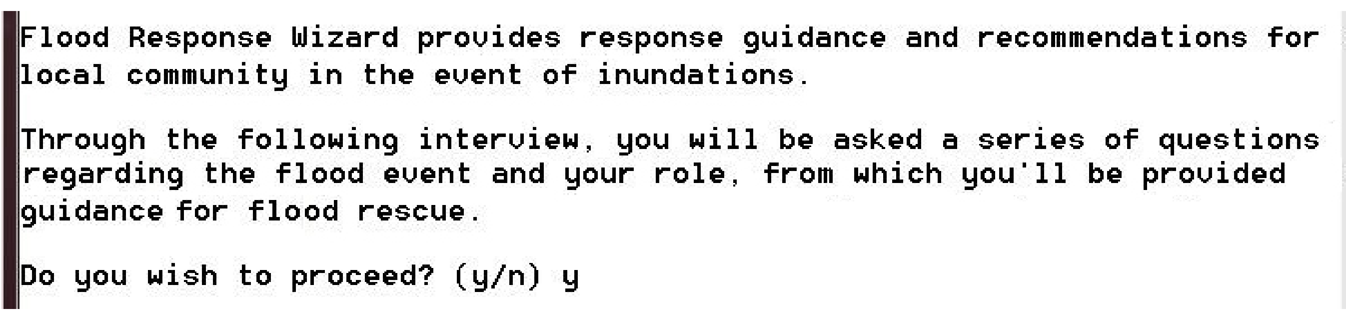

Figure 4.3. Banner screen

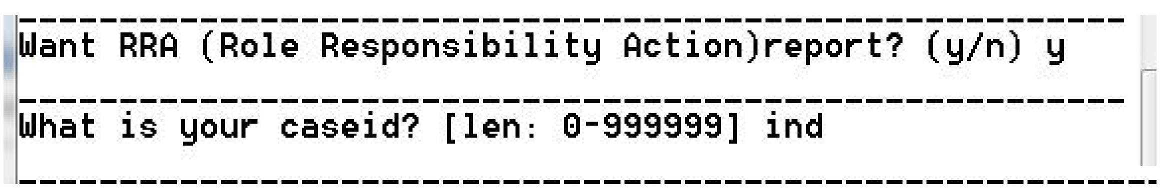

Figure 4.4. Module selecting screen

```
Effective response depends on integration of the whole community and all partners
executing their roles and responsibilities. This section describes those roles
and responsibilities and sharpens the focus on identifying who is involved with
the Response mission area. It also addresses what the various partners must do
to deliver the response core capabilities and to integrate successfully with the
Prevention, Protection, Mitigation, and Recovery mission areas.

An effective, unified national response requires layered, mutually supporting
capabilities. Individuals and families, communities, the private sector, NGOs, and
local, state, tribal, territorial, insular area, and Federal governments should
each understand their respective roles and responsibilities and how to complement
each other in achieving shared goals. All elements of the whole community play
prominent roles in developing the core capabilities needed to respond to incidents.
This includes developing plans, conducting assessments and exercises, providing
and directing resources and capabilities, and gathering lessons learned. These
activities require that all partners understand how they fit within and are
supported by the structures described in the NRF.

Emergency management staff in all jurisdictions have a fundamental responsibility
to consider the needs of all members of the whole community, including children;
individuals with disabilities and others with access and functional needs; those
from religious, racial, and ethnically diverse backgrounds;and people with limited
English proficiency. The potential contributions of all these individuals toward
delivering core capabilities during incident response (e.g., through associations
and alliances that serve these populations) should be incorporated into planning
efforts.
```

Figure 4.5. Introduction to RRA module

```
What is your role?
  1. Individuals, Families, and Households
  2. Communities
  3. Nongovernmental Organizations
  4. Private Sector Entities
  5. Local Governments
  6. State, Tribal, Territorial, and Insular Area Governments
  7. Federal Government
? [1-7] 1
```

Figure 4.6. Start screen of interview in RRA module

```
Responsibilities of Individuals, Families, and Households :

Although not formally part of emergency management operations, individuals,
families, and households play an important role in emergency preparedness and
response. By reducing hazards in and around their homes by efforts such as
raising utilities above flood level or securing unanchored objects against the
threatof high winds, individuals reduce potential emergency response requirements.
Individuals, families, and households should also prepare emergency supply kits
and emergency plans so they can take care of themselves and their neighbors until
assistance arrives. Information on emergency preparedness can be found at many
community, state, and Federal emergency management Web sites, such as
http://www.ready.gov.
Individuals can also contribute to the preparedness and resilience of their
households and communities by volunteering with emergency organizations (e.g.,
the local chapter of the American Red Cross, Medical Reserve Corps, or Community
Emergency Response Teams [CERTs]) and completing emergency response training
courses. Individuals, families, and households should make preparations with
family members who have access and functional needs or medical needs. Their
plans should also include provisions for their animals, including household pets
or service animals. During an actual disaster, emergency, or threat,individuals,
households, and families should monitor emergency communications and follow
guidance and instructions provided by local authorities.

Actions:

Many individuals have talents and experience that can be tapped to support core
capabilities. Individuals can contribute to the delivery of response core
capabilities through community organizations, by participating in community
preparedness activities, such as CERT, and by ensuring that they have
household/family emergency plans.Individual and household preparedness
information can be located at http://www.ready.gov/make-a-plan


****************     Report   End    *****************

Job Done! For more details, see your Role_Response_Action Report: RRA_ind in
frs_reports\
```

