GUI – Enhanced Activity Diagarams with Application to the Design of AVISTED

Likhitha Ravi, Sergiu M. Dascalu, Frederick C. Harris
College of Engineering
University of Nevada
Reno, USA
(ravi, dascalu, fredh)@cse.unr.edu

Abstract

Software design is worthwhile if it paves the way to a good and useful functional software. In this paper, we propose a new tool entitled GUI-Enhanced Activity diagrams (GEAD). This tool facilitates a new approach of software design consisting of UML activity diagrams with links to their respective interface snapshots. Notably, GEAD is more efficient for web applications that have numerous graphical user interfaces. To illustrate GEAD, we introduce a new toolset entitled 'Analysis and Visualization Toolset for Environmental Data' (AVISTED). AVISTED is a generic web-based visualization toolset which is used by climate researchers for visualizing large climate datasets. This paper introduces the design aspects of AVISTED by leveraging the concepts and strategies of GEAD. The links between the designs of the webpages and the activities in UML diagrams help showing clear connections between the backend and frontend activities. This comprehensive view helps developing the desired software that meets the user requirements more efficiently by improving collaboration and decreasing development time.

Keywords: GUI-Enhanced Activity Diagrams; UML; Software Design; Data Visualization, Web application;

I. INTRODUCTION

Software Design is one of the most important steps in the process of software development. It helps the developers, testers, users and the stockholders in understanding the functions of the software system thoroughly [1]. It also helps in planning a solution and approach for the software system.

Software systems are composed of components, subsystems, modules, design patterns, interfaces, and activities. These individual elements and their interdependencies can be shown through UML Diagrams. There are two types of UML diagrams: Static and Behavior Diagrams. A static diagram depicts the structure of the software system where as the behavior diagrams shows the dynamic behavior of the components in the system [2]. Examples of static diagrams include class diagrams, object diagrams, system level diagrams and user interface snapshots. Examples of behavior diagrams include use-case diagrams, sequence diagrams, and activity diagrams. With GEAD we intending to link the static and behavior diagrams. The snapshots of user interface and the activity diagrams are linked. It is a very new technique which is not used in designing software systems.

The purpose of GEAD is to provide a comprehensive view for developers and testers. It will provide them with a user interface view for each significant activity. This helps in identifying activities that belong to a single view and activities that are common in a group of views. These connections are very helpful in software systems that are implemented using MVC architecture. This technique will also help users and stakeholders in getting a modest link between the execution and the interface. GEAD is applicable in many fields such as web development, video game development and mobile/desktop apps. In this paper, we apply this technique to a web based application AVISTED.

AVISTED is a web based visualization toolset. It helps climate researchers in finding the trends, change of variables with time, and dependencies between variables. AVISTED supports User Authentication, User Authorization, Model Output Management, Model Operation, Tools Upload, and Archives of Visualization and Download. It supports the datasets in netCDF, JSON, and CSV formats. Users can perform extraction, visualization, view, download and save operations on the dataset. Visualizations that are developed using the toolset can also be saved in the user accounts for future use. Also, to support extensibility users are allowed to upload their own tools for data processing with the administrator's approval.

The rest of the paper is structured in five sections. Section II presents the motivation and concepts for the GEAD technique. Section III gives an overview of AVISTED. Section IV discusses how we applied the GEAD technique for designing AVISTED. Section V gives a summary of related tools used for designing software systems. Finally, Section VI concludes the paper by presenting some directions of future work for this approach.

II. MOTIVATION AND CONCEPTS FOR THE GEAD TECHNIQUE

The design aspects of the AVISTED led to the requirement of the new technique GEAD. AVISTED is a web application. It allows users to visualize climate data using several visualization techniques. This functionality presents the user with a different view for each visualization technique but some actions are common regardless of the view. For example getting user input, data extraction from NetCDF files, and performing statistics. And some actions are common among the all views, for instance scaling the data based on the visualization technique. In such a big application with many GUI designs that have similar activities and some page specific activities, it is very important to identify all the activities and actions before the development phase.

