SMILE Visualization with Flash Technologies

Anucha Tungkasthan[1], Pittaya Poompuang[2], Assoc.Prof.Dr.Wichian Premchaiswadi[3]
Graduate School of Information Technology in Business, Siam University, Thailand
aimdala@hotmail.com[1], pitaya_pp@yahoo.co.th[2], wichian@siam.edu[3]

Abstract

The principles of decision-analytic decision support, implemented in GeNIe (Graphical Network Interface) and SMILE (Structural Modeling, Inference, and Learning Engine) can be applied in practical decision support systems (DSSs). GeNIe plays the role of a development environment and SMILE plays the role of a reasoning engine. A decision support system based on SMILE can be equipped with a customized user interface. GeNIe's name and its uncommon capitalization originate from the name Graphical Network Interface, given to the original simple interface to SMILE which is a library of functions for graphical probabilistic and decision-theoretic models. GeNIe only runs under one of the most popular computing platform,: the Windows operating systems, which makes it not easily portable. GeNIe is therefore limited in its graphical representation across multiple system platforms. This paper is composed of two parts. The first part discusses a development environment for building graphical decision-theoretic models, an influence diagram, on a website by using an newly developed engine called "SMILE". The second part of the paper discusses the visualization of SMILE decision-theoretic models on a website using Flash technologies.

1. Introduction

Decision analysis is the art and practice of decision theory, an axiomatic theory prescribing how decisions should be made. Decision analysis is based on the premise that humans are reasonably capable of framing a decision problem, listing possible decision options, determining relevant factors, and quantifying uncertainty and preferences, but are rather weak in combining this information into a rational decision.

Decision analysis comes with a set of empirically tested tools for framing decisions, structuring decision problems, quantifying uncertainty and preferences, discovering those factors in a decision model that are critical for the decision, and computing the value of information that reduces uncertainty. Probability theory and decision theory provide tools for combining observations and optimizing decisions. While decision analysis is based on two quantitative theories, probability theory and decision theory, its foundations are qualitative and based on axioms of rational choice. The purpose of decision analysis is to gain insight into a decision and not to make the decision and provide a recommendation.

SMILE [1] (Structural Modeling, Inference, and Learning Engine) is a fully platform independent library of functions implementing graphical probabilistic and decision-theoretic models, such as Bayesian networks, influence diagrams, and structural equation models. Its individual functions, defined in the SMILE Applications Programmer Interface (API), allows for creating, editing, saving, and loading graphical models, and using them for probabilistic reasoning and decision making under uncertainty. SMILE is implemented in C++ in a platform independent fashion. SMILE can be embedded in programs that use graphical probabilistic models as their reasoning engines. Models developed in SMILE can be equipped with a user interface that best suits the user of the resulting application.

GeNIe [2] (Graphical Network Interface) and SMILE (Structural Modeling, Inference, and Learning Engine) are designed and used as tools for framing decisions, structuring decision problems, quantifying uncertainty and preferences, discovering those factors in a decision model that are critical for the decision, and computing the value of information that reduces uncertainty.

One disadvantage and serious limitation of using GeNIe is its platform dependence [3] and its limited visualization capabilities which limit its use on a global basis. It only runs on computers. This paper presents the development of visualization capabilities for SMILE on a web site using Flash technologies in order to overcome the limitations of GeNIe discussed previously. The standard web browser can load the models and the user or decision modeler can experiment with the model and compute the values for decision information. The powerful technologies and

978-0-7695-3263-9/08 $25.00 © 2008 IEEE
DOI 10.1109/SNPD.2008.77

Authorized licensed use limited to: Wichian Premchaiswadi. Downloaded on June 09,2010 at 04:22:06 UTC from IEEE Xplore. Restrictions apply.

tools in a Flash application can create a dynamic model and present a friendly user interface for a model.

2. Fundamentals

GeNIe is a development environment for building graphical decision-theoretic models [2]. GeNIe's name and its uncommon capitalization originate from the name Graphical Network Interface, given to the original rather simple interface to SMILE which is a library of functions for graphical probabilistic and decision-theoretic models. GeNIe is an outer shell to SMILE. GeNIe allows for building models of any size and complexity, limited only by the capacity of the operating memory of the computer on which it is run. GeNIe is implemented in Visual C++ and draws heavily on the MFC (Microsoft Foundation Classes). This makes it not easily portable, although it runs under one of the most popular computing platforms: the Windows operating systems. GeNIe is platform dependent and runs only on Windows computers.