Figure 4.7. RRA report screen for the role of individual

```
Want RRA (Role Responsibility Action)report? (y/n) n
______________________________________________________________________________
Emergency management and incident response refer to the broad spectrum of activities
and organizations providing effective and efficient operations, coordination, and
support. Incident management, by distinction, includes directing specific incident
operations; acquiring, coordinating, and delivering resources to incid  to incident
sites; and sharing information about the incident with the public.

Reset your Response Operational Structure? (y/n) y
Resetting your Response Operational Structure...
______________________________________________________________________________
What is your caseid? [len: 0-999999] local
______________________________________________________________________________
```

Figure 4.8. Introduction to ROS module

```
______________________________________________________________________________
The Incident Command System (ICS), Multiagency Coordination System (MACS), and
Public Information are the fundamental elements of incident management used to
facilitate incident Command and Management operations.

Want to know the relationships among those elements? (y/n) y
Although described herein as separate elements of Command and Management within
NIMS. However, NIMS relies on the relationships among these elements along with
the elements themselves. Some relationships are specifically defined.For example,
an Area Command or Incident Command coordinates with Public Information on incident
specific public information through an incident Public Information Officer
within the JIS. The relationship between Area Command or Incident Command and
MACS is primarily defined by a communications link between Command and/or field
level personnel with resource management responsibilities and a particular staff
position within multiagency coordination.

These relationships-along with other relationships among Command and Management
elements that are not as clearly defined in advance-must be clearly defined and
documented as each element evolves during an incident.
```

Figure 4.9. Additional information on the relationships among ICS, MACS, and PI

```
ICS is used by all levels of government-Federal, State, tribal, and local-as well
as by many NGOs and the private sector.ICS is also applicable across disciplines.
It is normally structured to facilitate activities in five major functional areas:
Command, Operations, Planning, Logistics, and Finance/Administration. In
telligence/Investigations is an optional sixth functional area that is activated
on a case-by-case basis.

Want to know management charateristics before reset? (y/n) y
--------------------------------------------------------------------------------
ICS is based on 14 proven management characteristics, each of which contributes
to the strength and efficiency of the overall system.

Which one(s) do you want to know more? (use comma to seperate your choices)
  1. Command Terminology
  2. Modular Organization
  3. Management by Objectives
  4. Incident Action Planning
  5. Manageable Span of Control
  6. Incident Facilities and Locations
  7. Comprehensive Resource Management
  8. Integrated Communications
  9. Establishment and Transfer of Command
 10. Chain of Command and Unity of Command
 11. Unified Command
 12. Accountability
 13. Dispatch/Deployment
 14. Information and Intelligence Management
? [1-14, ...] 1, 3
Command Terminology:
ICS establishes common terminology that allows diverse incident management and
support organizations to work together across a wide variety of incident management
functions and hazard scenarios. This common terminology covers the following:

(1) Organizational Functions
Major functions and functional units with incident management responsibilities are
named and defined. Terminology for the organizational elements is standard and
consistent.
(2) Resource Descriptions
Major resources-including personnel, facilities, and major equipment and supply
items-that support incident management activities are given common names and are
'typed' with respect to their capabilities,to help avoid confusion and to enhance
interoperability.
(3) Incident Facilities
Common terminology is used to designate the facilities in the vicinity of the
incident area that will be used during the course of the incident.

Management by Objectives:
Management by objectives is communicated throughout the entire ICS organization
```

Figure 4.10. Additional information on the 14 characteristics of ICS

```
---------------------------------------------------------------------------
Incident Command is responsible for overall management of the incident. Overall
management includes Command Staff assignments required to support the command
function. The Command and General Staffs are typically located at the Incident
Command Post (ICP).The command function may be conducted in one of two general
ways : Single Incident Commander(IC) and Unified Command(UC).

Want more information about IC and UC? (y/n) y

---------------------------------------------------------------------------
Which one(s) do you want to know more? (use comma to seperate )

  1. Single Incident Commander:
  2. Unified Commander:
  3. Comparison of Single IC and UC
? [1-3, ...] 2,3
UC is an important element in multijurisdictional or multiagency incident management.
It provides guidelines to enable agencies with different legal, geographic , and
functional responsibilities to coordinate, plan, and interact effectively.  As a
team effort, UC allows all agencies with jurisdictional authority or functional
responsibility for the incident to jointly provide management direction through
a common set of incident objectives and strategies and a single Incident Action
Plan (IAP). Each participating agency maintains its authority, responsibility,
and accountability.
```

