2008

Application of Geographic Information System and Probabilistic Analysis for Water Pipeline Renewal Prioritization

Shu Kai Kong
Western University

Application of Geographic Information System and Probabilistic Analysis for Water Pipeline Renewal Prioritization

(Spine title: GIS and Probabilistic Analysis for Water Pipeline)

(Thesis format: Monograph)

by

Shu Kai Kong

Graduate Program in Engineering Science

Department of Civil and Environmental Engineering

A thesis submitted in partial fulfillment of the requirement for the degree of Master of Engineering Science

The School of Graduate and Postdoctoral Studies

The University of Western Ontario

London, Ontario, Canada

September, 2008

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Supervisor	Examiners
Dr. Hanping Hong	Dr. Greyory A. Kopp
Co-Supervisor	Dr. Sean D. Hinchberger
Dr. Ian D. Moore	Dr. Jinfei Wang

The thesis by

Shu Kai Kong

entitled:

Application of Geographic Information System and Probabilistic Analysis for Water Pipeline Renewal Prioritization

is accepted in partial fulfillment of the
requirements for the degree of
Master of Engineering Science

Date______________________

____
Chair of the Thesis Examination Board

Abstract

Aging infrastructure such as the water distribution system needs to be rehabilitated and maintained to provide uninterrupted and intended service, and to minimize possible failure with minimum expenditure. The failure of water distribution system is inherently uncertain and cannot be predicted deterministically. The theories and methodologies for rational maintenance decisions under uncertainty for water distribution systems have been well developed. The methodologies take into account the uncertainty in the break occurrence and consider the cost associated with rehabilitation and renewal. Some of these methodologies have been implemented in the decision support systems (DSS), and the presentation of the analysis results for the spatially distributed water main network is facilitated through the use of a Geographic Information System (GIS). However, in many studies, the integration of the probabilistic analysis models and a GIS to prioritize the rehabilitation scheme for water distribution systems is often either loose coupling, which is often considered to be cumbersome in data exchange among the GIS and other programs, or embedded computing system, which is often considered to be superficial in problem solving and expensive due to the complex system development. Therefore, in those studies, a GIS is often used as a tool for spatial query, spatial selection or thematic mapping and they do not take the full advantage of the features, such as the spatial analysis functionalities, available within the GIS environment.

To facilitate the municipal engineers in using probabilistic based DSS to make informed decisions, in this study, development of integrated application of the GIS and probabilistic analysis for water pipeline renewal prioritization are carried out. The development includes the implementation of necessary subroutines and functions, and the incorporation of optimization algorithm for the probabilistic analysis in Visual Basic Application (VBA) environment that is supported by the commercially available GIS software, ArcView 9.1 by ESRI. In addition, user friendly Graphic User Interfaces (GUIs) are built and within the ArcView 9.1 to assist the user with data processing (e.g., cost information) and result visualization in map format. The developed system uses the expected total cost during a service period or the expected cost per unit service period as objective functions, and takes into account stochastic break occurrence modeling, pipe material and surrounding soil condition. An illustrative application of the developed system is given with the water distribution system obtained from an industry partner, the City of Hamilton.

Through the example application, it is shown that the developed GIS-integrated DSS is not only an effective and efficient tight coupling system to implement the analysis and provide optimal replacement schedules to decision makers and city planners, but also a user-friendly system with the developed GUIs in ArcGIS.

Acknowledgment

I would like to express my deepest sense of gratitude to my supervisors, Dr. Hanping Hong and Dr. Ian D. Moore, for their patient guidance, support, encouragement and excellent advice. Without their help, this study would not be possible. The information provided by Mr. K. Bainbridge about the City of Hamilton for the present study is greatfully acknowledged.

The assistance provided by Tom Lee for the FORTRAN programming and by Michael Evan on the programming task under the geographic information system environment is gratefully acknowledged. Furthermore, I would like to thank Charissa Chan and Celina Ho who helped me to proofread my thesis.

I am also very grateful to my uncle, Raymond Chung and my aunty, Lily Kong, who encouraged me studying in Canada and have provided assistances since my arrival in Canada.

At last, I am grateful to my family for their unstinted and unconditional support. I would also like to thank my sister who helped me to take care of our parents as I was unable to be with them while I was working on this study.

I dedicate this thesis to my mother and father.

Table of Contents Page

List of Tables Page

List of Figures Page

Nomenclature

A	= model coefficient of break occurrence rate equation (Shaimr and Howard, 1979)
a	= model constant
a_n	= lateral dimension of corrosion pit
b	= model constant
$\mathbf{b}$	= a vector of coefficients in hazard function $h(\tau, Z)$
B_d	= width of bedding ditch
c	= model constant
C_d	= calculation coefficient
C_t	= surface load coefficient
$C_T(t)$	= the present value of the maintenance cost to be incurred over the next t years
D	= nominal pipe diameter
$D(T)$	= depth of corrosion
d	= maximum defect depth
E_p	= pipe material elastic modulus
$E(C_T(t))$	= expected total cost
h	= nominal pipe wall thickness
h_0	= a arbitrary base break occurrence rate
$h(\tau, Z)$	= instantaneous break occurrence rate (hazard function)
I_c	= impact factor
k	= regression parameters
k_1	= model parameter of break occurrence rate equation (Pelletier et al, 2003)
k_2	= model parameter of break occurrence rate equation (Pelletier et al, 2003)
K_d	= deflection coefficient
K_m	= bending moment coefficient

K_q	= fracture toughness
l	= total length of defect
L	= pipe effective length
m_{C_N}	= the expected replacement cost
m_{C_R}	= the expected repair cost
$m(l)$	= Folias factor
$N(t)$	= occurrence of number of breaks during a service period t
$N_S(s)$	= $N(V^{-1}(s))$, normalized homogeneous Poisson process with occurrence rate equal to one
n	= number of breaks
p	= internal pipe pressure or model parameter of break occurrence rate equation (Pelletier et al, 2003)
P_c	= corrosion pit dimension
p_C	= remaining pressure strength of steel pipe
P_{fT}	= tolerable failure probability level
$P(N(t)=n)$	= probability of occurrence of n breaks within (0,t]
$P(\tau)$	= probability of at least one break before the end of a future time τ
r	= pipe outside radius
s	= model constant
$V(t)$	$= \int_0^t \nu(\tau \mid t_0) d\tau$
T	= time
T_1	= replacement time from present
T_{op}	= optimum replacement time
Z	= a vector of covariates in hazard function $h(\tau, Z)$
t_0	= duration of sustained service period of a water pipeline segment
α	= model constant
β	= geometric factor

β^{upper}	= upper bound of β
β^{lower}	= lower bound of β
γ_0	= unit weight of soil or model parameter of break occurrence rate equation (Longanathan et al, 2002)
γ	= the discount rate
δ	= model parameter of break occurrence rate equation (Longanathan et al, 2002)
λ	= break occurrence rate
λ_0	= break occurrence rate of a newly installed pipeline segment
σ_F	= internal fluid pressure
σ_n	= normal tensile stress
σ_S	= external soil pressure
σ_V	= traffic loads
σ_y	= specified minimum yield strength of steel
σ_θ	= circumferential tensile stress
τ	= a value of time

Chapter 1 Introduction

1.1 Background

Economic loss and loss of life due to the deterioration and collapse of infrastructure are well-known problems to engineers and planners. Bridge collapses and water pipe bursts (resulting in sinkholes) are increasing incidents in our daily news. For example, two recent water main break incidents that occurred in North America are: a 70-year-old, 12-inch water main broke beneath a major downtown intersection in London, Ontario on October 31, 2007 (McDermott, 2007); and a 128-year-old, 30-inch water main broke in downtown Cleveland, Ohio on March 6, 2008 (Julie, 2008). Both incidents caused huge sinkholes, road closures, disruption of economic activities, and flooding of nearby buildings. The maintenance and the rehabilitation of these infrastructures need to be optimized because many of them are facing the risk of failure due to their aging and the available funding for their rehabilitation and repair is always limited or insufficient. In other words, although the number of water main breaks has been on the rise, the available funds are limited for Water Utilities. Therefore, the selection of optimum decisions for renewal and/or rehabilitation of water distribution systems must consider budgetary constraints, and must be prioritized given that the available resources for these repairs are limited.

Pipeline break occurrences cannot be predicted deterministically because the physical process leading to the break is both spatially and temporally uncertain. Therefore, to cope with such uncertainty, the selection of the optimal prioritization for rehabilitating water distribution system must carried out under probabilistic framework. Studies for the rational maintenance and rehabilitation scheduling that took into account the uncertainty, have been carried out and presented in the literature (e.g., Pelletier et al., 2003; Hong et al., 2006; and Moglia et al., 2006). These studies established simple and sophisticated methodologies and models for predicting break occurrence probabilistically. Statistical and physical models are the most common models for predicting pipeline deterioration or the frequency of break occurrence. Reviews of several of these models can be found in Kleiner and Rajani (2001), and Rajani and Kleiner (2001). These reviews clearly indicated that although sophisticated physical models could be used to deal with the interaction of environmental actions on deterioration of pipe, pipe residual strength against different loads, and also aid our understanding of the essential physical process of pipe failure, they are very expensive to use for water pipeline renewal prioritization since they often require costly data for their elaborate numerical modeling. However, simple empirical or mathematical statistical models for estimating the pipe break occurrence rate (e.g., Shamir and Howard, 1979; Loganath et al., 2002) could be an alternative for predicting the failure of water pipe and developing prioritization systems for the renewal and/or rehabilitation of water distribution systems. More details will be presented and discussed in Chapter 2 of this study. If the prioritization is to be decided based on benefit/cost analysis or economic efficiency, cost models must also be developed in addition to the use of a mathematical statistical model for break occurrence, and decision rules under uncertainty must be adopted. It should be noted that ideally the cost information should include the cost of rehabilitation, replacement and maintenance, as well as the social cost if possible. Some of the cost information for Canadian cities given by Zhao and Rajani (2002) and Rahman et al. (2005) are adopted by the present study.

The combination of the stochastic break occurrence model and the cost model, together with decision rules, forms the basis for selecting the optimal prioritization schemes to rehabilitate or renew water utilities. It also provides the basis for the development of a decision support system (DSS) that helps the planner to make decisions to rehabilitate existing water distribution systems. Some of the DSSs, including those developed by Herz (1998), Kleiner et al. (1998a,b), Hadzilacos et al. (2000), Burn et al. (2003) and Moglia et al. (2006) are reviewed by Stone et al. (2002) and Moglia et al. (2006). More recently, a system developed for modeling the deterioration rate of water distribution systems and planning their renewal has been described by Kleiner et al. (2007). Some of these systems focus on long term planning, while others concentrate on prioritization, each with different levels of sophistication, objectives, and targeted users. Some of these systems are integrated with a geographic information system (GIS). However, none of the mentioned systems can be considered as tight coupling with the GIS, and rather they simply used the visualization tool and/or data information query capability in GIS. Therefore, they are either cumbersome in exchanging files, or superficial in solving problems or expensive in their complex system development.

1.2 Objectives and thesis outline

The main objective of the present study is to develop a decision support system (DSS) by integrating the GIS and probabilistic analysis for prioritizing the rehabilitation of water distribution system. For the developed DSS, probabilistic models, cost models and decision rules are embedded within the ArcGIS system (ArcGIS 9.1 by ESRI) to form a spatial decision support system (SDSS). User-friendly Graphic User Interfaces (GUIs) are also implemented in the VBA environment that is supported by the ArcGIS. The developed system can be used to evaluate and illustrate (or map) the preferred decisions on prioritized water pipeline renewal plan. This developed system uses probabilistic models, economical criteria, and/or decision rules that are not available in the afore-mentioned systems in the literature. The system is illustrated for the water distribution system from an industry partner, the City of Hamilton.

Details of the adopted probabilistic approach for break occurrence modeling, objective functions, and the decision rules are described in Chapter 2, and its integration into GIS and the illustration of the water distribution system with GIS-integrated DSS are described in Chapter 3. This is followed by summary and conclusions which are presented in Chapter 4. Finally, the implemented program (subroutines, functions and button controls of GUIs) is listed in an appendix.

Chapter 2 Probabilistic models for break occurrence and cost information

2.1 Introduction

The deterioration of water distribution system caused by various corrosion mechanisms such as soil corrosion, bimetallic corrosion and stray currents is a well-known, well-documented and wide spread problem. The degradation of this system causes leaks and breaks, leading to pressure head losses, service interruption, higher pumping costs, erosion of roadways' subgrade, and repair/reconstruction costs. Furthermore, construction, maintenance, and retrofit of water distribution systems cause considerable disruption and inconvenience to the municipalities, and the cost associated with such activities are often difficult to quantify. Deb et al. (2003) pointed out that the replacement and rehabilitation of water distribution systems across the U.S. require many billions dollars; while CWWA (1997) indicated that the cost of upgrading municipal water distribution systems in Canada is estimated to be about 11.5 billion Canadian dollars over the next 15 years. This simply shows that the need for the rehabilitation and the retrofitting of water distribution systems has been apparent for some time.

Since the natural and environmental actions causing the pipeline to deteriorate is uncertain, consequently, the pipeline break occurrence is also uncertain, and this uncertainty must be taken into account in making decisions for the rehabilitation and retrofitting of the water distribution system. Moreover, the selection of optimum decisions for renewal and/or rehabilitation of water distribution systems must consider budgetary constraints and must be prioritized given that the available resources for these repairs are limited. Using such optimum decisions will enable agencies to maximize the benefit/cost ratio associated with capital expenditures on their water distribution systems.

The selection of rational and optimized rehabilitation and retrofitting decisions, and the development of prioritization systems incorporates the information on the break occurrence modeling, the rehabilitation and repair cost, and possibly intangible cost (e.g., interruption of business, inconvenience to motorist), and adopted decision rules under uncertainty.

A review of statistical and physical models for predicting pipeline deterioration or the frequency of break occurrence can be found in Kleiner and Rajani (2001), and Rajani and Kleiner (2001). Furthermore, Muhlbauer (2004) discussed extensively an index based approach in predicting the pipeline deterioration. Based on these studies, it is clear that the index based prediction approach is very simple to use, but subjective and semi quantitative. Although sophisticated physical models could be used to deal with the interaction of environmental actions and pipe strength and, aid our understanding of the essential physical process of pipe failure, they are very expensive to use for prioritizing water pipeline renewal, since they often require detailed information and elaborate numerical modeling. However, simple empirical or mathematical statistical models for estimating the pipe break occurrence rate could be adopted for such a purpose. Besides using a mathematical statistical model for break occurrence, cost models must also be developed if prioritization is to be based on cost/benefit analysis or economic efficiency. The break occurrence rate can be incorporated in stochastic processes in modeling the temporal variation of break occurrences and their associated probability, which will be discussed in more detail in the following sections.

Statistics of some of the cost information for Canadian cities have been given by Zhao and Rajani (2002), and Rahman et al. (2005). Najafi et al. (2005) also provided information for cost associated with trenchless and open-cut methods for pipeline systems, and social cost.

The integrated use of the stochastic model for break occurrence modeling and cost associated with reconstruction, repair and rehabilitation for pipeline systems has been proposed in the literature to facilitate the selection of optimum replacement strategies (e.g., Shamir and Howard, 1979; Kleiner et al., 2001; Loganathan et al., 2002; Hong et al., 2006). Most of these methods consider that the optimum decision under uncertainty can be carried out based on the minimization of the total cost that includes the cost of repair and replacement during a preselected planning period. However, use of optimum replacement time obtained by minimizing the expected total cost (at present value) may not lead to the minimum annual average cost during the service period. As indicated by Hong et al. (2006), a solution that minimizes the total cost might not provide the maximum benefit in terms of minimizing the cost (at present value) per unit service period (or time) (i.e., maximizing the service period per dollar spent).

The objective of the current chapter is to provide an overview as well as justification of the adopted pipeline break occurrence model, the cost for repair and replacement, and formulation for selecting the optimum prioritization scheme for water distribution systems. Formulation for selecting the optimum scheduling of replacement and rehabilitation of water distribution system presented in this chapter and that is implemented in the following chapter will be based on both the commonly used minimum expected life cycle cost criteria and the minimum expected annual average cost criterion. The adopted model, cost information and decision criteria are to be implemented in the next chapter.

2.2 Failure or break occurrence modeling

2.2.1 Break occurrence rate

It is intuitively true that one of the most important parameter that affects the selection of the optimum scheduling of replacement and rehabilitation time for water distribution system is the pipeline break occurrence rate. Many studies focused on the break occurrence rate have been reported in the literature (see e.g., Muhlbauer, 2004; Kleiner and Rajani, 2001; Rajani and Kleiner, 2001). In general, this rate is affected by many variables including the pipeline age, pipe material and geometry, and the surrounding soil condition. These models could be classified as index based models, statistical models, and physical models.

Use of index based models is extensively discussed and followed by Muhlbauer (2004). This type of model often uses scoring methods to assess the condition of the water pipelines, although justification of scales used for scoring is rarely given. In the model, there are sets of factors affecting the pipeline's failure rate or probability. Each factor is assigned to a numerical value and weighted according to its importance of the contribution to the failure. Those factors may include pipe property, soil property, coating and cathodic protection applications, inspection plan, external and internal loadings, and operations and maintenance of the pipeline. The number of factors is varied in different indexing models, ranging from simple models with one or two factors to models with hundreds of factors. Pipes with scores below the different selected target levels are good candidates for renewal, inspection and repair. However, pipes falling into the same score scope cannot be further prioritized and the predictive capability of the model is limited (Loganathan et al., 2002).

Further information on the index based model can be found in Maulbauer (2004) wherein a model with more than one hundred factors for general pipelines is discussed. These indices are grouped into the indexes of third-party damage, corrosion, design, incorrect operation and leak impact factor. This makes the indexing model very simple to use but extremely subjective. Therefore, the index based model is not considered further in the present study.

In general, physical models are developed based on the strength or residual strength of a pipeline and considering the applied external loads (external and/or internal loads) to predict the failure of the pipeline. Residual strength is the resistance or capacity of buried pipes with degrading geometry, and is a function of the material properties and the original wall thickness of the pipe. It decreases over time because of the thinning of pipe walls or deterioration of pipe walls and the growth of internal and external corrosion pits. Doyle et al. (2003) showed that buried cast iron water pipes surrounded by low resistivity soil are prone to external corrosion, while the application of coating or cathodic protection slows down the rate of deterioration. This is expected since the deterioration including the corrosion pit growth depends on the soil properties around the pipe.

For example, for a steel pipe under high pressure having a defect with maximum defect depth, d, and the total length of defect, l, the widely used equation for predicting the remaining pressure strength $p_C(d,l)$ given in ASME B31G (1991) is,

$$p_C(d,l) = 1.1\frac{h\sigma_y}{r}\left(\frac{1-(2/3)d/h}{1-(2/3)d/(hm(l))}\right) \qquad \text{if } m(l) \le 4.1 \qquad (2.1a)$$

where σ_y is the specified minimum yield strength of steel, h is the nominal pipe wall thickness, r is the pipe outside radius, and

$$p_C(d,l) = 1.1\frac{h\sigma_y}{r}(1-d/h) \qquad \text{if } m(l) > 4.1 \qquad (2.1b)$$

and,

$$m(l) = \sqrt{1+0.8\frac{l^2}{2rh}} \qquad (2.1c)$$

is known as the two term Folias factor. The maximum defect depth, d, and the total length of defect, l, increase with time.

Rajani and Makar (2000) investigated the strength of cast iron pipe by considering the mechanical properties of the pipe and soil properties that were obtained from experiments of exhumed cast iron water pipe samples, with or without corrosion pits, and surrounding soil samples collected from sixteen cities in Canada and US. They suggested an empirical equation for the normal tensile strength of cast iron pipes that takes into account the pit dimensions at which fracture takes place:

$$\sigma_n = \frac{\alpha K_q}{\beta\left[(d/h)\sqrt{a_n}\right]^s} \qquad (2.2a)$$

where σ_n is the normal tensile stress; K_q is the fracture toughness; β is the geometric factor; (d/h) is the pit depth and pipe wall thickness ratio; a_n is the lateral dimension of corrosion pit; α and s are the model constants. They also suggested a upper bound, β^{upper}, and a lower bound, β^{lower}, for β based on empirical data. These bounds are given by,

$$\beta^{upper} = 0.5(d/h)^{-0.3} \qquad (2.2b)$$

and,

$$\beta^{lower} = 0.3(d/h)^{-0.2} \tag{2.2c}$$

Furthermore, when (d/h) is greater than 0.6, the value of β should not exceed 0.4.

In order to obtain the residual tensile strength of the cast iron pipe at service time *T*, a corrosion pit dimensions growth model was also suggested for predicting the pit depth, *d*, and pit width, a_n:

$$P_c = aT + b\left(1 - e^{-cT}\right) \tag{2.3}$$

where P_c is the pit dimension; and *a*, *b*, and *c* are the model constants.

The residual strength is used to compare with the total axial stress or circumferential tensile stress in order to assess the pipe safety. Note that the total stress due to internal and external loads may be time-dependent or time-independent. The external loads including the time-independent statistic loads (e.g., earth load) and time-dependent dynamic loads (e.g., frost load and thermal load), and the internal loads due to the internal fluid pressure and surge pressure also affect the safety of pipeline. By considering the circumferential tensile stress, σ_θ due to external soil pressure, σ_S and traffic loads, σ_V, which are described by Spangler and Handy (1982) models, and the internal fluid pressure, σ_F, in the loading model for the underground pressured pipelines, Ahammed and Melchers (1994) concluded that σ_θ can be calculated from,

$$\sigma_\theta = \sigma_S + \sigma_V + \sigma_F = \frac{3K_m \gamma B_d^2 C_d E_p hD}{E_p h + 3K_d pD^3} + \frac{3K_m I_c C_t E_p hD}{L\left(E_p h + 3K_d pD^3\right)} + \frac{pD}{2h} \tag{2.4}$$

where K_m is the bending moment coefficient, γ is the unit weight of soil, B_d is the width of ditch, C_d is the calculation coefficient, E_p is the pipe material elastic modulus,

h is the pipe wall thickness, D is the nominal pipe diameter, K_d is the deflection coefficient, p is the internal pipe pressure, I_c is the impact factor, C_t is the surface load coefficient, and L is the pipe effective length.

If the depth of corrosion is assumed to be governed by the power law, $D(T) = kT^n$, where $D(T)$ is a decreasing function of T, and k and n are regression parameters, as the wall thickness reduced, the cross section area reduced, and the circumferential tensile stress, $\sigma_\theta(T)$, increased over time T with the same subjected loads (i.e. substitute h with $h - kT^n$ in Eq (2.4)). In such a case, a comparison of the material yield stress of the pipe σ_y, and $\sigma_\theta(T)$ can be used to assess the safety of the pipe. Failure occurs if σ_y is less than $\sigma_\theta(T)$. The remaining service time of the pipe can then also be evaluated by subtracting the age of the pipe from the time of failure.

It must be emphasized that this mentioned failure assessment procedure ignores the uncertainty in and bias associated with the calculated strength and load effects. However, if the uncertainty in strength or remaining strength and load effects is considered, their probabilistic models must be developed, and well-known reliability analysis methods such as simulation technique, the first-order reliability method, or second-order reliability method (Madsen et al., 1986) could be employed in estimating the probability of safety or reliability. Examples of such reliability analyses using the remaining strength model shown in Eqs. (2.1) to (2.3) for assessing the safety of the deteriorating pipe have been reported in the literature (Hong, 1997, 1999; Sadiq et al., 2004). However, such detailed assessment of the safety of a pipeline section considering all possible pipe defects and/or corrosion pits is a time consuming task since there are many corrosion pits or defects in a pipeline sections as well as their probabilistic models need to be considered. Hence, this approach is not adopted in the present study.

Besides the index based model and physical based model, empirical models inferred from results of statistical analysis have also been extensively employed (Shamir and Howard, 1979; Walski and Pelliccia, 1982; Kleiner and Rajani, 1999; Pelletier et al., 2003). The popularity of these models is partly because the required data for physical models are costly to obtain and often not available, and partly because the statistical models can cope with data of various levels of detail, and they are simpler to use. Statistical models predict the water main breaks by identifying breakage patterns from the available historical pipe failure data. They can incorporate statistical information from pipelines of similar vintage, operating conditions and material properties.