2.1. Bayesian networks

Bayesian networks [4][5] (also called *belief networks*, *Bayesian belief networks*, *causal probabilistic networks*, or *causal networks*) are acyclic directed graphs in which nodes represent random variables and arcs represent direct probabilistic dependencies among them. The structure of a Bayesian network is a graphical, qualitative illustration of the interactions among the set of variables that it models [6]. The structure of the directed graph can mimic the causal structure of the modeled domain, although this is not necessary. When the structure is causal, it gives a useful, modular insight into the interactions among the variables and allows for predicting the effects of external manipulation. Nodes of a Bayesian network are usually drawn as circles or ovals. The simple Bayesian network shown in Figure 1 represents two variables, *Success* and *Forecast*, and expresses the fact that they are directly dependent on each other.

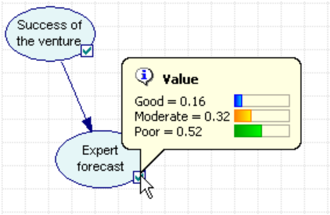


Figure 1. Bayesian network in GeNIe.

2.2. Influence diagrams

Influence diagrams [7], also called relevance diagrams, are acyclic directed graphs representing decision problems. The goal of influence diagrams is to choose a decision alternative that has the highest expected gain (utility). Similar to Bayesian networks, influence diagrams are very useful in showing the structure of the domain, i.e., the structure of the decision problem. Influence diagrams contain four types of nodes (Decision, Chance, Deterministic, and Value) and two types of arcs (influences and informational arcs). The influence diagram shown in Figure 2 models a decision related to an investment in a risky venture.

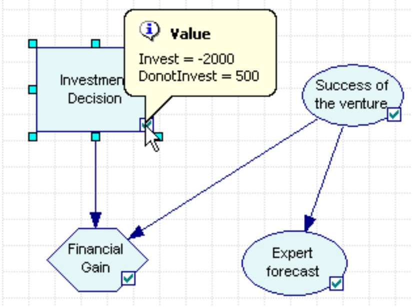


Figure 2. Influence diagram in GeNIe.

3. Design of SMILE Visualization on Web

The SMILE API is used to create the SMILE visualization on a web page that will help the user and modeler develop models more easily without installing any programs on the machine. In order to display the decision-theoretic models on the web page, two outer shells and one core named "SMILE" or "SMILE.dll" are required. See Figure 3.

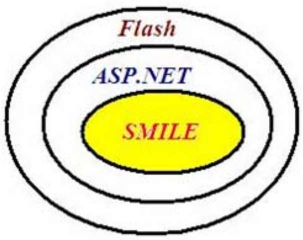


Figure 3. SMILE visualization on web architecture.

SMILE is in the core level of the designed application. SMILE provides useful functions such as

Authorized licensed use limited to: Wichian Premchaiswadi. Downloaded on June 09,2010 at 04:22:06 UTC from IEEE Xplore. Restrictions apply.

Bayesian networks and influence diagrams for solving probabilistic and decision-theoretic problems. SMILE is released as a dynamic link library (DLL) that implements graphical probabilistic and decision-theoretic models. SMILE.NET is used as a SMILE wrapper to the .NET layer. Actually there are several SMILE wrappers, such as SMILE.NET (.NET interface), SMILEX (Active X) [1], jSMILE (Java interface) [1], etc. This project uses SMILE.NET as a wrapper in order to interface with the .NET layer.

The ASP.NET layer is grouped into 2 layers, the inner and the outer layer. The inner layer of ASP.NET is developed using Visual Basic.NET to interface with SMILE.net and is compile as ClassSmileNet.dll. The outer layer of ASP.NET is developed using Visual C#.Net and acts as the interface between the ClassSmileNet.dll and Flash layer.

The FLASH layer is a presentation layer for SMILE visualization. The FLASH layer consists of two main sections. These are FLASH Remoting [8] and FLASH player. Flash Remoting provides a network communications channel between Flash applications and remote services (ASP.NET) and runs at the server. FLASH player is responsible for displaying the interactive model on a website. FLASH Remoting provides the following advantages [9]:

- Ease of use: Flash Remoting offers automatic data type conversion from native remote service code, such as Java, CFML, and C#, to ActionScript and back again. Also, Flash Remoting MX automatically performs logging, debugging, and security integration.
- Performance: Flash Remoting serializes messages between Flash applications and remote services using the Action Message Format (AMF) [10]. AMF is a binary format modeled on the Simple Object Access Protocol (SOAP) format.
- Extensibility: Flash Remoting is designed to integrate with established application design patterns and best practices to build well-designed Flash applications.