Figure 4.11. Additional information on UC and the comparison

```
---------------------------------------------------------------------------
What is the name of your Single Incident Commander? [len: 0-999999] SIC

---------------------------------------------------------------------------
What is the work phone number of SIC? [len: 0-999999] SIC WP

---------------------------------------------------------------------------
What is the cellphone number of SIC? [len: 0-999999] SIC CP

---------------------------------------------------------------------------
What is the email address of SIC? [len: 0-999999] SIC Email

---------------------------------------------------------------------------
Add one more person in Unified Command? (y/n) n
```

Figure 4.12. Acquisition of the contact information of the commander

```
--------------------------------------------------------------------------------
What is the name of your Single Incident Commander? [len: 0-999999] SIC
--------------------------------------------------------------------------------
What is the work phone number of SIC? [len: 0-999999] SIC WP
--------------------------------------------------------------------------------
What is the cellphone number of SIC? [len: 0-999999] SIC CP
--------------------------------------------------------------------------------
What is the email address of SIC? [len: 0-999999] SIC Email
--------------------------------------------------------------------------------
Add one more person in Unified Command? (y/n) n
--------------------------------------------------------------------------------
Does report of Command Staff of ROS already exist? (y/n) n
Reset Command Staff...
--------------------------------------------------------------------------------
In an incident command organization,the Command Staff typically includes a Public
Information Officer,a Safety Officer,and a Liaison Officer,who report directly
to the IC/UC and may have assistants as necessary. Additional positions may
 be required, depending on the nature, scope, complexity, and location(s) of the
 incident(s), or according to specific requirements established by the IC/UC.

Public Information Officer:
The Public Information Officer is responsible for interfacing with the public and
media and/or with other agencies with incident-related information requirements.
The Public Information Officer gathers, verifies, coordinates, and disseminates
accurate, accessible, and timely information on the incident's cause, size, and
current situation; resources committed; and other matters of general interest
for both internal and external audiences.The Public Information Officer may also
perform a key public information-monitoring role. Whether the command structure
is single or unified, only one Public Information Officer should be designated
per incident. Assistants may be assigned from other involved agencies, departments,
or organizations. The IC/UC must approve the release of all incident-related
information.In large-scale incidents or where multiple command posts are established,
the Public Information Officer should participate in or lead the Joint Information
Center (JIC) in order to ensure consistency in the provision of information
 to the public.

What is the name of your Public Information Officer? [len: 0-999999] PIO
```

Figure 4.13. Acquisition of the contact information of Public Information Officer

4.6. Verification and Validation

The technical goal of Verification and Validation (V&V) is determining whether the expert system conforms to the requirements and satisfy customers' needs (IEEE, 2012; O'Keefe & O'Leary, 1993). V&V are vital components to ensure the quality of developed expert systems through the processes of analysis, evaluation, review, inspection, assessment, and testing of products.

Typically, developers conduct a set of test cases to assure they are building the expert system correctly. These cases are either collected from real life situations or designed by domain experts to represent the possible problems in implementation (O'Leary, et al., 1990; Leondes, 2002). With the help of the debugging tools built in Python, we periodically verified the LFRS throughout the development stage by conducting a complete set of pre-defined tests. Specifically, the the RRA module was tested role by role. Similarly, the ROS module was primarily tested section by section. Based on the results of those tests, we modified or reprogrammed the necessary heuristic knowledge and inferential logic.

A common method to assure building the right prototype of expert systems combines face validation (the process by which the experts assess the prototype “at face value”) with component testing and system validation through cases or Turing tests (O'Leary et al., 1990). According to this paradigm, experts from the fields of water resources management and emergency management viewed the system’s operation, output, and documentation. In addition, the experts tested our system using selected cases from their experience.

4.7. Conclusions

To break through the bottleneck of knowledge management in local flood response coordination, we identified the knowledge necessary to incorporate into the LFRS and assimilated these knowledge primitives into the knowledge base. The case studies illustrate that both the RRA module and the ROS module work out the correct reports. The responsibilities and actions match with the various roles accurately. Hierarchies of response operational structure correctly link with each other. The contact information and capabilities of each staff lay out clearly. Introductions and all additional information pop up promptly. By repeated running either the RRA module or the ROS module, the emergency personnel is well trained.

Security and resilience work is never finished. The LFRS can be improved to face more challenges simply by blending more knowledge into the two existing modules or adding more modules into the LFRS. For example, the responsibilities, actions, and capabilities of the governmental roles above local level can be incorporated into the RRA module and the ROS module to prepare for larger scale and more complicated flood events. New modules cover other mission areas like planning and recovery can be built to provide better performance in flood response.