Perhaps, the most commonly used model for predicting the break occurrence is the one proposed by Shamir and Howard (1979). This model considers that the break occurrence increases with time, and the proposed mathematical form is as follows,

$$\lambda(\tau|t_0) = \lambda_0 \exp(A(\tau + t_0)) \tag{2.5}$$

where $\lambda(\tau|t_0)$ (breaks/yr/km) is the break occurrence rate in a pipeline segment at time τ as shown in Figure 2.1, λ_0 represents the break occurrence rate for a new pipeline segment that is just commissioned for service, t_0 is the duration of already sustained service period with necessary maintenance repairs, and A is the model coefficient to be determined through regression analysis. This model presents the exponential relationship between the breaks and the age of the pipe group and was adopted by several researchers for prioritizing rehabilitation and repair schemes (Kleiner and Rajani 1999; Hong et al. 2006).

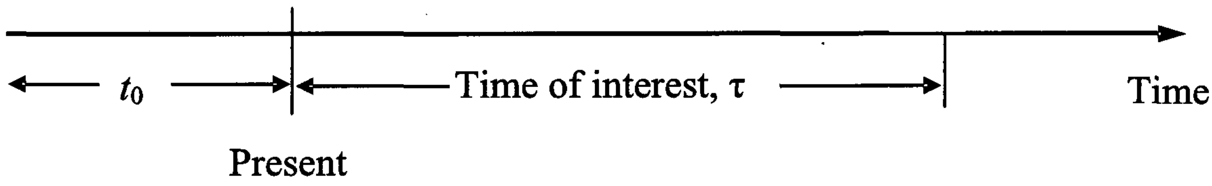


Figure 2.1. Time line for Eq. (2.5)

Kleiner and Rajani (1999) discussed extensively a set of collected historical water main break data that are grouped into homogeneous subsets. More specifically, the data for 2,430 water main segments with total length of 298 km are considered. These data were collected from one region for 24 years, from 1973 to 1996, where the data from other regions were considered insufficient since they were collected only in a 4-year period. The model parameters A and λ_0 that depend on the soil classification and pipe geometry can be estimated using the statistics given by Kleiner and Rajani (1999). The obtained values are shown in Table 2.1.

The simplicity of the exponential model of Shamir and Howard (1979) makes it easy to implement, however, it needs a careful classification on grouping the model data homogeneously in the application of this model. Furthermore, since the water main breaks within a group are assumed to be uniformly distributed, the location of a particular failure cannot be identified and a larger scale of inspection may be needed for locating the failure among the water mains in that group.

Table 2.1. Predicted break occurrence rates

Group (actual installation period) for different soil type	Assumed Installation time	A	λ_0
Group A (Before 1929)			
Clay	1920	0.188	1.42E-7
Sand	1920	0.018	0.0162
Group B (1930-1949)			
All soil types	1940	0.176	1.2E-5
Group C (1950-1959)			
Clay	1955	0.024	0.0565
Sand	1955	0.044	0.0362
Group D (1960-1979)			
Clay	1970	0.104	0.0161
Sand	1970	0.028	0.1232
Group E (1980-1996)			
All soil types	1990	0.144	0.1041

It must be emphasized that the data given by Kleiner and Rajani (1999) were grouped based on intervals defining the installation period. Therefore, in assessing the initial break occurrence rate, a single installation time shown in the table is assumed. Furthermore, the break occurrence rate is affected by the pipe diameter, soil moisture, temperature, and frost penetration. Therefore, if sufficient statistical data on break history is available, a more accurate model discussed in the following could be adopted.

The model proposed by Shamir and Howard (1979) was enhanced by Walski and Pelliccia (1982) by including two additional parameters that are used to scale the rate shown in Eq. (2.5). These factors are introduced to reflect previously observed break occurrence rates and the breakage rate of larger diameter pit cast iron pipes.

Another model for break occurrence rate was proposed by Clark et al. (1982). Their model included a linear equation to predict the time elapsed to the first break and an exponential equation to predict the subsequence breaks. The model has more stringent data requirements, including the consideration of pipe history, breakage history, material, diameter of pipe, pipe operating pressure, soil corrosivity, and zoning composition. The model is more sophisticated and could provide more accurate prediction of the break occurrence rate if historical data for estimating the model coefficients are available. Unfortunately, the data is often too scarce to allow the use of this sophisticated model.

Note that other mathematical forms for predicting the break occurrence rate $\lambda(\tau)$ include the one proposed by Loganathan et al. (2002) and the one proposed by Pelletier et al (2003). The former can be expressed as,

$$\lambda(\tau) = \gamma_0 \delta \tau^{\delta-1} \quad (2.6)$$

where γ_0 and δ are model parameters to be determined using historical break occurrence data and regression analysis. This mathematical form (Eq. 2.6) is related to the Weibull distribution. The mathematical form proposed by Pelletier et al (2003) is based on the assumption that the time interval between the pipe installation and its first break has different statistical time distribution from the time intervals between the subsequent breaks. They assumed the hazard function of the former behaved as Weibull failure rate function, $\lambda(\tau)=k_1 p(k_1 \tau)^{p-1}$, and the latter were exponential failure rate function, $\lambda(\tau)=k_2$, where k_1, p and k_2 are model parameters. The model was applied to three municipalities in Quebec. With the obtained limited records, updated until 1996, of pipe breaks, pipe lengths and diameters, and year of the pipe installation, for each municipality, they segmented the pipes into average length and periods of urbanization associated with the year of installation of the pipe. These data were then used for the simulation of the annual pipe breaks before 1996, and the prediction of the numbers of annual average pipe breaks after 1996. However, it is not clear whether this model is applicable to other regions. It is of interest to note that Pelletier et al (2003) observed that the municipality would have high breakage rate for pipe laid in the period of rapid urban growth. This could have economical and social impacts and deserve more scrutiny if extensive data from other municipalities become available.

One more model that has been discussed in the statistical literature is the so-called proportional hazard model proposed by Cox (1972) and adopted by Andreou et al. (1987a, 1987b) for the modeling of the break occurrence rate. The model can be expressed as

$$h(\tau, \mathbf{Z}) = h_0(\tau)\exp(\mathbf{b}^T\mathbf{Z}) \tag{2.7}$$

where $h(\tau,\mathbf{Z})$ is the hazard function, representing the instantaneous break occurrence rate, $h_0(\tau)$ is a arbitrary base line function, $\mathbf{Z}$ is a vector of covariates and $\mathbf{b}$ is a vector of coefficients to be estimated using available historical break occurrence data. The model is extremely general and flexible. In fact, one may consider that the model proposed by Shamir and Howard (1979) and shown in Eq. (2.5) is a particular case of this model.

2.2.2 Stochastic model

Use of a break occurrence rate model alone such as Eq. (2.5) does not provide the complete probabilistic model for assessing pipeline breaks. To complete the model, one could consider that the break occurrence is a non-homogeneous Poisson process with occurrence rate shown in Eq. (2.5) (Hong et al., 2006). The use of the non-homogeneous Poisson model is also considered by Constantine and Darroch (1993) (see Kleiner and Rajani, 2001) but with a different break occurrence rate model. The non-homogenous Poisson model is adopted for the system described herein. The model can be used to estimate the time to failure probability and incorporated in the consequence and risk analysis. The selection of this model is partly based on the consideration of the "Principle of Consistent Crudeness" which was advocated by Elms (1985), indicating that the quality of the results of the system is dominated by the "crudest" input to the system, and based on the fact that both cost information and more detailed break histories are often difficult or impossible to obtain. As will be seen, the information on cost of inspection, rehabilitation, and maintenance is scarce, crude, and may be solely qualitative.

For completeness and easy reference, some of the relevant information on the non-Poisson process for the break occurrence modeling given in Hong et al. (2006) is summarized and described as follows. For this purpose, let $N(t)$ denote the occurrence of the number of breaks in a pipeline segment during a service period t. Consider that $N(t)$ can be treated as a non-homogeneous Poisson process since the age of the pipeline segment affects the break occurrence rate as shown in the previous section. Note that the difference between the occurrence rate for a homogeneous and a non-homogeneous Poisson process is a constant for the former, while it varies with time for a non-homogeneous Poisson process. For a non-homogeneous Poisson process the probability of occurrence of n breaks within (0,t], $P(N(t) = n)$, is given by the following equation (Parzen 1962),

$$P(N(t) = n) = \frac{1}{n!}\left(\int_0^t v(\tau|t_0)d\tau\right)^n \exp\left(-\int_0^t v(\tau|t_0)d\tau\right) \tag{2.8}$$

where $v(\tau|t_0)$ is the occurrence rate (such as that described in Eq. (2.5)), t_0 indicates that the pipeline segment has already been in service for t_0 years with necessary repairs and maintenance, the time t represents the future service period (i.e., the measure of t starts at present). Define,

$$s = V(t) = \int_0^t v(\tau|t_0)d\tau . \quad (2.9)$$

It can be shown (Parzen 1962) that $N_S(s) = N\left(V^{-1}(t)\right)$ is a normalized homogeneous Poisson process with an occurrence rate equal to one, and $V^{-1}(\bullet)$ denotes the inverse transformation of $V(\bullet)$. The probability of n breaks within (0,s], $P(N_S(s) = n)$, is given by,

$$P(N_S(s) = n) = (s)^n \exp(-s)/n! \quad (2.10)$$

In short, in this study, it is considered that the break occurrence follows the non-homogeneous Poisson process with occurrence rate given by Eq. (2.5) (i.e., $v(t|t_0) = \lambda(t|t_0)$). This model is to be incorporated in estimating the expected life cycle cost for selecting the optimum prioritization scheme for rehabilitating and/or repairing the water distribution systems.

Other stochastic models that could be used to model the failure of pipelines include the semi-Markov Process (Li and Haime, 1992) and the homogeneous and non-homogeneous Markov Processes (Hong, 1999). Although these models are useful, they are outside of the scope of the present study and therefore, they are not considered.

2.3 Maintenance and repair cost

Before the formulation of selecting optimum prioritization for rehabilitation/repair can be formulated and carried out, cost information must be collected. The cost information that is required includes the cost of construction, repair and damage.

Conventional construction costs include cost of materials, equipment, labour and administration fees, and municipal organizations may have their own cost manual for conventional construction methods. For example, in the study of Kleiner and Rajani (1999), the conventional construction cost information of replacement and failure is given by a utility with value of $442/m and $5,780/event, respectively. Unfortunately, the cost for rehabilitation and maintenance of water pipe distribution systems is scarce.

Due to the increase in construction projects that are taken place in both the above and below ground of congested urban environments, ideally, engineers should not only account for the total direct cost of a construction project, but also consider the social costs in order to reflect the disruptions and damages to the surface and subsurface activities. An effort has been made by Zhao and Rajani (2002), and Rahman et al. (2005) to assess the cost associated with the rehabilitation of buried pipes as being project dependent.

Zhao and Rajani (2002) compiled cost information from the literature focused on different trenchless technologies such as cure-in-place pipe (CIPP), horizontal directional drilling (HDD), slipping, relining, microtunneling, tunneling, pipe bursting, pipe jacking, and open-cut. A summary of those costs is shown in Table 2.2. The cost data are grouped into overall average cost (dollar per millimeter diameter per linear meter length) and costs in pipe diameter ranges (dollar per linear meter length). Note that all cost values are converted to the dollar value in 2006. In the overall average cost category, microtunneling is the most expensive ($12.32/mm/m). The overall average costs of microtunneling and pipe jacking ($5.55/mm/m) are even more expensive than the open-cut method ($4.98/mm/m). The relining method is the least expensive ($1.23/mm/m).

Table 2.2 Rehabilitation and replacement cost (after Zhao and Rajani, 2002)

Method	Overall average cost ($/mm dia./m length)	Diameter range (mm)				Categories
		Small (<=300) ($/m)	Medium (330-940) ($/m)	Large (960-1830) ($/m)	Very Large (>1830) ($/m)	
CIPP	1.79	387	687	3,433	-	Structural Lining
Sliplining	1.79	299	1,278	3,158	3,321	Structural Lining
Relining	1.23	382	-	-	-	Non-structural Lining
Microtunneling	12.32	3,382	6,171	19,921	60,669	Trenchless Replacement
Tunneling	4.84	-	2,538	9,176	10,309	Trenchless Replacement
HDD	3.84	343	2,317	8,071	-	Trenchless Replacement
Pipe Bursting	2.85	939	1,507	-	-	Trenchless Replacement
Pipe Jacking	5.55	-	-	9,754	12,309	Trenchless Replacement
Open-cut	4.98	788	2,993	2,878	-	Traditional Replacement

Costs are expressed in 2006

The results of the analysis by Zhao and Rajani (2002) suggested that the cost of trenchless rehabilitation or construction increase when the pipeline diameter is increased, however, when pipeline with diameter larger than 960mm, none of the trenchless methods is more economical than the method of open-cut. They also suggested that the cost of emergency repair is about three times the normal repair cost.

Zhao and Rajani (2002) also provided information on rehabilitation cost by using different rehabilitation or maintenance techniques and costs of some inspection methods. The inspection cost given by them is summarized and shown in Table 2.3.

Table 2.3 Inspection cost (after Zhao and Rajani, 2002)

Inspection method	Cost (2006)	Average Cost (2006)
CCTV	\$2.6-\$13/m	\$8/m
CCTV with Sonar	\$9-\$13/m	\$11/m
Person-entry	\$2.6-\$26/m	\$14/m
Rotary sonic device	\$19.3-\$24.2/m	\$22/m

The study of Rahman et al. (2005) was focused on the social cost due to infrastructure rehabilitation and maintenance actions. They classified the social cost into three categories, namely, direct social cost, indirect social cost, and intangible social cost. Direct social costs are the contractual costs including the preconstruction, planning and engineering and construction costs; indirect social costs are the hidden costs that indirectly affect the surface users (i.e. the road user, the business owner, and the residents) in economical ways (i.e. traffic disruptions, business loss); and the intangible social costs are the monetary value of the damage to the environment, and the health and safety (i.e. pollution, contamination) of the surface users. They also indicated that the social cost could be for up to 4 times of the construction cost on certain projects. Due to the minimal surface disruption when applying the trenchless construction methods, their social cost ratio is usually smaller than that of the traditional open-cut construction. Najafi et al. (2005) suggested that the social costs for open-cut construction can be as high as several times of the direct construction costs, while the social cost for the trenchless methods can be as low as around 5 percent of the direct construction costs. In order to account for the social costs, additional data, such as parking meter distribution, are required. As such data are not available, the social costs could be taken as the ratio of the direct construction costs to social costs as 1:1 for traditional open-cut methods and 1:0.05 for trenchless methods.

Table 2.4 Cost factors (after Najafi et al. 2005)

Cost Type	Cost Factors	Open-cut	Trenchless
Direct Costs	Planning, equipment, labour, site reinstatement	Major	Minor
	Design, pipe material, subsurface investigation	Minor	Major to minor
Social Costs	Traffic disruption, business loss, damage to environment and public health and safety	Major	Minor

Note that Najafi et al. (2005) also provided a qualitative comparison of the cost factors for trenchless and open-cut methods that are given by others. Table 2.4 presents some of the cost factors for trenchless and open-cut methods given by Najafi et al. (2005).

In short, the above mentioned three relatively extensive studies simply indicate that the cost information is very hard to obtain and failure to account social cost may result in making poor decisions for selecting optimum prioritization for rehabilitation/repair for water pipelines. Note that rather than using the actual cost, one could use of the ratio of the cost of repair to the cost of replacement in assessing the the preferred or optimum time for replacing and rehabilitating the existing water distribution system.

2.4 Criteria and Decision making under uncertainty

If the life cycle cost is not considered, perhaps the simple criterion that one could use for prioritizing water pipe renewal is a tolerable failure probability level, P_{fT}. For the adopted break occurrence model (i.e., the non-homogeneous Poisson model with break occurrence rate shown in Eq. (2.5)), the probability that there is at least one break, $P(\tau)$, before the end of a future time τ for a pipe that has already been in service for t_0 years, is given by,

$$P(\tau) = 1 - \exp\left(-\lambda(\tau|t_0)\right) = 1 - \exp\left(-\lambda_0 \exp(A(\tau + t_0))\right) \tag{2.11}$$

where the parameters A and λ_0 depend on the surrounding soil condition and the pipe material and geometry. Therefore, for a given water distribution system, one could calculate $P(\tau)$ for each pipe segment for a given future year τ, and display the probabilities $P(\tau)$ obtained for the whole water distribution system. Visualization methods could then be used to display critical segments defined as those segments whose $P(\tau)$ is greater than the tolerable level P_{fT}.

It is noted that many of these studies consider that the optimum prioritization scheme can be carried out by maximizing the expected benefit or minimizing the expected life cycle cost (e.g., Shamir and Howard, 1979; Andreou et al., 1987a, 1987b; Kleiner, 2001; Loganathan et al., 2002; Hong et al., 2006). The use of maximum expected value of benefit or minimum expected cost as criterion in decision making under uncertainty could be adequate if a decision maker is risk-neutral. Note that the minimum expected life cycle cost criterion does not consider the magnitude of the scatter associated with the estimated life cycle that could be characterized by using its variance or higher statistical moments. To overcome this, one could use the expected utility theory for decision making (von Neumann and Morgenstern, 1943). Use of the expected utility theory can cope with preferences of different decision makers since the selected utility function reflects the decision makers' risk attitudes such as risk-neutral, risk-averse and risk-seeking. Unfortunately, since the adopted utility function may not be universally accepted, use of a specific utility function for decision making may not be accepted by others with different risk preference. This shows that different decision criteria and rules can be adopted for making decisions, and each criterion is associated with its own arguable weakness. Therefore, for simplicity, the commonly used minimum expected cost criterion is adopted in the present study.

Note that as mentioned earlier, rather than minimizing the total cost that includes the cost of repair and replacement (at present value) during a preselected planning period to optimally prioritize rehabilitation and replacement needs, one could select the optimum prioritization scheme based on the minimum expected annual average cost (at present value) criterion during the service period. This is because a decision that minimizes the total cost might not provide the maximum benefit in terms of minimizing the cost (at present value) per unit service period (or time). Therefore, both minimization of the expected total and minimization of expected annual average cost during a predetermined service period (or time horizon) selected by the decision maker(s) are considered in the present study.

In order to find the optimum prioritization scheme, consider that one is interested in finding the expected total cost, $E(C_T(t))$, given that replacement of the pipeline segment is scheduled at T_{op}, and that repairs are carried out for the future service period t whenever break occurs (see Fig. 2.2).

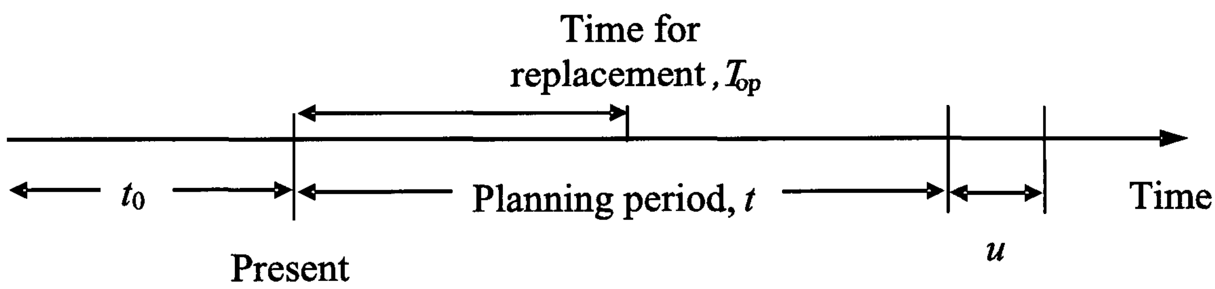


Figure 2.2. Illustration of the time line associated with pipe service period

By assuming that, after a replacement, the occurrence of the breaks for the pipeline segment follows a non-homogeneous Poisson process, it can be shown (Hong et al., 2006) that $E(C_T(T))$ with some simplification can be expressed as,

$$E(C_T(T)) = m_{C_R}\lambda_0\left\{\frac{e^{At_0}}{A-\gamma}\left(e^{(A-\gamma)T}-1\right)+\frac{m_{C_N}e^{-\gamma T}}{m_{C_R}\lambda_0}+\frac{e^{-\gamma T}}{A-\gamma}\left(e^{(A-\gamma)(t-T)}-1\right)\right\} \tag{2.12}$$

where m_{C_N} and m_{C_R} denote the expected cost for replacement and repair, and γ is the discount rate.

By considering the total cost criteria, the optimum replacement time T_{op} is the one that minimizes the expected life cycle cost, $E(C_T(t))$. The minimization of Eq. (2.12) can be carried out using one of the many efficient optimization algorithms found in the literature (e.g., Schittkowski, 1980; Fattler et al., 1982; Reklaitis et al., 1983). In this study, a FORTRAN subroutine called NLPQL, which implements the sequential quadratic programming (SQP) method for solving the nonlinearly constrained optimization problems presented by Schittkowski (1985), is adopted, compiled as a DLL (dynamic link library) and is included in the macro to be described in the following chapter. For the theoretical basis and details of implementation of NLPQL, the readers are referred to Schittkowski (1985). An illustration of the optimum T_{op} obtained in this way is shown in Figure 2.3.

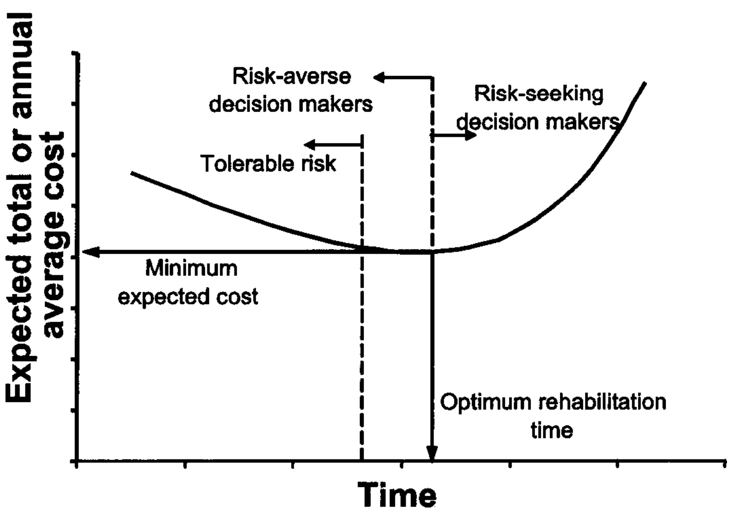


Figure 2.3. Illustration of optimum rehabilitation time (the position of the tolerable risk shown in the figure is selected arbitrarily).

Assuming the planning horizon T equals T_{op} (Shamir and Howard 1979) leads to,

$$T_{op} = \max\left(0,\ -t_0 + \frac{1}{A}\ln\left(\frac{m_{C_N}\gamma}{m_{C_R}\lambda_0}\right)\right) \tag{2.13}$$

where if T_{op} is equal to zero, this simply indicates that repair or replacement is long overdue. The corresponding critical break occurrence rate is obtained by substituting Eq. (2.13) into Eq. (2.5), yielding the critical occurrence rate,

$$\lambda(T_{op}|t_0) = (m_{C_N}\gamma)/m_{C_R} \tag{2.14}$$

for $T_{op} > 0$, and $\lambda(T_{op}|t_0)$ equals $\lambda_0 \exp(At_0)$ otherwise. Clearly, if the actual break occurrence rate of the pipe segment is greater than or equal to this rate, replacement is preferred. Therefore, the ratio of the actual to the critical break occurrence rate, R, can be used as an indicator for prioritizing the pipeline rehabilitation if the statistics of the actual break occurrence rates are available.

If the expected annual average cost instead of using the expected total cost is considered, Eq. (2.12) is replaced by,

$$E(C_T(T))/t = \frac{m_{C_R}\lambda_0}{t_0 + T_{op}}\left\{\frac{e^{At_0}}{A-\gamma}\left(e^{(A-\gamma)T}-1\right)+\frac{m_{C_N}e^{-\gamma T}}{m_{C_R}\lambda_0}+\frac{e^{-\gamma T}}{A-\gamma}\left(e^{(A-\gamma)t_0}-1\right)\right\} \tag{2.15}$$

and similar procedure as before is followed to obtain the optimum replacement time T_{op}.

2.5 Summary

Based on the previous discussion, the break occurrence is modeled as a non-homogeneous Poisson process with an occurrence rate given Eq. (2.5) and the model parameter shown in Table 2.1 in the present study. Since detailed cost information is not available and the cost could be project specific, therefore, this information could be considered as user defined information.

For the selection of the optimum replacement time T_{op}, equations shown in Eqs. (2.12), (2.13) and (2.15) will be implemented and integrated with the geographic information system.

Chapter 3 Integration of geographic information system and probabilistic analysis to prioritize pipeline renewal

3.1 Introduction

Infrastructure systems such as water distribution systems, transportation systems, transmission line systems are spatially distributed and are usually large and complex. To manage such a complex system, one needs many layers of information. For example, for a water distribution system, one needs to know the soil conditions surrounding the pipeline network, the geometric and material properties of the pipe segments, and the surface conditions of the area or region of interest which can be obtained from the aerial photos. The layers of information are most easily visualized and can be manipulated by using a Geographic Information System (GIS).

