BULLETIN

PRAGUE COLLEGE
CENTRE FOR RESEARCH & INTERDISCIPLINARY STUDIES

CRIS 2015/02

SCHOOL OF IT & COMPUTING
SELECTION SHOWCASE

7

BARTER ONLINE NETWORK

Van Ngoc Tran

25

COMPARISON OF BEAM-FORMING & RELAYING IN SPARSE SENSOR NETWORKS

Mikuláš Krebs

This study focuses on the differences in power consumption between beam-forming and relaying data transmission methods in a sparse wireless ad-hoc network. These two methods are observed for the same parameters using an identical network topology in a simulation programme that was developed as a part of this study.

41

MYETL: A JAVA SOFTWARE TOOL TO EXTRACT, TRANSFORM, & LOAD YOUR BUSINESS

Michele Nuovo

The paper follows the development of a Java Software Tool that extracts data from Flat File (Fixed Length Record Type), CSV (Comma Separated Values), and XLS (Microsoft Excel 97-2003 Worksheet file), apply transformation to those sources, and finally load the data into the end target RDBMS.

53

ONTOLOGY TRANSLATION: THE SEMIOTIC ENGINEERING OF CONTENT MANAGEMENT SYSTEMS

Alejandro Villamarin M.

The paper proposes the application of Semiotic Engineering theory to Content Management Systems (CMS) focusing on the analysis of how the use of different ontologies can affect the user's efficiency when performing tasks in a CMS. The analysis is performed using the W-SIDE model.

DESIGN & LAYOUT
BAB design studio | www.babds.com

To submit your work, please contact the project coordinator of CRIS:
Bohus Ziskal <cris@praguecollege.cz>

Printed in Prague, Czech Republic.

ISSN 1805-5109

PREFACE

The Centre for Research and Interdisciplinary Studies (CRIS) was founded with the idea of developing interdisciplinary research crossing several fields and subject areas underlying the academic curricula at Prague College, and its main purposes are:

- To promote a medium of participation and discussions by means of regular interdisciplinary workshops and seminars.
- To promote and to encourage the collaboration among different schools and programmes in the design and creation of multidisciplinary courses in the college.
- To provide a means of publishing research work for both students and staff as part of a quarterly academic bulletin and e-journal.
- To cooperate with other education institutions and organisations in the development of common projects of interest.

The Centre was developed from projects initiated by Stefano Cavagnetto in the context of his role as Head of the School of Business and the School of Computing, by Bruce Gahir, Principal Lecturer in the School of Business and Computing, and by Pascal Silondi, Director of Libat and Principal Lecturer in Interactive Media. Beginning in 2009, research in the following areas had been initiated:

1. Game theory and its application to economics, business, philosophy, and international relations.
2. The history of programming languages and history of computers.
3. Experimental media (Prague College and the CRIS, formerly PCRC, is an associate partner for Underground City XXI an international interdisciplinary EU project).
4. The history of cryptology and the science of enciphering.
5. Art and mathematics: a profitable relationship in history - from classical geometry to fractals and topology.

By combining academic study with practical training, the CRIS aims to create an environment where personal achievement goes hand-in-hand with social responsibility. Strategically, this offers students the chance to actively collaborate in several research areas with the support of faculty members and lecturers of the college.

Since 2010 a quarterly Bulletin has been published detailing progress in relevant research activities of lecturers and students. This bulletin forms an integral part of the CRIS and provides a medium whereby the research activities of the centre can be documented. Faculty members, lecturers and students belonging to every school of the college are welcome to submit their work for publication.

You can find the published Bulletins of CRIS on Ebrary (electronic library), in the college library, in six Prague libraries (Narodni knihovna, Knihovna Narodniho muzea v Praze, Ministerstvo kultury CR, Parlamentni knihovna, Mestska knihovna v Praze, Knihovna a tiskarna pro nevidome K.E. Macana), Moravska zemska knihovna in Brno, Stredoceska vedecka knihovna in Kladno, Jihoceska vedecka knihovna in Ceske Budejovice, Studijni a vedecka knihovna Plzenskeho kraje in Plzen, Severoceska vedecka knihovna in Usti nad Labem, Krajska vedecka knihovna in Liberec, Studijni a vedecka knihovna in Hradec Kralove, Moravskoslezska vedecka knihovna in Ostrava, Vedecka knihovna in Olomouc, Krajska knihovna in Pardubice, Havlickuv Brod, Zlin, and Karlovy Vary.

Deadlines for the next issue in 2016 is 31st July 2016.

BARTER ONLINE NETWORK

VAN NGOC TRAN

Barter is the direct exchange of goods or services without using a medium of exchange, such as money. Barter faced a number of limitations, and according to Smith (1776), these limitations led to the emergence of money. However, trading with money also exposes traders to the problems of monetary economy such as inflation, deflation, currency devaluation, and currency exchange fluctuation. According to Statista.com (2015), in 2016, global Business to Customer (B2C) e-commerce sales will reach 1.92 trillion US dollars. On the other hand, online barter solutions are rare on the market. The only attempts to tackle online barter are mobile applications, carried out by small businesses. The market gap is caused by the unsolved inefficiencies of barter. The aim of this thesis is to identify the problems of barter, propose an IT solution for the problems of barter, and finally, produce an artefact, which is the realisation of the proposed IT solution by utilising computer systems and computer algorithms.

1. INTRODUCTION

Before money, there was bartering. According to a journal article by Dalton (1982, p. 181), barter stands for a moneyless market exchange. Barter is defined as a direct exchange of goods and services without using a medium of exchange such as money. Barter inefficiencies are the problems of barter. These inefficiencies will be defined and solved via a proposed model.

2. PROBLEM DEFINITION: BARTER INEFFICIENCIES

The first problem of barter is the **coincidence of wants**. Coincidence of wants in barter is described by Jevons (1875) as "*the first difficulty in barter is to find two persons whose disposable possessions mutually suit each other's wants. There may be many people wanting, and many possessing those things wanted; but to allow of an act of barter, there must be a double coincidence, which will rarely happen.*" The coincidence of wants problem is a transaction cost which imposes limitations on trade under a barter system. To understand how severe this limitation is, let us examine how trade works under a traditional barter scenario (non-electronic barter).

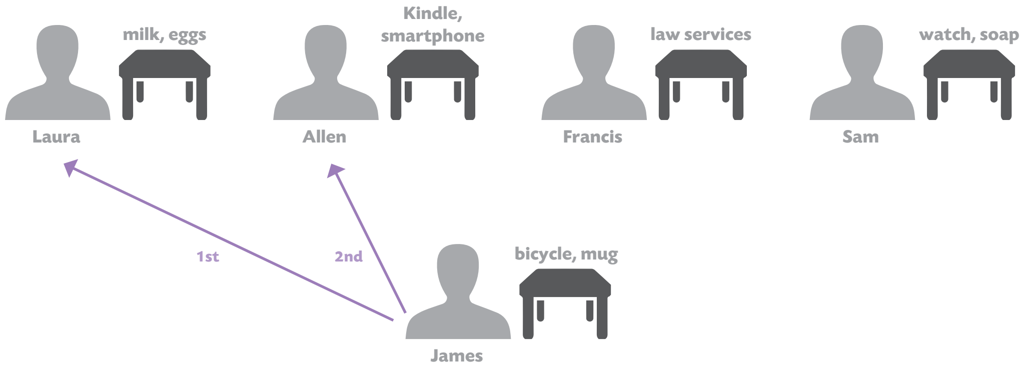


Figure 2.1: Trade under tradition barter scenarios

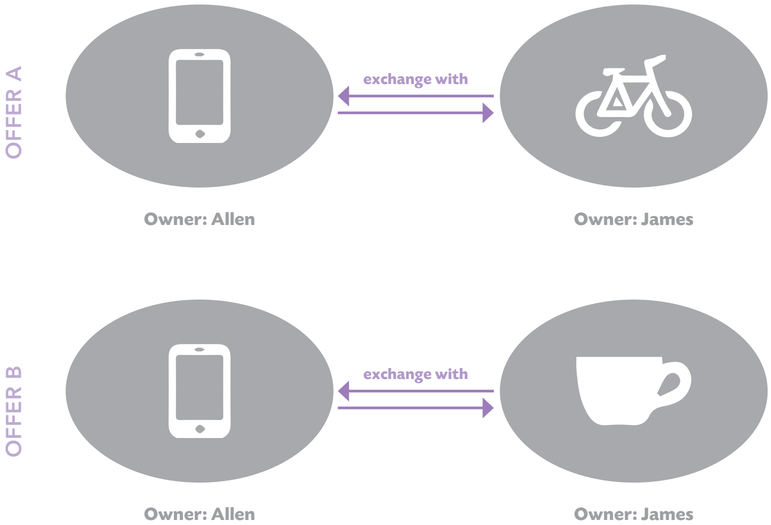


Figure 2.2: Barter offers from James to Allen

In a traditional barter scenario, trades happen in a physical market, as seen in Figure 2.1. People who wish to trade join the market with items they have. Let us follow the example seen in Figure 2.1. James wishes to trade, hence James joins the market with his items. James must go to one seller at a time to examine what they are selling and only when James identifies an item he wants, James then can offer an exchange with one of his items in return. First, James visits Laura. James is not interested in Laura's items. James then visits Allen. Suppose James finds an item he wants, which is the smartphone Allen is selling. James then offers Allen to exchange the smartphone for one of James' items. The two offers James makes to Allen are visualised in Figure 2.2.

