

Social Network For Citizen Scientist To Support The Development of Wise Management And Policy In Biodiversity

Fitri Nurjannah
Universitas Gunadarma
fitri_nurjannah@staff.gunadarma.ac.id

Kartika Dwintaputri S
Universitas Gunadarma
kartika_ds@staff.gunadarma.ac.id

Panchit Longpradit
Petchaburi Rajabat University
panchit@pbru.ac.th

Abstract

One of the existing problems in biodiversity is the species publishing species are quite complicated and time consuming. This is a cause that made a social network is used to support the publication in biodiversity. By using data from various sources of biodiversity, this social network is expected to help the development of data management and policies in biodiversity.

Keywords :Biodiversity, biodiversity informatics, social network, management, policy

1 Introduction

Biodiversity is an attribute of an area and specifically refers to the variety within and among living organisms, assemblages of living organisms, biotic communities, and biotic processes, whether naturally occurring or modified by humans. Biodiversity can be measured in terms of genetic diversity and the identity and number of different types of species, assemblages of species, biotic communities, and biotic processes, and the amount and structure of each. It can be observed and measured at any spatial scale ranging from microsites and habitat patches to the entire biosphere. [Don C. Delong(1996)]

Biodiversity informatics is an emerging field that applies information management tools to the management and analysis of species-occurrence, taxonomic character, and image data. The development and implementation of formal data exchange standards and query protocols have made it possible to integrate data holdings from collections around the world. [N.F.Johnson(2007)]

GTA (Groupware Task Analysis) which is used to support designing system is a method to design a system in which there is an agent (actor), task (task), and event (situation). This method is used to tell the interactions that occur between agents, tasks, and events. Recently, most people make the system without the using any particular method and the result is most of systems that has made

does not fit the needs of the target system manufacture. By using GTA, the system created will look more structured and focused, so that system builders can evaluate system deficiencies of the initial system was made.

Pattern is a term that refers to a general solution that can be used repeatedly to resolve common problems found in software design [Christopher Alexander, 1977]. In designing the interface design definitely need a method that does not damaging existing system or the system that has been previously designed. This method aims at designing interfaces according with a system which has been designed before, so the interface designer can design the according system design and not useless.

2 Methods

2.1 GTA

GTA is an approach to the early phases of the design of complex systems (task analysis, global design) that integrates ethnographic techniques and interaction analysis, analytic techniques from Human-Computer Interaction (HCI), formal task modelling, and design space analysis [van Welie(2001)]. Through GTA, it was that the present system is very rarely completely reliable by a user. Therefore, GTA paying attention to the study of a group or organization and their activities and not only examine a user when working. In essence,

the GTA consists of a concept that describes the relevant aspects of a set of jobs (tasks) that require attention when designing a groupware.

Aspect Of GTA

Task models for complex situations need to be composed of different aspects. Each describes the task world from a different viewpoint, and each relates to the others. Consequently, the resulting final task model will be redundant at the level of representation for human readers. [Gerrit C. van der Veer, Cristina Chisalita, martin van wellie]

1. **Agents.** The first aspect focuses on agents. "Agents" often indicates people, either individual or groups. Agents are considered in relation to the task world, hence, we need to make a distinction between agents as acting individuals or systems, and the roles they play.
2. **Works.** The concepts of task and goal in most frameworks have either a many to one or a one to one relation – several tasks may have the same goal, and each task has exactly one goal. In activity theory tasks are referred to as 'actions' (which are, like in HCI task analysis approaches, considered to be hierarchically structured), where long-term tasks are referred to as 'object' or 'motive'.
3. **Situations.** Analyzing a task world from the viewpoint of the situation means detecting and describing the environment (physical, conceptual, and social) and the objects in the environment. Object description includes an analysis of the object structure.

Euterpe

Euterpe <http://www.cs.vu.nl/~gerrit/gta/> is a Task Analysis tool developed at the Vrije Universiteit Amsterdam. It is based on an ontology for describing the task world in a structured way. The theory behind it is based of GTA. Euterpe helps to build task trees, object hierarchies and other important concepts such as event and roles. Templates allow detailed information to be specified and multimedia can be attached to concepts to clarify their nature. Documentation can be generated on paper and as HTML pages.

