GeoDialogue: A Software Agent Enabling Collaborative Dialogues between a User and a Conversational GIS

Hongmei Wang
Department of Computer Science
Northern Kentucky University
Highland Heights, KY 41099
wangh1@nku.edu

Guoray Cai
College of Information Sciences & Technology
Pennsylvania State University
University Park, PA 16802
cai@ist.psu.edu

Alan M. MacEachren
Department of Geography
Pennsylvania State University
University Park, PA 16802
MacEachren@psu.edu

Abstract

A fundamental challenge that must be met to achieve a usable conversational interface to Geographic Information System (GIS) is how to enable a more natural interaction between the user and the system. This paper presents a design of an agent-based computational model, PlanGraph, and implementation of this model in a software agent, GeoDialogue, as a dialogue manager for a conversational GIS. This dialogue agent enables intelligent collaborative human-GIS dialogues. The capabilities of this agent are demonstrated through its performance in a conversational GIS, Dave_G.

1. Introduction

A Geographic Information System (GIS) is a computer-based information system for managing and processing geographically referenced data. Development of transparent interfaces is one of the long term goals for GIS, which leads to development of natural interfaces (human communication modalities enabled interfaces, e.g. speech and gesture enabled interfaces) for GIS. Development of natural interfaces for geographical information use has evolved from early systems enabling speech only and/or simple mouse-simulated gesture to current systems enabling speech and free-hand/pen gesture [1-5]. Most of these systems are conversational (speech enabled). One of the major challenges that must be faced for development of conversational GIS is how to enable a more natural human-GIS interaction (that is, more like human-human interaction [6]). The goal of this study is to facilitate more natural human-GIS interaction through enabling collaborative human-GIS dialogues. This study takes the Human Emulation (HE) approach to design the human-GIS interaction, considering its advantages over the Human Complementary (HC) approach [7].

An agent-based computational model, PlanGraph, is developed in this study to support the HE approach. A software agent, GeoDialogue, which implements this model, is implemented as a dialogue manager to enable collaborative human-computer dialogues. The capabilities of this agent are demonstrated through its performance in a conversational GIS, Dave_G [1, 3]. Our previous papers [1, 8] related to GeoDialogue focus on conceptual descriptions of this agent. This paper will focus on its implementation details.

2. PlanGraph

The design of the PlanGraph model is mainly based on two important theories, the computational collaborative discourse theory [9] and the SharedPlan theory [10, 11], together with the agent-based computational model, the RecipeGraph (Rgraph) model [12]. The assumption underlying our design of the PlanGraph model is that collaborative behavior between the GIS and the user underlying the collaborative human-GIS discourse is centered on a collaborative plan, SharedPlan [10, 11]. Both the system and the user contribute to this plan. The system agent needs to retain knowledge of the SharedPlan in order to reason about how to help the user during the communication. The PlanGraph models the dynamic knowledge that, at a given time during the human-GIS collaboration, the GIS agent adheres to the SharedPlan.

The PlanGraph (see its structure from our previous paper [1]) consists of the root plan representing the common goal of the system agent and the user and all its sub-plans for sub-goals. Each node in the PlanGraph is centered around a plan/subplan, a complex data structure which records an action...
together with a set of mental states that the GIS agent holds on that action at that given time. The PlanGraph also records the attention focus that the system agent holds on the SharedPlan at that given time.

This model also provides a set of reasoning algorithms associated with the PlanGraph. Upon receiving each user input, the GIS agent needs to react to the input with at least three steps, including interpreting the input, elaborating and executing the current SharedPlan toward its success, and sending out responses to the user. The design of the Interpretation Algorithms (IAs) in this model is based on and extends those in [12]. The designs of the Elaboration Algorithms (EAs) and Response Algorithms (RAs) follow the plan evolution process of a SharedPlan [11] and extend those in [13].