For James, these offers are based on an identified need. James has seen, examined, and decided that he wants one of Allen's items. On the other hand, Allen sees James' items for the first time, thus it is possible that Allen could agree with one of James' offers, but it is also possible that Allen will not agree with any offers made by James. This is the problem of coincidence of wants. Whether Allen agrees to an exchange or not, is *a matter or coincidence*. As James goes around the market, making offers to other sellers, but there are *no guarantees* that an exchange will happen.

The second problem of barter is that **multilateral trade is less feasible**. Barter can be bilateral as well as multilateral. Bilateral barter is an exchange between two parties, visualised in Figure 2.3. User A wants an item of User B and User B wants an item of User A. Multilateral barter is an exchange between multiple parties (more than two), visualised in Figure 2.4. User A wants an item of User B, but User B doesn't want item of User A. However, User B wants an item of User C, and User C happens to want an item of User A, making it a full circle of wants.

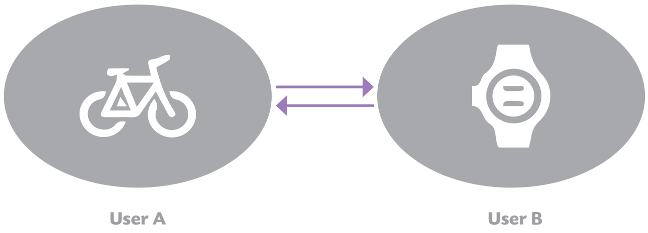


Figure 2.3: Bilateral exchange

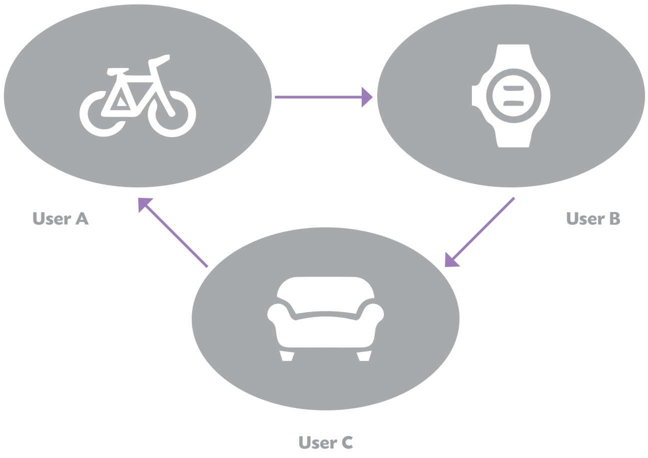


Figure 2.4: Multilateral exchange

Why is multilateral trade important? When using money to trade, multilateral trade happens regularly. Examine the following example. James is a doctor. James treats a patient and gets paid in return. James then buys bread from a baker and pays with his money. The baker then uses the money to pay for his car wash. The owner of the car wash service then uses the money to pay his rent to the landlord. This is a trade between multiple parties - a patient, a doctor, a baker, a car wash services owner, and a landlord. If trade was limited only to two parties, then right at the beginning, James cannot treat the patient if the patient does not have something James want in return. This would be a major inefficiency.

In barter, as the number of parties involved in an exchange grows, the problem of coincidence of wants *aggregates*, thus making multilateral trade less feasible than bilateral trade. This is a major inefficiency that needs to be solved.

The third problem of barter is **inefficient shopping experience**. When shopping using money, the shopping flow is separated into two modules. In the first module, the shopper browses all of the items available in a shop and chooses what he/she wants. In the second module, the shopper makes a transaction by paying with money. When bartering, the shopping experience can be also divided into two modules: browsing/picking items and making a transaction. However, when bartering, these modules are repeated every time the shopper visits a different seller. Another inefficient shopping experience is that in a physical market, sellers can communicate with another seller one at the time. For example, while James communicates with Laura, no one can approach James to check out his items. This means that in a market full of sellers, many people might have to wait in line until a person becomes available. Finally, another inefficient shopping experience is that items are displayed per seller, and not based on its functionality. All of these problems cause barter to be time consuming. It is concluded that the identified problems are: coincidence of wants, less feasible multilateral trade, and inefficient shopping experience.

3. PROBLEM SOLVING PROCESSES

The problem solving processes show the construction of how each identified problem was solved. First, problem solving techniques in computer science were chosen to set a general approach to solving problems. The techniques utilised are *divide and conquer* and *abductive reasoning*. The *divide and conquer* strategy decomposes a problem into subtasks. These subtasks are then conquered by *abductive reasoning*, which uses logic and reasoning to find the simplest and most likely solution to the problem.

The identified problems - *coincidence of wants*, *less feasible multilateral trade*, and *inefficient shopping experience*, were solved using the following processes:

1. *Analyse the problem* – understand what the problem is about.
2. *Decompose the problem* – use the divide and conquer technique to decompose the problem into subtasks.
3. *Solve each subtask* - find the simplest and most likely solution to the problem.
4. *Develop an algorithm (optional)* – develop an algorithm to solve a problem based on its solutions. The algorithm contains a step-by-step set of operations.
5. *Check algorithm efficiency (optional)* – calculate algorithm efficiency.
6. *Reflect on the whole process* – conclude in findings.

Steps 4 and 5 are optional. An algorithm is developed only when the solution demands a computer algorithm to solve a problem. The following sections show how each identified problem was solved.

3.1 COINCIDENCE OF WANTS

The first problem to be solved is *coincidence of wants*, which is defined as the difficulty in barter to find two persons whose disposable possessions mutually suit each other's wants. The problem can be understood as:

- Scenario: there are two people whose item mutually suits each other's wants.
- Problem: it is difficult to find (match) these people.

Let us decompose the problem into subtasks (see Table 3.1). Let us call the two people whose item mutually suit each other's wants as person A and person B.

Subtask	Problem
#1	How does person A find and show interest in an item of person B? How does person B find and show interest in an item of person A?
#2	How can person A and person B efficiently match their items that mutually suits each other's wants (if the market is composed of x number of sellers)?

Table 3.1: Subtasks to the problem of coincidence of wants

Let us find the simplest solutions to these subtasks.

Subtask #1 poses the same question, only from the perspective of two different people. The question can be abstracted to "how does any person find an item and show interest in an item of any other person?" The simplest solution to the problem is for the IT solution to collect all of the information of all the people, items, and interests into a database. Any person then can find any item belonging to any user in a table which collects data about items. When a person shows interest in an item, the interest is recorded in the interest table. Any software depends on a database. A database is a collection of data. The solution to build a database, which collects and organise all data necessary for barter, is concluded as the simplest and the most likely solution to the problem.

Subtask #2 poses the question "how can person A and person B efficiently match their items as mutually suiting each other's wants (if the market is composed of x number of sellers)?"

The key to understanding this question is the keyword ' efficiently', which means that the solution, aside from solving the problem, must also provide efficiency. The simplest solution to this problem is to develop a computer algorithm to match an item of person A and an item of person B that mutually suits each other's wants. As information about people, items, and interests are collected into a database, a computer algorithm can be developed to take the collected data as input and proceed to find matches of mutual wants. The purpose of a computer algorithm is to effectively calculate a task. The solution of using a computer algorithm for a calculation of tasks instead of leaving the calculation to humans is justified as humans are prone to error and the size of a database of a typical software is not feasible for a human to process. It is concluded that the solution is the simplest and the most likely solution to the problem.

3.2 MULTILATERAL TRADE FEASIBILITY

The second problem to be solved is that *multilateral trade is less feasible than bilateral trade*, due to the problem of coincidence of wants aggregating as the number of participants grows within a barter transaction. The root cause of the problem is the problem of coincidence of wants. If this root cause is solved, the problem is solved. The problem of coincidence of wants has been already solved, thus we already have a solution to the problem of multilateral trade feasibility.