In traditional UML activity diagrams, there is no link between the user interface snapshots and activity diagrams. As there is no link, it is very hard for the testers, users and the stakeholder's to relate things. Also, it is very time consuming for the developers to find the common tasks and view specific tasks. Figure 1 shows a traditional activity diagram of Visualization Toolset of Environmental Data (VISTED). VISTED is non-generic version of AVISTED with minimum functionalities. The main purpose of VISTED is to visualize the climate modeling data which is provided by the Nevada climate change portal. Climate modeling data is available from 1980 to 2009 in NetCDF format with variables like Precipitation, Temperature, surface winds, and solar radiation [3].

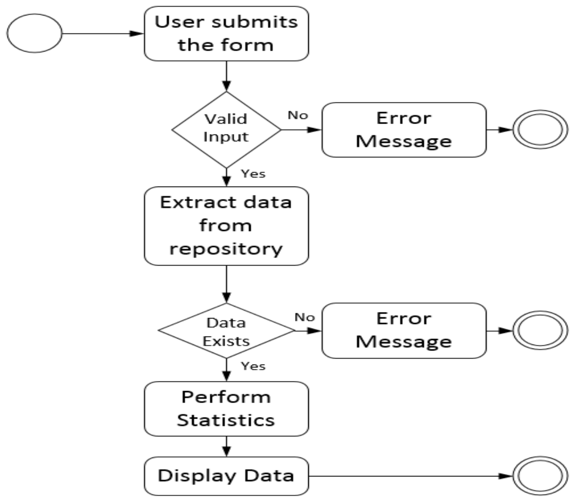
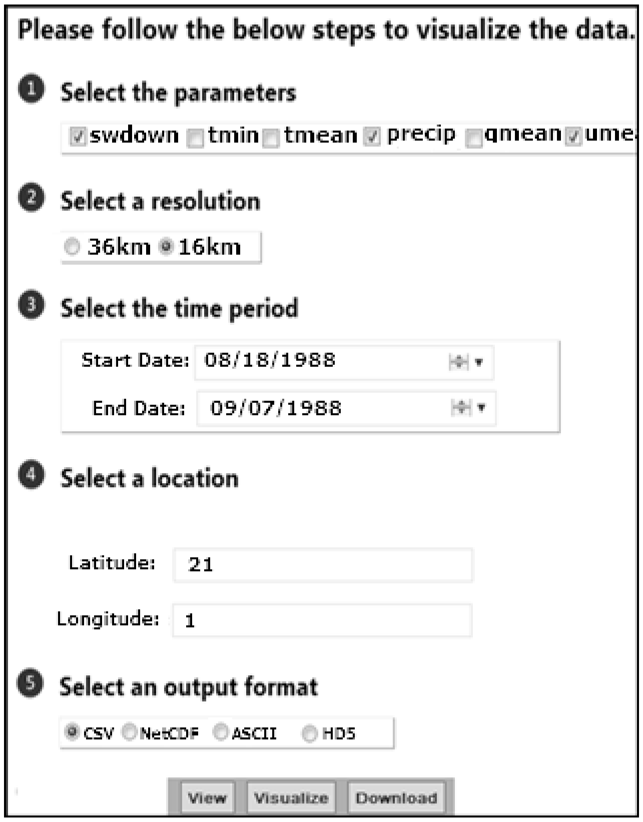


Figure 1: Traditional Activity diagram of VISTED

The activity diagram of VISTED in figure 1 shows the actions involved in displaying the data extracted by the AVISTED upon users request. User selects the required variables/parameters, resolution, time period, location and submits the form to view, download or visualize the selected data. If the user input is valid, data is extracted and presented it to the user else an error message is sent to the user. Although these actions can be clearly stated using the traditional activity diagrams it is time consuming for the team to search for the related designs associated with activity diagram.


Please follow the below steps to visualize the data.