A GIS is a system of hardware, software, and procedures specially designed to collect, maintain, maniputlate, analyze, display and distribute spatially referenced data and information for identifying and solving spatially related problems or for complex plannings (Lo and Yeung, 2007).

As our population grows, the infrastructure problems become more and more complex, and the use of the conventional methods to deal with the data management and handling has become a tedious and error-prone task. This could be alleviated if the GIS is integrated with civil engineering applications in dealing with data collection and representation, spatial functionalities, and illustration. For example, Doyle et al. (2003) adopted the GIS to display the collected and analyzed data of the maximum external average pitting rate in the district of Etobicoke and Toronto. In their study, three feature layers were used: the geocoded centre-line street map of the districts of Etobicoke and Toronto, the locations with the collected sample data of soil properties and the conditions of water main due to external corrosion, and the surficial soil type map. From the overlay of street map and water main condition map, it was shown that Etobicoke had higher corrosion rate than those of in Toronto. By comparing the soil property map which showed that Etobicoke had lower soil resistivity from the collected soil samples than those in Toronto, Doyle et al. (2003) suggested that areas with lower soil resistivity such as Etobicoke potentially had higher risk of external corrosion of water mains than those located in higher soil resistivity areas such as Toronto district.

A GIS can be used as a platform to develop a spatial decision support system (SDSS) or a decision support system (DSS). A DSS is a computerized information system to provide spatially oriented information to the decision makers to solve problems and make decisions. For an example, Sinske and Sietsman (2004) used a GIS as a supplement with an object query browser to establish a SDSS for analysis of the pipe break susceptibility, due to pipe age, air-pocket formation and pipe damaged by tree-roots, for municipal water distribution systems. In the SDSS, the object query browser is applied for the data operations, data queries, input configurations and also provides access to the GIS and a link to a fuzzy modeling program. The GIS is used for spatial queries, spatial selection and thematic mapping for the results to present a more specified output for the SDSS analyst, administrator and decision maker. Furturemore, a DSS, PARMS-PRIORITY, supplemented with a GIS, MapObject LT, was developed by Moglia et al. (2006) to help water utility companies making decisions on the prioritization of pipe replacement, pressure reduction or shut-off valve insertion. In this DSS, the GIS is used to display selected pipe segments and highlight them in a pipeline network. A user can also add new pipes into the pipeline network through the functionalities provided by this GIS. However, these studies are mainly focused on exploiting the built in query and display functionality of the GIS, and did not take to full advantage of a GIS, such as fully incorporate detailed engineering analysis and probabilistic analysis for decision making under uncertainty with the GIS.

As mentioned in Chapter 2, the combination of the stochastic modeling of the break occurrence and the cost model, together with decision rules, form the basis for the development of a decision support system for water pipeline renewal. Reviews of some of these systems, including those developed by Herz (1998), Kleiner et al. (1998a,b), Hadzilacos et al. (2000), Burn et al. (2003) and Moglia et al. (2006) are given in Stone et al. (2002) and Moglia et al. (2006). More recently, a system developed for modeling the deterioration rate of water distribution systems and their renewal planning has been described by Kleiner et al. (2007). Some of these systems focus on long term planning, while others concentrate on prioritization, each with a different level of sophistication, objectives, and targeted users.

The main objective of this chapter is to describe the development and integration of the powerful GIS software (i.e., ArcView 9.1 by ESRI) with a probabilistic model and decision rules, that are described in Chapter 2 for selecting preferred decisions to prioritize water pipeline renewal. Note that ArcView is a GIS software for visualizing, managing, creating, and analyzing geographic data. This development allows users to adopt probabilistic models, economical criteria, and/or decision rules that are not available in the afore-mentioned systems found in the literature.

Details concerning some basic information on GIS, the implementation of the probabilistic decision analysis described in Chapter 2 using the VBA, and its integration with ArcView are described hereafter. The implementation includes the incorporation of Dynamic Link Library (DLL) compiled under FORTRAN for optimization, the access of the data layers in the GIS system for probabilistic analysis, and the development of a simple user interface. The system is illustrated with the pipe network from the City of Hamilton.

3.2 Some basic information concerning GIS and its application for managing pipeline systems

According to Lo and Yeung (2007), the term GIS was used when the Canada Geographic Information System (CGIS) was developed by Roger Tomlinson, head of the Canada Land Invent ory (CLI), and IBM in the early 1960s in Ottawa, Ontario, Canada. At that time CGIS was used to analyze Canada's national inventory by creating digital maps showing the spatial data of agriculture, forestry, wildlife, and recreation on them.

The 1960s and 1970s are the formative years of GIS. Several active research and development programs occurred in universities in North America and Europe. Hundreds of software packages for geographic information handling and analysis were producted during these two decades. The development of GIS in these formative years was application driven, and mainly foucsed on map data processing rather than spatial analysis functionality.

The development of GIS was continued by Environmental Systems Research Institute, Inc. (ESRI) and other private sectors. In 1982, ESRI released ArcInfo which wass designed for minicomputers. ArcInfo is one of the first vector-based GIS using the topological data structure to store graphical data, which the attribute data are stored by the relational or tabular data structure. In the 1990s, the development of GIS rapidly grew and the growth of computer technology is one of the main factors. By the end of the 20^{th} century, GIS has been consolidated and standardized on relatively few platforms and data format convertion was required for geospatial data sharing, but this is now achieved through standardization. Recently, more and more open source (as well as proprietary) GIS and object-relational data models have been or are being developed for different operating systems and can be customized for some specified tasks.

Many studies or projects require the use of geographically referenced information, thus one needs to collect such data in order to carry out analysis or just simply use the data to represent the idea in visual format. Since the GIS has been used by governmental organizations, private commercial or non-commercial associations for decades, these organizations or associations have been collecting spatial and non-spatial data. However, many of these databases are only available for internal use or very expensive to obtain, one may need to create his/her own spatial data by digitizing the interested feature from the existing paper maps, scanned maps, aerial photographs, or satellite images and collect his/her own non-spatial data or information. If the data are already stored in digital forms, such as Computer-Aided Design (CAD), vector-based Triangulated Irregular Network (TIN), rasterized Digital Elevation Model (DEM), etc., one can import those data in their GIS with corresponding geo-references. Internet could be another source to gather geo-referenced data. These data include information such as city maps, building information, and addresses, but the obtained information may not guarantee to be accurate or up-to-date. Furthermore, one could also collect data from surveying instruments, such as a Global Positioning System (GPS).

In some cases, the gathered or imported data may require further editing, removing errors, and reorganizing for future uses. For example, the fault, which originally is a dirty mark on a paper map, needed to be removed from the scanned data. Therefore, for most of the GIS projects, the data collection is usually the most time consuming and tedious task.

GIS is often used as a supplement and externally combined with well-designed expert system for implementation of engineering models for water distribution systems. This is partly due to the fact that the engineering model may already have been implemented with existing computer programs or stand alone system. Therefore, sometimes it is simpler to supplement those models with the database capacity and the visualization ability of GIS to implement their spatial referenced data or display the exported in GIS.

In general, there are three basic methods for linking other programs to GIS-based system: loose coupling, tight coupling, and embedded computing system (Lo and Yeung 2007). With loose coupling, GIS and the computer programs work separately. Data and results are manually imported and exported among the GIS and the programs. Furthermore, it may require considerable amount of work and consume a lot of time when data format convertion is requried. In the embedded computing system, the GIS and the other programs share the same memory and a common menu interfacre. However, Goodchild et al. (1993) pointed out that embedded computing system is either too superficial in problem solving or too complex to be developed. Lastly, in the tight coupling linking method, GIS and other programs work together by integrating the other programs into GIS software, or the other way around. Furthermore, data exchange can be fully automatic (Iu et al., 2003) or GIS and other programs in the tight coupling system share the common database (Michener et al., 1994). With the other programs integrated into GIS software, the former works as the extensions or outside modules of the GIS software (Tyler, 2007).

GIS has been broadly used for civil engineering and infrastructure related problems due to its unique spatial database structure, spatial functionalities, illustration capability, and flexibility of integration with other expert systems. However, with costly and limited available data for water main break studies, only a few studies have explored the use of GIS system for managing the water distribution system as indicated in the introduction.

Water distribution system, an infrastructure asset, can be spatially stored in a GIS vector map according to their physical coordination, latitude and longitude, as spatial data. A map may contain layers with different features such as water pipes, values and fittings. Features in each layer are represented as geometric elements. For example, the water main pipes are represented as line elements, and the values and fittings are represented as point elements. Information or descriptions of an element, such as pipe diameter and material, are the non-spatial data and can be stored as attributes in the corresponding attribute tables. By overlaying these related layers, one can obtain a thematic GIS map, such as water main distribution map.

It must be emphasized that the integration of DSS with GIS in the literature is often either a loose integration or an embedded integration (e.g., Raterman et al., 2003; Sinske and Sietsman, 2004). The engineering algorithms (for probabilistic analysis and for selecting optimum decisions) in their study are often not fully integrated in the GIS, rather GIS is simply used as visualization or spatial query tool. For example, engineering algorithms can be implemented in a GIS with its spatial analysis and macro functionalitites and simple external modules can be created and tightly integated to the GIS to supplement the functionality of the GIS. In such cases, not only more accurate results can be obtained, but also the cumbersome data conversion in loose coupling sytems (Tyler, 2007) and complex and expensive development in embedded integation systems (Goodchild et al., 1993) can be eliminated.

3.3 Integration of GIS and probabilistic models

3.3.1 General consideration and data structure

There are two common approaches to be considered in the planning stage of developing the DSS. The first approach is to collect required data for the analysis based on the models we choose; second approach is to choose suitable models based on the available data we have. In this study, the second approach is being adopted due to limited available data, thus, the models which require simpler and fewer data are chosen. However, in either approach, the accuracy, consistency and integrity of the data are very important as they will affect the predicted results directly. A general structure of a GIS database is illustrated in Figure 3.1 for the integrated GIS and probabilistic model for selecting the optimum rehabilitation time.

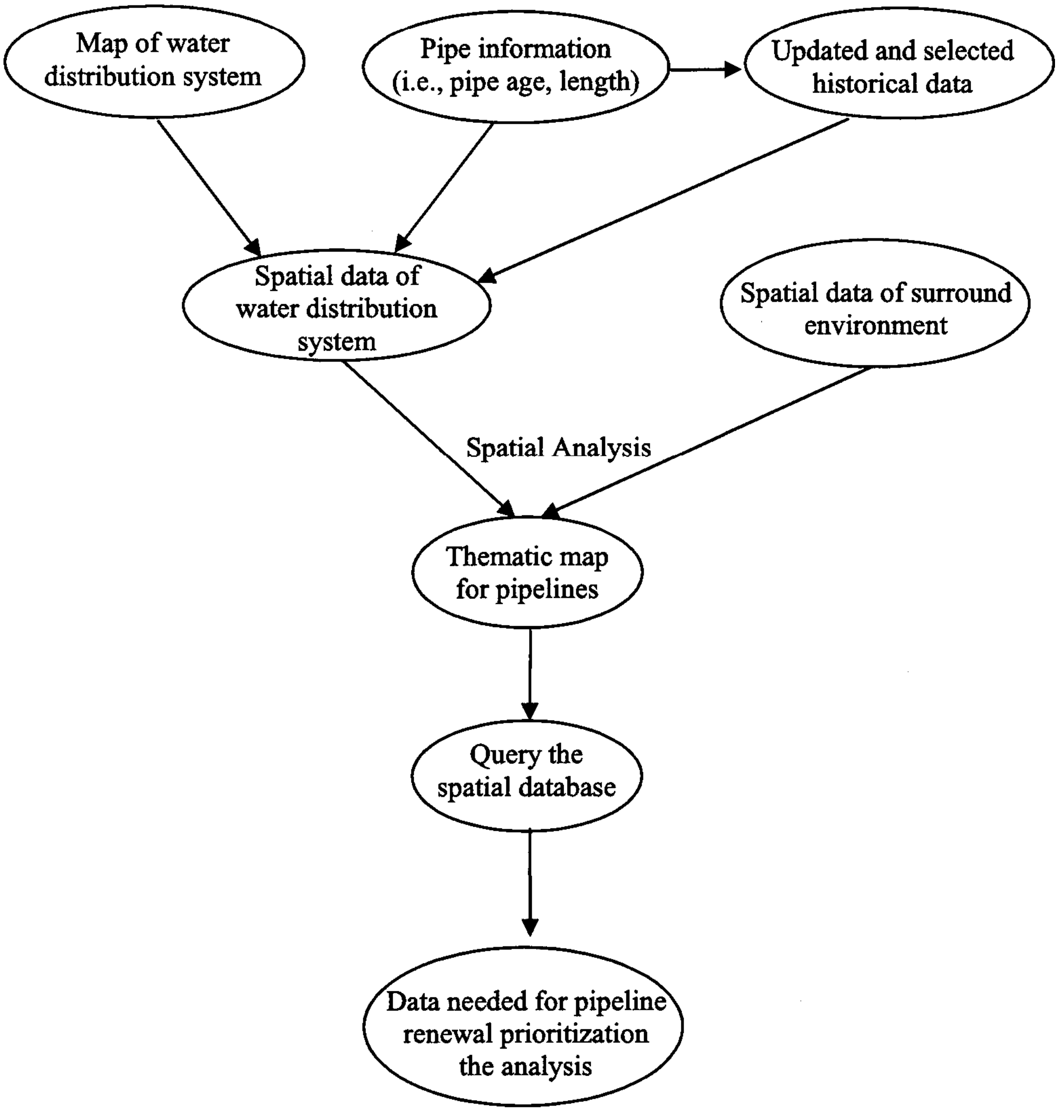


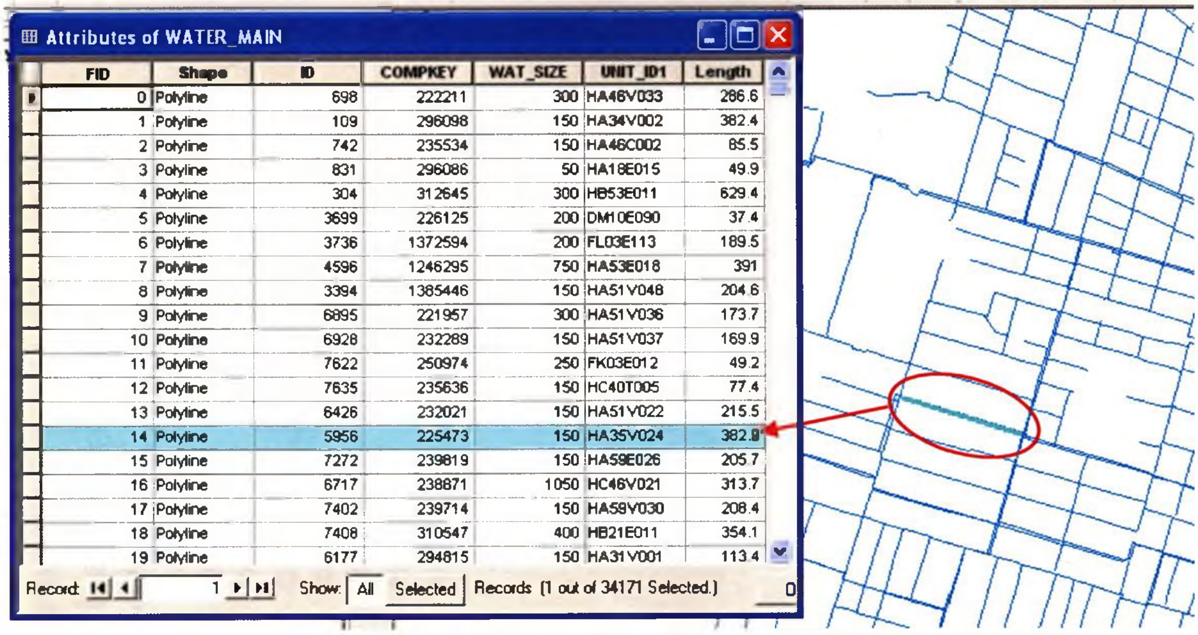
Figure 3.1 General structural of GIS database for the integrated GIS and probabilistic model

The information obtained from the City of Hamilton will be applied in the present study as an illustration of the developed DSS. The data are the maps in digital format known as the shapefiles, which are data sets for the spatial features, and can be directly used in the adopted GIS software. The layer of the shapefile of the water main network or the water main network feature layer (Burke, 2003) of the City of Hamilton is illustrated in Figure 3.2. The figure shows the map of the water main network of the City of

Hamilton by overlapping the feature layer of the water main network and the feature layer of the parcels of the City of Hamilton. Associated pipe information are defined in the attribute table of the feature layer of water main, apportion of which is shown in Figure 3.3.

Attributes, the descriptive data of a spatial feature, are stored in an attribute table of the corresponding shapefile as shown in Figure 3.3. In the WATER_MAIN shapefile, there are 34,171 pipeline segments. Each pipeline segment is a feature (or record) with attributes, namely the unique feature identification (FID), element type (Shape), pipe diameter (WAT_SIZE), pipe length (Length) and other information (ID, COMPKEY and UNIT_ID1). For example, the selected pipeline segment, with FID equals 14, is a 150 mm diameter water pipe and 382.9m in length.

Figure 3.2 Water distribution system from the City of Hamilton



FID	Shape	ID	COMPKEY	WAT_SIZE	UNIT_ID1	Length
0	Polyline	698	222211	300	HA46V033	286.6
1	Polyline	109	296098	150	HA34V002	382.4
2	Polyline	742	235534	150	HA46C002	85.5
3	Polyline	831	296086	50	HA18E015	49.9
4	Polyline	304	312645	300	HB53E011	629.4
5	Polyline	3699	226125	200	DM10E090	37.4
6	Polyline	3736	1372594	200	FL03E113	189.5
7	Polyline	4596	1246295	750	HA53E018	391
8	Polyline	3394	1385446	150	HA51V048	204.6
9	Polyline	6895	221957	300	HA51V036	173.7
10	Polyline	6928	232289	150	HA51V037	169.9
11	Polyline	7622	250974	250	FK03E012	49.2
12	Polyline	7635	235636	150	HC40T005	77.4
13	Polyline	6426	232021	150	HA51V022	215.5
14	Polyline	5956	225473	150	HA35V024	382.9
15	Polyline	7272	239819	150	HA59E026	205.7
16	Polyline	6717	238871	1050	HC46V021	313.7
17	Polyline	7402	239714	150	HA59V030	208.4
18	Polyline	7408	310547	400	HB21E011	354.1
19	Polyline	6177	294815	150	HA31V001	113.4

Figure 3.3 Attribute table of the shapefile of the city water main network.

However, not all the given data in the attribute table are useful, only the data of the pipe diameters and the length of the pipes are applicable for the present study. Since other essential data, such as the information on the pipe material properties and pipe history, are not entirely available or inaccurate, it was decided that for the purpose of demonstrating the developed integrated DSS for water distribution network, hypothetical information on pipe ages and pipe materials are added into the attribute table of the shapefile of the water main network. A water main network with the added information of hypothetical pipe ages and hypothetical materials are shown in Figure 3.4a and 3.4b, respectively.

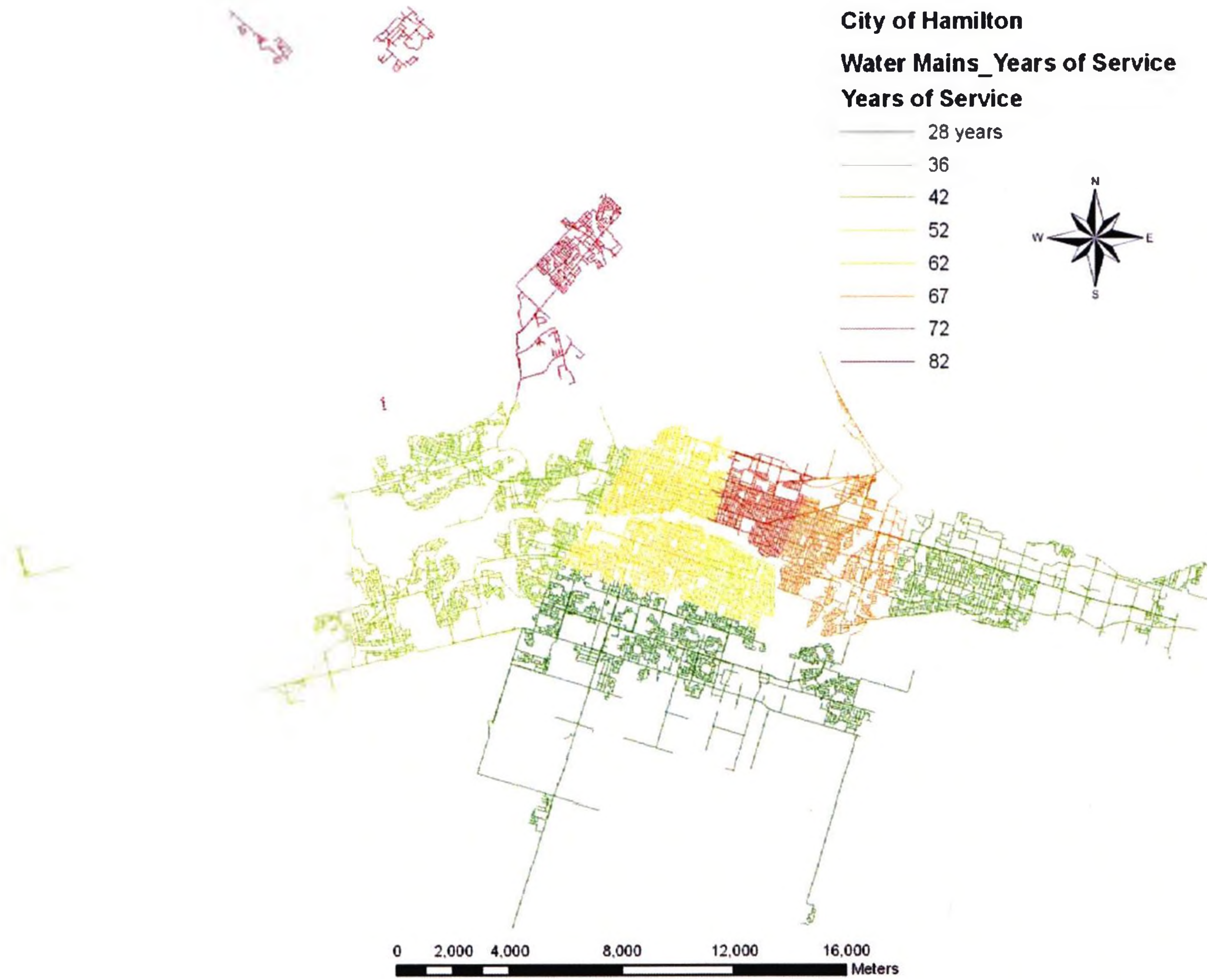
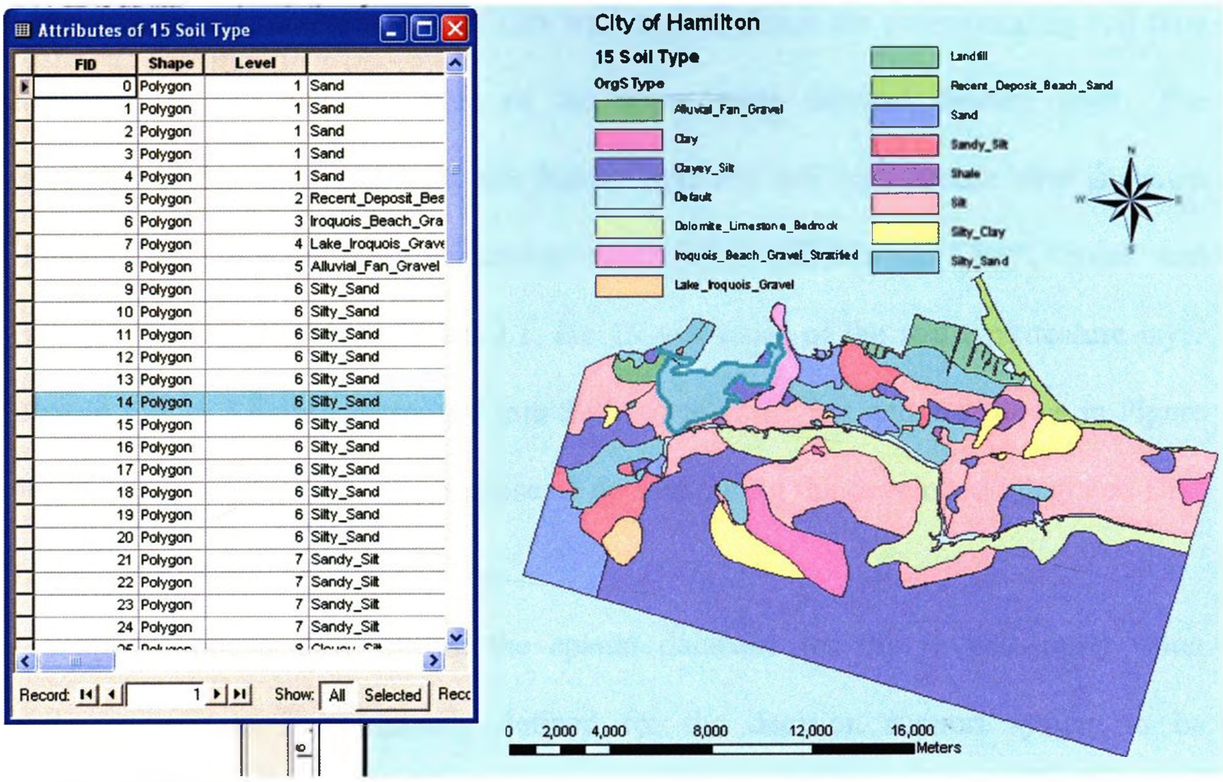