2.2 Pattern

Pattern is a term that refers to a general solution that can be used repeatedly to resolve common problems found in software design [Christopher Alexander, 1977]. A pattern is supposed to capture proven design knowledge and is described in terms of a problem, context and solution. The pattern does not form a final solution which can be directly translated into program code. Pattern design

for object-oriented usually show relationships and interactions between classes and objects, without explaining the end of classes and objects involved in an application.

Element of Pattern

Several different formats have been used to describe the pattern. Pattern's format which used in Alexander's work called "Alexandria form". Although the usage of patterns may differ, it is generally agreed that the pattern must contain certain essential elements. The following essential elements of pattern:

1. **Name.** Name is allows us to use a single word or short phrase to refer to the pattern, and the knowledge and structure it describes.
2. **Problem.** A statement of the problem which describes its intent.
3. **Context.** Context tells us the pattern's applicability. It can be thought of as the initial configuration of the system before the pattern is applied to it.
4. **Forces.** A description of the relevant forces and constraints and how they interact/conflict with one another and with goals we wish to achieve.
5. **Solution.** Solution is instructions which describe how to construct the necessary work products.
6. **Examples.** One or more sample applications of the pattern which illustrate.
7. **Resulting Context.** It describes the post conditions and side-effects of the pattern.
8. **Rationale.** This tells us how the pattern actually works, why it works, and why it is "good".
9. **Related Patterns.** The static and dynamic relationships between this pattern and others within the same pattern language or system.
10. **Known Uses.** This helps validate a pattern by verifying that it is indeed a proven solution to a recurring problem.

But there are often used, only five of the required elements are:

Pattern = {Name, Example, Problem, Context, Solution}

Even it could be called as a pattern when it already has five elements, it would be very convincing if all the elements have been found.

3 Design

The design is divided into 2 parts, the system design and user interface design. System design was made with GTA method and tools Euterpe meanwhile the user interface design was made with Pattern method.

3.1 System Design

System design includes the things that used to make applications run, whether it is theoretical or technical. Including things such as system architecture, design features, and database design.

Features Design

Design features designed to facilitate the design of the system being designed. As for the features that made, among others:

1. **Registers**. Registers used to register as a user application. Which must be done by the user is filling out a form that must be filled. Further fill out the form for additional information the user itself. This additional information is optional.
2. **Login/Logout**. Users who already have an account on the application, can directly log in through the website or mobile app. Data is required for login can be either username or email and password. Once logged in, the user will be redirected to the next page. Logout function will redirect the user to the index page of the web.
3. **Profile**. This page is a view to look at user data. On this page displayed data contributing to the user such as name, email, date of birth, and so on.
4. **New Report**. This feature is presented in 3 stages. In the first stage the user will be presented with a form to fill in report titles, taxonomy, and description. After that in the second stage the user can insert the photos they want to display in the report. The photos can be uploaded at this time is limited according with the provisions made. In the third stage the user is required to select a photo caption that will be used to display the report on page feed.
5. **Retrieve Taxon**. This function is found on the first page of the new report stage. The user can press the button which will take the data taxonomy from GBIF website <http://data.gbif.org/> according with the input species are included.
6. **XML Input**. XML input is provided for those users who are no longer common in the world

of biodiversity. Here the user can enter tags into xml form naming taxonomy of a species that does not exist in the form.

7. **Read Report**. On this page the user can view the complete information contained in a report, start from photo gallery, taxonomy, latitude and longitude (if any), and so on. On this page the user can also make comments and resource in the report. If users like the report that is being viewed, users can add a "like" on the report.

Tasks Design

Task can be done by the user or system. Each task will have a performing agent or agents that perform the task. The task can be performed by one or more agent / role. Some examples of the design task is:

- **Register Task**. Register Task consist of one subtask, ie fill out a form. Complete the form on the subtasks performed by the user agent. After the user fills in the data in the task form, the system will send the data to be stored into a database which can then be called back its data to other tasks.