1 Select the parameters

[✓ swdown] tmin tmean [✓ precip] qmean [✓ ume

2 Select a resolution

36km [✓ 16km]

3 Select the time period

Start Date: 08/18/1988

End Date: 09/07/1988

4 Select a location

Latitude: 21

Longitude: 1

5 Select an output format

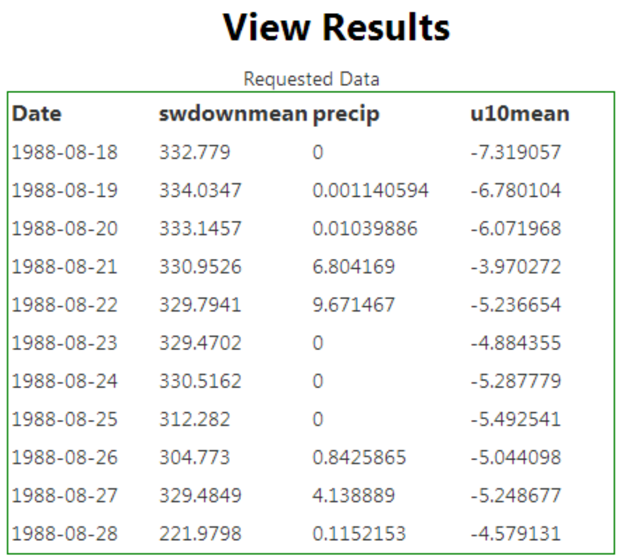
[✓ CSV] NetCDF ASCII HD5

Figure 2: User Input Form of VISTED

A new technique GEAD that allows connections between actions in the activity diagrams and their corresponding GUI designs of the user interface will help magnificently in identifying the common actions and view specific actions of a software system. Linking the designs in Figure 2, 3 to the actions of the activity diagram in the Figure 1 will allow the team to check the related designs quickly.

GEAD provides them a comprehensive view of the system so the team will understand the process much clearly. Figure 4 shows an activity diagram of AVISTED with GEAD approach. The small rectangular box at the bottom of each action has a link to the corresponding GUI design of the application. The actions Delete tool, Select tool, and Sort tool are linked to the same design shown in the Figure 5.

If an action has more than one design linked to it, it can be considered as a common task for the linked designs. So the implementation for such actions can be placed in a global scope for reuse. With MVC architecture all the common actions can be placed in a common controller and the actions that are specific to a view can be placed in the controller of that view. This saves a lot of development time for the developers.



View Results

Requested Data

Date	swdownmean	precip	u10mean
1988-08-18	332.779	0	-7.319057
1988-08-19	334.0347	0.001140594	-6.780104
1988-08-20	333.1457	0.01039886	-6.071968
1988-08-21	330.9526	6.804169	-3.970272
1988-08-22	329.7941	9.671467	-5.236654
1988-08-23	329.4702	0	-4.884355
1988-08-24	330.5162	0	-5.287779
1988-08-25	312.282	0	-5.492541
1988-08-26	304.773	0.8425865	-5.044098
1988-08-27	329.4849	4.138889	-5.248677
1988-08-28	221.9798	0.1152153	-4.579131

Figure 3: Data View of VISTED

Using GEAD designers can identify if more designs are needed for a particular state of the application. Imagine a game application where the design of the game changes with the state of the game. As GEAD necessitates the designs for major actions there is less probability of missing a design for the application. GEAD can also identify missing activity diagrams when the significant designs have not been linked to any actions.

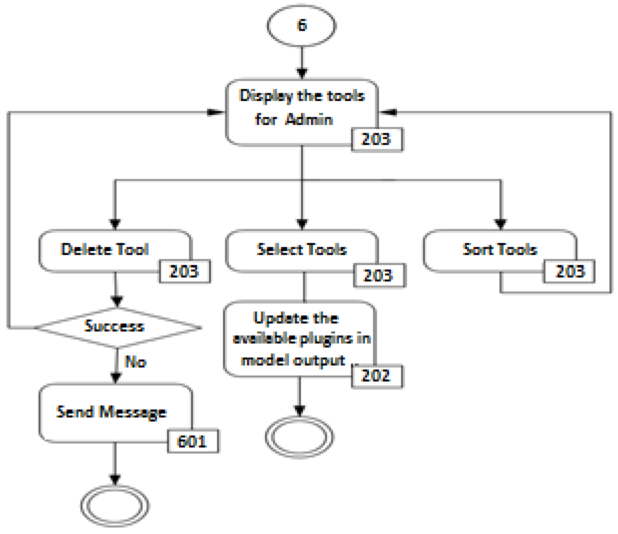
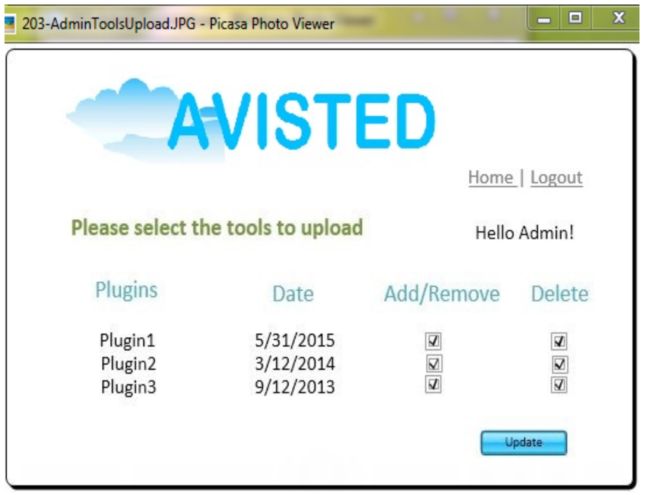