3.3 INEFFICIENT SHOPPING EXPERIENCE

The third problem of barter is *inefficient shopping experience*, which includes: repetition of tasks as a person move of one seller to another seller, one seller can communicate with another seller one at the time, and items are displayed per seller, not based on the functionality of an item. These problems cause barter to be more time consuming than shopping with money. Let us decompose the problems into subtasks (see Table 3.2).

Subtask	Problem
#1	How can a person avoid repeating the same barter steps every time the person visits a different seller?
#2	How can a person avoid being limited to bartering with one person at a time?
#3	How can a person browse items based on functionality of the item rather than to whom the item belongs to?

Table 3.2: Subtasks of the problem of inefficient shopping experience

Subtask #1 poses the question "how can a person avoid repeating the same barter steps every time the person visits a different seller?"

The simplest solution to this problem is for the IT solution to break the barter process into stages that mimics the shopping experience. The stages are:

1. Person X browses all items.
2. Person X shows interest in items.
3. A computer algorithm finds mutual wants (internal step, performed by the IT solution itself)
4. Person X makes a barter transaction by: browsing mutual wants, choosing what he/she likes, and performing the barter.

Subtask #2 poses the question "how can a person avoid being limited to bartering with one person at a time?"

The solution to Subtask #1 already solves the problem of Subtask #2. By breaking the barter process into the above mentioned stages, a person is not limited to another person one at the time. Instead, any person can browse all items at any time, no matter who is the owner. Any person should be able to show interest in any item as well.

Subtask #3 poses the question "how can a person browse items based on functionality of the item rather than to whom the item belongs to?"

The simplest solution to the problem is to display items to people based on functionality (just as items in a shop are displayed based on its category) rather than based who owns them. The IT solution will be consisted of a presentation layer, where data is fetched from the database but presented to people based on a set of business logic. Display items to people based on functionality shall be one of the business logic aspects.

It is concluded that mimicking a standard shopping experience is the simplest and most likely solution to the problems.

4. PROPOSED MODEL

The proposed model is **an electronic barter (e-barter) system**, hereinafter referred to as *the system*. The system is a barter network of people who wish to trade products or services for other products or services, hereinafter referred to as *an item*. The system implements the solutions to the problems of barter inefficiency. The system access point shall be a web application, accessible through desktop computers and mobile devices. The back-end is consisted of a database. The objective is to showcase the system as a proof of concept.

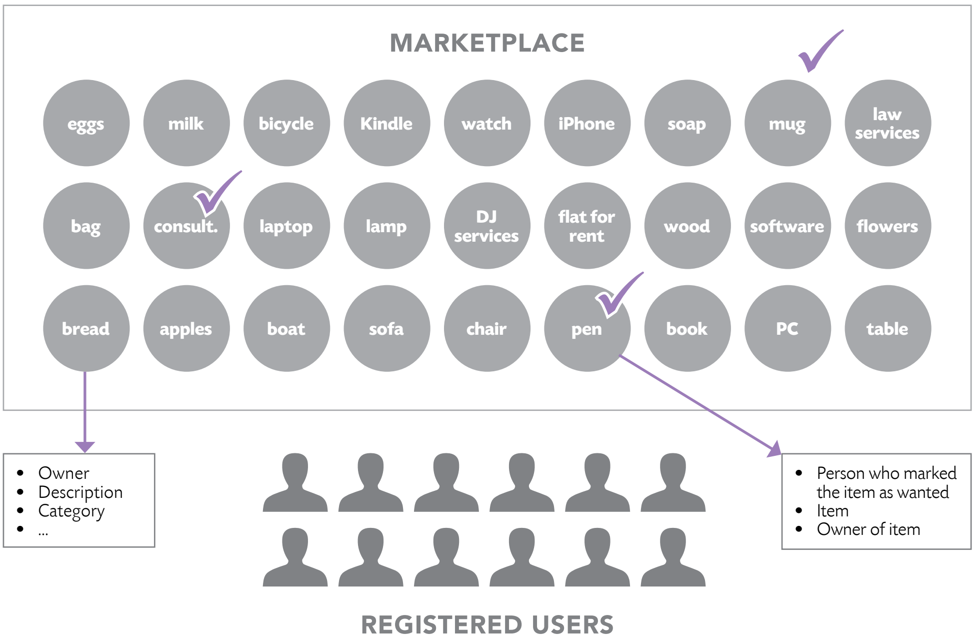


Figure 4.1: Proposed model – marketplace

Just like a physical barter market, the system is consisted of participants, referred to as users. Anyone can become a user by registering into the system using their email address.

Just like a physical barter market, a user has items. The first shift from the traditional barter scenario is that here, users' items are not presented on separate tables belonging to individual sellers; instead they are presented together in a pool of items. This is called a **marketplace**, as visualised in Figure 4.1. The marketplace is comprised of all items from all users. Each item contains information such as: owner of the item, title, description, category, subcategory, and images. The marketplace serves one purpose, which is getting users identify their items of interest by letting them find and mark any items they want as wanted. The marketplace shall have search and filtering functionalities, making it more convenient for users to find a specific item under specific criteria. The introduction of the marketplace removes identified problems of barter described in Chapter 2 by:

- Making shopping experience modular. A user does not need to complete all phases of trade before moving to another seller. The marketplace resembles a typical shop, where users pick items they want, regardless of who owns the items and what they have to offer in return.
- Making the shopping experience more efficient due to the search and filter functionalities (items can be displayed based on their functionality).
- The virtual market also lifts the restraint of sellers only communicating with each other one at a time, as seen in a physical market.

When a user marks an item as wanted (represented as a checkmark in Figure 4.1), the system collects **a buy request**. A buy request contains information, such as the person who marked the item as wanted, the item itself, and the owner of the item. From this table of information, the system shall run the algorithm to find items of mutual wants. Mutual wants can be in one-to-one, one-to-many or many-to-many relationship, visualised in Figure 4.2.

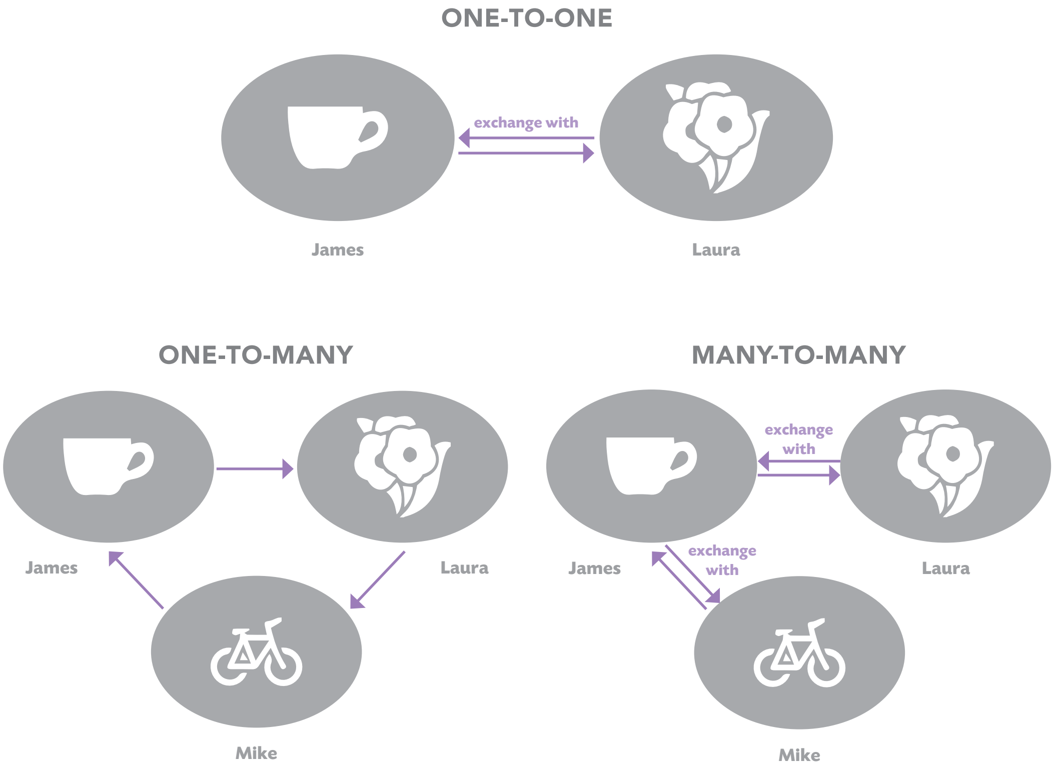


Figure 4.2: Mutual wants

These mutual wants found by the system are barter options, which are presented to a user in his/her **shopping cart**. For each of his/her item, barter option/s will be shown. These barter options are no longer dependent on coincidence of wants; rather each option consists of **identified wants**. For example, supposed James and Laura find a barter option seen in Figure 4.2 (one-to-one example) in their shopping cart. This option can tell us that James marked item flowers from Laura as wanted and Laura marked item mug from James as wanted. By letting the system find mutual wants based on users' expressed interest, another problem of barter, *coincidence of wants*, is removed.