4.8. References

Chau, K. W., & Phil, M. (2004). Knowledge-based system on water resources management in coastal waters. *Water and Environmental Journal, 18*(1), 25-28. https://doi.org/10.1111/j.1747-6593.2004.tb00488.x

U.S. Department of Homeland Security (2008). *National incident management system.* Retrieved from https://www.fema.gov/pdf/emergency/nims/NIMS_core.pdf

U.S. Department of Homeland Security (2013). *National response framework.* Retrieved from http://www.fema.gov/media-library-data/20130726-1914-25045-1246/final_national_response_framework_20130501.pdf

U.S. Department of Education (2009, January 1). *Action guide for emergency management at institutions of higher education.* Retrieved from https://www.edpubs.gov/document/ed005103p.pdf?ck=5

U.S. Department of Labor (2016). *What is an Incident Command System?* Retrieved November 12, 2016, from https://www.osha.gov/SLTC/etools/ics/what_is_ics.html

Durkin, J. (1994). *Expert systems: Design and development.* Macmillan.

Feigenbaum, E. (1977). *The art of artificial intelligence: Themes and case studies of knowledge engineering.* School of Humanities and Sciences, Stanford University, Computer Science Department, Stanford.

Federal Emergency Management Agency (2014, September 5). *Prepare your organization for a flood playbook.* Retrieved from https://www.fema.gov/media-library/assets/documents/98407

Federal Emergency Management Agency (2016, March 30). *America's PrepareAthon! Additional playbook materials.* Retrieved from https://www.fema.gov/media-library-data/1409002852888-3c5d1f64f12df02aa801901cc7c311ca/how_to_prepare_flood_033014_508.pdf

Federal Emergency Management Agency (2016). *Incident command system resources*. Retrieved 11 15, 2016, from https://www.fema.gov/incident-command-system-resources

Federal Emergency Management Agency (2017, February 22). *National flood insurance program.* Retrieved from https://www.fema.gov/national-flood-insurance-program

Flin, R., O'Connor, P., & Crichton, M. (2008). *Safety at the sharp end: A guide to non-technical skills.* Farnham, Surrey, United Kindom: Ashgate Publishing, Ltd.

Hernandez, J. Z., & Serrano, J. M. (2001). Knowledge-based models for emergency management systems. *Expert Systems with Applications, 20*, 173-186. https://doi.org/10.1016/S0957-4174(00)00057-9

Institute of Electrical and Electronics Engineers Standards Association (2012, 10 12). *IEEE Standard for System and Software Verification and Validation.* Retrieved from IEEE Standards Association: https://standards.ieee.org/findstds/standard/1012-2012.html

Jackson, P. (1998). Introduction to expert systems. In P. Jackson, *Introduction to expert systems* (3ed ed., p. 2). Addison Wesley.

Jennex, M. E. (2007). Modeling emergency response systems. *Proceedings of the 40th Annual Hawaii International Conference on System Sciences* (pp. 22-22). Hawaii: Hawaii International International Conference on System Sciences. https://doi.org/10.1109/HICSS.2007.386

Lee, W., Wang, Y., Wang, W., & Cheung, C. (2012). An unstructured information management system (UIMS) for emergency management. *Expert Systems with Applications, 39*, 12743-12758. https://doi.org/10.1016/j.eswa.2012.02.037

Leon, C., Martin, S., Elena, J. M., & Luque, J. (2000). Explore: Hybrid expert system for water networks management. *Journal of Water Resources Planning and Management, 126*(2), 65-74. http://dx.doi.org/10.1061/(ASCE)0733-9496(2000)126:2(65)#sthash.15WONqGB.dpuf

Leondes, C. T. (2002). *Expert systems: The technology of knowledge management and decision making for the 21st century.* Academic Press.

Liou, Y. I. (1990). Knowledge acquisition: Issues, techniques, and methodology. *Conference on Trends and Directions in Expert Systems* (pp. 36-212). Orlando: ACM. https://doi.org/10.1145/97709.97726

Liu, K. F. (2004). Agent-based resource discovery architecture for environmental emergency management. *Expert Systems with Applications, 27*, 77-95. 10.1016/j.eswa.2003.12.016

Liu, X., Li, W., Tu, Y., & Zhang, W. (2011). An expert system for an emergency response management in networked safe service systems. *Expert Systems with Applications,* 11928-11938. https://doi.org/10.1016/j.eswa.2011.03.087

Mounce, S., Boxall, J., & Machell, J. (2010). Development and verification of an online artificial intelligence system for detection of bursts and other abnormal flows. *Journal of Water Resources Planning and Management, 136*(3), 309-318. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000030

National Hurricane Center. (2011, August). *NOAA Technical Memorandum NWS HHC-6.* Retrieved from The Deadliest, Costliest, and Most Intense United States Tropical

Cyclones From 1851 to 2010 (And Other Frequently Requested Hurricane Facts): http://www.nhc.noaa.gov/pdf/nws-nhc-6.pdf

National Integration Center (2011, September). *NIMS training program.* Retrieved from https://www.fema.gov/pdf/emergency/nims/nims_training_program.pdf

National Weather Service (2015, March). *Turn around don't drown.* Retrieved from http://www.nws.noaa.gov/os/water/tadd/

O'Keefe, R. M., & O'Leary, D. E. (1993). Expert system verification and validation: a survey and tutorial. *Artificial Intelligence Review, 7*, 3-42. https://doi.org/10.1007/BF00849196