Figure 4: Activity diagram with GEAD approach

In large organizations, GEAD allows the development team to plan the development effectively and improves the collaboration within the team. For an instance, a backend developer can easily find the front end developer who is working on the view related to the actions he is working at the backend. A manager can easily divide the tasks among the team members without any duplication in actions.



Please select the tools to upload

Plugins	Date	Add/Remove	Delete
Plugin1	5/31/2015	✓	✓
Plugin2	3/12/2014	✓	✓
Plugin3	9/12/2013	✓	✓

Figure 5: Snapshot of the design of AVISTED

It should be noted that all actions may not have links to their corresponding designs. And the GUIS may not exactly look as GUIS that will be developed.

III. OVERVIEW OF AVISTED

VISTED[4] is a web based visualization toolset for visualizing climate dataset. It allows users to view, download, and visualize climate datasets. Based on the users selection, data is extracted from NetCDF files. It also supports input data formats such as CSV, and ASCII. The backend is implemented using C# and frontend logic is implemented using HTML, JavaScript, and D3.

AVISTED is an extended version of VISTED. The extension was needed to make the application applicable to other models in environmental sciences. Features such as Model Output Management, Tools upload, Archives and Download make AVISTED more generic and unique. Model Output is a common feature of AVISTED and VISTED. Figure 6, 7 shows the activity diagram of AVISTED with all the features. Users can use AVISTED in four modes. They are User Mode, Admin Mode, New User, and Guest. Guests have limited options.

Model Output Management provides a list of models to the user. The author, date of creation, file format, size and description of all the models are also provided. Administrators are allowed to select and delete a model but users and guests are not allowed to delete a model. Guests are only provided with few default models.

The selected model is displayed to the user by Model Output. It will extract all the parameters from the selected Model. To further extract data from the Model a time range and location options are provided based on the availability in the model. Also, to further extend this feature user can select a plug-in from the available set of tools. After the data extraction some statistics are performed on the data based on the selected plugins. User can download, view or visualize the extracted data.