When compared to traditional HTML-based browser applications, Flash applications provide unique abilities to create dynamic and sophisticated user interactions, including the following:

- Flash Player runtime to execute code, transmit data, and invoke remote services
- Separation of client-side presentation logic from the server-side application logic
- Efficient use of bandwidth by removing the need to refresh the entire page and employing vector-based graphics
- Easy deployment on multiple platforms and devices on the server side because Flash Remoting runs as an assembly in .NET servers. Depending on the application server platform, Flash Remoting on the server contains a series of filters that perform logging, error handling, and security authentication, as well as automatically mapping the service function requests to the appropriate server technology. The image below shows an overview of the Flash Remoting control flow.

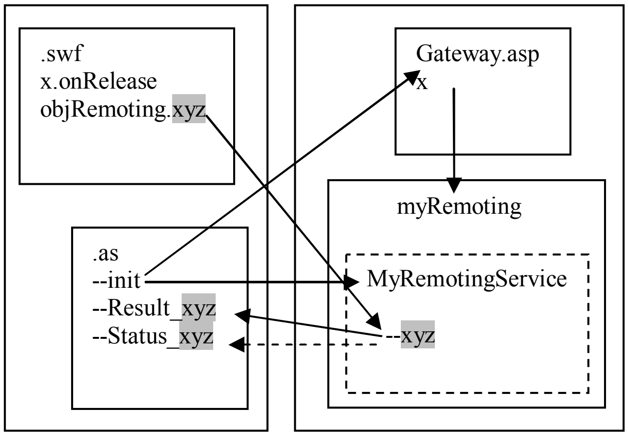


Figure 4. The overview of the flash remoting control flow.

In order to understand how Flash Remoting works in SMILE visualization, figure 5 depicts a simplified representation of Communication between a web Browser and a Web Server

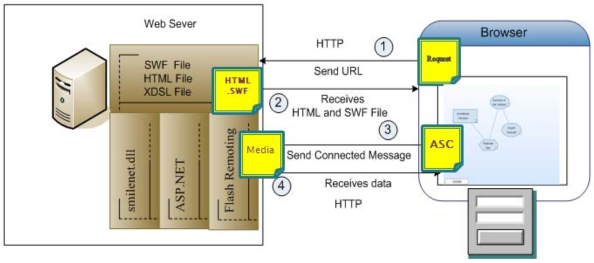


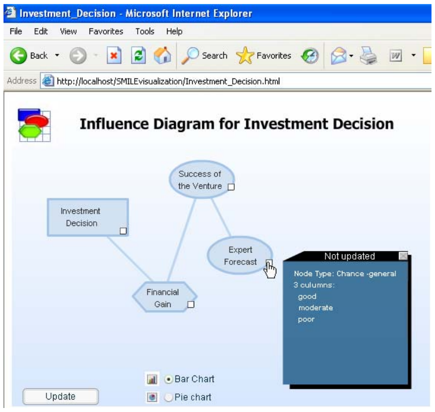
Figure 5. Sequence in communication between web browser and web server.

In the first and second steps, the client makes requests for a web page from the web server. The web server returns a flash movie back to client side. In the third and last step, Flash player will handle and process an actionScript to communicate with Flash Remoting embedded on the web server.

Authorized licensed use limited to: Wichian Premchaiswadi. Downloaded on June 09,2010 at 04:22:06 UTC from IEEE Xplore. Restrictions apply.

4. Implementation

The SMILE visualization of an Influence Diagram for making an investment decision is developed in the following by using the technologies mentioned above. The client site must contain the flash player application in order to display a flash movie. Flash Remoting need to be installed in Internet Information Service (IIS) in order to communicate with several functions in the .NET environment. We start with the flash application that is developed in actionScript 3.0. An influence diagram for an investment decision is created by defining the meaning of each element of the xsdl file format [11] and drawing all nodes individually. There are three types of nodes in an influence diagram. The first one is a decision node. Figure 6 to Figure 9 shows screenshots of the user interface implemented by Flash technologies on a website for the case of an influence diagram for an investment decision.