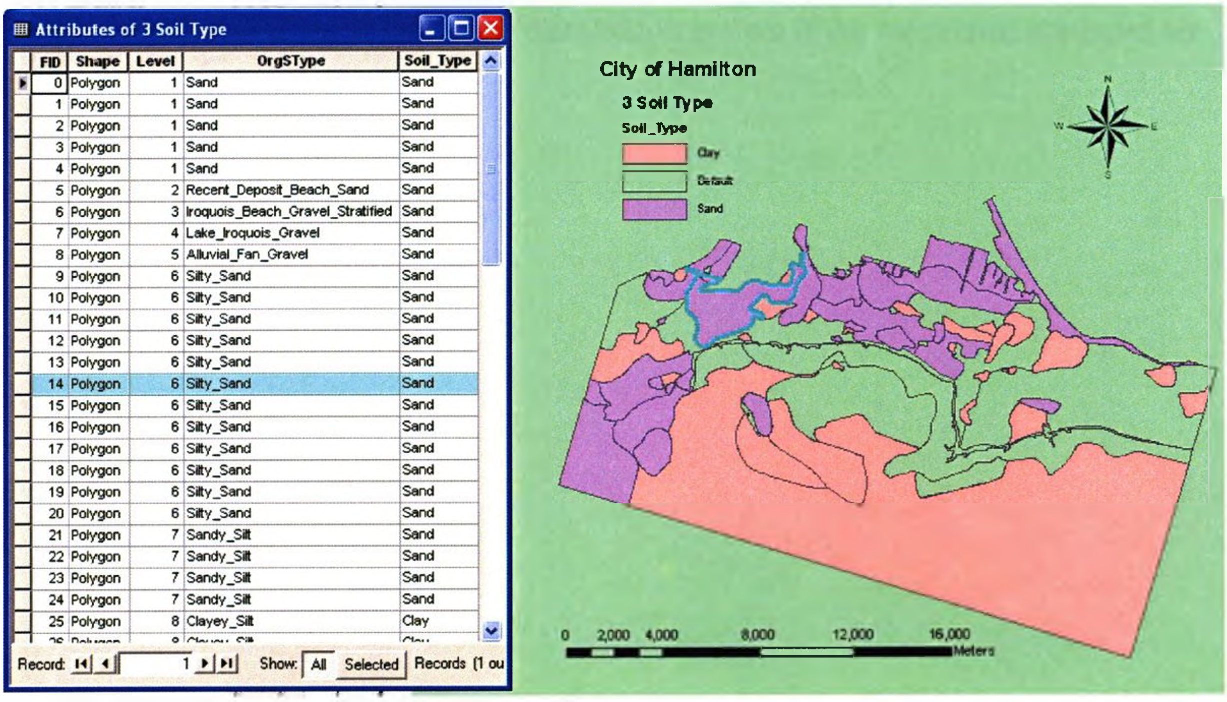
Figure 3.4a Hypothetical information on pipe ages.

Figure 3.4b Hypothetical information on pipe materials.

Fourteen shapefiles of different soil types of the City of Hamilton are also given in the database in vector form. The soil types are alluvial fan gravel, clay, clayey silt, dolomite limestone bedrock, stratified Iroquois beach gravel, lake Iroquis gravel, landfill, recent deposits of beach sand, sand, sandy silt, shale, silt, silty clay, and silty sand. The spatial function in GIS software was used to create a new shapefile of overall soil type of the City of Hamilton by combining these fourteen shapefiles of different soil types. A "default" soil type was added to the areas that are not covered by these fourteen soil type shapefiles. Feature layers of the combined soil type shapefile and their corresponding attribute table are displayed in Figure 3.5.



a) With 15 soil types



b) With three simplified "soil types"

Figure 3.5 Feature layer of the combined soil type shapefile and corresponding attribute table

Spatial analysis functionality of GIS was used to insert the corresponding soil type information to the attribute table of the water main network feature layer. The intersection of the water main network feature layer and soil type feature layer is shown in Figure 3.6a. Since the break occurrence rate is only available for "Clay", "Sand" and "All" soil types as shown in Table 2.1, the 15 soil types of the soil type feature layer shown in Figure 3.6a are regrouped into these three soil types and are shown in Figure 3.6b. The re-grouping is for the purpose of demonstrating the developed integrated DSS for water distribution system.

With the above descriptions, the spatial database for the (hypothetical) water distribution system is completely defined for the decision support system to be developed. It must be emphasized that the developed integrated decision support system is equally applicable to an actual water distribution system if the mentioned attributes are available.

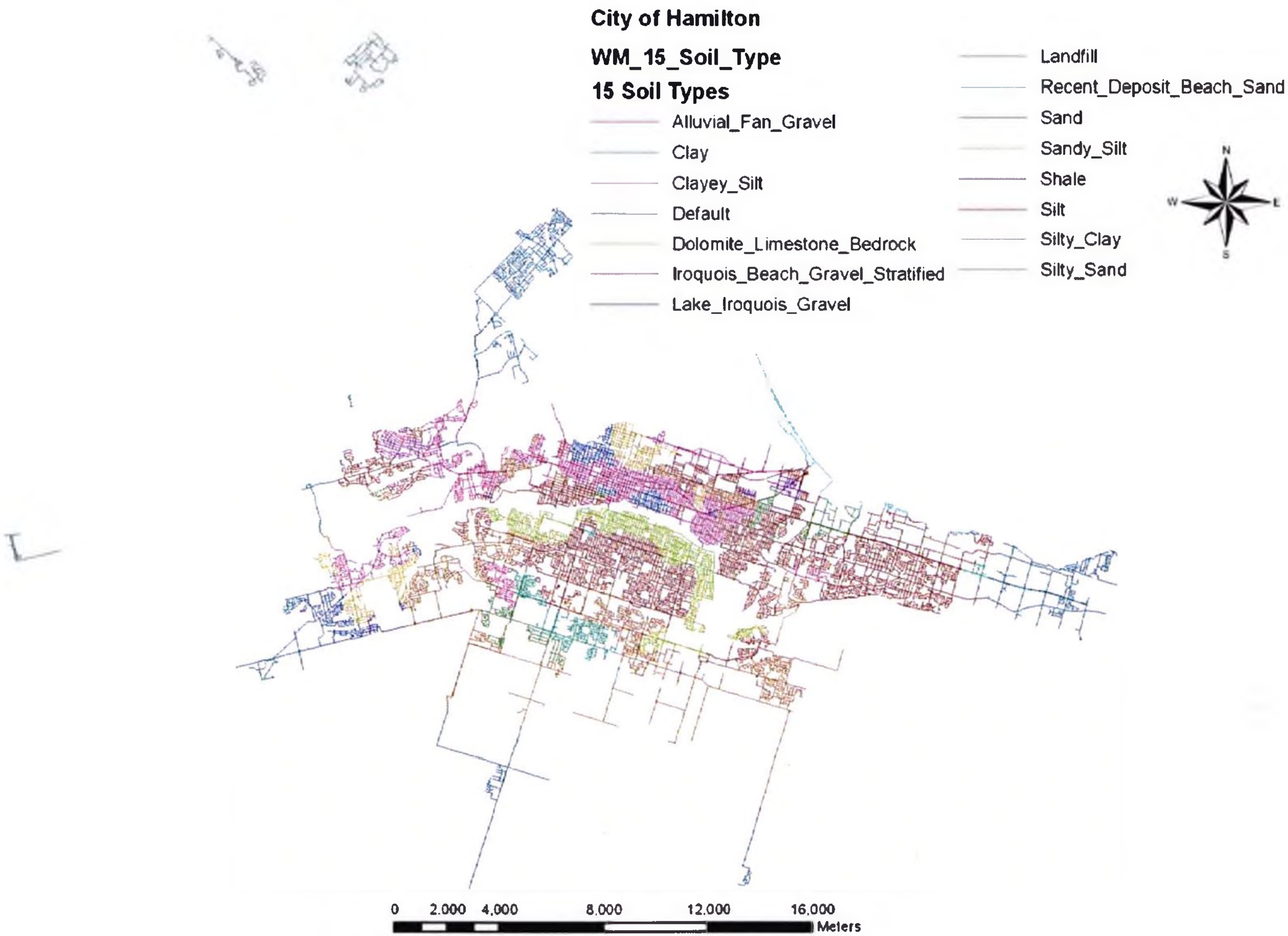


Figure 3.6a Intersection of the water main network feature layer and the 15 soil type feature layer.

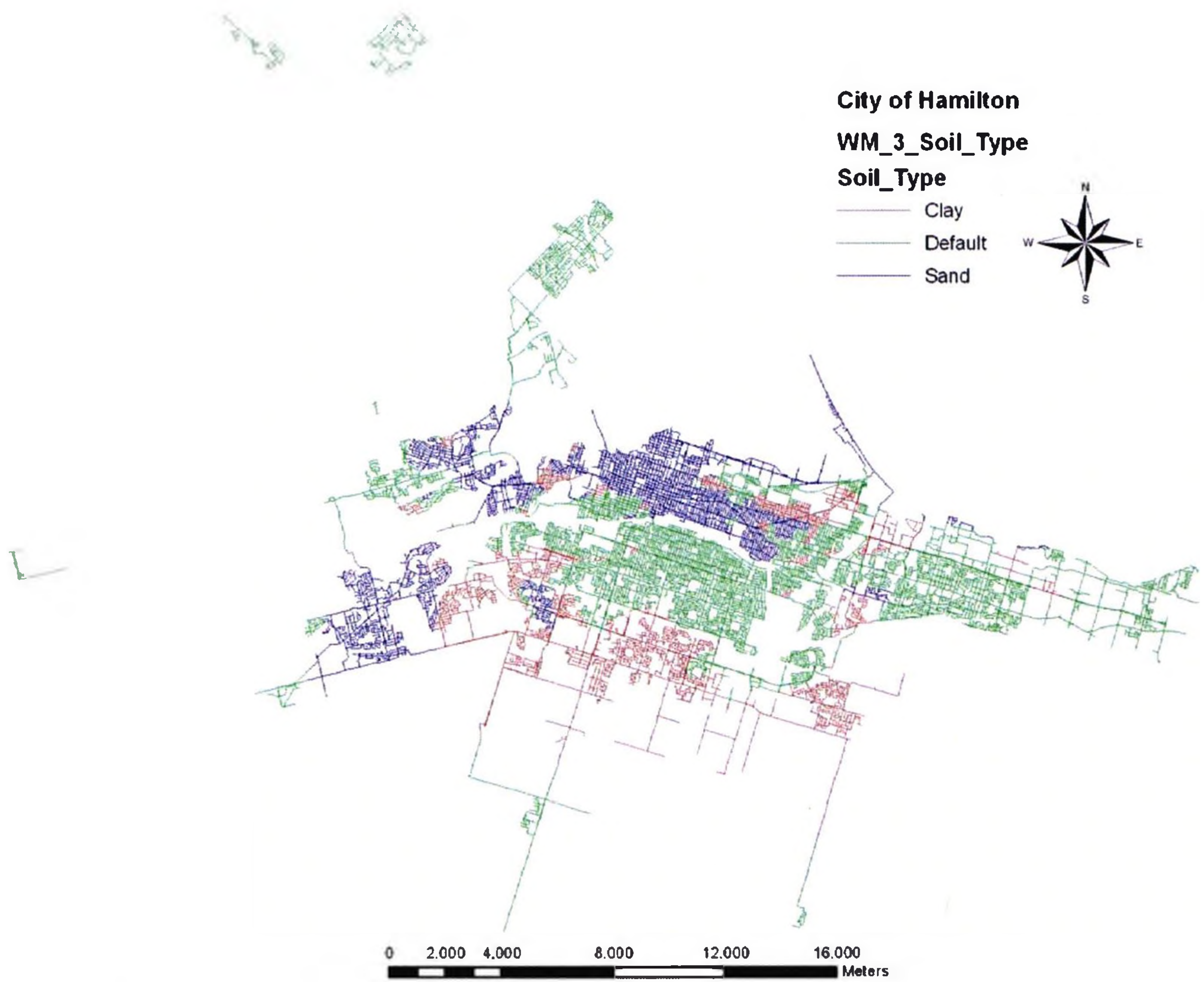


Figure 3.6b Intersection of the water main network feature layer and the 3 simplified "soil type" feature layer.

3.3.2 Development of macro under VBA and accessing information in GIS

Given the spatial database of the water distribution system is developed with the necessary attributes discussed in the previous section, the probabilistic model described in Chapter 2 can be employed in evaluating the optimum replacement time. To carry out this evaluation automatically, one needs to implement or integrate the probabilistic model into the GIS for the water distribution system. The implementation and integration requires the following steps:

1) Access the detailed information on the water distribution system including the pipe geometries, pipe material properties, pipe history, the surrounding soil environment, and other attributes of the water main network feature layer, possibly its operational characteristics;
2) Implement the algorithm discussed in Chapter 2 and summarized in Section 2.5. This includes the assessment of the probability of failure, and the evaluation of the optimum replacement time based on the minimum expected life cycle cost and the minimum expected annual average cost for each individual pipeline segment. The implementation should be carried out using a language that is supported by the adopted GIS software (i.e., ArcView 9.1 from ESRI); and
3) Save or store the obtained results in Step 2) in a data layer in the GIS, which can then be shown as an output thematic map on the water distribution system. That is, the GIS application is then used to display and geographically reference the ranked or preferred decisions.

In order to gain flexibility, one could also develop a user interface to allow user defined parameters for the break occurrence model and/or the cost related information for the optimum replacement scheduling models.

For the present study, ArcGIS from ESRI is adopted since it is one of the most widely used GIS software. The access to the attributes and manipulation of the information is written in Visual Basic Application (VBA) which is one of the programming languages supported by ArcGIS. The algorithms written for this purpose are provided in the Appendix A.

It should be noted that the algorithm for selecting the optimum replacement scheduling described in Chapter 2 requires the use of an optimization algorithm to find the optimum replacement time. Rather than re-write an optimization algorithm in VBA, we use an existing optimization program written in FORTRAN using the nonlinear sequential quadratic programming method (Schittkowski, 1985). This program is integrated through the use of the dynamic link library (DLL). The interface between the written macro and the DLL is also included in Appendix A. The closely integrated method implemented in this study speeds up the numerical analysis when comparing with a simple implementation of loose coupling method of using Microsoft Excel spreadsheet and Solver included in Microsoft Excel.

It should be mentioned that the obtained results need to be written or saved to the database. This is described in the subroutine named, CalOptReplacementTime(), in the macro implemented in VBA which is also included in Appendix A.

3.3.3 Use of existing and developed graphic user interface (GUI)

The purpose of the development of the GUI is to make the software more user-friendly. User has two different ways to execute the developed macros mentioned previously through GUI by using: 1) Optimum Replacement Scheduling (ORS) toolbar, as illustrated in Figure 3.7 or 2) feature layer context menu, which only appears when a feature layer has been right-clicked (Burke, 2007), as illustrated in Figure 3.8.

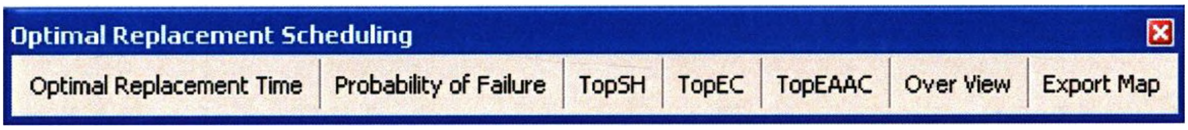


Figure 3.7 Optimum Replacement Scheduling (ORS) toolbar

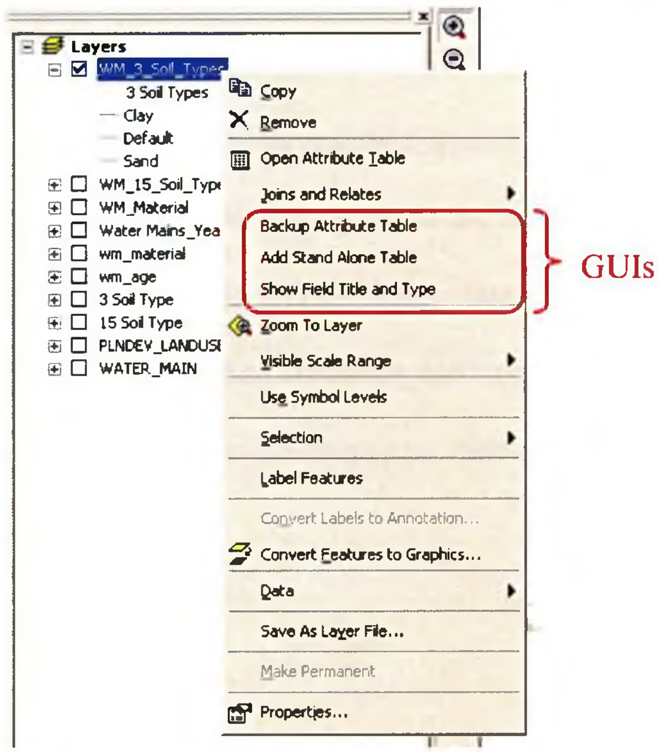


Figure 3.8 GUIs in feature layer context menu

GUIs in the ORS toolbar are developed in this study for accessing the optimum replacement scheduling calculations (Optimum Replacement Time), result presentations (Probability of Failure, TopSH, TopEC, and Top EAAC), result exploration (Over View), and map exportation (Export Map), while the GUIs in the right click menu of the feature layer, i.e. Backup Attribute Table, Add Stand Alone Table, and Show Field Title and Type, are related to the attribute table data.

When the Backup Attribute Table GUI is executed, a comma-separated value (csv) format text file will be created and it will store all the data from the attribute table of the selected feature layer. This backup csv format text file can be explored in Microsoft Excel spreadsheet. The Add Stand Alone Table GUI is used to insert extra tabulated data to the attribute table of the selected feature layer from a tabulated format file, such as csv, txt, or dbf files. The first column of the input tabulated file and the attribute table of the target feature layer are required to have the title and data type as the key data for the table insertion process. This ensures that the data is inserted in the right place. When the Add Stand Alone Table GUI is clicked, an Add Data dialog box as shown in Figure 3.9 appears, displaying only the ArcGIS specified tabulated file types. This allows the user to select the desired tabulated file that contains the data to be inserted.

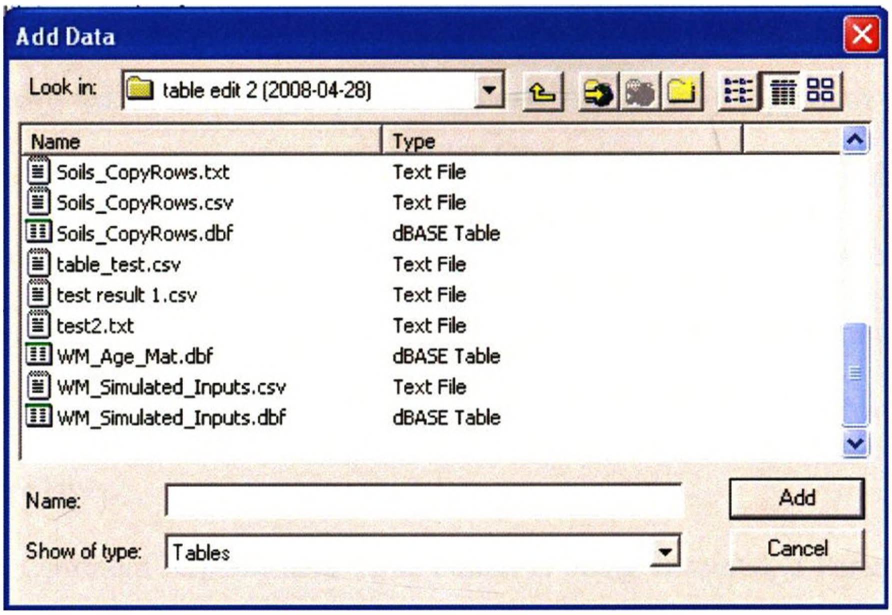


Figure 3.9 Add Data dialog box for tabulated file selection.

To make it easier for the user to find out fields title and their corresponding field type in an attribute table of a feature layer, the Show Field Title and Type GUI was created. This GUI is presented in Figure 3.10, showing the typical message box for the field title and field type of a feature layer.

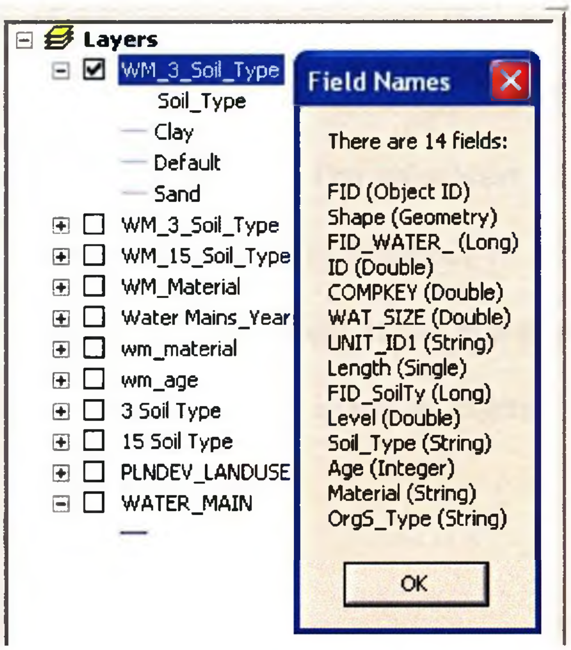
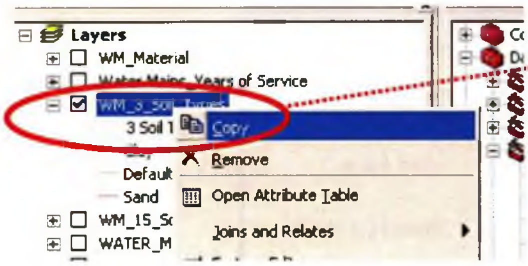


Figure 3.10 Message box showing field titles and field types of a feature layer.

Before using the ORS GUIs to activate the calculation of the optimum replacement time and produce result presentations, a base feature layer with a completed spatial database, as mentioned in section 3.3.1, must be provided. A working feature layer could be created by copy and paste of the base feature layer to the layers as illustrated in Figure 3.11. Use of a working feature layer is advantageous since it will not overwrite any existing feature layer.

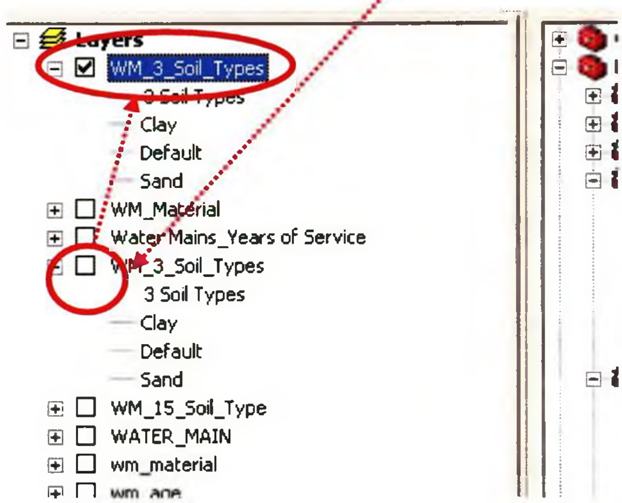
When the Optimum Replacement Time button is being executed, a Parameter Inputs dialog box shown in Figure 3.12 pops up to allow user to provide some basic input parameters including current year, planning horizon, discount rate, and cost information. Note that there are two input methods for the cost information. User can provide either the cost data including the repair cost and replacement cost or the cost rate data including the repair cost and cost rate; where cost rate is the ratio of the repair cost to the replacement cost.