When a user picks an option, a **transaction** is created. A transaction is considered completed when all parties involved in the exchange picked the same option and all parties involved in the exchange performed handover their items. An optional step in the transaction is rating the item received.

The problems of barter identified in Chapter 2 have been tackled and removed. Barter in the proposed model is based on identified wants. As the problem of coincidence of wants was removed, multilateral exchanges are feasible and increase the probability of exchanges occurring. Lastly, the business logic and proposed features improve the barter shopping experience.

5. DESIGN

The e-barter system is designed according to the proposed model. The e-barter system follows three-tier architecture, visualised in Figure 5:

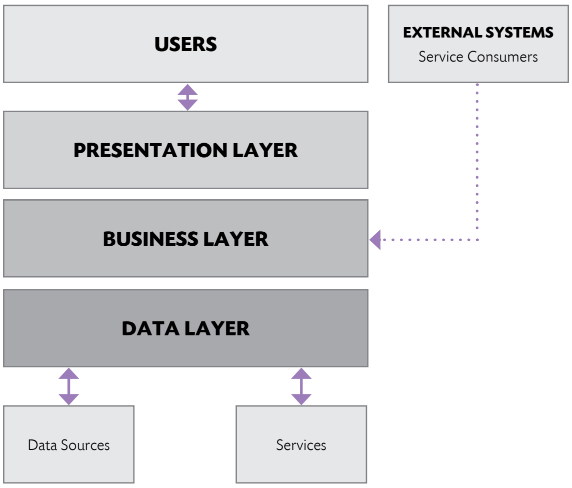


Figure 5: Three tier architecture

The three-tier architecture separates the system into three logical layers.

- The **presentation layer** is the user's access point. The presentation layer can be presented in the form of web applications or native mobile applications. The presentation layer manages the user interaction with the system via a user-interface (UI).
- The **business logic layer** enforces business logic, business rules, and business algorithms. The business logic layer implements core functionalities.
- The **data layer** provides access to the system data, in the form of a database.

5.1. DATA LAYER DESIGN

The database design is seen in Figure 5.1.

The Proposed Model was reviewed to determine what information is needed to be collected. Data which can form real-world objects were identified and then turned into tables. These tables are seen as full-lined in Figure 5.1. Each row of data in these tables represent an elementary object, meaning not further decomposable. This ensures that when the database is being expanded, it expands outwards, and existing tables will not have to be broken down into more tables. Relationships between these tables were identified and junction tables were created, wherever necessary. Junction tables are seen as dotted-lined in Figure 5.1.

The database is then checked for violations of database normal forms; this process is called *database normalisation*. Database normalisation is the process of organising data in relational databases, which reduces or eliminates data redundancy and prevent insert, update, and deletion anomalies. The database is in *First Normal Form (1NF)*:

- Each table has a unique identifier in the form of primary key.
- Each field contains one value.
- No repeating groups of columns.

The database is in *Second Normal Form (2NF)*, which means that any non-key fields are dependent on the entire primary key. No tables uses compound keys, thus the 2NF is satisfied.

The database is in *Third Normal Form (3NF)*, which states that no non-key fields are dependent on another non-key field. No tables violate the 3NF.

The UML has been colour-coded to show the four entities (an entity refers to tables which are in a close relations with each other), which make up the database.

5.2 BUSINESS LOGIC LAYER DESIGN

The business logic layer concerns with the retrieval, processing, transformation, and management of application data and the application of business rules. The business logic layer consists of:

- **Business entities**, which are business objects that represent real world elements, such as login, customers, or orders.
- **Business workflows**, which are patterns of a user's activity, dictating how a user's tasks are completed to satisfy business process and rules.

The proposed model and software requirement specification were examined and the business entities have been designed to reflect these requirements. The application is consisted of the following entities: *Register, Login, My Profile, Profile, My Items, Marketplace, My Cart,* and *My Transactions*.

The *Register entity* concerns with user registration. An unregistered user must register in order to access the application. A user registers via a form, filling in their name, email address, and password. After registering, the user can immediately log into the system.

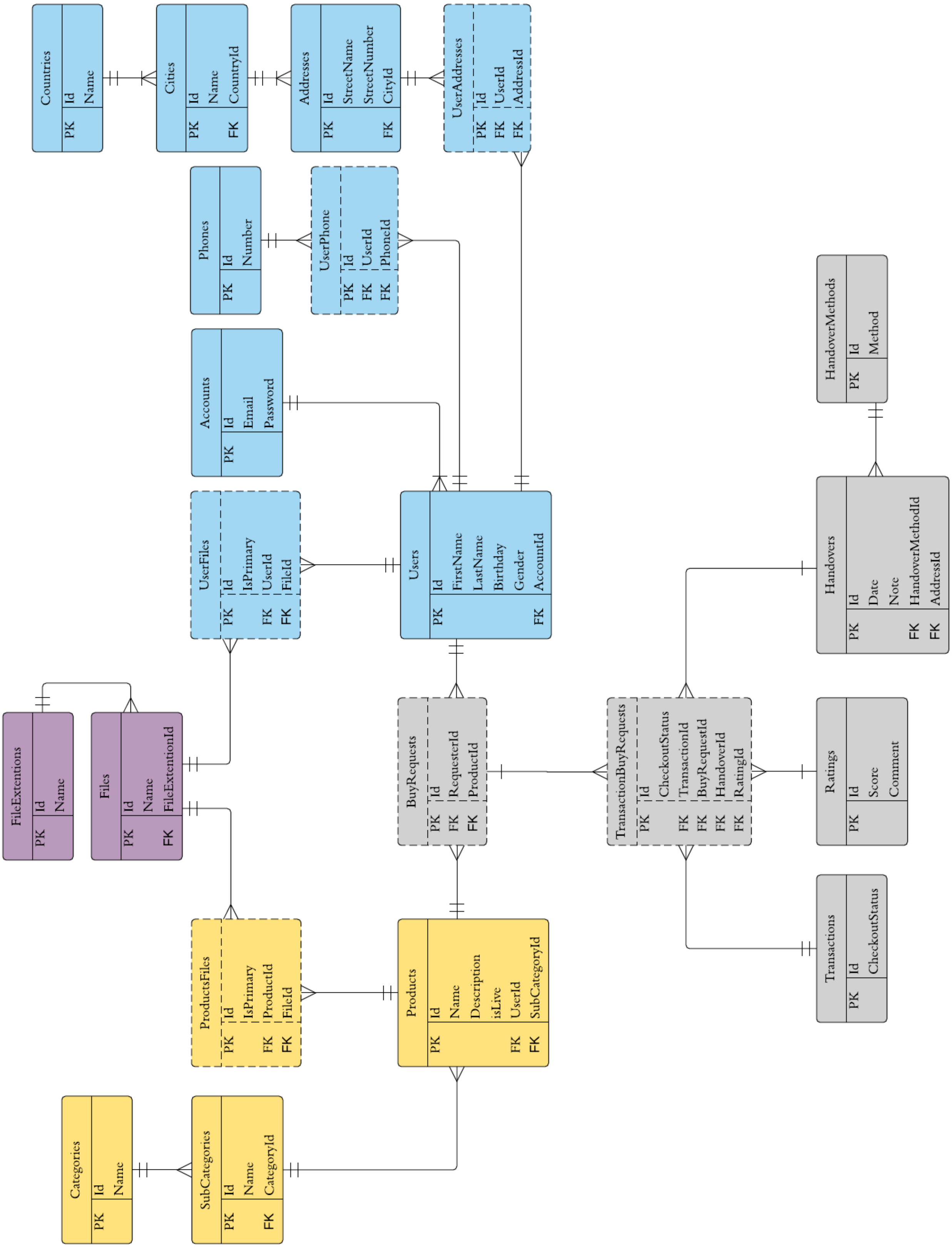


Figure 5.1 Database design, UM

The *Login entity* is concerned with user login and logout processes. A user can login to the system, by providing user credentials, as well as log out of the system. While logged into the system, the user is associated with an ID and can perform any of the functionalities offered by the application while being recognised as a particular user.