O'Leary, T. J., Goul, M., Moffitt, K. E., & Radwan, A. E. (1990). Validating expert systems. *IEEE Intelligent Systems, 5*(3), 51-58. https://doi.org/10.1109/64.54673

Ooshaksaraie, L., & Basri, N. E. (2011). An expert system applied in construction water quality monitoring. *American Journal of Environmental Sciences, 7*(11), 75-81. https://doi.org/10.3844/ajessp.2011.75.81

PyKE. (2015). Retrieved from Welcome to Pyke: http://pyke.sourceforge.net/

Python. (2015). Retrieved from Python: https://www.python.org/

Robindro, K., & Sarma, S. K. (2013). JESS based expert system architecture for diagnosis of rice plant diseases: Design and prototype development. *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, (pp. 674-676). 10.1109/ISMS.2013.54

Select Bipartisan Committee of the U.S. House of Representatives. (2006, February 15). *A failure of initiative: The final report of the select bipartisan committee to investigate the preparation for and response to Hurricane Katrina.* Retrieved from A failure of initiative: The final report of the select bipartisan committee to investigate the preparation for and response to Hurricane Katrina: https://www.gpo.gov/fdsys/pkg/CRPT-109hrpt377/pdf/CRPT-109hrpt377.pdf

Spyridakos, T., Pierakos, G., Metaxas, V., & Logotheti, S. (2005). Supporting the management of measurement network with an expert system: The NeMO system. *Operational Research An International Journal, 5*(2), 273-288. https://doi.org/10.1007/BF02944313

Storey, V. C., Chiang, R. H., & Chua, C. E. (2012, January). Knowledge representation: A conceptual modeling approach. *Journal of Database Management, 23*(1), 1-30. https://doi.org/10.4018/jdm.2012010101

Turoff, M., Chumer, M., Walle, B. V., & Yao, X. (2004). The design of a dynamic emergency response management information system (DERMIS). *The Journal of Information Technology Theory and Application (JITTA), 5*(4), 1-35. Retrieved

from
https://pdfs.semanticscholar.org/3dcc/6657c3d895a3e03b888bdce1f9465d7a29ee.pdf

Williams, L. (2014, August 1). *It started like any other day*. Retrieved from https://www.fema.gov/media-library/assets/videos/97885

Wong, B. K., & Monaco, J. A. (1995, September). Expert system applications in business: a review and analysis of the literature (1977-1993). *Information and Management, 29*(3), 141-152. https://doi.org/10.1016/0378-7206(95)00023-P

5. CONCLUSIONS

This thesis, written in a journal article form, proposes three papers focusing on expert systems for disaster forecasting, warning, recovery, and response in water resources management. Each article discusses the subject from a different perspective (WDN decontamination, local flood forecasting and warning, and local flood response coordination and training). As a whole, the three papers form a coherent thesis that proposes a methodology, illustrates the implementation, and investigate the benefits by comparison with others.

Research Directions

Our study establishes and automates the necessary knowledge for to break through the bottleneck of knowledge acquisition in disaster warning, recovery, and response in water resources management. After assimilation, knowledge bases incorporating both dynamic and static knowledge primitives are developed for those three scenarios, respectively. In this work, we describe how the dynamic knowledge primitives of particular cases are learned from specific users and how the general human implicit knowledge is distilled from official documents, classical journal papers, and interviews. We also explain how the knowledge is compiled into computer language and how the compiled knowledge is stored, shared and driven by both inferential logic and procedural logic.

We selected an expert system shell-PyKE that was capable of both forward and backward chaining inferential logic. Based on PyKE, we developed three conceptual systems: a conventional procedural pseudo-code, a forward chaining expert system framework, and a backward chaining expert system framework for each scenario. We found that developing expert systems to support decision-making for WDN decontamination, local flood forecasting & warning, and local flood response coordination & training is more effective than procedural codes through our analyses. The system architectures of expert systems initiate the rewards of speedy development and convenient maintenance. The explicit knowledge bases and packaged expert system shell enable that 1) new facts, rules, questions, and goals can be easily added to the extensible knowledge base by domain experts, such as environmental engineers, or even skilled users to adapt to more scenarios; 2) all goals can be skipped or proved in various sequences automatically (forward chaining) or according to users' demands (backward chaining); 3) partially developed expert systems can be functional; 4) logic or syntax errors and outdated data can be rapidly identified and corrected; and 5) the users can understand and learn the reasoning simultaneously when they obtain the reports.

Furthermore, the backward chaining method performs more effectively than forward chaining to satisfy users' evolving demands and growing new information in the three scenarios. Backward chaining enables the inference engine of the three expert systems to start from incomplete information and to keep running as more and more information becomes available. With backward chaining inference engines, our expert systems can quickly figure out and optimally collect the necessary data from previous analyses or user interviews based on the developing goals, and efficiently develop reports and recommendations with the accumulating information at hand. Therefore, we decided to turn the backward chaining frameworks into complete backward chaining expert systems for WDN decontamination, local flood forecasting & warning, and local flood response coordination & training.