When the calculation of the optimum replacement time and corresponding annual costs is in progress, a progress status bar, presented in Figure 3.13, shows the progress at the bottom left-hand corner of the window. The calculated results are stored in the corresponding fields as illustrated in Figure 3.14 created in the attribute table of the selected working feature layer. The results are also stored in a tabulated output file in csv format to facilitate the user for possible transfer of results to other software platform.



a) Copy the base feature layer

b) Paste the layer back to Layers as a working feature layer



c) Uncheck the original base feature layer and select and check the working feature layer

Figure 3.11 Illustration of creating a working feature layer from a base feature layer.

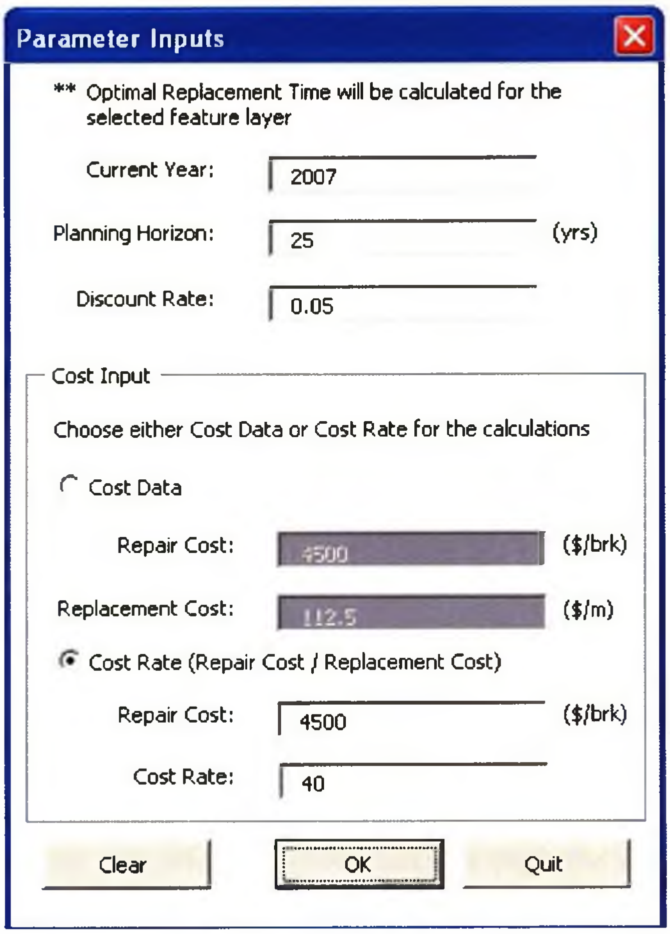


Figure 3.12 Developed Parameter Inputs dialog box.

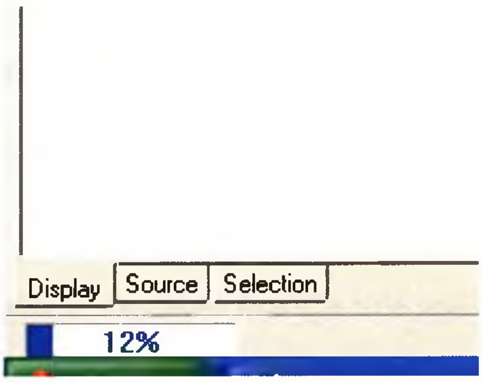


Figure 3.13 Progress status bar.

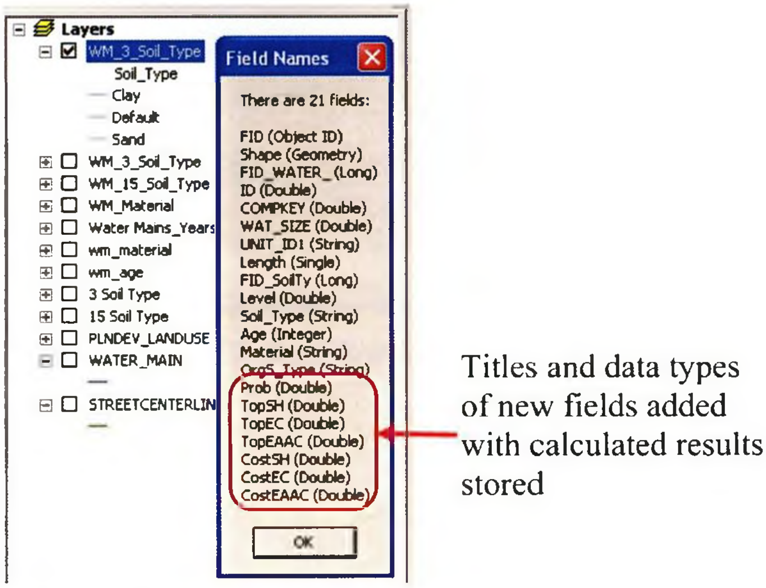


Figure 3.14 New fields added to the attribute table of the selected feature layer after the execution of ORS GUI.

After the calculations of the optimum replacement time, results presented in map format can be obtained by executing the procedures through the buttons of Probability of Failure, TopSH, TopEC, and TopEAAC, respectively, for the probability of failure of water pipe segments, optimum replacement time calculated based on Eq. (2.13), the expected total cost theory and the expected annual average cost theory. As an example, when the Probability of Failure button is clicked, an Add Probability of Failure Renderer dialog box shown as Figure 3.15 pops up. This dialog box allows user to select either colour or gray scales for the rendering. Examples of colour and gray scale maps are presented in Figure 3.16a and 3.16b. Note that the title and the renderer of the selected feature layer are changed and, a renderer is added to the corresponding map in the layout page.

Since identical user procedure and similar mapping options for other calculated values (i.e., add renderer buttons, namely the TopSH, TopEC and TopEAAC buttons), are developed, they are not explained again.

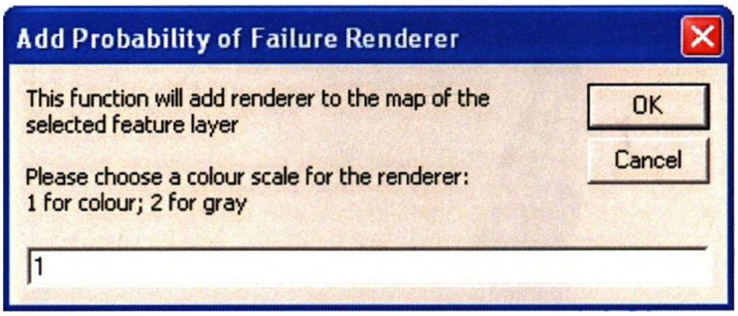


Figure 3.15 Add Probability of Failure Renderer dialog box.

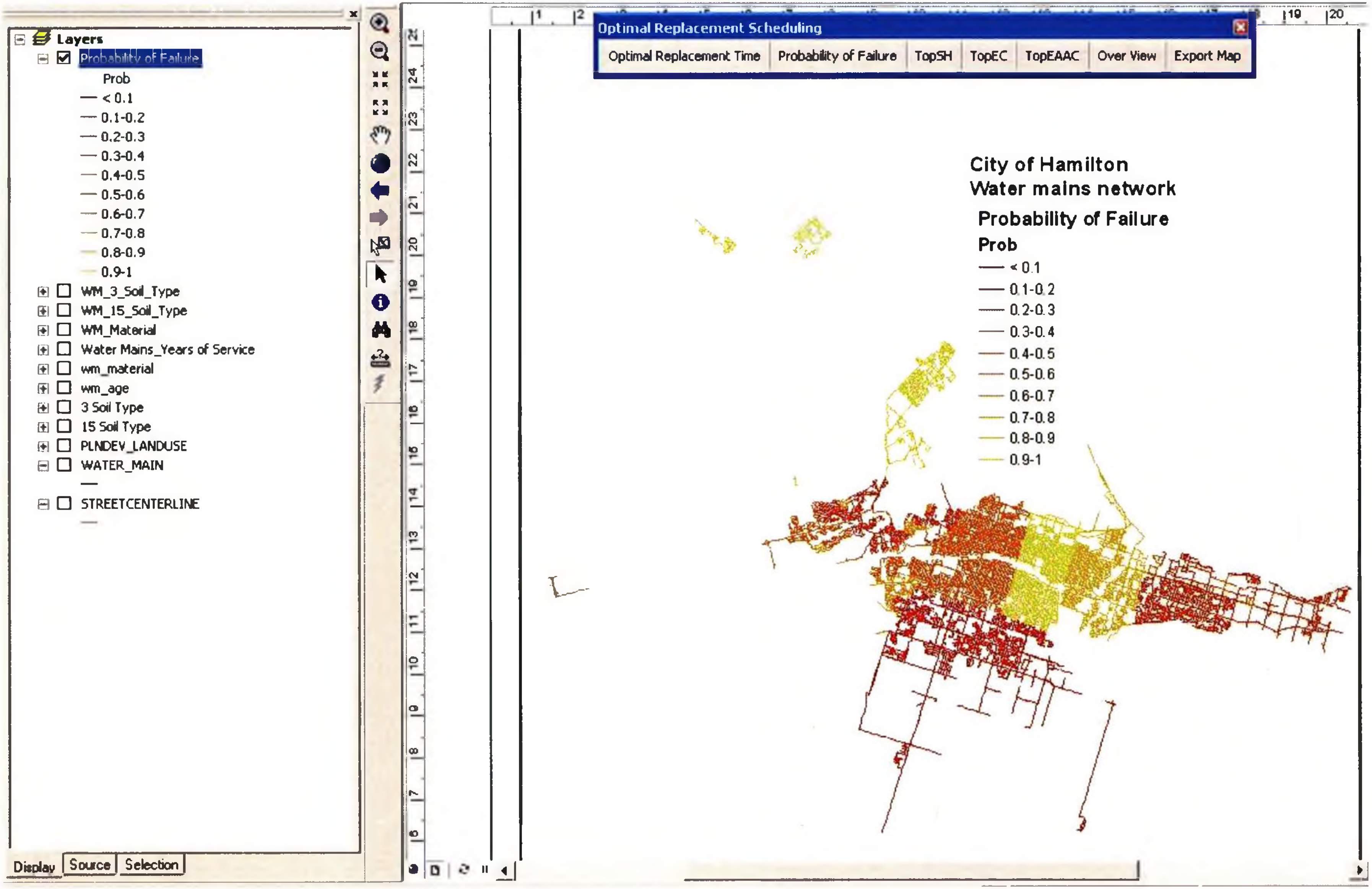


Figure 3.16a Example map results with renderer in colour scale

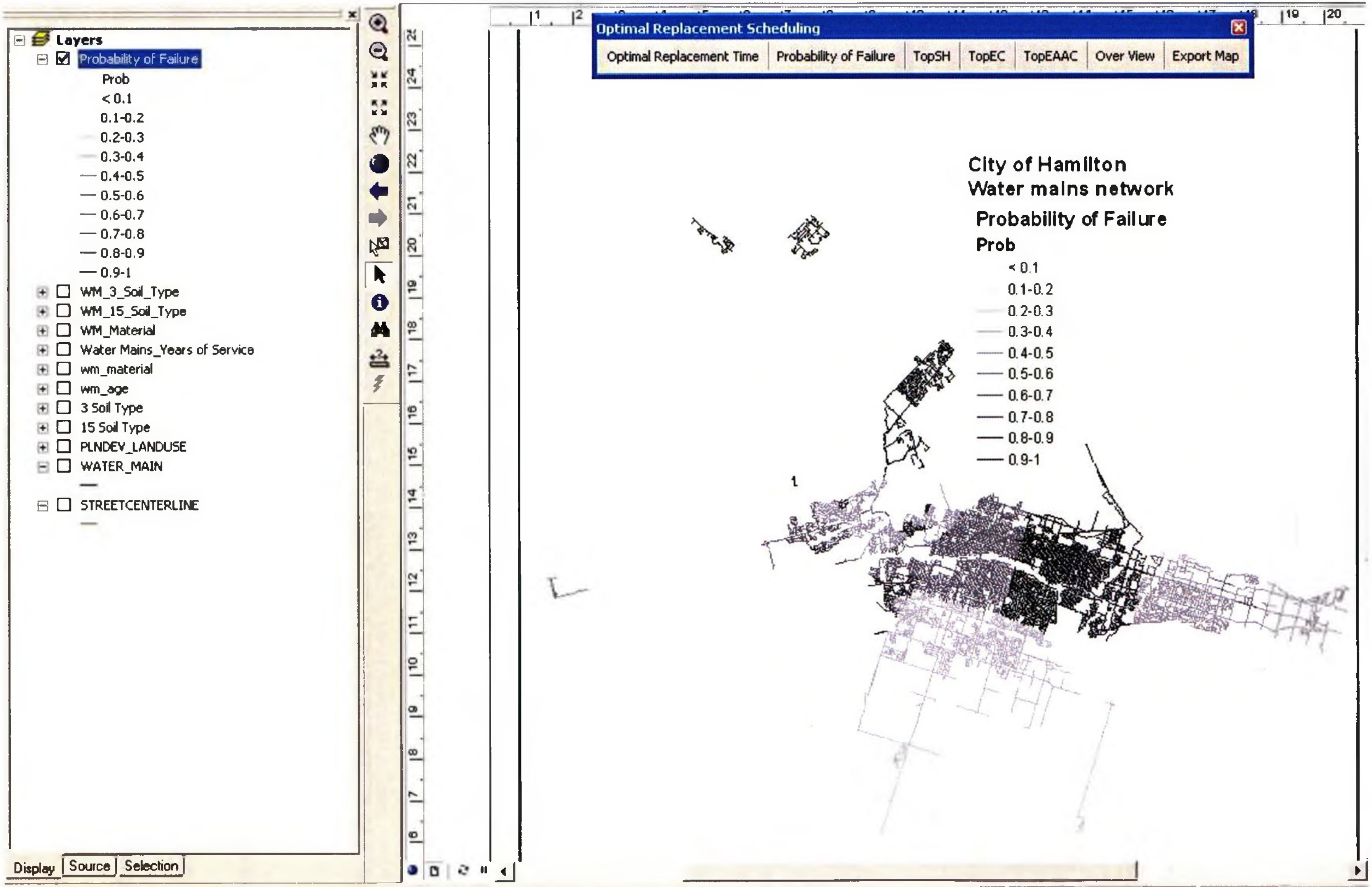


Figure 3.16b Example map results with renderer in grey scale

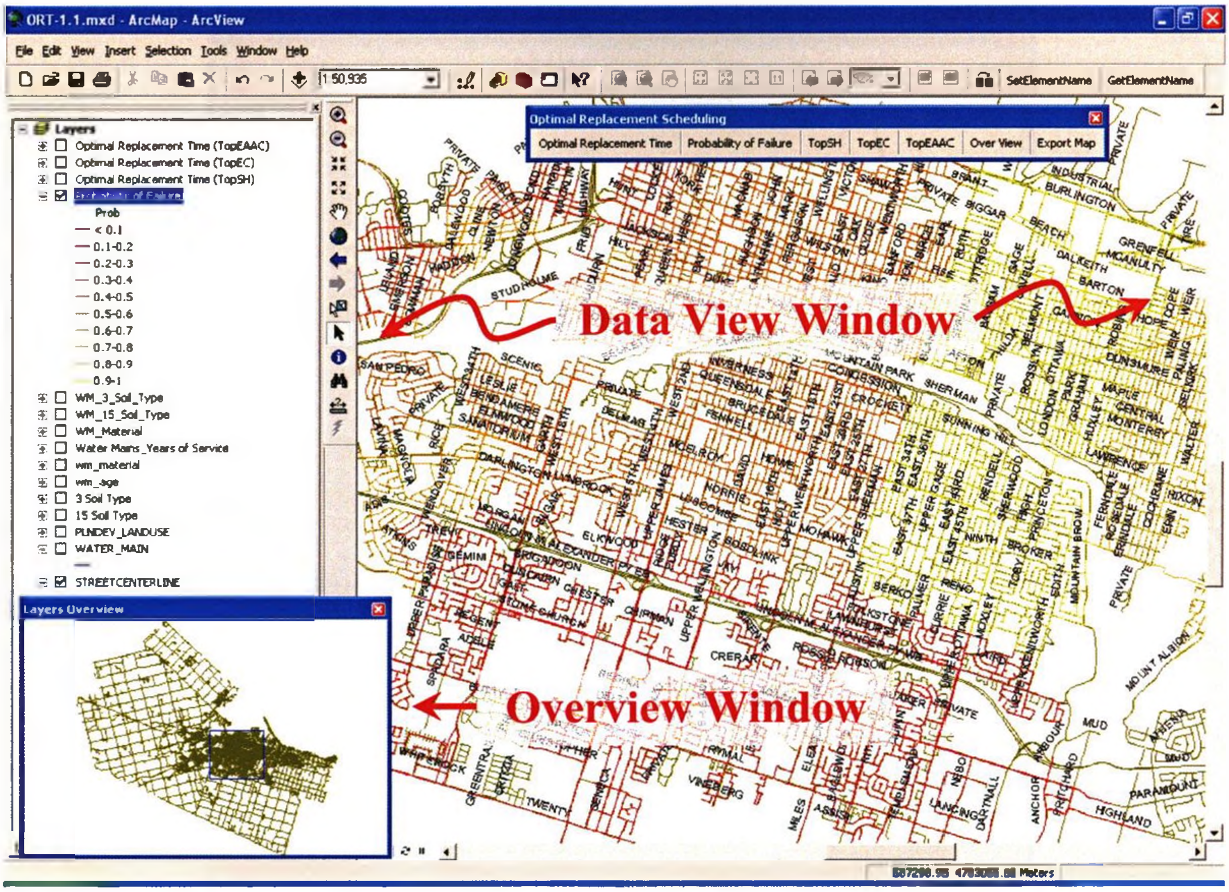


Figure 3.17 Overview window

When a closer view of the result map is desirable, user can use the developed Over View GUI to turn on an overview window as illustrated in Figure 3.17. The area within the blue box in the overview window is displayed in the data view window. User can move around the blue box to explore different place within the overview window. Furthermore, the user can change the size of the blue box in order to zoom in or zoom out of the map in the data view window.

The last developed GUI in the ORS toolbar is the Export Map button. By clicking this button, a build-in Export Map dialog box as shown in Figure 3.18 pops up. This dialog allows the user to save the map in the data view window with a desired name and resolution.

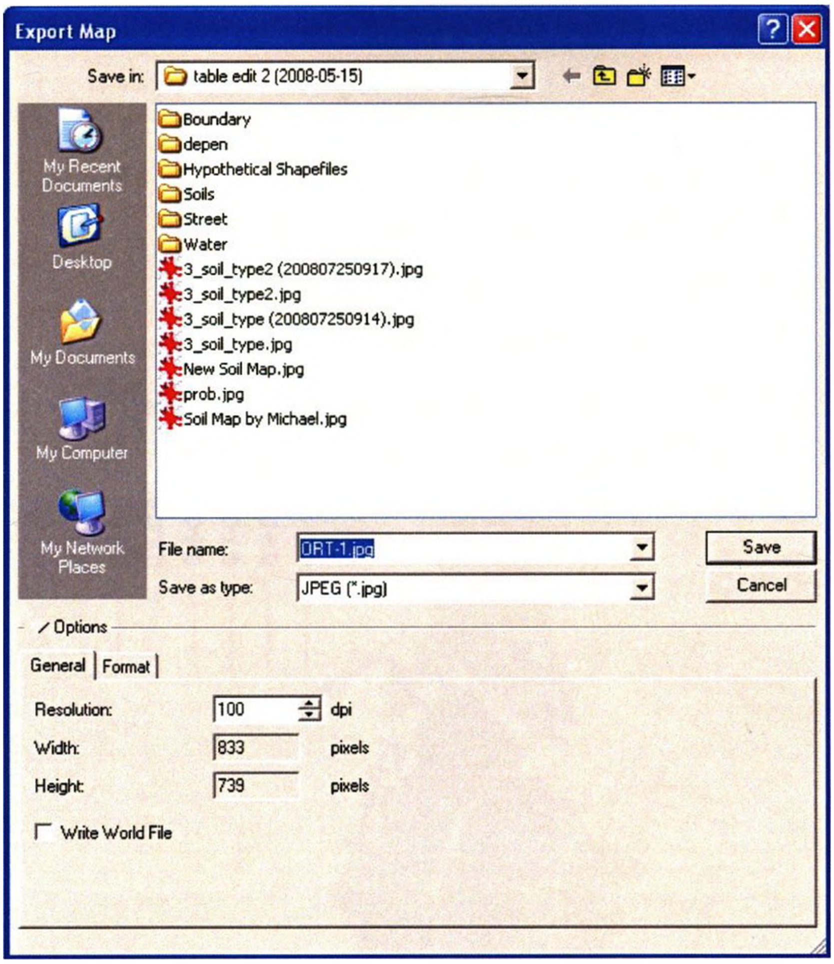


Figure 3.18 Build-in Export Map dialog box

3.4 Illustrative application and numerical results

Using the developed and implemented software (which are included in Appendix A), and the already established feature layer for the water distribution system discussed previously, the probabilities of at least one break before the end of service period for the water main network is calculated. The results are displayed across the network in Figure 3.19. For a selected tolerable failure probability, the program can be used to identify the segments of water distribution system with failure probability less than this tolerable level, i.e. those are in most urgent need for rehabilitation.

Figure 3.19 Calculated failure probability $P(\tau)$ according to Eq. (2.11)

Rather than using the failure probability as an indicator for prioritizing rehabilitation schedules, as discussed in the previous section, the optimum replacement time T_{op} can be evaluated by minimizing $E(C(T_{op},T))$ as shown in Eq. (2.12) for a selected decision horizon T, or by minimizing $E(C(T_{op},T))/T$ shown in Eq. (2.15). Results for three such cases are shown in Figure 3.20 and Figure 3.21. These results can be used as an aid for decision makers or planners to prioritize the rehabilitation effort for the water distribution network. The figures indicate that the obtained optimum rehabilitation scheduling depends on the adopted planning period and/or the objective function.

The example application shows clearly that the GIS-integrated decision support system not only provides efficient and powerful storage, management, manipulation and query data in the spatial database for the geographically linked data, but also enhances the ability of handling complex analysis. This system makes the process of analysis run smoother and simpler to operate. Furthermore, with the visual illustration capability from the GIS, the decision maker or planner gets an easy to understand map representing the analyzed results.