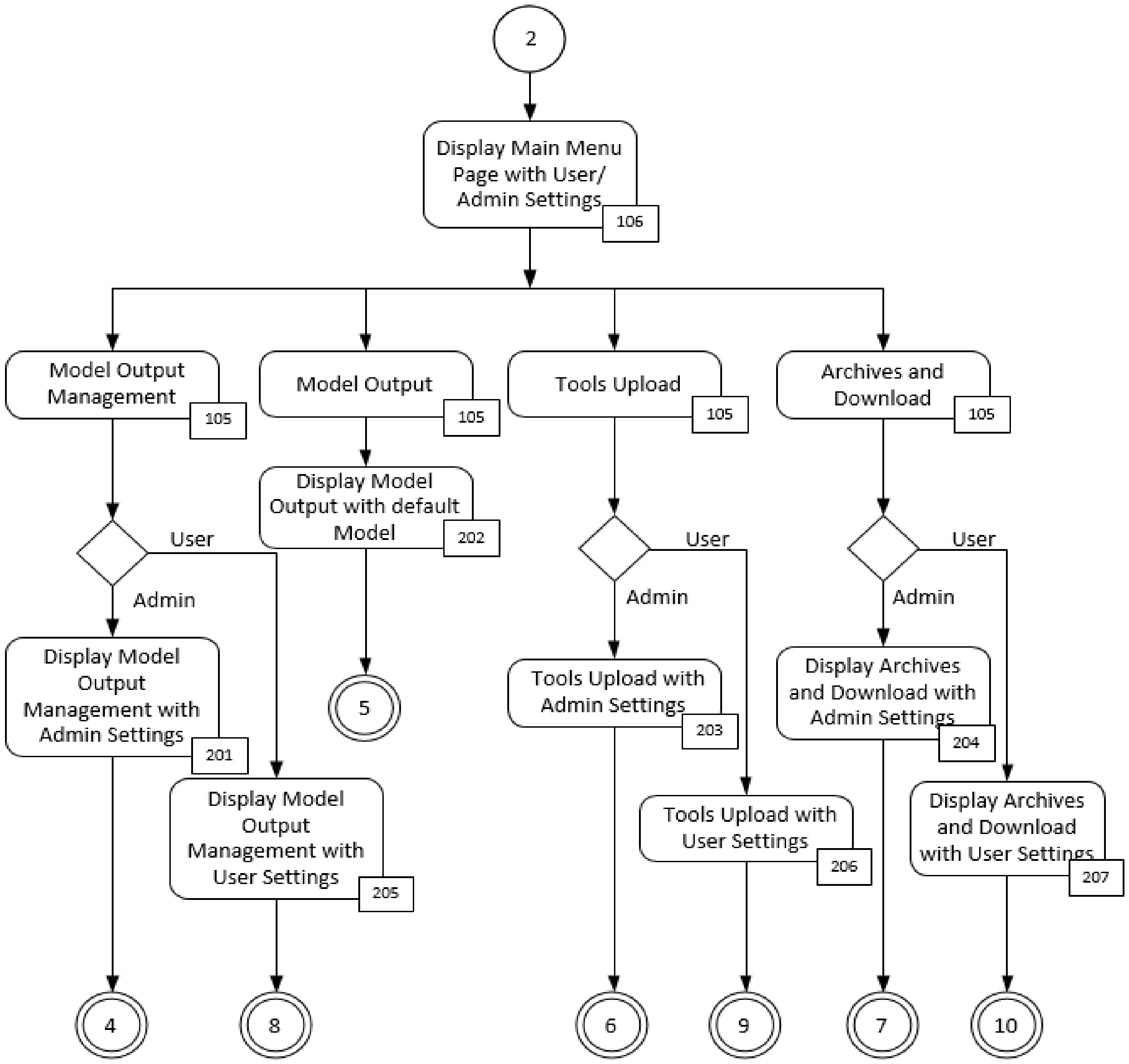

2
Display Main Menu Page with User/ Admin Settings
106
Model Output Management
105
Model Output
105
Tools Upload
105
Archives and Download
105
Display Model Output with default Model
202
User
Admin
User
Admin
User
Admin
Display Model Output Management with Admin Settings
201
5
Tools Upload with Admin Settings
203
Display Archives and Download with Admin Settings
204
Display Model Output Management with User Settings
205
Tools Upload with User Settings
206
Display Archives and Download with User Settings
207
4
8
6
9
7
10

Figure 6: Activity Diagram with all features

Tools upload is another significant feature of AVISTED. It makes the application plug-in based which allows it to be extended without any recompilation. Administrator of the application manages all the plugins. He or she can either upload or delete a plugin. Users can select a plugin from the available list. Guests are not provided with this feature.

The last and exciting feature of AVISTED is Archiving and Download. A user can view the saved images of the visualizations and download a preselected data. Administrator manages this feature. He or she can add or delete an image or user download selection.