We periodically verified our systems throughout the development stage by conducting complete sets of pre-defined tests, with the assistance of debugging tools built in Python. These tests were either collected from real life situations or designed by domain experts to represent the possible problems in implementation. To make sure that we built the correct expert systems, experts from a water resources management area reviewed the systems' operations, outputs, and documentation. In addition, the experts tested our systems using the cases they are familiar. The accuracy of the systems are evaluated by comparing the results provided by the systems with the documented test cases. Based on the results of the tests of Validation and Verification (V & V) (O'Keefe & O'Leary, 1993), we redesigned and reprogrammed the necessary heuristic knowledge and inferential logic.

The Three Journal Articles

The three expert systems work efficiently with extensive knowledge bases and backward chaining inference logic.

Article One-Evaluation of the Benefits of Using a Backward Chaining Expert System for Water Distribution Networks Decontamination

The first proposed expert system, Decon provides reasoning routines and recommendations on the type of contamination event and consequences on the water operators, the public in general, the environment, and the potential threat from the different interactions with the network pipe material. It also gives users the guidance on the currently available technologies and their effectiveness along the optimized quick solution.

Article Two-Evaluation of the Benefits of Using a Backward Chaining Expert System for Local Flood Forecasting and Warning

The second expert system, LFFWS provides reasoning procedures and forecasting on the flood magnitude, warning stages, potential damage, and recommendations for community authorities, landowners, or public in general. LFFWS can decrease the engagement of computational resources by minimizing the boundary of the interested area and decreasing the usage of complicated numerical models. This makes flood forecasting more economical and therefore realistic for small communities to fit their tight flood management budgets. This research represents an advance in the applicability of expert systems to solve flood prediction and management problems.

Article Three-Evaluation of the Benefits of Using a Backward Chaining Expert System for Local Flood Response Coordination and Training

LFRS, the third expert system can help emergency managers construct scalable, flexible, and adaptable coordination structures and support mentoring and drilling flood responders such as individuals, communities, nongovernmental organizations, private sector entities, and local governments. The prototype expert system products two CSV formatted reports as well as prompt screens. The operational structure report hierarchically depicts the crisscross linkages among all response entities, their primary functions, and contact information. Another report is a review of the responsibilities and actions of a certain role of flood responders from authorities to individuals.

Limitations and Future Research Directions

Security and resilience work is never finished. Our expert systems are always ready for breaking their limits and get through. More goals and their corresponding knowledge related to other key issues can be easily updated to face more challenges. For example, one cost-effective analysis goal can be added to give the users guidance on the optimization of the currently available and possible technologies for Decon; Interactions between multiple contaminants can be taken into consideration, as well. Moreover, other software like EPANET can be linked with the current system to work for a whole WDN rather than one isolated node. In other words, this expert system can quickly "learn" more and therefore, becomes more robust to recommend a thorough and expedited solution to WDN decontamination. Regarding LFFWS and LFRS, this research represents an advance in the applicability of expert systems to solve flood prediction and management problems. The historical records, responsibilities, actions, and capabilities of the governmental roles above local level can be incorporated to prepare for larger scale and more complicated flood events.

In addition, the framework of our expert systems can easily be duplicated to create other expert systems. Domain experts in other areas can make use of our framework to record their valuable expertise and undocumented "rules of thumb" in computer-readable language and create more expert systems to perform repeated work efficiently.

Key Benefits

In summary, we provide an advanced solution to water resources management. Specifically, we evaluate the utility of using expert systems with backward chaining inference engine in WDN decontamination, flood forecasting and warning, and flood response coordination and training. We overcome the block of knowledge acquisition and build three prototypes for the proposed research fields. The accurate results show that with less computational resources, the three expert systems efficiently help the water resources managers and community authorities make critical decisions and give timely recommendations and guidance for all related people corresponding to their roles. The convenience of implying the framework of our prototypes to other research areas will make our investigation to be prosperous in the near future.

REFERENCES

Afshar, A., Massoumi, F., Afshar, A., & Mariño, M. A. (2015, September). State of the Art Review of Ant Colony Optimization Applications in Water Resource Management. *Water Resources Management, 29*(11), 3891-3904.

Altenburger, R., Ait-Aissa, S., Antczak, P., Backhaus, T., Seiler, D. B.-B., Brion, F., . . . Faust, M. (2015, April 15). Future water quality monitoring — Adapting tools to deal with mixtures of pollutants in water resource management. *Science of The Total Environment, 512-513*, 540-551. doi:https://doi.org/10.1016/j.scitotenv.2014.12.057

Altunkaynak, A. (2014). Predicting Water Level Fluctuation in Lake Michigan-Huroon Using Wavelet-Expert System Methods. *Water Resource Manage, 28*(8), 2293-2314.