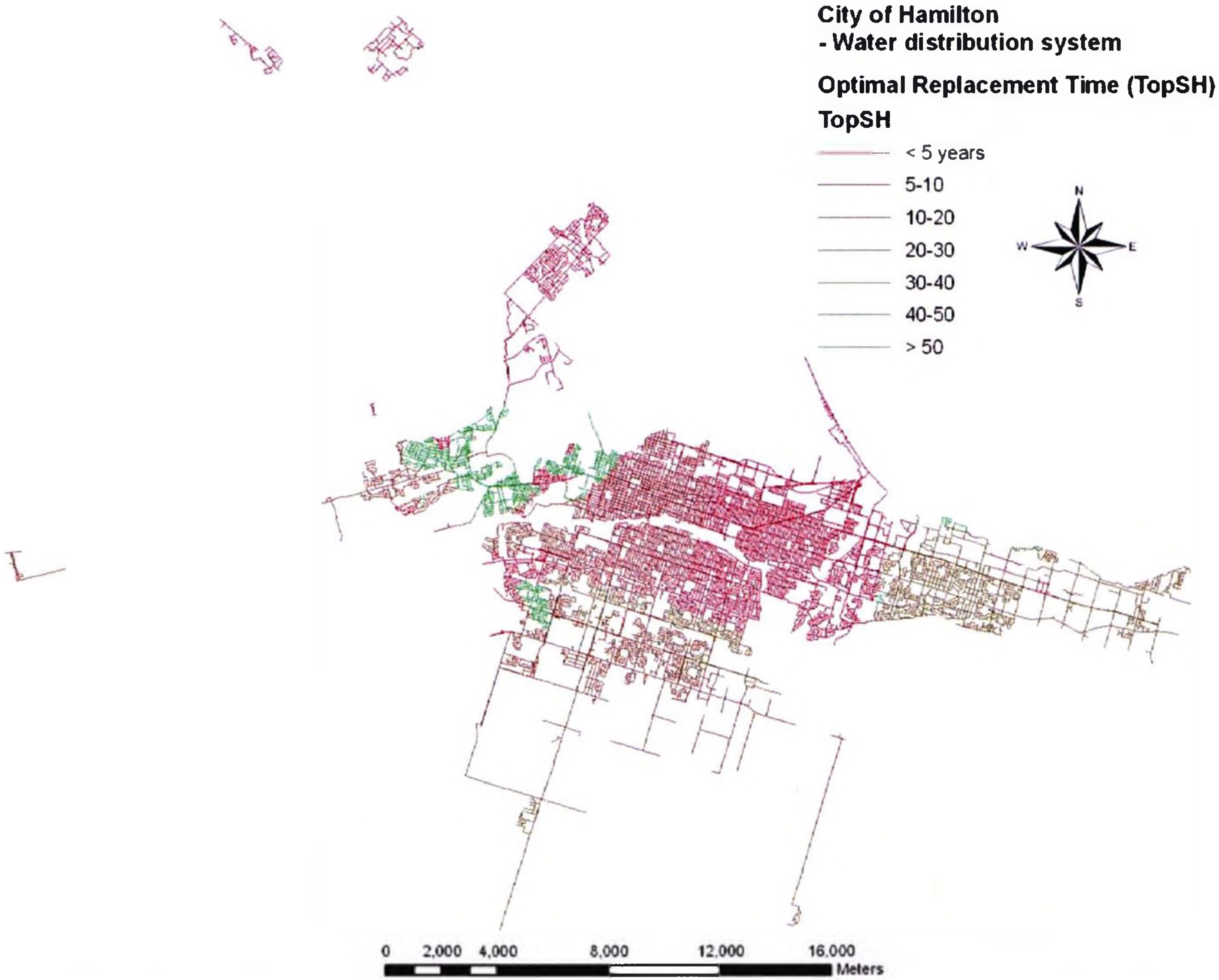


Figure 3.20a Optimum rehabilitation scheduling for Eq. (2.13) based on minimum expected total cost.

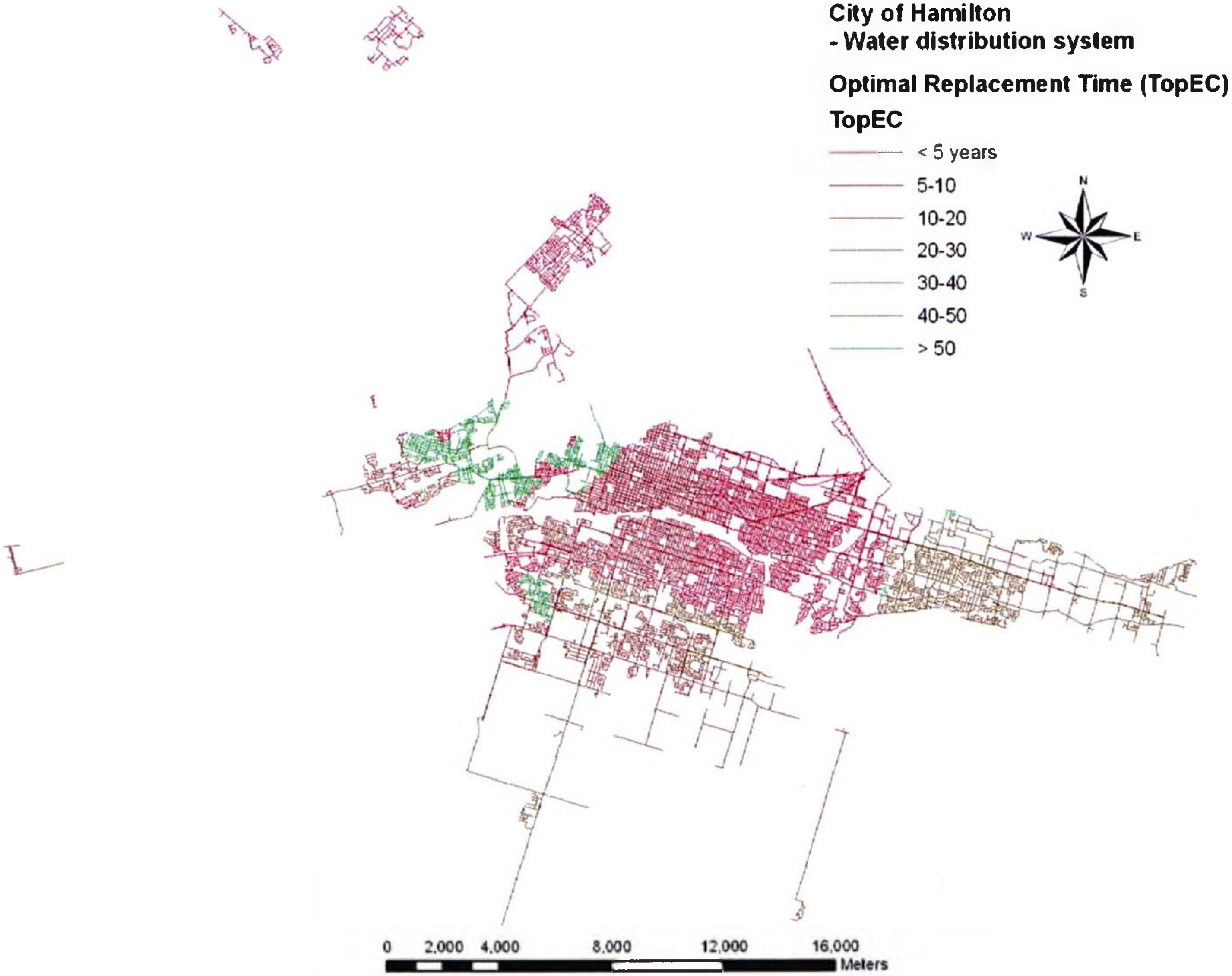


Figure 3.20b Optimum rehabilitation scheduling for 25 years of planning horizon based on minimum expected total cost.

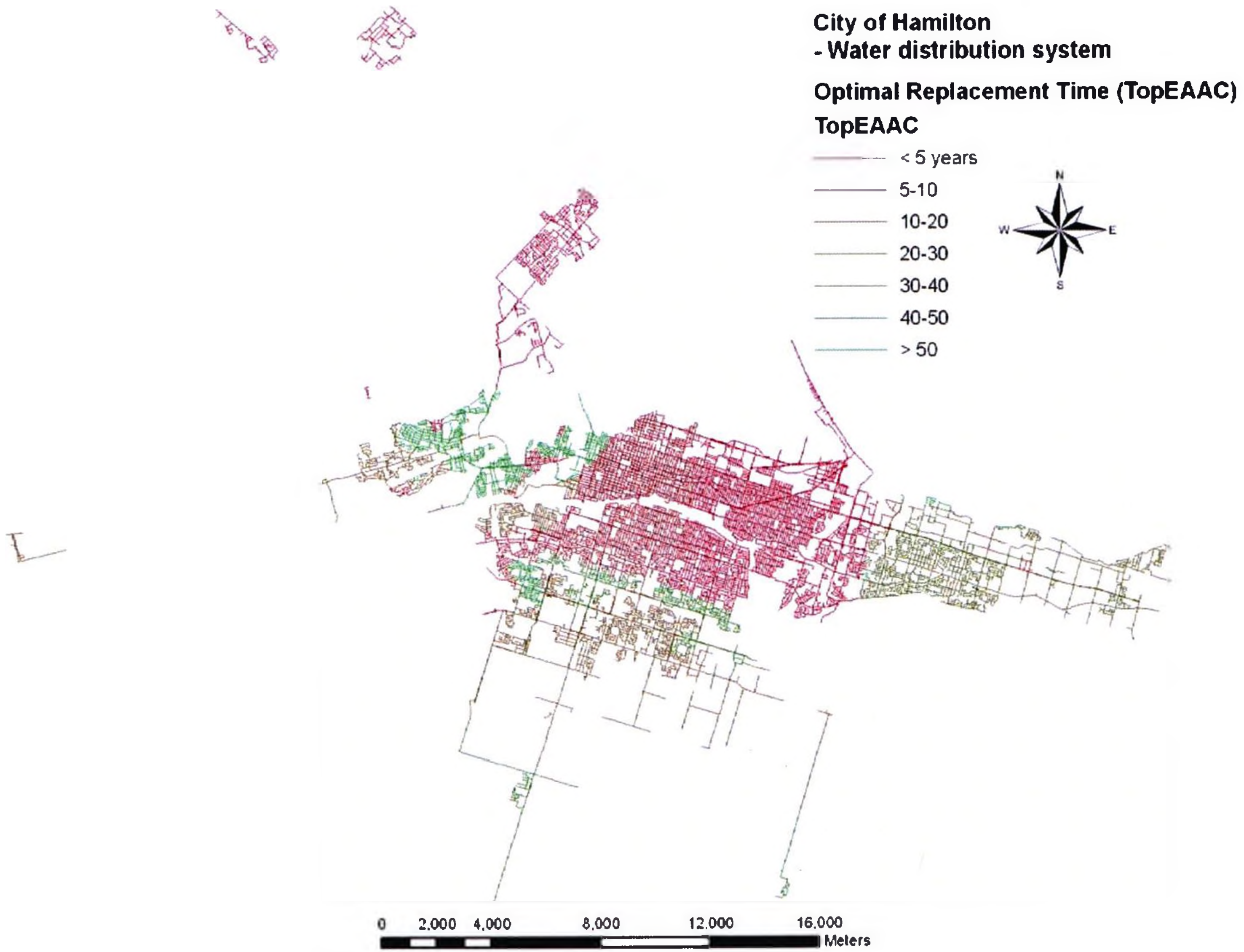


Figure 3.21 Optimum rehabilitation scheduling by minimizing the expected annual average cost..

3.5 Summary

A water distribution system is a complex infrastructure system that can be georeferenced. Therefore, it is advantageous to use the GIS in dealing with such a system for data management, layering management, visualization, and spatial data analysis. The use of GIS system also allows different levels of integration, development of macro for numerical analysis and user interfaces, and map visualization in 2D and 3D environments.

In this chapter, an integration of probabilistic analysis method with the ArcGIS for water pipeline renewal prioritization is carried out. The adoption of the ArcGIS rather than other GIS system is justified since it is most often used GIS system in the industry. The developed decision support system is illustrated through the use of the water distribution system obtained from the City of Hamilton. Data, such as soil profile of the city, and the hypothetical pipe age and material are combined with the water main network feature layer through the spatial analysis to construct a spatial database for the water distribution system of the City of Hamilton. The use of the developed decision support system and the developed graphic user interface (GUI) is illustrated as well. The obtained optimum rehabilitation scheduling, based on failure probability level, and/or cost/benefit analysis are presented in the form of maps.

Chapter 4 Summary, conclusions and future work

4.1 Summary and conclusions

Existing water distribution system is deteriorating. Some of these buried pipes in the systems are meeting or even exceeding the intended or design service life. Failure of water distribution systems is often reported in technical publications as well as in the news media. The consequences of the failures caused economic losses and many social problems including water supply disruptions, inconveniences to public, damages to adjacent utilities, road subgrade, and nearby buildings or structures. Since the occurrence of the failure is both spatially and temporally uncertain, a probabilistic approach should be employed for water distribution system renewal prioritization. For these reasons, the presented study is focused on the development of a decision support system for optimally selecting the prioritization scheme for renewal of existing water distribution system.

A new decision support system for prioritized water distribution system renewal is developed. The decision support system integrates the probabilistic method with the ArcGIS. It is shown how a thematic feature layer with spatial database can be created for the water distribution system by combining the data of soil types and water pipe age and material. Macros are written in VBA and are integrated with the ArcGIS. These macros implement the probabilistic models, process the data from the thematic feature layer and, incorporate optimization algorithms for numerical analysis. To facilitate its use, some graphic user interfaces (GUIs) are developed and can be used to execute the macros.

As an example application, the developed system is illustrated using the water distribution system obtained from the City of Hamilton. The application successfully demonstrates the usefulness and user friendliness of the developed decision support system, even though some hypothetical information is employed for the numerical analysis. The results, presented in the form of maps with failure probability of the spatially distributed pipe segments and optimum rehabilitation scheduling, facilitate the representation of the results. It also promotes the decision makers and planners to make informed decisions and allocate limited available resources.

The developed GIS-integrated DSS for water distribution system renewal planning, with a tight integration of the program for optimization algorithms, is not only an effective and efficient system to provide optimal replacement schedules of a water distribution system to the decision makers or planners, but also provide them a uer-friendly environment for the analysis implementation and result visualizations by the developed GUIs in ArcGIS.

4.2 Suggested future works

Similar to many other research projects and software developments, there is always room for new development and enhancement. Throughout this study, it was identified that more extensive pipe break occurrence history data would be extremely valuable for calibrating the probabilistic models and model parameters used to predict the break occurrence. Also, it was noted that the realistic cost information on pipe rehabilitation and renewal, which is scarce in the literature, is extremely important for the analysis in order to obtain more relevant results for the decision makers and planners. Therefore, extensive statistical data gathering and analysis on the water mian breaks and costs can be a very valuable project.

Other tasks that can enhance the developed decision support system include the incorporation of more accurate physical-mathematical model for predicting the break occurrence, 3D instead of 2D visualization of the water distribution system, and the access of condition of pipe in real time.

References

Ahammed, M. and Melchers, R. E. (1994). “Reliability of underground pipelines subject to corrosion.” Journal Transportation Engineering, ASCE, 120(6), 989-1002.

American Society of Mechanic Engineers (1991). ASME B31G Manual for detecting the remaining strength of corroded pipelines, A supplement to ASME B312G code for pressure piping, New York, N.Y.

Andreou, S. A., Marks, D. H., and Clark, R. M. (1987a). “A new methodology for modelling break failure patterns in deteriorating water distribution systems: Theory.” *Advance in Water Resources*, 10, 2-10.

Andreou, S. A., Marks, D. H., and Clark, R. M. (1987b). “A new methodology for modelling break failure patterns in deteriorating water distribution systems: Applications.” Advance in Water Resources, 10, 11-20.

Burn, L. S., Tucker, S. N., Rahilly, M., Davis, P., Jarrett, R., and Po, M. (2003). “Asset planning for water reticulation systems - the PARMS model.” Water science & technology: Water Supply 3, 55-62.

Bruke, R. (2003) “Getting to know ArcObjects™ – Programming ArcGIS® with VBA.” ESRI Press, Redlands, California

Canadian Water and Wastewater (CWWA). (1997). “Municipal water and wastewater infrastructure: Estimated investment needs 1997-2012.” Partially sponsored by the Canadian Mortgage and Housing Corp., Ottawa

Clark, R. M., Stafford, C. L. and Goodrich, J. A. (1982). “Water Distribution Systems: A Spatial and Cost Evaluation.” Journal of Water Resources Planning and Management Division, ASCE, 108(3), 243-256.

Constantine, A. G. and Darroch, J. N. (1993). "Pipeline reliability: stochastic models in engineering technology and management." S. Osaki, D.N.P. Murthy, eds., World Scientific Publishing Co., River Edge, N.J.

Cox, D. R. (1972). "Regression models and life tables." J. Royal Statistic Society, 34(B), 187-220.

Deb, A. K., Snyder, J. K., Logananthan, G. V., Agbenowski, N., and Grablutz, F. (2003) "Prioritizing water main renewal program." Proc., New Pipeline Technologies, security and Safety, ASCE, Reston, Va., 1002-1011.

Doyle, G. Seica, M. V. and Grabinsky, M. W. F. (2003). "The Role of Soil in the External Corrosion of Cast Iron Water Mains in Toronto, Canada." Canadian Geotech. Journal, 40, 225-236.

Elms, D. (1985). "The principle of consistent crudeness." Proc., NSF Workshop on Civil Engineering Application of Fuzzy Sets." Purdue Univ., West Lafayette, Ind., 35-44.

Fattler, J. E., Sin, Y. T., Root, R. R., Ragsdell, K. M., and Reklaitis, G. V. (1982). "On the computational utility of polynomial programming solution method." Mathematical Programming, 22, 163–201.

Goodchild, M. F., Parks, B. O., and Styaert, L. T. (1993). "Environmental modeling with GIS." Oxford Univeristy Press US.

Hadzilacos T., Kalles D., Preston N., Melbourne P., Camarinopoulos L., Eimermacher M., Kallidromitis V., Frondistou-Yannas S., and Saegrov S. (2000). "UtilNets: a water mains rehabilitation decision-support system." Computers, environment and urban systems 24, 215-232.

Herz R. K. (1998). "Exploring rehabilitation needs and strategies for water distribution networks." J Water SRT -Aqua 47, 275-283.

Hong, H. P. (1997). "Reliability based optimal inspection and maintenance for pipeline under corrosion." Civil Engineering Systems, 14, 313-334.

Hong, H. P. (1999). "Inspection and maintenance planning of pipeline under external corrosion considering generation of new defects." Structural Safety, 21, 203-222.

Hong, H. P., Allouche, E. N., and Trivedi, M. (2006). "Optimal scheduling of replacement and rehabilitation of water distribution systems." Journal of Infrastructure Systems, ASCE. 12(3), September: 184-191.

Iu, V. P., Lamas, L. N., Li, Y. P., and Mok, K. M. (2003). "Computational methods in engineering and science." Taylor & Francis.

Julie (2008). "Water main break downtown causes road collapse, threaten gas line." Cleveland Leader, Cleveland Media Group LLC

Kleiner, Y., and Rajani, B. B. (1999). "Using limited data to assess future need." Journal of American Water Works Association, 91(7), 47-62.

Kleiner Y., Adams B. J., and Rogers J. S. (1998a). "Long-term planning methodology for water distribution system rehabilitation." Water resources research 34, 2039-2051.

Kleiner Y., Adams B. J., and Rogers J. S. (1998b). "Selection and scheduling of rehabilitation alternatives for water distribution systems." Water resources research 34, 2053-2061.

Kleiner, Y. (2001). "Optimal scheduling of rehabilitation and inspection/condition assessment in large buried pipes." 4th International Conference on Water Pipeline Systems – Managing Pipeline Assets in an Evolving Market, pp. 181-197.

Kleiner, Y. and Rajani, B. B. (2001) "Comprehensive review of structural deterioration of water mains: statistical models." Urban Water, 3(3), Oct. 2001: 131-150.

Kleiner, Y., Adams, B. J., and Rogers, J. S. (2001). "Water distribution network renewal planning." J. Comput, Civ. Eng., 15(1), 15-26.

Kleiner, Y., Rajani, B. B., and Wang, S. (2007). "Consideration of static and dynamic effects to plan water main renewal." National Research Council, Ottawa, Canada, NRCC-49269

Li, D., and Haims, Y. Y. (1992). "Optimal maintenance-related decision making for deteriorating water distribution systems 1. Semi-Markovian model for a water main." Water Resources Research, 28(4), 1053-1061.

Lo, C.P. and Yeung, A.K.W. (2007). "Concepts and Techniques in Geographic Information Systems." *Pearson Education, Inc.*

Loganathan, G. V., Park, S. and Sherail, H. D. (2002). "Threshold Break Rate for Pipeline Replacement in Water Distribution Systems." Journal of Water Resources Planning and Management, July-August 2002, pp. 271-279.

Madsen, H. O., Krenk, S., and Lind, N. C. (1986). "Methods of Structural Safety." Prentice-Hall. Inc. Englewood Cliffs, New Jersey

McDermott, R. (2007). "SINKHOLE AFTERMATH: Mayor calls for government support." London Topic.ca, News London Ontario.

Michener, W. K., Brunt, J. W., and Stafford, S. G. (1994). "Environmental information management and analysis." CRC Press.

Moglia, M, Burn, S. Meddings S. (2006). "Decision support systems for water pipeline renewal prioritization." *Journal of Information Technology in Construction*, Vol. 2, pp.237-256.

Muhlbauer, W. K. (2004). "Pipeline Risk Management Manual – Ideas, Techniques, and Resources, Third Edition." Elsevier Inc. 200 Wheeler Road, Burlington, MA 01803, USA.

Najafi, M. and Gokhale, S. (2005). "Trenchless Technology: Pipeline and Utility Design, Consstruction, and Renewal." McGraw-Hill Professional.

Parzen, E. (1962). "Stochastic Processes." Holden-Day, San Francisco.

Pelletier, G., Mailhot, A. and Villeneuve, J. (2003). "Modeling Water Pipe Breaks – Three Case Studies." Journal of Water Resources Planning and Management, March-April 2003, pp. 115-123.

Rahman S., Vanier D.J., and Newton, L. (2005). "Municipal infrastructure planning: Social Cost Considerations for Municipal Infrastructure Management, Report." National Research Council, Ottawa, Canada, NRC B-5123.8.

Rajani, B. and Makar, J. (2000). "A Methodology to Estimate Remaining Service Life of Grey Cast Iron Water Mains." Canadian Journal of Civil Engineering, 27, 1259-1272.

Rajani, B. B. and Kleiner, Y. (2001) "Comprehensive review of structural deterioration of water mains: physically based models." Urban Water, 3(3), 151-164.

Raterman, B., Schaars, F.W. and Griffioen, M. (2003). "GIS and MATLAB Integrated for Groundwater Modelling."

http://gis.esri.com/library/userconf/proc01/professional/papers/pap600/p600.htm

Reklaitis, G. V., Ravindran, A., and Ragsdell, K. M. (1983). "Engineering optimization, method and applications." John Wiley & Sons, Inc., New York.

Sadiq, R., Rajani, B. and Kleiner, Y. (2004). "Probabilistic Risk Analysis of corrosion Associated Failures in Cast Iron Water Mains." Reliability Engineering and System Safety, v. 86, pp. 1-10.

Schittkowski, K. (1980). "Nonlinear programming codes: information, tests, performance." Lecture notes in Economics and Mathematical systems, Vol. 183, Springer-Verlag, New York.

Schittkowski, K. (1985). "NLPQL: a FORTRAN subroutine solving constrained nonlinear programming problems." Annals of Operations Research, 5: 485–500.

Shamir, U. and Howard, C. D. D. (1979). "An Analytic Approach to Scheduling Pipe Replacement." Journal of American Water Works Association, 71(5), 248-258.

Sinske, S.A and Zietsman, H.L. (2004). "A Spatial Decision Support System for Pipe-break Susceptibility Analysis of Municipal Water Distribution Systems." Water SA, 30(1), 71-79.

Spangler, M. G. and Handy, R. L. (1982). "Soil Engineering: 4th edition." Harpercollins College Div

Stone, S., Dzuray, E. J. Meisegeier, D., Dahlborg, A. S. and Erickson, M. (2002). "Decision-Support Tools for Predicting the Performance of Water Distribution and Wastewater Collection Systems", Technical report EPA/600/R-02/029, U.S. Environmental Protection Agency

Tyler, A. R. (2007). "Expert systems research trends." Nova Publishers.

von Neumann, J., and Morgenstern, O. (1943). "Theory of games and economic behavior." Princeton Univ. Press, Princeton, NJ.

Walski, T. M. and Pelliccia, A. (1982). "Economic Analysis of Water Main Breaks." Journal of American Water Works Association, 74(3), 140-147.

Zhao, J. Q. and Rajani, B. (2002). "Construction and rehabilitation costs for buried pipe with a focus on trenchless technologies, Research Report No. 101." Institute for Research in Construction, National Research Council, Ottawa, Canada.