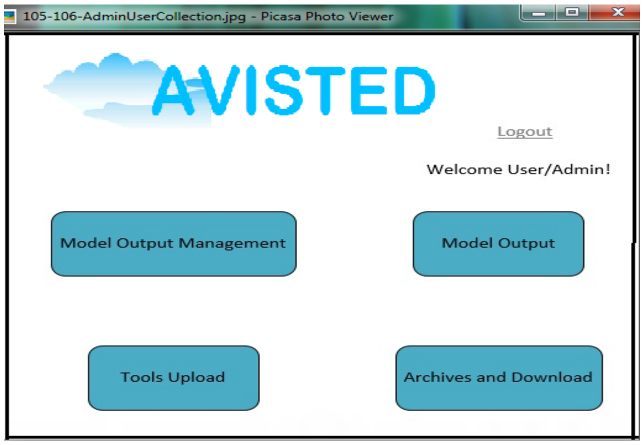


Figure 7: Snapshot of the GUI design with all the features of the AVISTED

IV. Applying GEAD to Designing AVISTED

AVISTED has been designed using the GEAD approach. Figures 6, 8, 10 show the activity diagrams with their respective designs in Figure 7,9,11.

All features of AVISTED are shown in the activity diagram in Figure 6. All the actions are linked to a GUI design. Interestingly all the four features of AVISTED are linked to the same design 105 because they are presented to the user on the same web page. Figure 7 shows the GUI design of the webpage. The GUI name starts with the 105. This naming convention links it back to the actions in the activity diagram.

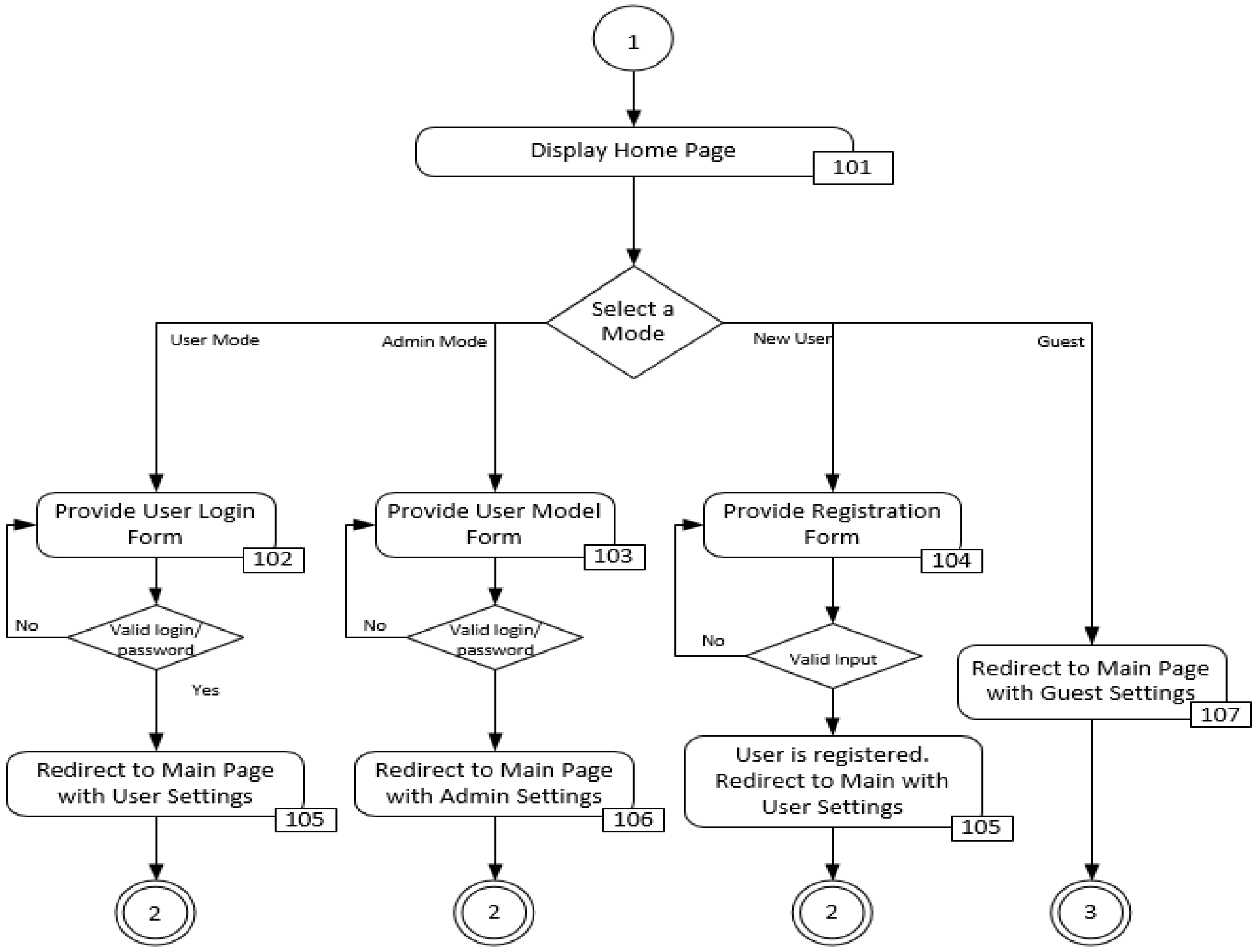
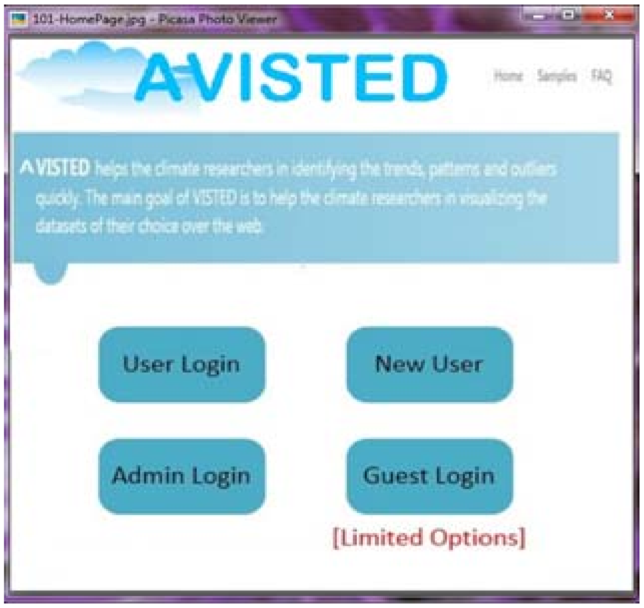