APFM. (2013). *Integrated Flood Management Tools Series Flood Forecasting and Early Warning.* Retrieved from Associated Programme on Flood Management: https://library.wmo.int/pmb_ged/ifmts_19.pdf

Benson, D., Gain, A. K., & Rouillard, J. J. (2015). Water Governance in a Comparative Perspective: From IWRM to a 'Nexus' Approach? *Water Alternatives, 8*(1), 756-773.

Cardwell, H. E., Cole, R. A., Cartwright, L. A., & Martin, L. A. (2009, May 1). Integrated Water Resources Management: Definitions and Conceptual Musings. *Journal of Contemporary Water Research & Education, 135*(1), 8-18.

CBC News. (2010). *Inside Walkerton: Canada's Worst-Ever E. coli Contamination.* CBC News. Retrieved from http://www.cbc.ca/news/canada/inside-walkerton-canada-s-worst-ever-e-coli-contamination-1.887200

Certescu, I., Craciun, I., Benchea, R. E., Kovács, Z., Iavorschi, A., Sontea, V., & Macoveanu, M. (2013). Development of An Expert System for Surface Water Quality Monitoring in The Context of Sustainable Mangement of Water Resources. *Environmental Engineering and Management Journal, 12*(8), 1721-1734.

Chau, K. W., & Phil, M. (2004). Knowledge-Based System on Water Resource Management in Coastal Waters. *Water and Environmental Journal, 18*(1), 25-28.

Chiketo, B. (2012, October 5). *Zimbabwe's Troubled Waters: Chemical Pollution in the Marange District*. Retrieved from Think Africa Press: https://www.rapaportfairtrade.com/Docs/100512_ZimbabwesTroubledWaters.pdf

Decision Making. (2015, 12 22). Retrieved from BusinessDictionary.com: http://www.businessdictionary.com/definition/decision-making.html

EPA. (2015, April). *Understanding the Safe Drinking Water Act.* Retrieved July 8, 2017, from Overview of the Safe Drinking Water Act: https://www.epa.gov/sites/production/files/2015-04/documents/epa816f04030.pdf

Fang, S., Xu, L., Pei, H., Liu, Y., Liu, Z., & Zhu, Y. (2014, February). An Integrated Approach to Snowmelt Flood Forecasting in Water Resource Management. *IEEE Transactions on Industrial Informatics*, 548-558.

Feigenbaum, E. (1977). *The art of artificial intelligence: themes and case studies of knowledge engineering.* School of Humanities and Sciences, Stanford University, Computer Science Department, Stanford.

Flin, R., O'Connor, P., & Crichton, M. (2008). *Safety at the sharp end: A guide to non-technical skills.* Farnham, Surrey, United Kindom: Ashgate Publishing, Ltd.

Francis, R., Guikema, S., & Henneman, L. (2014). Bayesian Belief Networks for predicting drinking water distribution system pipe breaks. *Reliability Engineering and System Safety*, *130*, 1-11.

Fulton, D. (2015, April 22). *North Carolina Residents Warned Not to Drink Water Near Coal Ash Dumps*. Retrieved from CommonDreams: Breaking News & Views for the Progressive Community: https://www.commondreams.org/news/2015/04/22/north-carolina-residents-warned-not-drink-water-near-coal-ash-dumps

Geissen, V., Mol, H., Klumpp, E., Umlauf, G., Nadal, M., Ploeg, M. v., . . . Ritsema, C. J. (2015, March 1). Emerging pollutants in the environment: A challenge for water resource management. *International Soil and Water Conservation Research*, *3*(1), 57-65. doi:https://doi.org/10.1016/j.iswcr.2015.03.002

Giupponi, C., & Sgobbi, A. (2013, June 19). Decision Support Systems for Water Resources Management in Developing Countries: Learning from Experiences in Africa. *Journal of Hydrologic Engineering*, *5*(2), 798-818. doi:10.3390/w5020798

Halbe, J., Pahl-Wostl, C., Sendzimir, J., & Adamowski, J. (2013, June). Towards adaptive and integrated management paradigms to meet the challenges of water governance. *Water Science & Technology*, *67*(11), 2651-2660. doi:10.2166/wst.2013.146

Heckerman, D., Horvitz, E., & Nathwani, B. N. (1992, June). Toward Normative Expert Systems. *Methods of Information in Medicine*, *31*, 90-116.

Jackson, P. (1998). Introduction To Expert Systems. In P. Jackson, *Introduction To Expert Systems* (3ed ed., p. 2). Addison Wesley.

Kenov, K. N., & Ramos, H. M. (2012). Water and energy sustainable management in irrigation systems network. *International Journal of Energy and Environment, 3*(6), 833-860.

Kolinjivadi, V., & Kosoy, N. (2014, December). Recasting payments for ecosystem services (PES) in water resource management: A novel institutional approach. *Ecosystem Services, 10*, 144-154. doi:https://doi.org/10.1016/j.ecoser.2014.08.008

Ladopoulos, E. G. (2013). Non-linear Pipe Networks Water Management Real-Time Expert Telematics System. *Journal of Water Resource and Hydraulic Engineering, 2*(1), 13-20.