Appendix A Program implementation in VBA

There are several developed function and subroutine procedures, as well as the button controls and input controls of GUIs and Parameter Inputs dialog box, listed in this appendix. These function and subroutine procedures are:

1) Sub CalOptReplacementTime()
2) Function minCostMethod(FFX2 As Double, Z As Double) As Double
3) Sub LayerBackup()
4) Sub CreateOverviewWindow()
5) Sub AddTopRenderer(intBreakNum As Integer, strFieldName As String)
6) Sub AddProbRenderer(intBreakNum As Integer, strFieldName As String)
7) Sub ListTableFields(pFields As IFields)
8) Function FieldNameList(pFields As IFields) As Variant

Button controls of GUIs:
9) Sub OptimalReplacementTime_Click()
10) Sub OverView_Click()
11) Sub TopEAACRendererDisplay_Click()
12) Sub TopECRendererDisplay_Click()
13) Sub TopSHRendererDisplay_Click()
14) Sub ProbRendererDisplay_Click()
15) Sub AddStandAloneTable_Click()
16) Sub ShowFieldTitle_Click()

Parameter Inputs dialog box button controls and input controls:
17) Sub cmdClear_Click()
18) Sub cmdOK_Click()
19) Sub cmdQuit_Click()
20) Sub optCostData_Change()
21) Sub optCostData_Click()
22) Sub optCostRate_Change()
23) Sub OptCostRate_Click()
24) Sub txtCostRate_Change()
25) Sub txtCurrentYear_Change()
26) Sub txtDiscountRate_Change()
27) Sub txtPlanHorizon_Change()
28) Sub txtRepairCost_Change()
29) Sub txtRepairCost2_Change()
30) Sub txtReplacementCost_Change()

This listed subroutines, functions and user input form button controls are listed in the following.

```
Private Declare Sub test2 Lib "C:\Documents and Settings\Jackson\My
Documents\GIS\Thesis\Nlp\nlp_ver2_ok\dll\Release\TopTest1\Release\TopTe
st1.dll" (EPS2 As Double, ACC As Double, SCBOU As Double, MAXIT As
```

```
Integer, MAXFUN As Integer, IPRINT As Integer, N As Integer, M As
Integer, ME2 As Integer, IFAIL As Integer, MODE As Integer, FFX2 As
Double, X As Double, XL As Double, XU As Double, Z As Double)

Public Sub CalOptReplacementTime()
'Parent subroutine: Private Sub cmdOK_Click() @ User_Inputs
'2008-07-24: added input options for the cost inputs.

    'time stamp
    Dim strTimeStamp As String
    strTimeStamp = Format(Now, "yyyymmddhhmm")

    'Open a new text file to write to
    Open "./outfile" & "(" & strTimeStamp & ").csv" For Output As #1

    ' The arc document
    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' the Layers in the document
    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMxDoc.SelectedLayer

    ' the feature class in the layer
    Dim pFClass As IFeatureClass
    Set pFClass = pFLayer.FeatureClass

    'the fields in the attribute table
    Dim pFields As IFields
    Set pFields = pFClass.Fields

    'the index of the new field
    Dim intPosProb As Integer
    intPosProb = pFields.FindField("Prob")
    Dim intPosTopSH As Integer
    intPosTopSH = pFields.FindField("TopSH")
    Dim intPosTopEC As Integer
    intPosTopEC = pFields.FindField("TopEC")
    Dim intPosTopEAAC As Integer
    intPosTopEAAC = pFields.FindField("TopEAAC")
    Dim intPosCostSH As Integer
    intPosCostSH = pFields.FindField("CostSH")
    Dim intPosCostEC As Integer
    intPosCostEC = pFields.FindField("CostEC")
    Dim intPosCostEAAC As Integer
    intPosCostEAAC = pFields.FindField("CostEAAC")

    'delcare parameters
    Dim presentYear, installYear As Single
    Dim dbtTopSH, probability As Double
    Dim dbtA, dbtNamda, dbtCritBrkRate, dbtCostRate, dbtHorizon As
Double
    Dim dbtRplCost, dbtRprCost, dbtDisRate As Currency
    Dim strDia, strLength, strMaterial, strOpYear, strSoil As String
    Dim strTopSH, strTopEC, strTopEAAC, strCostSH, strCostEC,
strCostEAAC As String
```

```
'pompt for values for parameters from user
With User_Inputs
    presentYear = .txtCurrentYear.Value
    dbtHorizon = .txtPlanHorizon.Value
    dbtDisRate = .txtDiscountRate.Value
    If .optCostData.Value Then
        dbtRprCost = .txtRepairCost.Value
        dbtRplCost = .txtReplacementCost.Value
        dbtCostRate = dbtRprCost / dbtRplCost
    ElseIf .optCostRate.Value Then
        dbtRprCost = .txtRepairCost2.Value
        dbtCostRate = .txtCostRate.Value
    End If
End With

'set Field Names to strings variables
strDia = "WAT_SIZE"
strLength = "Length"
strMaterial = "Material"
strOpYear = "Age"
strSoil = "Soil_Type"
strProb = "Prob"
strTopSH = "TopSH"
strTopEC = "TopEC"
strTopEAAC = "TopEAAC"
strCostSH = "CostSH"
strCostEC = "CostEC"
strCostEAAC = "CostEAAC"

' set up type and information for each Field if not exist
Dim pFieldEdit As IFieldEdit

' if the feild (Prob) doesn't exist
If intPosProb = -1 Then

   'create a new field object
   Set pFieldEdit = New Field

   'set the field properties
   pFieldEdit.Name = strProb
   pFieldEdit.Type = esriFieldTypeDouble

   'add the field
   pFClass.AddField pFieldEdit

   'reset the field index, because it now exists
   intPosProb = pFields.FindField(strProb)

End If

' if the feild (TopSH) doesn't exist
If intPosTopSH = -1 Then

   'create a new field object
   Set pFieldEdit = New Field

   'set the field properties
```

```
        pFieldEdit.Name = strTopSH
        pFieldEdit.Type = esriFieldTypeDouble

        'add the field
        pFClass.AddField pFieldEdit

        'reset the field index, because it now exists
        intPosTopSH = pFields.FindField(strTopSH)

    End If

    ' if the feild (TopEC) doesn't exist
    If intPosTopEC = -1 Then

        'create a new field object
        Set pFieldEdit = New Field

        'set the field properties
        pFieldEdit.Name = strTopEC
        pFieldEdit.Type = esriFieldTypeDouble

        'add the field
        pFClass.AddField pFieldEdit

        'reset the field index, because it now exists
        intPosTopEC = pFields.FindField(strTopEC)

    End If

    ' if the feild (TopEAAC) doesn't exist
    If intPosTopEAAC = -1 Then

        'create a new field object
        Set pFieldEdit = New Field

        'set the field properties
        pFieldEdit.Name = strTopEAAC
        pFieldEdit.Type = esriFieldTypeDouble

        'add the field
        pFClass.AddField pFieldEdit

        'reset the field index, because it now exists
        intPosTopEAAC = pFields.FindField(strTopEAAC)

    End If

    ' if the feild (CostSH) doesn't exist
    If intPosCostSH = -1 Then

        'create a new field object
        Set pFieldEdit = New Field

        'set the field properties
        pFieldEdit.Name = strCostSH
        pFieldEdit.Type = esriFieldTypeDouble
```

```
   'add the field
   pFClass.AddField pFieldEdit

   'reset the field index, because it now exists
   intPosCostSH = pFields.FindField(strCostSH)

End If

' if the feild (CostEC) doesn't exist
If intPosCostEC = -1 Then

    'create a new field object
    Set pFieldEdit = New Field

    'set the field properties
    pFieldEdit.Name = strCostEC
    pFieldEdit.Type = esriFieldTypeDouble

    'add the field
    pFClass.AddField pFieldEdit

    'reset the field index, because it now exists
    intPosCostEC = pFields.FindField(strCostEC)

End If

' if the feild (CostEAAC) doesn't exist
If intPosCostEAAC = -1 Then

    'create a new field object
    Set pFieldEdit = New Field

    'set the field properties
    pFieldEdit.Name = strCostEAAC
    pFieldEdit.Type = esriFieldTypeDouble

    'add the field
    pFClass.AddField pFieldEdit

    'reset the field index, because it now exists
    intPosCostEAAC = pFields.FindField(strCostEAAC)

End If


'get field position for water mains size
Dim intPosWAT_SIZE As Integer
intPosWAT_SIZE = pFields.FindField(strDia)

'get field position for water mains Length
Dim intPosLength As Integer
intPosLength = pFields.FindField(strLength)

'get field position for water mains year of operation
Dim intPosAge As Integer
intPosAge = pFields.FindField(strOpYear)
```

```
'get field position for soil type
Dim intPosSoil_Type As Integer
intPosSoil_Type = pFields.FindField(strSoil)

'get field position for pipe material
Dim intPosMaterial As Integer
intPosMaterial = pFields.FindField(strMaterial)

're-define field position for probability of pipe failure
intPosProb = pFields.FindField(strProb)

're-define field position for optimal replacement time (SH)
intPosTopSH = pFields.FindField(strTopSH)

're-define field position for optimal replacement time (EC)
intPosTopEC = pFields.FindField(strTopEC)

're-define field position for optimal replacement time (EAAC)
intPosTopEAAC = pFields.FindField(strTopEAAC)

're-define field position for Expected Cost of (SH)
intPosCostSH = pFields.FindField(strCostSH)

're-define field position for Expected Cost of (EC)
intPosCostEC = pFields.FindField(strCostEC)

're-define field position for Expected Cost of (EAAC)
intPosCostEAAC = pFields.FindField(strCostEAAC)

'Get cursor points to the Feature Class
Dim pFCursor As IFeatureCursor
Set pFCursor = pFClass.Update(Nothing, False)

'move cursor to the first row
Dim pFeature As IFeature
Set pFeature = pFCursor.NextFeature

'progress bar set up
Dim lNumPipe, lProPosition As Long
Dim dProInterval As Double

'define a status bar at the bottom of the application window
Dim pSBar As IStatusBar
Set pSBar = StatusBar
Dim pProgressor As IStepProgressor
Set pProgressor = pSBar.ProgressBar

lNumPipe = pFClass.FeatureCount(Nothing)
dinterval = lNumPipe / 100
pProgressor.MinRange = 1
pProgressor.MaxRange = lNumPipe
pProgressor.StepValue = dinterval
lProPosition = 1

'define variable to record the execution time
Dim Start, Finish, TotalTime
```

```
    'start the timer
    Start = Timer

    'write headings to file #1
    Print #1, pFields.Field(0).Name;
    For ii = 1 To pFields.FieldCount - 1
        If pFields.Field(ii).Type <> esriFieldTypeGeometry Then
            Print #1, "," & pFields.Field(ii).Name;
        End If
    Next

    'For all the Feature in the attribute table of the feature class
(layer)
    Do Until pFeature Is Nothing

        'set current progress status and show progress bar
        pProgressor.position = lProPosition
        pProgressor.Step
        pProgressor.Show

        'define A and Namda
        With pFeature
            If .Value(intPosMaterial) = "CI" Then
                installYear = presentYear - .Value(intPosAge)
                Select Case installYear
                Case Is < 1930
                    Select Case .Value(intPosSoil_Type)
                    Case Is = "Clay"
                        dbtA = 0.188
                        dbtNamda = 0.000000142
                    Case Is = "Sand"
                        dbtA = 0.018
                        dbtNamda = 0.0162
                    Case Else
                        dbtA = 0.091
                        dbtNamda = 0.008
                    End Select
                Case 1930 To 1950
                        dbtA = 0.176
                        dbtNamda = 0.000012
                Case 1950 To 1960
                    Select Case .Value(intPosSoil_Type)
                    Case Is = "Clay"
                        dbtA = 0.024
                        dbtNamda = 0.0565
                    Case Is = "Sand"
                        dbtA = 0.044
                        dbtNamda = 0.0362
                    Case Else
                        dbtA = 0.074
                        dbtNamda = 0.07
                    End Select
                Case 1960 To 1980
                    Select Case .Value(intPosSoil_Type)
                    Case Is = "Clay"
                        dbtA = 0.104
                        dbtNamda = 0.0161
```

```
                Case Is = "Sand"
                    dbtA = 0.028
                    dbtNamda = 0.1232
                Case Else
                    dbtA = 0.065
                    dbtNamda = 0.045
                End Select
            Case Is > 1980
                    dbtA = 0.144
                    dbtNamda = 0.1041
            End Select

            'calculate Probability of Failure
            probability = 1# - Exp(-dbtNamda * Exp(dbtA *
(.Value(intPosAge) + 1)))
            .Value(intPosProb) = probability

            'calculate TopSH
            If .Value(intPosLength) = 0 Then
                .Value(intPosLength) = 1
            End If
            'Calculate TopSH with Cost Rate
            dbtTopSH = -.Value(intPosAge) + (Log(dbtCostRate *
dbtDisRate / dbtNamda) / dbtA)

            'write results to Field TopSH in the attribute table
            If dbtTopSH < 0 Then
                dbtTopSH = 0#
            End If
            .Value(intPosTopSH) = dbtTopSH

            'calculate the Expect Cost for TopSH
            .Value(intPosCostSH) = dbtRprCost * dbtNamda *
(Exp(dbtA * .Value(intPosAge)) _
                                        * (Exp((dbtA - dbtNamda) *
dbtTopSH) - 1) / (dbtA - dbtNamda) _
                                        + Exp(-dbtNamda * dbtTopSH) /
(dbtCostRate * dbtNamda) _
                                        + Exp(-dbtNamda * dbtTopSH) *
(Exp((dbtA - dbtNamda) _
                                        * (dbtHorizon - dbtTopSH)) - 1)
/ (dbtA - dbtNamda))

            '.Value(intPosCostSH) = dbtRprCost * Exp(-dbtNamda *
dbtTopSH)

            'calculate TopEC
            ReDim Z(1 To 31) As Double
            Dim FFX2 As Double

            'parameters for dll fille
            Z(1) = 1# 'Equation for TopEC
            Z(2) = dbtA
            Z(3) = dbtNamda
            Z(4) = dbtDisRate
            Z(5) = dbtCostRate
            ''Z(5) = dbtRplCost / dbtRprCost
```

```
                Z(6) = .Value(intPosAge)
                Z(7) = dbtHorizon

                'calculate TopEC with dll file
                .Value(intPosTopEC) = minCostMethod(FFX2, Z(1))

                'calculate Expected Cost for TopEC
                '.Value(intPosCostEC) = FFX2 * dbtRprCost * Z(3)
                .Value(intPosCostEC) = dbtRprCost * dbtNamda *
(Exp(dbtA * .Value(intPosAge)) _
                                        * (Exp((dbtA - dbtNamda) *
.Value(intPosTopEC)) - 1) / (dbtA - dbtNamda) _
                                        + Exp(-dbtNamda *
.Value(intPosTopEC)) / (dbtCostRate * dbtNamda) _
                                        + Exp(-dbtNamda *
.Value(intPosTopEC)) * (Exp((dbtA - dbtNamda) _
                                        * (dbtHorizon -
.Value(intPosTopEC))) - 1) / (dbtA - dbtNamda))

                'calculate TopEAAC woth dll file
                Z(1) = 2# 'Equation for TopEAAC
                .Value(intPosTopEAAC) = minCostMethod(FFX2, Z(1))
                If .Value(intPosTopEAAC) > 100 Then
                    .Value(intPosTopEAAC) = 100
                End If

                'calculate Expected Cost for TopEAAC
                .Value(intPosCostEAAC) = FFX2 * dbtRprCost * Z(3)

                'output all data of a record (row of data) in the
attribute table to file #1
                Print #1, .Value(0);
                For jj = 1 To pFields.FieldCount - 1
                    If pFields.Field(jj).Type <> esriFieldTypeGeometry
Then
                        Print #1, "," & .Value(jj);
                    End If
                Next

                'update values in the table
                pFCursor.UpdateFeature pFeature

            End If
        End With

        'move to next value in table
        Set pFeature = pFCursor.NextFeature
        lProPosition = lProPosition + 1

    Loop

    'Stop the timer
    Finish = Timer

    'Calculate total execution time (in seconds)
    TotalTime = Finish - Start
```

```
    'Checking execution time
    Debug.Print "Total Time: " & TotalTime & " seconds, and analysised
" _
                 & lProPosition - 1 & " of " & lNumPipe & " pipes
segments"

    'Close file #1
    Close #1

    'Refresh the data screen and the content on.the left side
    pMxDoc.ActiveView.Refresh
    pMxDoc.UpdateContents

    'Hide the status bar
    pProgressor.Hide

End Sub '-----------------------------------------

Public Function minCostMethod(FFX2 As Double, Z As Double) As Double
'Parent subroutine: CalOptReplacementTime()

    'parameter setup for dll
    EPS2 = 0.00000000000001
    ACC = 0.0000000001
    SCBOU = 100000#
    MAXIT = 1000
    MAXFUN = 10
    IPRINT = 2
    N = 1
    M = 0
    ME2 = 0
    IFAIL = 0
    MODE = 0

    ReDim X(1 To 31) As Double
    ReDim XL(1 To 31) As Double
    ReDim XU(1 To 31) As Double

    For i = 1 To N
        X(i) = 25#  'initial guess
        XL(i) = 0#  'lower bound
        XU(i) = 1000#   'upper bound
    Next

    Call test2(EPS2, ACC, SCBOU, MAXIT, MAXFUN, IPRINT, N, M, ME2,
IFAIL, MODE, FFX2, X(1), XL(1), XU(1), Z)

    'return value of TopEC or TopEAAC
    minCostMethod = X(1)

End Function '-------------------------------------------

Public Sub LayerBackup()
'Backup attribute table of selected layer
'2008-07-23 changed method of creating a output file name with a
current time stamp
```

```
    ' The arc document
    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' the Layers in the document
    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMxDoc.SelectedLayer

    'time stamp
    Dim strFileName As String
    strFileName = Format(Now, "yyyymmddhhmm")

    'Open a new text file to write to
    Open "./Backup_" & pFLayer.Name & " (" & strFileName & ").csv" For
Output As #1

    ' the feature class in the layer
    Dim pFClass As IFeatureClass
    Set pFClass = pFLayer.FeatureClass

    'the fields in the attribute table
    Dim pFields As IFields
    Set pFields = pFClass.Fields

    'Get cursor points to the Feature Class
    Dim pFCursor As IFeatureCursor
    Set pFCursor = pFClass.Update(Nothing, False)

    'move cursor to the first row
    Dim pFeature As IFeature
    Set pFeature = pFCursor.NextFeature

    Print #1, pFields.Field(0).Name;
    For ii = 1 To pFields.FieldCount - 1
        If pFields.Field(ii).Type <> esriFieldTypeGeometry Then
            Print #1, "," & pFields.Field(ii).Name;
        End If
    Next

    Print #1,

    Do Until pFeature Is Nothing

        With pFeature

        Print #1, .Value(0);
        For jj = 1 To pFields.FieldCount - 1
            If pFields.Field(jj).Type <> esriFieldTypeGeometry Then
                Print #1, "," & .Value(jj);
            End If
        Next
        Print #1,

        End With

        'move to next record in table
        Set pFeature = pFCursor.NextFeature
```

```
    Loop

    Close #1

End Sub '----------------------------------------------------

Public Sub CreateOverviewWindow()
'Parent subroutine:  OverView_Click() @ ThisDocument

    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument
    Dim pActiveView As IActiveView
    Set pActiveView = pMxDoc.ActiveView

    Dim pCommandBars As ICommandBars
    Set pCommandBars = ThisDocument.CommandBars

    Dim pCommandItem As ICommandItem

    If pActiveView.IsMapActivated Then
        Debug.Print "Map is Activated"

    Else
        Debug.Print "Map is NOT Activated"

        Set pCommandItem = _
            pCommandBars.Find(arcid.View_Geographic)
        pCommandItem.Execute

    End If

    pMxDoc.FocusMap.MapScale = 50000
    pMxDoc.ActiveView.Refresh
    pMxDoc.UpdateContents

    'Declare variables
    Dim pOverview As IOverview
    Dim pOverviewWindow As IOverviewWindow
    Dim pDataWindowFactory As IDataWindowFactory
    Dim pFillSymbol As ISimpleFillSymbol
    Dim pLineSymbol As ISimpleLineSymbol
    Dim pRgbColor As IRgbColor

    'Set Variables
    Set pDataWindowFactory = New OverviewWindowFactory
    If Not pDataWindowFactory.CanCreate(Application) Then Exit Sub

    'Create a new overview window
    Set pOverviewWindow = pDataWindowFactory.Create(Application)

    'Change the area of interest fill symbol
    'to a hollow fill with a blue border
    Set pOverview = pOverviewWindow.OverView
    Set pFillSymbol = New SimpleFillSymbol
    Set pLineSymbol = New SimpleLineSymbol
    Set pRgbColor = New RgbColor
```

```
    'Set object properties
    pRgbColor.Blue = 255
    pLineSymbol.Color = pRgbColor
    pFillSymbol.Style = esriSFSNull
    pFillSymbol.Outline = pLineSymbol
    pOverview.AoiFillSymbol = pFillSymbol

End Sub '-------------------------------------------

Public Sub AddTopRenderer(intBreakNum As Integer, strFieldName As
String)
'Parent subroutine: TopECRendererDisplay_Click() @ ThisDocument
'Parent subroutine: TopEAACRendererDisplay_Click() @ ThisDocument
'Parent subroutine: TopSHRendererDisplay_Click() @ ThisDocument

    Dim intColourChoice As String
    intColourChoice = InputBox("This function will add renderer to the
map of " _
                              & "the selected feature layer" _
                              & Chr(13) & Chr(13) _
                              & "Please choose a colour scale for the
renderer:" _
                              & Chr(13) & "1 for colour; 2 for gray",
_
                              "Add " & strFieldName & " Renderer")

    If intColourChoice = "" Then
        Exit Sub
    End If

    ' The arc document
    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' the Layers in the document
    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMxDoc.SelectedLayer

    'Add class breaks renderer code here.
    'create a class breaks renderer
    Dim pCBR As IClassBreaksRenderer
    Set pCBR = New ClassBreaksRenderer

    'set name of the render and number of classes
    pCBR.Field = strFieldName
    pCBR.BreakCount = intBreakNum

    'set the break point values with 0-5, 5-10,
    '10-20, 20-30, 30-40, 40-50, and 50-100 years
    pCBR.Break(0) = 5
    For i = 1 To intBreakNum - 2
        pCBR.Break(i) = 10 * i
    Next i
    pCBR.Break(intBreakNum - 1) = 100

    'set the legend label values for the 3 classes
```

```
    pCBR.Label(0) = "< " & pCBR.Break(0) & " years"
    For i = 1 To intBreakNum - 1
        pCBR.Label(i) = pCBR.Break(i - 1) & "-" & pCBR.Break(i)
    Next i
    pCBR.Label(intBreakNum - 1) = "> " & pCBR.Break(intBreakNum - 2)

    'make RGB color objects
    ReDim pRgbColor(0 To (intBreakNum - 1)) As IRgbColor
    'make Gray color objects
    ReDim pGrayColor(0 To (intBreakNum - 1)) As IGrayColor

    For i = 0 To intBreakNum - 1
        Set pRgbColor(i) = New RgbColor
    Next i
    For i = 0 To intBreakNum - 1
        Set pGrayColor(i) = New GrayColor
    Next i

    'set the color's RGB property with the RGB function
    For i = 0 To intBreakNum - 1
        pRgbColor(i).RGB = RGB(225 - i * (225 / intBreakNum), 225 /
intBreakNum * i, 225 / intBreakNum)
    Next i
    'set the gray color levels
    For i = 0 To intBreakNum - 1
        pGrayColor(i).Level = 255 * (i / intBreakNum)
    Next i

    'set the fill symbol for each class with different level of gray
(or colour)
    Dim pLine As ISimpleLineSymbol
    Set pLine = New SimpleLineSymbol

    'define proporties of each rendener class
    For i = 0 To intBreakNum - 1
        If intColourChoice = "1" Then
            pLine.Color = pRgbColor(i)
        ElseIf intColourChoice = "2" Then
            pLine.Color = pGrayColor(i)
        End If
        pLine.Width = 0.01
        pCBR.Symbol(i) = pLine
    Next i

    'assign the renderer to the Feature layer
    Dim pGFLayer As IGeoFeatureLayer
    Set pGFLayer = pFLayer

    Set pGFLayer.Renderer = pCBR

    pFLayer.Name = "Optimal Replacement Time (" & strFieldName & ")"

    Dim pLayerDef As IFeatureLayerDefinition
    Set pLayerDef = pFLayer

    'display the map with only CI waterpipes
    pLayerDef.DefinitionExpression = "Material = 'CI'"
```

```
    'Refresh the map by changing views between data view and layout
view
    Dim pCommandBars As ICommandBars
    Set pCommandBars = ThisDocument.CommandBars

    Dim pCommandItem As ICommandItem

    Set pCommandItem = _
        pCommandBars.Find(arcid.View_Geographic)
    pCommandItem.Execute

    Set pCommandItem = _
        pCommandBars.Find(arcid.View_LayoutView)
    pCommandItem.Execute

    pMxDoc.ActiveView.Refresh
    pMxDoc.UpdateContents

End Sub '-------------------------------------------

Public Sub AddProbRenderer(intBreakNum As Integer, strFieldName As
String)
'Parent subroutine: ProbRendererDisplay_Click() @ ThisDocument

    Dim intColourChoice As String
    intColourChoice = InputBox("This function will add renderer to the
map of " _
                                & "the selected feature layer" _
                                & Chr(13) & Chr(13) _
                                & "Please choose a colour scale for the
renderer:" _
                                & Chr(13) & "1 for colour; 2 for gray",
_
                                "Add Probability of Failure Renderer")
    If intColourChoice = "" Then
        Exit Sub
    End If

    ' The arc document
    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' the Layers in the document
    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMxDoc.SelectedLayer

    'create a class breaks renderer
    Dim pCBR As IClassBreaksRenderer
    Set pCBR = New ClassBreaksRenderer

    'set name of the render and number of classes
    pCBR.Field = strFieldName
    pCBR.BreakCount = intBreakNum

    'set the break point values
    For i = 0 To intBreakNum - 1
```

```
        pCBR.Break(i) = (1 + i) * 1 / intBreakNum
    Next i

    'set the legend label values for the N classes
    pCBR.Label(0) = "< " & pCBR.Break(0)
    For i = 1 To intBreakNum - 1
        pCBR.Label(i) = pCBR.Break(i - 1) & "-" & pCBR.Break(i)
    Next i

    'make RGB color objects
    ReDim pRgbColor(0 To (intBreakNum - 1)) As IRgbColor
    'make Gray color objects
    ReDim pGrayColor(0 To (intBreakNum - 1)) As IGrayColor

    For i = 0 To intBreakNum - 1
        Set pRgbColor(i) = New RgbColor
    Next i
    For i = 0 To intBreakNum - 1
        Set pGrayColor(i) = New GrayColor
    Next i

    'set the color's RGB property with the RGB function
    For i = 0 To intBreakNum - 1
        pRgbColor(i).RGB = RGB(225, 25 * i, 15 * 1)
    Next i
    'set the gray color levels
    For i = 0 To intBreakNum - 1
        pGrayColor(i).Level = 255 * (1 - (i / intBreakNum))
    Next i

    'set the fill symbol for each class with different level of gray
(or colour)
    Dim pLine As ISimpleLineSymbol
    Set pLine = New SimpleLineSymbol

    'define proporties of each rendener class
    For i = 0 To intBreakNum - 1
        If intColourChoice = "1" Then
            pLine.Color = pRgbColor(i)
        ElseIf intColourChoice = "2" Then
            pLine.Color = pGrayColor(i)
        End If
        pLine.Width = 0.01
        pCBR.Symbol(i) = pLine
    Next i

    'assign the renderer to the Feature layer
    Dim pGFLayer As IGeoFeatureLayer
    Set pGFLayer = pFLayer

    Set pGFLayer.Renderer = pCBR

    pFLayer.Name = "Probability of Failure"

    Dim pLayerDef As IFeatureLayerDefinition
    Set pLayerDef = pFLayer
```

```
    'display the map with only CI waterpipes
    pLayerDef.DefinitionExpression = "Material = 'CI'"

    'Refresh the map by changing views between data view and layout
view
    Dim pCommandBars As ICommandBars
    Set pCommandBars = ThisDocument.CommandBars

    Dim pCommandItem As ICommandItem

    Set pCommandItem = _
        pCommandBars.Find(arcid.View_Geographic)
    pCommandItem.Execute

    Set pCommandItem = _
        pCommandBars.Find(arcid.View_LayoutView)
    pCommandItem.Execute

    pMxDoc.ActiveView.Refresh
    pMxDoc.UpdateContents

End Sub '-----------------------------------------

Public Sub ListTableFields(pFields As IFields)
'Parent subroutine: ShowFieldTitle_Click() @ ThisDocument

    Dim NameList As Variant
    NameList = FieldNameList(pFields)
    MsgBox NameList, , "Field Names"

End Sub '-----------------------------------------

Public Function FieldNameList(pFields As IFields) As Variant
'Parent subroutine: ListTableFields(pFields As IFields)

    Dim count As Integer
    count = pFields.FieldCount

    Dim ii As Integer
    Dim pField As IField
    Dim fieldName As Variant
    Dim theList As New Collection

    For ii = 0 To pFields.FieldCount - 1
        Set pField = pFields.Field(ii)
        fieldName = pField.Name
        Select Case pFields.Field(ii).Type
            Case esriFieldTypeSmallInteger
                fieldName = fieldName & " (Integer)"
            Case esriFieldTypeInteger
                fieldName = fieldName & " (Long)"
            Case esriFieldTypeSingle
                fieldName = fieldName & " (Single)"
            Case esriFieldTypeDouble
                fieldName = fieldName & " (Double)"
            Case esriFieldTypeString
                fieldName = fieldName & " (String)"
```

```
            Case esriFieldTypeDate
                fieldName = fieldName & " (Date)"
            Case esriFieldTypeOID
                fieldName = fieldName & " (Object ID)"
            Case esriFieldTypeGeometry
                fieldName = fieldName & " (Geometry)"
            Case Else
                fieldName = fieldName & " (" & pFields.Field(ii).Type &
")"
        End Select
        theList.Add (fieldName)
    Next

        FieldNameList = FieldNameList & "There are " & count & "
fields:" & Chr(13) & Chr(13)
    For Each fieldName In theList
        FieldNameList = FieldNameList & fieldName & Chr(13)
    Next fieldName

End Function '-------------------------------------------

Private Sub OptimalReplacementTime_Click()

    With User_Inputs
        If (IsNumeric(.txtCurrentYear.Text) And _
            IsNumeric(.txtPlanHorizon.Text) And _
            IsNumeric(.txtDiscountRate.Text) And _
            IsNumeric(.txtRepairCost.Text) And _
            IsNumeric(.txtReplacementCost.Text) And _
            .optCostData.Value) Then
            .cmdOK.Enabled = True
        ElseIf (IsNumeric(.txtCurrentYear.Text) And _
            IsNumeric(.txtPlanHorizon.Text) And _
            IsNumeric(.txtDiscountRate.Text) And _
            IsNumeric(.txtCostRate.Text) And _
            IsNumeric(.txtRepairCost2.Text) And _
            .optCostRate.Value) Then
            .cmdOK.Enabled = True
        Else
            .cmdOK.Enabled = False
        End If
        If .optCostData.Value Then
            .txtCostRate.Enabled = False
            .txtRepairCost2.Enabled = False
            .txtRepairCost.Enabled = True
            .txtReplacementCost.Enabled = True
            .txtCostRate.BackColor = RGB(170, 170, 170)
            .txtCostRate.BackStyle = fmBackStyleOpaque
            .txtRepairCost2.BackColor = RGB(170, 170, 170)
            .txtRepairCost2.BackStyle = fmBackStyleOpaque
            .txtRepairCost.BackColor = vbWhite
            .txtRepairCost.BackStyle = fmBackStyleTransparent
            .txtReplacementCost.BackColor = vbWhite
            .txtReplacementCost.BackStyle = fmBackStyleTransparent
        ElseIf .optCostRate.Value Then
            .txtCostRate.Enabled = True
            .txtRepairCost2.Enabled = True
```

```
            .txtRepairCost.Enabled = False
            .txtReplacementCost.Enabled = False
            .txtCostRate.BackColor = vbWhite
            .txtCostRate.BackStyle = fmBackStyleTransparent
            .txtRepairCost2.BackColor = vbWhite
            .txtRepairCost2.BackStyle = fmBackStyleTransparent
            .txtRepairCost.BackColor = RGB(170, 170, 170)
            .txtRepairCost.BackStyle = fmBackStyleOpaque
            .txtReplacementCost.BackColor = RGB(170, 170, 170)
            .txtReplacementCost.BackStyle = fmBackStyleOpaque
        End If
        '.optCostData.Enabled = True
        '.optCostRate.Enabled = True
        '.txtCostRate.Enabled = True
        '.txtRepairCost2.Enabled = True
        '.txtRepairCost.Enabled = True
        '.txtReplacementCost.Enabled = True
        '.txtCostRate.BackStyle = fmBackStyleTransparent
        '.txtRepairCost2.BackStyle = fmBackStyleTransparent
        '.txtRepairCost.BackStyle = fmBackStyleTransparent
        '.txtReplacementCost.BackStyle = fmBackStyleTransparent
        .Show
    End With

End Sub '--------------------------------------------

Private Sub OverView_Click()

    Call CreateOverviewWindow

End Sub '--------------------------------------------

Private Sub TopEAACRendererDisplay_Click()

    Call AddTopRenderer(7, "TopEAAC")

End Sub '--------------------------------------------

Private Sub TopECRendererDisplay_Click()

    Call AddTopRenderer(7, "TopEC")

End Sub '--------------------------------------------

Private Sub TopSHRendererDisplay_Click()

    Call AddTopRenderer(7, "TopSH")

End Sub '--------------------------------------------
Private Sub ProbRendererDisplay_Click()

    Call AddProbRenderer(10, "Prob")

End Sub '--------------------------------------------

Private Sub AddStandAloneTable_Click()
```

```
'2008-07-23 added input dialog box feature; improved the method to get
the input table;
'           change array system: set max number of items in arrays,
looping only according
'                                    the exact number of column of the
input table;
'           remove the added standalone table when the job is done
'2008-07-25  fixed bug for the cancel click of the "Add Data" input
dialog box;
'            changed from adding the standalone table from the top of
the list to the
'            bottom of the list;
'            added code at the end of the sub to refresh the contenct
window

    ' The arc document
    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' the Layers in the document
    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMxDoc.ContextItem

    ' the feature class in the layer
    Dim pFClass As IFeatureClass
    Set pFClass = pFLayer.FeatureClass

    'the fields in the attribute table
    Dim pFields As IFields
    Set pFields = pFClass.Fields

    MsgBox pFLayer.Name & " has " & pFClass.FeatureCount(Nothing) & "
records(rows)"
    Call ListTableFields(pFields)

    'Prompt a input dialog box for User to choose a standalone table
    Dim pGxDialog As IGxDialog
    Set pGxDialog = New GxDialog

    'Set input dialog box messages and open location
    pGxDialog.ButtonCaption = "Add"
    pGxDialog.StartingLocation = ".\"
    pGxDialog.Title = "Add Data"

    'only one data file is allowed
    pGxDialog.AllowMultiSelect = False

    'define the file type to be shown and for selection
    Dim pTFilter As IGxObjectFilter
    Set pTFilter = New GxFilterTables

    Set pGxDialog.ObjectFilter = pTFilter

    Dim pTableFiles As IEnumGxObject

    pGxDialog.DoModalOpen 0, pTableFiles
```

```
    ' Get the ITable from the geodatabase
    Dim pFact As IWorkspaceFactory
    Set pFact = New TextFileWorkspaceFactory

    Dim pWorkspace As IWorkspace
    Set pWorkspace =
pFact.OpenFromFile(pGxDialog.FinalLocation.FullName, 0)

    Dim pFeatws As IFeatureWorkspace
    Set pFeatws = pWorkspace

    'if cancel button is hitted, pTableFiles.Next is nothing, then exit
sub
    If pTableFiles.Next Is Nothing Then
        Exit Sub
    Else
        pTableFiles.Reset 'reset the cursor of pTableFiles to the top
    End If

    Dim pInputTable As ITable
    Set pInputTable = pFeatws.OpenTable(pTableFiles.Next.Name)

    ' Add the table
    Add_Table_TOC pInputTable

    Dim pMap As IMap
    Set pMap = pMxDoc.FocusMap

    Dim pStTableCollection As IStandaloneTableCollection
    Set pStTableCollection = pMap

    'last standalone table position
    Dim tablePostion
    tablePosition = pStTableCollection.StandaloneTableCount - 1

    'last standalone table in the list
    Dim pStTable As IStandaloneTable
    Set pStTable = pStTableCollection.StandaloneTable(tablePosition)

    Dim pTable As ITable
    Set pTable = pStTable.Table

    ''MsgBox pStTable.Name & " has " & pTable.RowCount(Nothing) & "
records(rows)"
    ''Call ListTableFields(pTable.Fields)

    'Set base field for joining standalone table
    Dim strBaseField_Table As String
    strBaseField_Table = pTable.Fields.Field(0).Name

    'temp assumed that same base field name for both table and layer
    Dim strBaseField_Layer As String
    strBaseField_Layer = strBaseField_Table

    Dim intPosBaseField_Layer As Integer
    intPosBaseField_Layer = pFields.FindField(strBaseField_Layer)
```

```
    Dim intTableFieldNum
    intTableFieldNum = pTable.Fields.FieldCount

    'set maximum number of fields (columns) to be added
    Const maxNumColumn = 20

    Dim strAddField_Table(1 To maxNumColumn) As String
    Dim intPosAddField_Table(1 To maxNumColumn) As Integer
    Dim intPosAddField_Layer(1 To maxNumColumn) As Integer
    Dim pFieldEdit(1 To maxNumColumn) As IFieldEdit

    For i = 1 To intTableFieldNum - 1

        strAddField_Table(i) = Mid(pTable.Fields.Field(i).Name, 1, 10)
        intPosAddField_Table(i) =
pTable.Fields.FindField(pTable.Fields.Field(i).Name)
        intPosAddField_Layer(i) =
pFields.FindField(strAddField_Table(i))

        If intPosAddField_Layer(i) = -1 Then

            'create a new field object
            Set pFieldEdit(i) = New Field

            'set the field properties
            pFieldEdit(i).Name = strAddField_Table(i)
            pFieldEdit(i).Type = pTable.Fields.Field(i).Type
            pFieldEdit(i).Length = 10

            'add the field to the attribute table
            pFClass.AddField pFieldEdit(i)

            'reset the field index after the new field is created
            intPosAddField_Layer(i) =
pFields.FindField(strAddField_Table(i))

        End If
    Next i

    'Add items from Standalone table to attribute table of selected
feature layer
    Dim pFCursor As IFeatureCursor
    Set pFCursor = pFClass.Update(Nothing, False)

    Dim pFeature As IFeature
    Set pFeature = pFCursor.NextFeature

    Dim pTCursor As ICursor
    Set pTCursor = pTable.Search(Nothing, False)

    Dim pSATFeature As IRow
    Set pSATFeature = pTCursor.NextRow

    Dim Start, Finish, TotalTime
    Start = Timer

    Do Until pSATFeature Is Nothing
```

```
        For i = 1 To intTableFieldNum - 1
            pFeature.Value(intPosAddField_Layer(i)) =
pSATFeature.Value(intPosAddField_Table(i))
            pFCursor.UpdateFeature pFeature
            TotalTime = Finish - Start
        Next i

        Set pFeature = pFCursor.NextFeature
        Set pSATFeature = pTCursor.NextRow

    Loop

    'remove the inserted standalone table
    pStTableCollection.RemoveStandaloneTable pStTable

    'Refresh the content on the left side
    pMxDoc.UpdateContents

    Finish = Timer
    TotalTime = Finish - Start
    MsgBox "Adding data for " & TotalTime & " seconds"

End Sub '-------------------------------------------

Private Sub ShowFieldTitle_Click()

    ' The arc document
    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' the Layers in the document
    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMxDoc.ContextItem

    ' the feature class in the layer
    Dim pFClass As IFeatureClass
    Set pFClass = pFLayer.FeatureClass

    'the fields in the attribute table
    Dim pFields As IFields
    Set pFields = pFClass.Fields

    MsgBox pFLayer.Name & " has " & pFClass.FeatureCount(Nothing) & "
records(rows)"
    Call ListTableFields(pFields)

End Sub '-------------------------------------------

Private Sub cmdClear_Click()
    txtCurrentYear.Text = " "
    txtPlanHorizon.Text = " "
    txtDiscountRate.Text = " "
    txtRepairCost.Text = " "
    txtReplacementCost.Text = " "
    txtCostRate.Text = " "
    optCostData.Enabled = True
```

```
    optCostRate.Enabled = True
    optCostData.Value = 0
    optCostRate.Value = 0
    txtCostRate.Enabled = True
    txtRepairCost.Enabled = True
    txtReplacementCost.Enabled = True
    txtRepairCost2.Enabled = True
    txtCostRate.BackStyle = fmBackStyleTransparent
    txtRepairCost.BackStyle = fmBackStyleTransparent
    txtReplacementCost.BackStyle = fmBackStyleTransparent
    txtRepairCost2.BackStyle = fmBackStyleTransparent
End Sub '--------------------------------------------

Private Sub cmdOK_Click()
    Call CalOptReplacementTime
    User_Inputs.Hide
End Sub '--------------------------------------------

Private Sub cmdQuit_Click()
    User_Inputs.Hide
End Sub '--------------------------------------------


Private Sub optCostData_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub '--------------------------------------------

Private Sub optCostData_Click()
    optCostRate.Value = 0
    txtCostRate.Enabled = False
    txtRepairCost2.Enabled = False
    txtRepairCost.Enabled = True
    txtReplacementCost.Enabled = True
    txtCostRate.BackColor = RGB(170, 170, 170)
    txtCostRate.BackStyle = fmBackStyleOpaque
    txtRepairCost2.BackColor = RGB(170, 170, 170)
    txtRepairCost2.BackStyle = fmBackStyleOpaque
    txtRepairCost.BackColor = vbWhite
    txtRepairCost.BackStyle = fmBackStyleTransparent
    txtReplacementCost.BackColor = vbWhite
    txtReplacementCost.BackStyle = fmBackStyleTransparent
```

```
End Sub '------------------------------------------

Private Sub optCostRate_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub '------------------------------------------

Private Sub OptCostRate_Click()
    optCostData.Value = 0
    txtCostRate.Enabled = True
    txtRepairCost2.Enabled = True
    txtRepairCost.Enabled = False
    txtReplacementCost.Enabled = False
    txtCostRate.BackColor = vbWhite
    txtCostRate.BackStyle = fmBackStyleTransparent
    txtRepairCost2.BackColor = vbWhite
    txtRepairCost2.BackStyle = fmBackStyleTransparent
    txtRepairCost.BackColor = RGB(170, 170, 170)
    txtRepairCost.BackStyle = fmBackStyleOpaque
    txtReplacementCost.BackColor = RGB(170, 170, 170)
    txtReplacementCost.BackStyle = fmBackStyleOpaque
End Sub '------------------------------------------

Private Sub txtCostRate_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub '------------------------------------------
```

```
Private Sub txtCurrentYear_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub '-----------------------------------------

Private Sub txtDiscountRate_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub '-----------------------------------------

Private Sub txtPlanHorizon_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
```

```
        cmdOK.Enabled = False
    End If
End Sub '-----------------------------------------------

Private Sub txtRepairCost_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub '-----------------------------------------------

Private Sub txtRepairCost2_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub '-----------------------------------------------

Private Sub txtReplacementCost_Change()
    If (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtRepairCost.Text) And _
        IsNumeric(txtReplacementCost.Text) And _
        optCostData.Value) Then
        cmdOK.Enabled = True
    ElseIf (IsNumeric(txtCurrentYear.Text) And _
        IsNumeric(txtPlanHorizon.Text) And _
        IsNumeric(txtDiscountRate.Text) And _
        IsNumeric(txtCostRate.Text) And _
        IsNumeric(txtRepairCost2.Text) And _
```

```
        optCostRate.Value) Then
        cmdOK.Enabled = True
    Else
        cmdOK.Enabled = False
    End If
End Sub '-----------------------------------------
```