To illustrate how the PlanGraph model works, we have implemented this model in the prototype software agent, GeoDialogue, a human-GIS dialogue manager for the conversational GIS, Dave_G [1, 3].

3.1 Structure

The prototype software agent, GeoDialogue, includes five major modules (Figure 1), including Semantic Analysis, Interpretation, Dialogue Control, Elaboration, and Response Control. In addition, another two modules, Knowledge Base and GIS Component, are also included to support performance of the five major modules. The Dialogue Control module is the central intelligence module in the dialogue agent. Its sub component, PlanGraph, helps the agent to maintain the dynamic knowledge on the SharedPlan developed by the system and the user. Another sub component, Reasoning Engine, provides automated reasoning for the three modules, Interpretation, Elaboration and Response Control. This sub component also controls the dialogue process flow in the order shown in Figure 1.

3.2 Implementation of actions and recipes

The basic actions are implemented as public functions in GeoDialogue. Complex actions are elaborated after their recipes are decomposed in the PlanGraph. They are further executed as the basic actions involved in these complex actions are executed. The recipes are implemented in XML (see an example in Figure 2) and stored in Knowledge Base.

3.3 Implementation of mental states

For convenience in programming, a set of numerical labels (see Table 1), called Mental State Number (MSN), are implemented to represent mental states that the GIS agent holds on an action or a parameter in the PlanGraph.

3.4 Implementation of PlanGraph

The PlanGraph is implemented as a recursive dynamic data structure in GeoDialogue. The data structure itself defines the root plan and current attention focus only. Its recursive structure is achieved through the recursive structure of an action/parameter node (see the example PlanGraph in Section 4).

Figure 1. Structure of GeoDialogue

![Figure 1](image1.png)

Figure 2. An example recipe, ShowMap1

```xml
<?xml version="1.0"?>
<DAVE_GXML version="0.0.1">
<RECIPE Name="ShowMap1" Act="Show Map" >
<Constraints Order="false"> </Constraints>
<Parameters Order="false">
  <PARA name="layers" multiple="yes" option="true">
    <TYPE type="Layer"/>
  </PARA>
  <PARA name="Extent" multiple="no" option="true">
    <TYPE type="envelop, shape, polygon, feature"/>
  </PARA>
</Parameters>
<Subactions order="true">
  <SUBACT act="Generate Map" />
</Subactions>
</RECIPE>
</DAVE_GXML>
```
3.5 Implementation of reasoning algorithms

All individual reasoning algorithms associated with the PlanGraph are implemented as separate functions in GeoDialogue. These individual functions are organized in Reasoning Engine as three interleaved module functions for Interpretation, Elaboration and Response Control.

<table>
<thead>
<tr>
<th>MN</th>
<th>Meaning of an MSN on an action α</th>
<th>Meaning of an MSN on a parameter p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The goal of α is proposed for α.</td>
<td>p is proposed in an action involved in the collaboration.</td>
</tr>
<tr>
<td>1</td>
<td>The goal of α is agreed by all agents in collaboration.</td>
<td>p is necessary for the collaboration and is waiting for instantiation.</td>
</tr>
<tr>
<td>2</td>
<td>α is a basic action ready to be executed.</td>
<td>p under the instantiation process is optional.</td>
</tr>
<tr>
<td>3</td>
<td>α is a complex action, and one plan is selected for this action.</td>
<td>p under the instantiation process is required.</td>
</tr>
<tr>
<td>4</td>
<td>The plan selected for α is successfully executed, and the plan execution result contains some uncertainties.</td>
<td>p is instantiated with a parameter value involving certain uncertainties.</td>
</tr>
<tr>
<td>5</td>
<td>The plan selected for α fails.</td>
<td>Instantiation of p fails.</td>
</tr>
<tr>
<td>6</td>
<td>The plan selected for α is successfully executed, and the plan execution result does not involve any uncertainties.</td>
<td>p is successfully instantiated with a parameter value without any uncertainties.</td>
</tr>
</tbody>
</table>

4. A sample human-GIS dialogue

A sample dialogue (Figure 3) between a human user and GeoDialogue is explained in this section to illustrate functionalities of PlanGraph and GeoDialogue. Figure 4 shows the final PlanGraph. The dotted section in this figure is added between U2 – U3, and all others are generated between U1 and G1.