1
Display Home Page
101
Select a Mode
User Mode
Admin Mode
New User
Guest
Provide User Login Form
102
Provide User Model Form
103
Provide Registration Form
104
No
Valid login/ password
Yes
No
Valid login/ password
No
Valid Input
Redirect to Main Page with Guest Settings
107
Redirect to Main Page with User Settings
105
Redirect to Main Page with Admin Settings
106
User is registered. Redirect to Main with User Settings
105
2
2
2
3

Figure 8: Activity Diagram with all the Modes

The four modes of the AVISTED are shown in the Figure 8. The action “Display Home Page” is linked to the GUI design of Home page with link number 101. The design in the Figure 9 has been inspired from VISTED and is relinked back to the activity diagram by naming it with number 101.

Similarly, Figure 10 shows another activity diagram with the Guest features. The guests of AVISTED are only provided with Model Output Management and Model Output Features. The design of guest Main page is shown in the Figure 11. If guest wants to select a model from Model Output Management it will be navigated to the webpage with design name starting with 301 but if he/she wants to
select a default Model it will be navigated to the webpage with design name starting with 202.



AVISTED helps the climate researchers in identifying the trends, patterns and outliers quickly. The main goal of VISTED is to help the climate researchers in visualizing the datasets of their choice over the web.

Figure 9: Snapshot of the AVISTED home page

Currently there is no tool to make these links between the actions and the interfaces. We are intending to implement the technique by following a naming convention. A tool with this feature will greatly help the software development team.

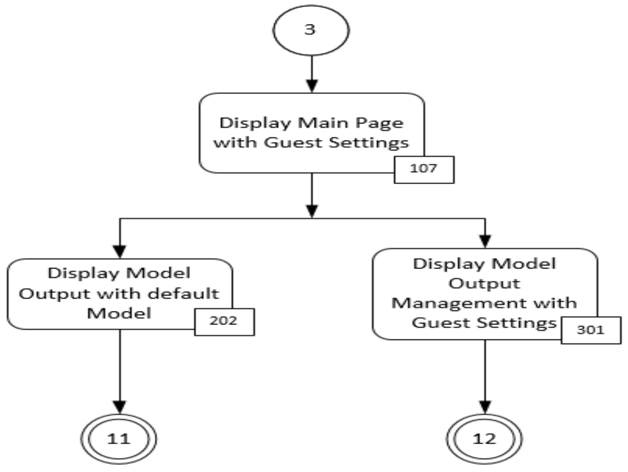

3
Display Main Page with Guest Settings
107
Display Model Output with default Model
202
Display Model Output Management with Guest Settings
301
11
12