**** Programming codes Optimal Replacement time calculations implemented with Excel spreadsheet

```
Private Sub OptimalReplacementTime_Click()
    ' The arc document
    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' the Layers in the document
    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMxDoc.ContextItem

    ' the feature class in the layer
    Dim pFClass As IFeatureClass
    Set pFClass = pFLayer.FeatureClass

    'the fields in the attribute table
    Dim pFields As IFields
    Set pFields = pFClass.Fields

    'the index of the new field
    Dim intPosOptRplTime As Integer
    intPosOptRplTime = pFields.FindField("OptRplTime")

    ' if the feild doesn't exist
    If intPosInput = -1 Then

       'create a new field object
       Dim pFieldEdit As IFieldEdit
       Set pFieldEdit = New Field

       'set the field properties
       pFieldEdit.Name = "OptRplTime"
       pFieldEdit.Type = esriFieldTypeInteger

       'add the field
       pFClass.AddField pFieldEdit

       'reset the field index, because it now exists
       intPosOptRplTime = pFields.FindField("OptRplTime")

    End If

    Dim presentYear, installYear, optRplYear As Single
    Dim dbtA, dbtNamda, dbtCritBrkRate As Double
    Dim dbtRplCost, dbtRprCost, dbtDisRate As Currency
    Dim strDia, strLength, strMaterial_1, strOpYear, strSoil,
strOptRplTime As String

    presentYear = 2007
```

```
dbtRplCost = 442
dbtRprCost = 5780
dbtDisRate = 0.05

dbtCritBrkRate = dbtRplCost / dbtRprCost * dbtDisRate

strDia = "WAT_SIZE"
strLength = "Length"
strMaterial_1 = "Material_1"
strOpYear = "Age"
strSoil = "Soil_Type"
strOptRplTime = "OptRplTime"

Debug.Print presentYear & ", " & dbtCritBrkRate

'define field position for water mains size
Dim intPosWAT_SIZE As Integer
intPosWAT_SIZE = pFields.FindField(strDia)

'define field position for water mains Length
Dim intPosLength As Integer
intPosLength = pFields.FindField(strLength)

'define field position for water mains year of operation
Dim intPosAge As Integer
intPosAge = pFields.FindField(strOpYear)

'define field position for soil type
Dim intPosSoil_Type As Integer
intPosSoil_Type = pFields.FindField(strSoil)

're-define field position for optimal replacement time
intPosOptRplTime = pFields.FindField(strOptRplTime)

'define field position for soil type
Dim intPosMaterial_1 As Integer
intPosMaterial_1 = pFields.FindField(strMaterial_1)

'Get cursor points to the Feature Class
Dim pFCursor As IFeatureCursor
Set pFCursor = pFClass.Update(Nothing, False)

'move cursor to the first row
Dim pFeature As IFeature
Set pFeature = pFCursor.NextFeature

Dim Start, Finish, TotalTime

'Dim i As Integer

Start = Timer

Do Until pFeature Is Nothing

    'define A and Namda
    With pFeature
        If .Value(intPosMaterial_1) = "CI" Then
```

```
                installYear = presentYear - .Value(intPosAge)
                Select Case installYear
                Case Is < 1930
                    'Debug.Print installYear & " is earlier than 1930,
Material is " & .Value(intPosMaterial_1)
                    Select Case .Value(intPosSoil_Type)
                    Case Is = "Clay"
                        dbtA = 0.188
                        dbtNamda = 0.00000014
                    Case Is = "Sand"
                        dbtA = 0.018
                        dbtNamda = 0.0162
                    Case Else
                        dbtA = 0.02
                        dbtNamda = 0.02
                    End Select
                Case 1930 To 1950
                    'Debug.Print installYear & " is between 1930 and
1950, Material is " & .Value(intPosMaterial_1)
                        dbtA = 0.176
                        dbtNamda = 0.00000014
                Case 1950 To 1960
                    'Debug.Print installYear & " is between 1950 and
1960, Material is " & .Value(intPosMaterial_1)
                    Select Case .Value(intPosSoil_Type)
                    Case Is = "Clay"
                        dbtA = 0.024
                        dbtNamda = 0.0565
                    Case Is = "Sand"
                        dbtA = 0.044
                        dbtNamda = 0.0362
                    Case Else
                        dbtA = 0.02
                        dbtNamda = 0.02
                    End Select
                Case 1960 To 1980
                    'Debug.Print installYear & " is between 1960 and
1980, Material is " & .Value(intPosMaterial_1)
                    Select Case .Value(intPosSoil_Type)
                    Case Is = "Clay"
                        dbtA = 0.104
                        dbtNamda = 0.0161
                    Case Is = "Sand"
                        dbtA = 0.028
                        dbtNamda = 0.1232
                    Case Else
                        dbtA = 0.02
                        dbtNamda = 0.02
                    End Select
                Case Is > 1980
                    'Debug.Print installYear & " is after 1980,
Material is " & .Value(intPosMaterial_1)
                        dbtA = 0.144
                        dbtNamda = 0.1041
                End Select
                If .Value(intPosLength) = 0 Then
                    .Value(intPosLength) = 1
```

```
                End If
                'optRplYear = Int(Rnd * (120 - .Value(intPosAge)))
                'optRplYear = -.Value(intPosAge) + (Log(dbtRplCost *
.Value(intPosLength) / dbtRprCost / dbtNamda) / Log(2.718282)) / dbtA
                optRplYear = -.Value(intPosAge) + Log(40 * 0.05 /
dbtNamda) / dbtA
                .Value(intPosOptRplTime) = optRplYear

                'Debug.Print i & ". Install in " & installYear & ", A =
" & dbtA & ", Namda = " & dbtNamda & _
                          ", Replaced in " & optRplYear
                'Debug.Print i & ". Replaced in " & installYear &
.Value(intPosOptRplTime)

                pFCursor.UpdateFeature pFeature

            End If
        End With

        'move to next value in table
        Set pFeature = pFCursor.NextFeature

    Loop

    Finish = Timer

    TotalTime = Finish - Start

    Debug.Print TotalTime

End Sub
```