The *My Profile entity* deals with the profile of the logged in user. User profile stores personal information and account information. A user profile is public, thus accessible to both the owner and other users. A user should be able to edit their profile.

The *Profile entity* deals with the profiles of other users. All profiles are public, thus any user can view the profile of other users; however, only with view permission. Profile shows basic information, the items belonging to that particular user and how the user is rated.

The *My Items entity* is concerned with the management of users' items. The My Item entity shows the logged-in user their items, and the user can perform CRUD actions on these items, such as add, edit, view, and delete items. When individual items are viewed, the user should be able to see a list of people who marked the item as wanted.

The *Marketplace entity* is concerned with the display of items of other sellers. The one action a user can perform is to mark any of these items as wanted, as well as undo the action.

The *My Cart entity* is concerned with the display of barter options, found by the system algorithm. My cart will list all the items which are ready to be checked out. There can be one barter option or multiple barter options for each item. The user then must checkout (pick) one option to start the transaction process.

The *My Transaction entity* concerns with the display of user transactions. When a particular transaction is viewed, we shall see a three step process: accept transaction, handover items, and rate transaction. A transaction is considered finished if all of the participants within a transaction finished the first two steps. The third step – rating – is an option.

By examining the business entities, the business workflow has been worked out as seen in Figure 5.2.

5.3 PRESENTATION LAYER DESIGN

In order to design the presentation layer, the application type must be determined. The proposed model established that the application type shall be a web application. To satisfy the requirement of the application being accessible via desktop computers and mobile devices, a responsive web design approach will be utilised. The responsive web design technique adapts application layout to any device's resolution, hence providing multi-device experience across smartphones, tablets, laptops, and desktop computers while running a single web application (as opposed to having a separate web version and a separate mobile version).

The presentation layer is the access point for users. The user interacts with the system via the application UI, consisted of UI components, which accept user input and controls what information is display to the user. The UI has been designed to maximise user experience, as well as to reflect the requirements in Chapter 2. Some of the UI designs are seen in the next pages.

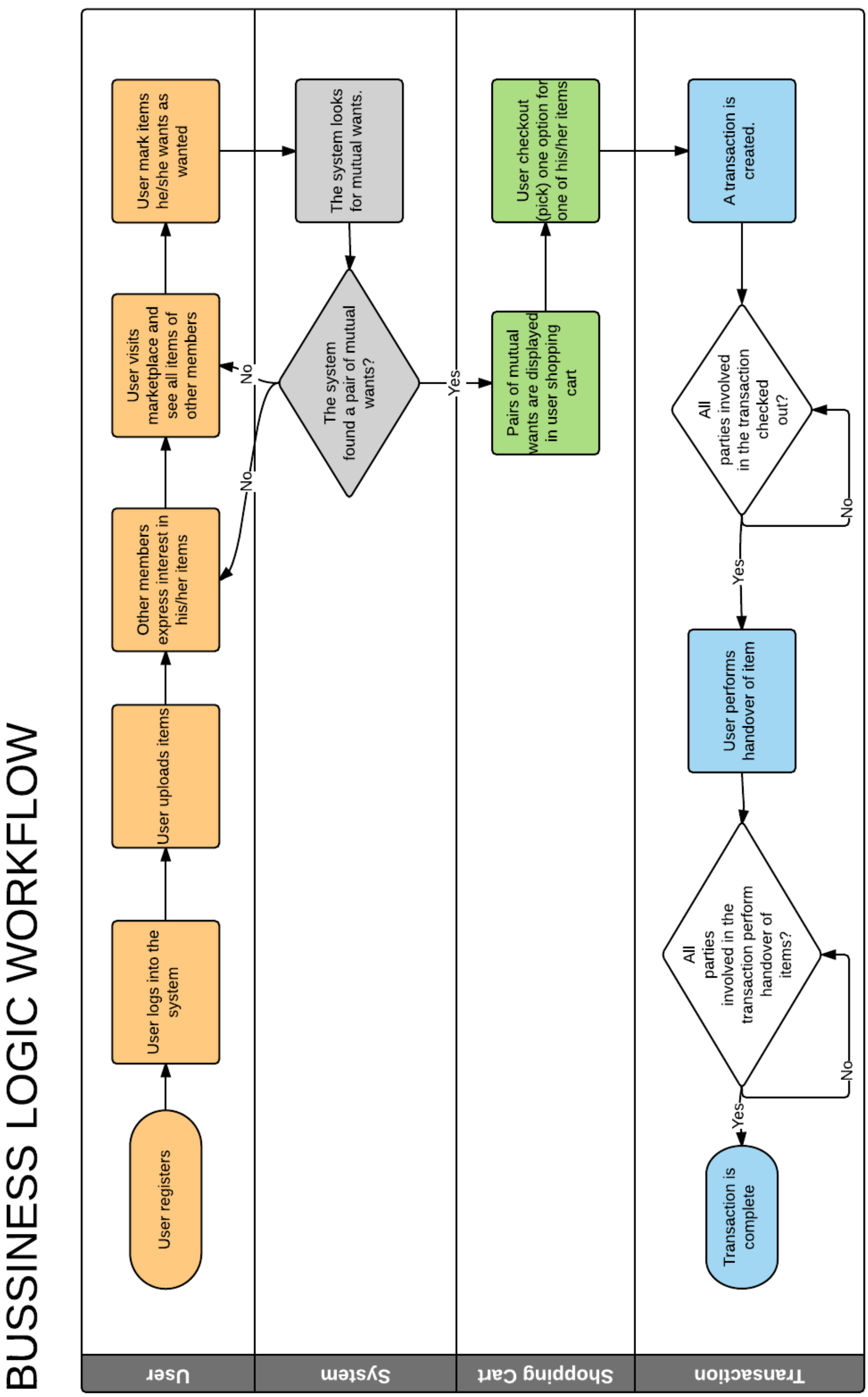

BUSSINESS LOGIC WORKFLOW
User
System
Shopping Cart
Transaction
User registers
User logs into the system
User uploads items
Other members express interest in his/her items
User visits marketplace and see all items of other members
User mark items he/she wants as wanted
The system looks for mutual wants.
The system found a pair of mutual wants?
No
No
Yes
Pairs of mutual wants are displayed in user shopping cart
User checkout (pick) one option for one of his/her items
A transaction is created.
All parties involved in the transaction checked out?
No
Yes
User performs handover of item
All parties involved in the transaction perform handover of items?
No
Yes
Transaction is complete

Figure 5.2: Business logic workflow

6. IMPLEMENTATION

The e-barter system was developed using C#, ASP.NET MVC, Entity Framework, and SQL Server as server-side technologies. Client-side technologies used were *Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), Jquery (Javascript Framework), Asynchronous JavaScript,* and *XML (AJAX).*

7. TESTING

The e-barter system has been tested against functional tests and usability tests. Functional tests were written and performed to test the system functionality - what it does. Usability tests were written and performed by users to evaluate the product. Functional tests were written and performed by the author; 19 were met and 4 were not met. 4 test cases were written to test the barter algorithm, all 4 tests were met. Test cases that were not met will be added to the backlog as part of things to be fixed during future development.

Usability testing was planned and executed as follows. 5 participants were selected to test the e-barter system. Each participant was asked to provide basic information such as age, gender, occupation, his/her computer skills, and familiarity with application usage and whether or not he/she has bartered before; user characterisation is seen in Table 7.1. Participants were then asked to be part of a test session. During the test session, each participant was given a task, corresponding to their level of computer skills and familiarity with application usage; tasks can be seen on Table 7.2. A task asks a user to complete at least one barter transaction. Easy and medium level tasks are accompanied by a set of instructions, helping the user to complete the task. No instructions were given to task at a difficult level. The intention is to test users based on their user group. After a test session, each participant is asked to answer 5 questions in a survey, survey questions are seen on Table 7.3. The results of the test session were collected. Survey answers were also collected.