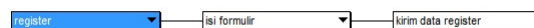


Figure 1: Register Task

- **Login Task**. Login task consists of several subtasks that horizontal / one direction. On this task the user will write down your username and password and then presses a button. When the button is pressed, the system will work to send data to the server, then match the data sent with the existing data on the server. Then the server will return a true or false value.



Figure 2: Login Task

- **Read Report Task**. This task is described by NOR constructor. Agent that can do one or more of the existing subtasks. Subtask post comments, write a re-source, and the like are run by the user agent (typing or pressing the button), and then the next subtask is executed by the system that is sending the data to be processed.

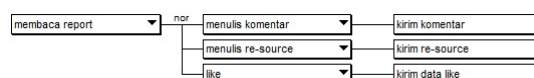


Figure 3: Read Report Task

System Architecture

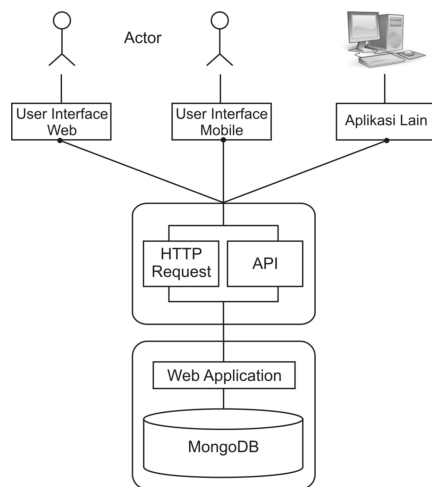


Figure 4: System Architecture

Figure 4 is a system architecture that has been built. Application user interface is on the client side that communicates with the server as the data provider via a web protocol HTTP (Hyper Text Transfer Protocol). The components in the system include:

- **Actors/Users.**
- **API.** API (Application Programming Interface) is a set of functions, components, and protocols provided by the system to replace the language used in the system calls in a language that is more structured and easily understood by programmers or other applications.
- **User Interface.** The user interface is divided into two parts, for web applications and mobile applications.
- **HTTP Request.** A request made by the client to the server is a web application for downloading files which will be shown a page that can be accessed / read.
- **Web Server.** Web server is a server software that functions accept HTTP or HTTPS requests from the client and sends the results back in the form of web pages. Server software that is used in this application is apache.
- **MongoDB.** Is a document-oriented database that consists of a key and a value. MongoDB does not recognize tables, columns, and rows that do not exist in a MongoDB schema (schema-less).

Data Structure

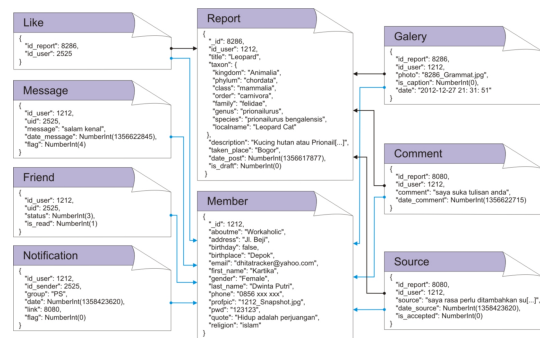


Figure 5: Data Structure

Basically, MongoDB does not recognize tables, columns and rows so that there is no schema (schema-less). So illustrative only use simple structure as shown in Figure 5.

3.2 User Interface Design

In the process of designing, bridging the gap between system analysis and detail of design is a difficult step. Details of design is also a difficult step when designers had to create a design that is useful to the user. In addition to the results of the analysis has been made, design knowledge also plays as an important role in site development. Therefore pattern is needed in the search for solutions to the problems encountered and applied in the design and implementation. Pattern here focuses on documentation and use of design knowledge by the designer.

The Target Audience & Their Needs

- **Biodiversity Scientist,** The biodiversity scientist can search biodiversity data that have been published.
- **Photographer,** photographer can channel their hobby by uploading photos while helping completing biodiversity data.
- **Student,** student who needs biodiversity data to fulfill their task.
- **Common People and Citizen Scientist,** In this case, both the people who have the ability to use a computer/mobile and access to the internet as well as scientists who are not focused on biodiversity can also use this application.