Leon, C., Martin, S., Elena, J. M., & Luque, J. (2000). Explore: Hybrid Expert System For Water Networks Management. *Journal of Water Resources Planning and Management, 126*(2), 65-74.

Marti, B., Bauser, G., Stauffer, F., Kuhlmann, U., Kaiser, H.-P., & Kinzelbach, W. (2012). An Expert System for Real-time Well Field Management. *Water Science & Technology: Water Supply, 12*(5), 699-706.

Mavrommati, G., Bithas, K., & Panayiotidis, P. (2013). Operationalizing Sustainability in Urban Coastal Systems: A System Dynamics Analysis. *Water Research, 47*(20), 7235-7250.

McNabb, D. E. (2017, June 3). Integrated Water Resource Management. *Water Resource Management*, 329-349.

Medsker, L., & Liebowitz, J. (1994). *Design and Development of Expert Systems and Neural Networks.* New York: Macmillan.

Mounce, S., Boxall, J., & Machell, J. (2010). Development and Verification of an Online Artificial Intelligence System for Detection of Bursts and Other Abnormal Flows. *Journal of Water Resources Planning and Management, 136*(3), 309-318.

O'Keefe, R. M., & O'Leary, D. E. (1993). Expert system verification and validation: a survey and tutorial. *Artificial Intelligence Review, 7*, 3-42. doi:https://doi.org/10.1007/BF00849196

Ooshaksaraie, L., & Basri, N. E. (2011). An Expert System Applied in Construction Water Quality Monitoring. *American Journal of Environmental Sciences, 7*(11), 75-81.

Pahl-Wostl, C., & Knieper, C. (2014, November). The capacity of water governance to deal with the climate change adaptation challenge: Using fuzzy set Qualitative Comparative Analysis to distinguish between polycentric, fragmented and centralized regimes. *Global Environmental Change, 29*, 139-154. doi:https://doi.org/10.1016/j.gloenvcha.2014.09.003

Panetto, H., & Cecil, J. (2013). Information systems for enterprise integration, interoperability and networking: theory and applications. *Enterprise Information*

Systems, 7(1), 1-6.

PyKE. (2015). Retrieved from Welcome to Pyke: http://pyke.sourceforge.net/

Robindro, K., & Sarma, S. K. (2013). JESS Based Expert System Architecture For Diagnosis Of Rice Plant Diseases: Design And Prototype Development. *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, (pp. 674-676).

Sankary, N., & Ostfeld, A. (2017, September). Scaled Multiobjective Optimization of an Intensive Early Warning System for Water Distribution System Security. *Journal of Hydraulic Engineering, 143*(9). doi:https://doi.org/10.1061/(ASCE)HY.1943-7900.0001317

Seckler, D., Barker, R., & Amarasinghe, U. (2010, August). Water Scarcity in the Twenty-first Century. *International Journal of Water Resources Development*, 29-42. doi:http://dx.doi.org/10.1080/07900629948916

Sharma, D., & Kansal, A. (2011, December). Water Quality Analysis of River Yamuna Using Water Quality Index in the National Captital Territory, India(2000-2009). *Applied Water Science, 1*(3-4), 147-157.

Spyridakos, T., Pierakos, G., Metaxas, V., & Logotheti, S. (2005). Supporting the Management of Measurement Network with an Expert System: The NeMO System. *Operational Research An International Journal, 5*(2), 273-288.

The World Bank. (2017, August 18). *Water Resources Management*. Retrieved from The World Bank: http://www.worldbank.org/en/topic/waterresourcesmanagement

UN ESCAP. (2016, April 25). *Flood Forecasting and Early Warning in Transboundary River Basins: A Toolkit.* Retrieved from United Nations Economic and Social Commission for Asia and the Pacific: http://www.unescap.org/sites/default/files/Flood_toolkit_HighRes.pdf

United States Geological Survey. (2017, August 18). *Where is Earth's Water*. Retrieved from USGS: https://water.usgs.gov/edu/earthwherewater.html

Velasquez, M., & Hester, P. T. (2013). An Analysis of Multi-Criteria Decision Making Methods. *International Journal of Operations Research, 10*(2), 56-66.

Wang, Z., Zhu, J., & Zheng, H. (2015). Improvement of Duration-Based Water Rights Management with Optimal Water Intake On/Off Events. *Water Resources Management*, *29*(8), 2927-2945.

Wei, X. (2014, April 16). *Lanzhou Admits to Problems in Water Supply.* Retrieved from ChinaDaily: http://all-about-water-filters.com/biggest-water-contamination-disasters/

Xin, K.-l., Tao, T., Li, S., & Yan, H. (2017, February). Contamination Accidents in China's Drinking Water Distribution Networks: Status and Countermeasures. *Water Policy, 19*(1), 13-27. doi:DOI: 10.2166/wp.2016.157

Zwass, V. (2017, August). *Information System*. Retrieved from Encyclopedia Britannica: https://global.britannica.com/topic/information-system