User	Gender	Age	Occupation	Computer skills & Familiarity with applications	Experience with barter
#1	Male	30	IT Technical Lead	High	Yes
#2	Female	27	Master Graduate	Medium	Yes
#3	Male	54	Salesman	Entry	Yes
#4	Male	15	Secondary School Student	Medium	No
#5	Female	23	Software Tester	High	Yes

Table 7.1: Participant characterisation

Task	Difficulty level
You are given a calendar an a guitar. Your task is to exchange the calendar for a pen and the guitar for a smartphone. Register and login as a new user and perform barter. When you see your transactions marked as completed, your task has been successfully completed. No instructions are given.	**High level**
You are given a calendar an a guitar. Your task is to exchange the calendar for a pen and the guitar for a smartphone. Register and login as a new user and perform barter. When you see your transactions marked as completed, your task has been successfully completed. Instruction: Register and login. Add two new items: calendar and guitar. Browse the marketplace, find items you want and mark them as wanted. When barter options are ready, proceed to checkout. After checkout, a transaction is created. Follow all three steps till both transaction are finished.	**Medium level**
You are given a calendar. Your task is to exchange the calendar for another item of your choice. Register and login as a new user and perform barter. When you see your transaction marked as completed, your task has been successfully completed. Instruction: Go to Login. Register as a new user. Login with the credentials. Take a picture of your given item. Go to My Items, add the calendar as a new item. Then, go to marketplace, find any items you like and mark them as wanted. Wait for your shopping bag to fill with barter options. Then, choose one option and click checkout. You are now redirect to the transaction page. Click on the transaction a go through all 3 steps. Step 1 – you must wait for the opponent to checkout the barter option you just checked out. Step 2 proceed to the Handover Item form and confirm your handover of the item. Wait for the opponent to do the same. Step 3 – Rate the item (optional). Your transaction should be marked as completed now.	**Entry level**

Table 7.2: Test scenarios

Id	Question	Answer Options
1	How easy or difficult was it to complete he given task?	A. Easy B. Medium C. Difficult
2	How would you describe your experience with the application while completing the given tasks?	A. Unpleasant experience B. Good experience C. Excellent experience
3	Compare the efficiency of barter using this e-barter system against the previous method you have used to barter.	A. Using e-barter system is more efficient B. My previous method of barter was more efficient
4	Describe your previous experience with barter.	No predefined options
5	How do you compare your previous experience with barter against the experience with the e-barter system?	No predefined options

Table 7.3: Survey questions

The results of the test sessions reveal the following conclusions:

- All participants were able to complete a given task. (See Figure 7.4)
- Users at entry level of computer skills (CS) and application familiarity (AF) took the most time to complete their task. The higher the level of CS and AF, the shorter time at takes a user to complete a task. (See Figure 7.5)
- 60% of participants said their task was easy to complete. (See Figure 7.6)
- 60% of participants said the application user experience was excellent. The remainder said the user experience was good. No negative user experience occurred. (See Figure 7.7)
- All participants (80%) said the e-barter system is more efficient for barter than their previous method of barter, one participant (20%) did not give answer as they have never bartered before. (See Figure 7.8)

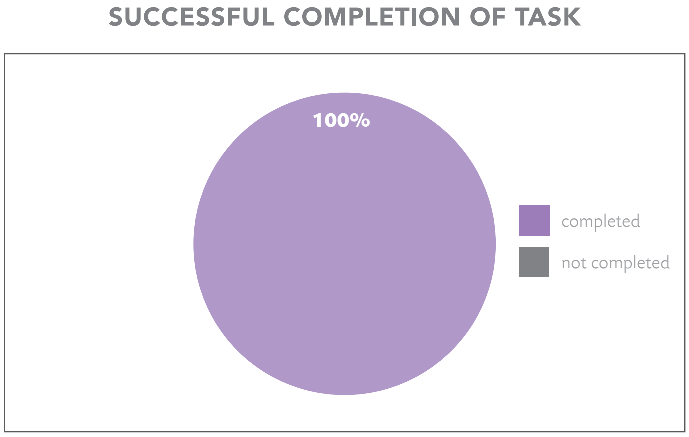


Figure 7.4: Successful completion of task

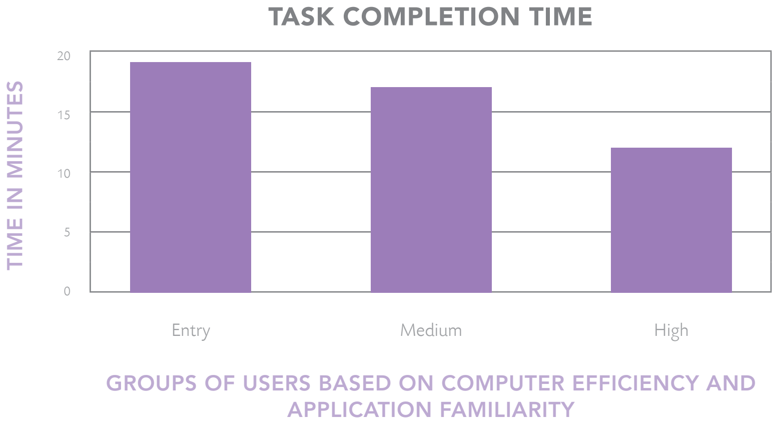


Figure 7.5: Task completion time based on different groups of users

HOW EASY OR DIFFICULT WAS IT COMPLETE THE GIVEN TASK?

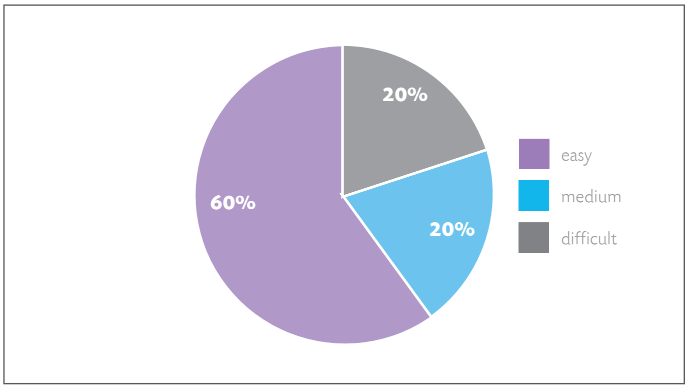


Figure 7.6: Survey question 1

HOW WOULD YOU DESCRIBE YOUR USER EXPERIENCE USING THE APPLICATION?

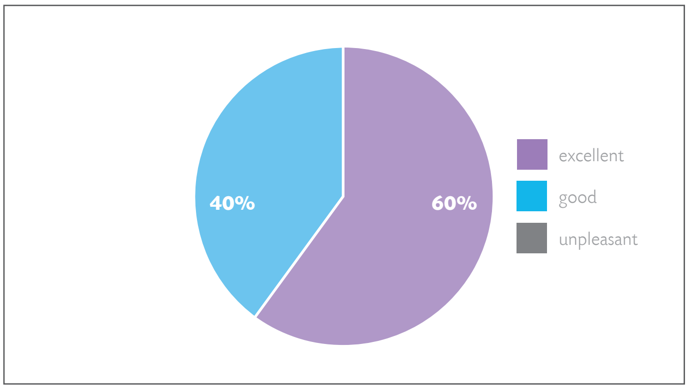


Figure 7.7: Survey question 2

COMPARE THE EFFICIENCY OF BARTER USING THIS E-BARTER SYSTEM AGAINST THE PREVIOUS METHOD YOU HAVE USED TO BARTER.

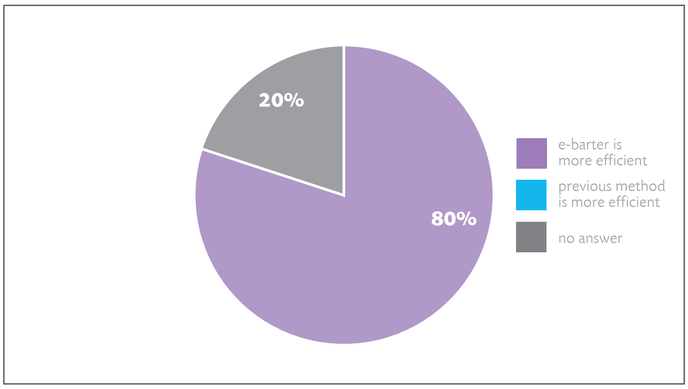


Figure 7.8: Survey question 3

8. CONCLUSION

This last chapter concludes and evaluates the whole project and the artefact produced. Possible further development plans will also be discussed.

Barter, the purest form of trade, has been always challenged by its hindrances. As money came to replace barter as the dominant infrastructure of trade, barter remained unexplored. The emergence of technology, computers, and the digital age has given us new ways of solving problems. The goal of this thesis was to explore barter, the problems of barter, and use computers to bring a solution that solves some problems of barter. It is concluded that the goal was fully met. Barter has been studied, the problems of barter identified, a solution to the problem in form of a proposed model has been delivered, and an e-barter system has been implemented. The produced e-barter system solves all three identified problems of barter: problem of coincidence of wants, problem of non-feasibility of multilateral barter, and the problem of an inefficient shopping experience. Further, all the requirements for the software has been met and the e-barter system has been tested against a functional test as well as a usability test.