Application Design Considerations

- **User and Report.** How users can comfortably when creating a new report without any confusion.

- **Personalization and Customization.** The design should be flexible so that users can easily edit the report.
- **Social Network Aspect.** The main thing that is required is a connection between the users.
- **Biodiversity Research Support.** Biodiversity research requires publication in the naming of species, fast information delivery, and a database with clear standardization in the naming of species.
- **License.** This website will eventually accommodate many authentic picture of each report. This becomes a problem of its license.

Pattern Example

This is one pattern example that have been used.

Name: Tabs

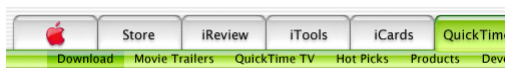


Figure 6: Pattern Example Tabs

Example:

Context: The main navigational design with consideration of horizontal form. Tab is a common form of primary navigation structure information for up to 10 items or less.

Problems: Users need to access a particular section of the entire available information.

Forces: Tabs are inspired by their use in file cabinets where they separate sections of files. Although there are no real differences with a normal horizontal bar, the shape of the tabs makes the menu less boring and more visually outstanding.

Solution: Show a horizontal row of tabs with the section label in the tab. Highlight the currently selected tab and make all other tabs clickable.

4 Implementation

The system was created represented with a user interface designed using pattern method. One example of the use of tabs pattern in the read report system:

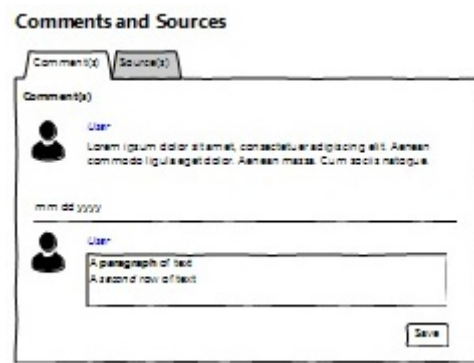


Figure 7: Prototype Design

COMMENTS AND RE-SOURCES



Figure 8: Pattern Implementation

5 Conclusion

System that has been created is applied in the form of web and mobile applications. But at this writing are discussed only in the system design and user interface design.

The system starts from user login till the user logout. Design System made by using aspects in the GTA with Euterpe tools. And User Interface Design made using Pattern made using pattern method.

System and usability evaluation results show that the system design and interface design created have fulfilled most of the needs of users and easy to use.

With the development of this social network, the data collected can create a new policy that allows the citizen scientist assisted in the development and data management of biodiversity.

References

- [eut(2012)] , 2012. Euterpe. [Online; accessed 20-October-2012].
URL <http://www.cs.vu.nl/gerrit/gta/>
- [wel(2012)] , 2012. Wellie.com patterns in interaction design. [Online; accessed 06-October-2012].
URL <http://welie.com/patterns/index.php>
- [D. Pinelle(2003)] D. Pinelle, C. Gutwin, S. G., Dec. 2003. Task analysis for groupware usability evaluation. ACM Trans. Comput.-Hum.

- Interact. 10 (4), 281–311.
URL <http://portal.acm.org/citation.cfm?doid=966930.966932>
- [Don C. Delong(1996)] Don C. Delong, J., 1996. Defining biodiversity. *Wildlife Society Bulletin* 24, 738–749.
- [Lehti and Ruokonen(2012)] Lehti, L., Ruokonen, A., 2012. Foundation of the patterns. [Downloaded; at 06-October-2012].
URL <http://www.cs.tut.fi/~kk/webstuff/Foundationofpatterns.pdf>
- [N.F.Johnson(2007)] N.F.Johnson, 2007. Biodiversity in. *Annual Review of Ent* 52, 421–438.
URL <http://www.annualreviews.org/doi/abs/10.1146/annurev.ento.52.110405.091259>
- [v. W. Gerrit C. van der Veer(2007)] v. W. Gerrit C. van der Veer, M., 2007. Groupware task analysis. Tech. rep., Department of Computer Science, Vrije University.
- [van Welie(2001)] van Welie, M., 2001. Task-based user interface design. Ph.D. thesis, SIKS.