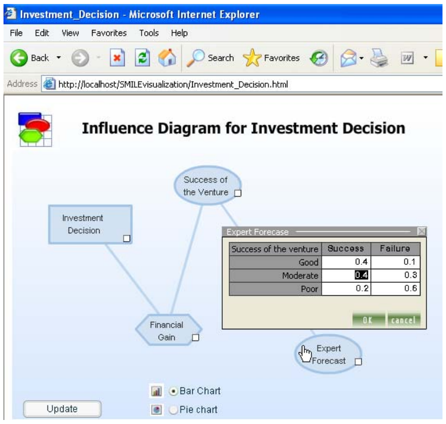

Investment_Decision - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/SMILEvisualization/Investment_Decision.html
Influence Diagram for Investment Decision
Success of the Venture
Investment Decision
Expert Forecast
Financial Gain
Bar Chart
Pie chart
Update

Not updated

Node Type: Chance -general
3 culumns:
good
moderate
poor

● Bar Chart
○ Pie chart

Figure 6. An influence diagram for investment decision.

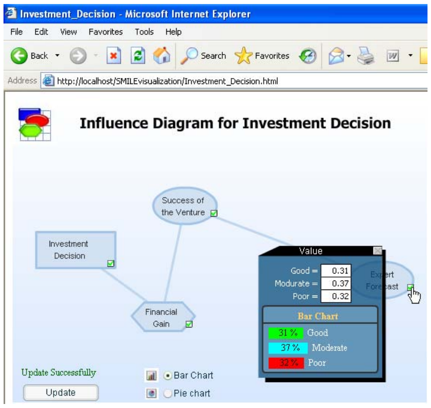

Investment_Decision - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/SMILEvisualization/Investment_Decision.html
Influence Diagram for Investment Decision
Success of the Venture
Investment Decision
Financial Gain
Expert Forecast
OK
cancel
Bar Chart
Pie chart
Update

Expert Forecase

Success of the venture	Success	Failure
Good	0.4	0.1
Moderate	0.4	0.3
Poor	0.2	0.6

● Bar Chart
○ Pie chart

Figure 7. Setting values for expert forecast node.


Investment_Decision - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/SMILEvisualization/Investment_Decision.html
Influence Diagram for Investment Decision
Success of the Venture
Investment Decision
Financial Gain
Expert Forecast
Bar Chart
Pie chart
Update

Value

Good = 0.31
Modurate = 0.37
Poor = 0.32

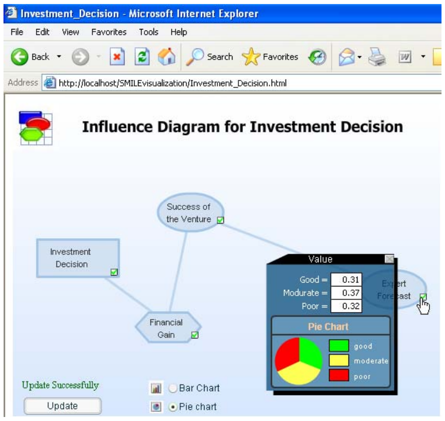
Bar Chart

31 % Good
37 % Moderate
32 % Poor

Update Successfully

● Bar Chart
○ Pie chart

Figure 8 Bar chart visualization after updating beliefs.


Investment_Decision - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/SMILEvisualization/Investment_Decision.html
Influence Diagram for Investment Decision
Success of the Venture
Investment Decision
Financial Gain
Expert Forecast
Bar Chart
Pie chart
Update

Value

Good = 0.31
Modurate = 0.37
Poor = 0.32

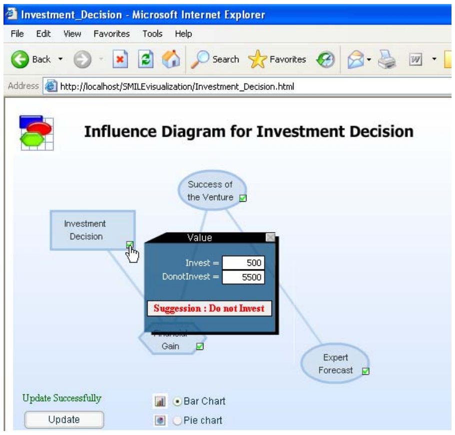
Pie Chart

🟩 good
🟨 moderate
🟥 poor

Update Successfully

○ Bar Chart
● Pie chart

Figure 9. Pie chart visualization after updating beliefs.


Investment_Decision - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/SMILEvisualization/Investment_Decision.html
Influence Diagram for Investment Decision
Success of the Venture
Investment Decision
Financial Gain
Expert Forecast
Bar Chart
Pie chart
Update

Value

Invest = 500
DonotInvest = 5500

Suggession : Do not Invest

Update Successfully

● Bar Chart
○ Pie chart

Figure 10. SMILE visualization result after updating beliefs.

Authorized licensed use limited to: Wichian Premchaiswadi. Downloaded on June 09,2010 at 04:22:06 UTC from IEEE Xplore. Restrictions apply.