The produced e-barter system serves as a proof of concept, demonstrating the feasibility of the proposed theory on how to solve inefficiencies of barter. Further development would include approaching the e-barter system from a user perspective. Research shall be conducted, examining user demands, user experience, and user satisfaction in the realm of barter. By taking user input into consideration, there is proof a concept can be further developed and turned into a usable product, ready to be published to the public. ■

REFERENCES

Smith, A. (1776) *An Inquiry Into The Nature And Causes Of The Wealth Of Nations.* London: W. Strahan and T. Cadell.

Statista.com. (2015) ' B2C E-Commerce Sales Worldwide 2018', *Statistic* [Online]. Available at: http://www.statista.com/statistics/261245/b2c-e-commerce-sales-worldwide/ (Accessed: 16 November 2015).

Dalton, G. (1982) ' Barter', *Journal of Economic Issues*, [Online]. Available at: http://www.jstor.org/stable/4225147 [Accessed 23 June 2015]

Jevons, William Stanley (1875) *Money and The Mechanism Of Exchange.* New York: D. Appleton.

COMPARISON OF BEAM-FORMING & RELAYING IN SPARSE SENSOR NETWORKS

MIKULÁŠ KREBS

This study focuses on the differences in power consumption between beam-forming and relaying data transmission methods in a sparse wireless ad-hoc network. These two methods are observed for the same parameters using an identical network topology in a simulation programme that was developed as a part of this study.

This paper introduces the reader the background of sensor networks and exposes the aims of this study and methods used to simulate these networks. Finally, the results of a simulation are analysed and the two methods are compared, followed by the conclusion of the study and the project itself.

INTRODUCTION

This paper explains and informs the reader about the background, the methods used, and the decision-making process that had to be implemented during this project. It is structured in four chapters. Each chapter focuses on a different aspect of the work while always exposing the research behind it and the decision-making process. The introduction explains the project background and then its general aims. The first chapter explains the project requirements and the way they were verified. This is followed by a chapter on the general design of the project. The third chapter goes into a more detailed description of the processes behind the simulation and the reasoning behind them followed by a chapter on testing the solution. The fourth chapter shows the results of the project including the data generated and its analysis. While the last chapter is focused on reflecting about the whole project, what possibilities there were, which were used, and what could be done with the resulting product in the future.

PROJECT BACKGROUND

"A wireless ad hoc [or sensor] network (WANET) is a decentralised type of wireless network. The network is ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data from other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity. In addition to the classic routing, ad hoc networks can use flooding for forwarding data"(Gupta and Kumar, 2000).

This project is focused on a particular type of network called an ad-hoc network. Ad-hoc and sensor networks are a particular type of wireless networks composed of nodes (or modules) which require no infrastructure and can connect without any pre-established hierarchy. Such networks are very important for environmental, civil, and military applications. For example, the sensors can measure the temperature or humidity in the soil (environmental), can measure tiny vibrations in the debris of a collapsed building, thus signalling the existence of survivors (civil), or can measure vibrations in the ground signalling a moving person in a forbidden area (military). These types of networks have huge advantages over classical wireless networks because they require no infrastructure or pre-setup, but with this comes inherent problems which have to be solved. This project focuses on a specific problem present in this type of a network.

GENERAL AIMS

There are times in which the modules cannot be placed in a specific area (e.g. because there is a lake or enemy territory). In such cases, communication can be disrupted by the lack of modules in that region creating so-called holes in the network due to the modules' physical limitation of wireless transmission. An ad-hoc network with holes is called a sparse network, thus an ad-hoc network without holes and with a uniform distribution of modules are called dense networks. This problem is currently solved by relaying information over multi-hop paths around the region with no modules (Fig. 1). But this method is problematic. In fact, relaying the information along the same multi-hop path will in time deplete the power source of the relaying modules disconnecting them in the end and therefore disrupting the network. Although these modules do not actually generate any new information, they only carry information generated by modules which could not directly communicate due to the huge distance between them. An alternative method of transmitting through these holes could be theoretically used. This method used distributed phase-shift beam-forming (further DPSBF) which enables clusters (groups) of modules to tune their transmission signal in such a way that their signals perfectly overlaps at the receiver (resonate). This allows the groups of wireless transmitters to behave as a single, virtual, more powerful radio transmitter and thus cover a larger radio communication distance (Fig. 2). The goal of this project is to construct an application that will be able to simulate these two different methods of information transmission in a specifically defined ad-hoc network to compare the difference of these two methods for an identical setup.

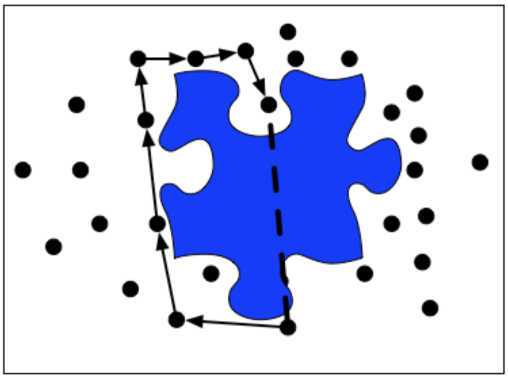


Figure 1: Relaying information over a multi-hop path around the zone with no modules (a lake for example)

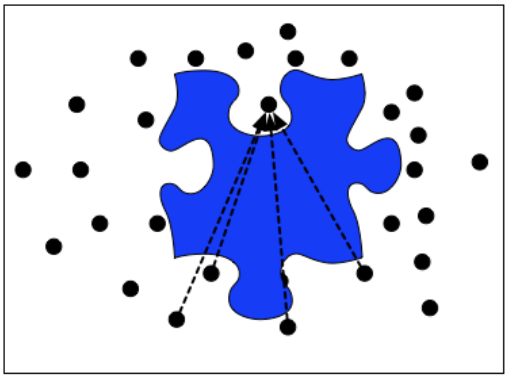


Figure 2: Modules transmit in the same time the same information using DPSBF and achieve together higher communication distance

1. GENERAL DESIGN

This section presents the general structure of the application with the inputs, the general processing structure and the outputs of the programme.

1.1. DATA INPUT AND DEFINITION

This section presents and explains the design of the simulator and the motivations behind the design. The main research in this area was about the use of graphs in representation of the network and the use of the coordinate system. The use of graph system to study a network is very natural in fact the network theory is a subset of the graph theory and is used in many scientific articles such as Newman's "The Structure and Function of Complex Networks."

The simulator application takes many input parameters, but they can all be divided in two main categories:

- Network Definition
- Module Definition

The first category – *Network Definition* – contains everything that is concerned in the topology of the network, while the second category – *Module Definition* – contains all the physical parameters for the simulation.

To define the topology of the network and therefore of the simulation, we need the location of all the nodes that will participate at the transmissions. For this purpose a simple two dimensional Cartesian coordinate system. This was done because it is a very simple and efficient way to locate objects in this system and work with them.

In this system a module/node is represented by a point which the location is expressed using two coordinates: the abscissa representing the value on the x axis and the ordinate representing the value on the y axis. The hole in the network is a square and it can therefore be sufficiently represented by only two points which define the diagonal. The user can thereby input only the top left point and the bottom right point coordinates to define the hole, and this simulation calculates the square from this simple input, using simple geometry. The simulator uses the graph naming convention; in graph theory, vertex and node are synonymous and can be considered as equivalent in this project. The last topological parameter is the starting and the ending node. These are the nodes where the information transmission respectively starts and ends. This means that the first module in the transmission path is the starting node and the last receiving node is the ending node.

To define the module specification means that we need several physical parameters describing this module. These parameters are as follows:

- data-rate
- battery capacity
- transmitter power
- transmitter frequency
- receiver sensitivity
- antenna gain
- packet size
- voltage

The data-rate or also bit rate is the number of bits per second. It is the speed of data transmission. The battery capacity is the number of charges that the battery contains and is expressed in mAh. The transmitter power is the maximal power that the transmitter can output in the antenna and is expressed in mW. The transmitter frequency is the frequency of the waves that are transmitted by the module and is expressed in MHz. The transmitter sensitivity is the minimal magnitude of input signal required for the signal to be recognisable by the receiver. The antenna gain is conveying how much it focuses the signal in comparison to an ideal dipole and is expressed in dBi. The packet is a data unit in networking, packet size is its size in bytes. Voltage is the potential difference of the current in the module circuits expressed in V.