Figure 10: Activity diagram with Guest Features of AVISTED

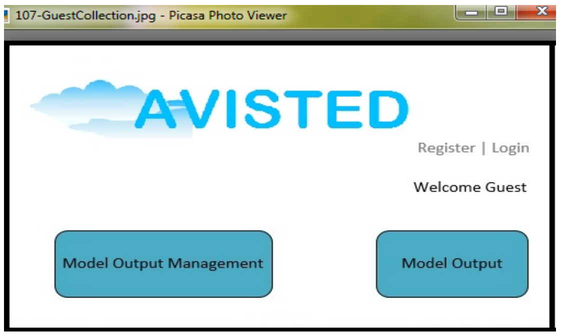


Figure 11: Designs of AVISTED with GEAD approach

In the next section we will discuss about the existing tools used in designing software systems.

V. Related Work

At present there are many tools available for designing software applications. They can be classified into categories based on their features like general purpose, special purpose, specific language, code generation, executable UML, desktop application, mobile application, online application, open source, commercial and freeware [5].

Some of the popular UML drawing tools include Rational Rose, SmartDraw, Microsoft Visio, Rational Rose, LucidDraw, and ArgoUML [6]. Some have support for activity diagrams and GUI builders, some have for GUI builder and some have for both like Visio, but they are static. None of them provide links between activity diagrams and GUI. These links are possible by building a HMTL based UML tool which allows drawing UML diagrams, GUI and providing HTML links between them. A similar HTML web application that allows drawing UML diagrams online is Diagramo [7]. It allows building UML designs online but it does not support links. Tools used in building such applications are Dreamweaver, sublime, Coda 2, and Brackets [8]. Tenzer [9] developed an interactive game to improve the UML designs. The user of the game is allowed to explore several variations of the designs while playing the game. This allows the users to improve their designs interactively.

VI. Future Work and Conclusion

To conclude, we have presented a new approach GEAD for software design. This approach links the static and behavioral diagrams. We have also introduced the software application AVISTED and illustrated the GEAD approach using it. And, finally we have discussed about tools used for software design. In future, we will be working on the development of a supporting tool for GEAD. By using the new tool in other software development projects GEAD approach can be refined further in terms of usability. Additionally, we will be completing the implementation of AVISTED.

ACKNOWLEDGMENT

This material is based in part upon work supported by the National Science Foundation under grant number IIA-1329469. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

1. "Software Design " Accessed on July 6, 2015, available at <https://en.wikipedia.org/wiki/Software_design>.
2. "UML 2.4 Diagrams Overview" Accessed on July 7, 2015, available at <http://www.uml-diagrams.org/uml-24-diagrams.html>.
3. "Modeling Output", Accessed on July 7, 2015, available at <http://sensor.nevada.edu/NCCP/Downloads/Modeling%20Output.aspx>.
4. Likhitha Ravi, Sergiu M. Dascalu, Frederick C. Harris, Jr., John Mejia, and Noureddine Belkhatir, " VISTED: A Visualization Toolset for Environmental Data", Proceedings of The 2015 International Conference on Computers and Their Application (CATA 2015), pp. 335-342, March 9-11, 2015.
5. "UML tools classified by categories", Accessed on July 13, 2015, available at <http://modeling-languages.com/uml-tools/#generic>.
6. "List of Unified Modeling Language tools", Accessed on July 12, 2015, available at <https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools>.
7. "HTML5 diagram editor", Accessed on July 13, 2015, available at < http://diagramo.com/editor/editor.php>.
8. "Choosing the Right Text Editor | Brackets, Sublime Text, Coda, and more" , Accessed on July 13, 2015, available at < http://blog.digitaltutors.com/brackets-coda-sublime-text-text-editor-choose/>.
9. Tenzer, J. "Improving UML design tools by formal games", IEEE International Conference on Software Engineering, pp.75 - 77, May 2004.