The flash application, shown in Figure 6, presents an Influence Diagrams for an investment decision for a risky venture. It is an acyclic directed graph representing the decision problem. The goal of an influence diagram is to choose a decision alternative that has the highest expected gain (utility). Influence diagrams contain four types of nodes (Decision, Chance, Deterministic, and Value) and two types of arcs (influences and informational arcs).

Decision nodes, usually drawn as rectangles (such as the node, Investment Decision, above), represents variables that are under control of the decision maker and model the decision alternatives available to the decision maker. Decision nodes include a specification of the available decision options. The Node, Investment Decision, above has two alternatives, namely, Invest and DoNotInvest.

Chance nodes, usually drawn as circles or ovals (such as the nodes, Expert Forecast and Success of the venture, above), are random variables and they represent uncertain quantities that are relevant to the decision problem. They are quantified by conditional probability distributions. Deterministic nodes, usually drawn as double-circles or double-ovals, represent either constant values or values that are algebraically determined from the states of their parents. In other words, if the values of their parents are known, then the value of a deterministic node is also known with certainty. Deterministic nodes are quantified similar to Chance nodes. The only difference is that their probability tables contain all zeros or ones (note that there is no uncertainty about the outcome of a deterministic node once all its parents are known).

Value nodes, usually drawn as diamonds (such as the node, Financial Gain, above), represent utility, i.e., a measure of desirability of the outcomes of the decision process. They are quantified by the utility of each of the possible combinations of outcomes of the parent nodes.

Normally, an arc in an influence diagram denotes an influence, i.e., the fact that the node at the tail of the arc influences the value (or the probability distribution over the possible values) of the node at the head of the arc. Some arcs in influence diagrams have clearly a causal meaning. In particular, a directed path from a decision node to a chance node means that the decision (i.e., a manipulation of the graph) will impact that chance node by changing its probability distribution.

5. Conclusions and Future Work

The SMILE visualization on a website is designed and presented in order to overcome the limitations of the graphical representation in GeNIe. All functions in this application represented by Flash are similar to those in GeNIe. Users, who are familiar with the GeNIe interface, can simply use the application. Flash has the capability to generate good graphical user interfaces and interactive application capabilities but it does not support the .NET environment. We still need to use the flash technology called “Flash Remoting” as middleware in communication with the core of SMILE and the Flash application.

This paper discussed how to use flash technology to extend the scalability and capability of SMILE and GeNIe for global users or modelers. In the next phase of our development, we will create more dynamic models by using several flash technologies such as Adobe AIR [12] and Flash Media Server [13] that will help to improve and increase performance for the flash application in SMILE visualizations.

References

[1] http://genie.sis.pitt.edu/wiki/SMILE:_Probabilistic_Inference_in_Bayesian_Networks

[2] http://genie.sis.pitt.edu

[3] Marek J. Druzdzel and Roger R. Flynn, “Decision Support Systems”, Encyclopedia of Library and Information Science, Second Edition, 2002.

[4] Pearl, J., “Probabilistic Reasoning in Intelligent System”, Networks of plausible Inference, Morgan Kaufmann, 1988

[5] http://genie.sis.pitt.edu/wiki/Probabilistic_Decision_Support_System:_Bayesian_Networks

[6] Agniezka Onisko, Marek J. Druzdzel, Hanna Wasyluk, and Warsaw, “Learning Bayesian Network Parameters from Small Data Sets”, International Journal of approximate Reasoning, 27(2):165-182, 2001

[7] Howard, Ronald A., James E. Matheson, “Readings on the Princinples and Applications of Decision Analysis II”, Strategic Decision Group, Menlo Park, CA, 719–762. Reprinted, Decision Anal. 2(3) 127–143, 1984/2005

[8] http://www.adobe.com/products/flashremoting

[9]http://livedocs.adobe.com/flashremoting/mx/Using_Flash_Remoting_MX/intro2.htm

[10]http://en.wikipedia.org/wiki/Action_Message_Format

Authorized licensed use limited to: Wichian Premchaiswadi. Downloaded on June 09,2010 at 04:22:06 UTC from IEEE Xplore. Restrictions apply.

[11] http://genie.sis.pitt.edu/wiki/Appendices:_XDSL_
File_Format_-_XML_Schema_Definitions

[12] http://www.adobe.com/products/air/

[13] http://www.adobe.com/products/flashmediaserver/

Authorized licensed use limited to: Wichian Premchaiswadi. Downloaded on June 09,2010 at 04:22:06 UTC from IEEE Xplore. Restrictions apply.