After receiving the first recognized and parsed input, U1, GeoDialogue assigns semantic meanings to the input through Semantic Analysis and then sends the analysis results to Interpretation. By calling the IAs from Reasoning Engine, the agent initiates the PlanGraph with the user’s intention, See Map (with MSN 0), underlying U1 and focuses its attention on this action.

The dialogue begins with an empty map on the screen

U: The user; G: The dialogue agent, GeoDialogue

U1: Show me a map of Florida.
G1: {show a map with state boundaries and interstates in the area of Florida}. Is this OK?
U2: No, please also add major cities.
G2: {show a map with state boundaries, interstates and major cities in the area of Florida} Here it is.
U3: Thanks.

Figure 3. A sample human-GIS dialogue

Figure 4. The final PlanGraph

The dialogue agent starts to advance the focus action in the PlanGraph (See Map) toward its success in Elaboration by calling the recursive EAs. It uses the recipe SeeMap1 (see Figure 4) to elaborate See Map and updates its MSN on See Map from 0 to 3. Then, it moves its attention to See Map’s first child node, Show Map, and elaborates this node with its recipe ShowMap1 (see Figure 4). The agent successfully identifies the map extent (for the parameter Extent) from U1 and infers the map layers (for the parameter Layers) that the user may be interested in as the state boundary and interstates for the Florida map. By calling the GIS component to generate a map response based on the two parameter values, the agent generates a map response and a speech response “Is this map
OK?” through executing the basic action, *Generate Map*.

Because *Generate Map* is a communicative action done by the system, by calling RAs from *Reasoning Engine, Response Control* sends out all responses in G1; the dialogue agent then focuses its attention on the root action *See Map* (under the assumption that the user usually can successfully execute the action *Watch Map*) and waits for the user’s feedback on the collaboration result—the map. Because the value of the parameter node, *Layers*, is inferred, the MSNs on all its parent nodes and the nodes that use the inferred value are updated as 4, which indicates the existence of uncertainties on these nodes. See the un-dotted section of the *PlanGraph* in Figure 4 for the current *PlanGraph*.

Upon receiving the user’s negative reply in U2, *GeoDialogue* modifies its MSN on *See Map* as 3 and re-elaborates this plan by tracing all its sub plans with uncertainties. The agent interprets the second part of U2 as *Communicate Value* by *Informing* (see the dotted section in Figure 4), by which the user actively provides modification information on the parameter *Layers*. Then, the agent re-generates a new map by adding another map layer, *major cities*, and a new speech response, “Here it is,” in G2. The user’s positive feedback in U3 confirms the success of human-GIS collaboration for the user’s request, *See Map* (see the final *PlanGraph* in Figure 4).

5. Conclusion and future work

In this paper, we offer two major contributions: 1) the agent-based computational model, *PlanGraph*, can help the system to keep track of the *SharedPlan* developed during the human-computer communication process; and 2) the prototype software agent, *GeoDialogue*, can facilitate the conversational GIS to have a more natural interaction with the user through collaborative dialogues. The dialogue agent fails in cases that 1) it can not interpret the recognized input, and 2) all recipes of a focus action have been tried, and all of them have failed. Future work will deal with evaluating the naturalness of human-GIS interaction supported by *GeoDialogue* and improving the knowledge of actions and recipes as dynamically accumulated and updated during human-GIS interaction.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grants No. BCS-0113030, and EIA-0306845. This work is also supported by the National Visualization and Analytics Center, a U.S. Department of Homeland Security program operated by the Pacific Northwest National Laboratory (PNNL). PNNL is a U.S. Department of Energy Office of Science laboratory.

References