These physical characteristics are expressed in units most commonly found in the networking industry. They are not equivalent, and the application therefore needs to perform several unit conversion procedures when dealing with these inputs. This is so that the user does not have to make the conversions by himself/herself.

I had to decide what programming platform to use for the development. As I knew I would need to be able to develop in a fast way and the application should preferably be available for multiple operating systems (Win and OSX), I chose Java. Java has the big advantage of great free libraries and running on the Java Virtual Machine (JVM) can be easily run on many platforms.

1.2. SIMULATION PROCESSING STRUCTURE

After the input data has been defined by the user through the UI menu, the programme can start the simulation process. This section walks the reader through the logical structure of procedures followed in the application.

I needed a simple but effective way to save the output of the simulation. In fact the output of the simulation was clearly a big amount of data as each of the networking modules needed an evolution of energy consumption over time. After considering multiple options such as xml or even a database system, I chose a simple comma separated value system (CSV), not only for its simplicity, but also for the overwhelming compatibility with every system imaginable. In fact, a CSV file is a simple text file and only by the way that the data is formatted, i.e. separated by commas over multiple lines, can be understood by programmes such as Microsoft Excel, Libre Office, and others. This is very important because it means that the output of my programme is not locked to my application, i.e. it can be understood, opened, and visualised with a standard programme. It also enables the user to evaluate the data in many more ways that would be able to do normally. In fact, this format allows the user to easily visualise the trends and make graphs out of it in a spreadsheet application of his choice.

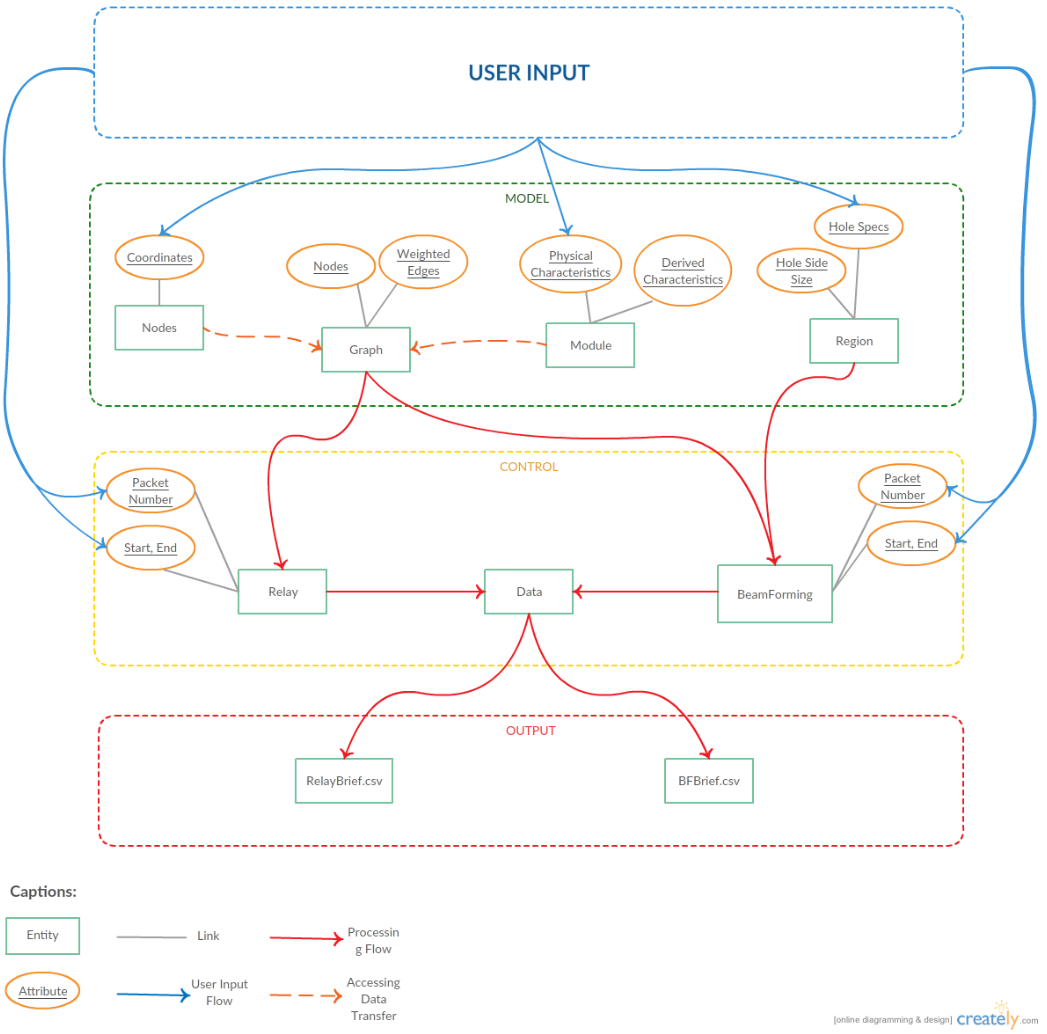


Figure 3: Simulator internal structure

The application follows loosely the structure of model/controller where the model is mostly responsible for object storage and the control performs most of the actions. The procedure starts with the module entity calculating additional module characteristics from the user input such as maximal transmission distance or the wavelength; these are then stored in this model entity for further use and reference. Then the relay entity steps in by calling on the graph entity which constructs a weighted graph that the relay method needs for its operations. The graph entity uses the information from the module and the node entities to construct this weighted graph that represents the modules in the network with the connections representing the possibility of communication and their weight the distance between the respective modules. The relay entity uses the graph as a basis for the relay method of information transmission in the network.

The beam-forming entity has a very similar function to the relay entity, but instead of the relay method it takes care of the beam-forming method. To do this, it also uses the same weighted graph constructed by the graph entity. In addition, it needs the information from the region entity which tells it the location of the hole/lake in the network. The two entities relay and beam-forming construct and fill the power consumption tables and send them to the data entity which saves them respectively as RelayBrief.csv and BFBrief.csv in the working directory.

1.3. RELAY DESIGN

This section provides more concrete information on the inner workings of the relaying method. I also needed to deepen my knowledge of graph theory which was really important for the path-finding in the network. Then I needed to research Java libraries that are focused on this field. I found multiple of them including Jung, JgraphT, and JgraphX. I chose JgraphT as it is focused on the processing of graphs and contains a lot of useful algorithms such as a collection of algorithms used to find the shortest possible path in a graph. This meant that I would be able to use the library to implement the path-finding in the network.

I had to research the possible algorithms used to find the path as each excels in a different aspect. I started my search on Wikipedia as it gives a nice and concise overview continuing by searching the information about the specific algorithms and their implementation in Java on various programming forums. I also looked at the site and knowledge repository of Cisco, a major manufacturer of networking technologies, to find what algorithms are they using in their hardware. Finally, I chose the Djikstras's algorithm as it is used in real networking protocols such as OSPF (Cisco, 2003).

The relay method has a starting node and an ending node. There is a set amount of packets that need to be transported from the starting note to the end node. In the relay method this is done by relaying the packets from node to node in a specific manner. The path for the transport is found by a particular form of the breadth-first searching algorithm called Dijkstra's algorithm. This algorithm searches the graph and finds the shortest path between two nodes. In the search for the shortest path, only the number of transmissions is taken into account. This is an algorithm that is really used in networking technologies such as the RIP networking protocol. The data is then sent packet per packet through the path each time calculating the power consumption for pairs of modules participating at the transmission in the proper path step. This means that the end node is only receiving.

> ***The modules in this project use a single transmission range this means that a node cannot transmit and receive at the same time. This is why the transmission time is basically halved and there is always a transmission period for a module and then a receiving period.***

This means that a module first receives a packet and then it transmits the packet further. Because of this, the simulator needs to calculate the proper transmission time from the absolute time.

1.4. BEAM-FORMING DESIGN

This section provides more concrete information on the inner workings of the beam-forming method. The beam-forming method has the same starting and ending node as the relay method and the same number of packets needs to be transported from the start node to the end node. But the process is quite different.

In fact the beam-forming method can be divided into two main phases:

- setup/preparation phase
- transportation phase

The setup phase consists in finding the paths that the information will take in the network including the finding of the beam-forming pairs and the transportation phase assists in the simulation calculations and data construction.

SETUP/PREPARATION PHASE

1. Calculate the shore distance D
2. Find beam-forming pairs using combinatorial from the red zone (Fig.4)
3. Calculate the range for all pairs
4. Choose the pair that can reach farthest
5. Choose beam-forming target from the green zone (Fig.4) and closest to the end node

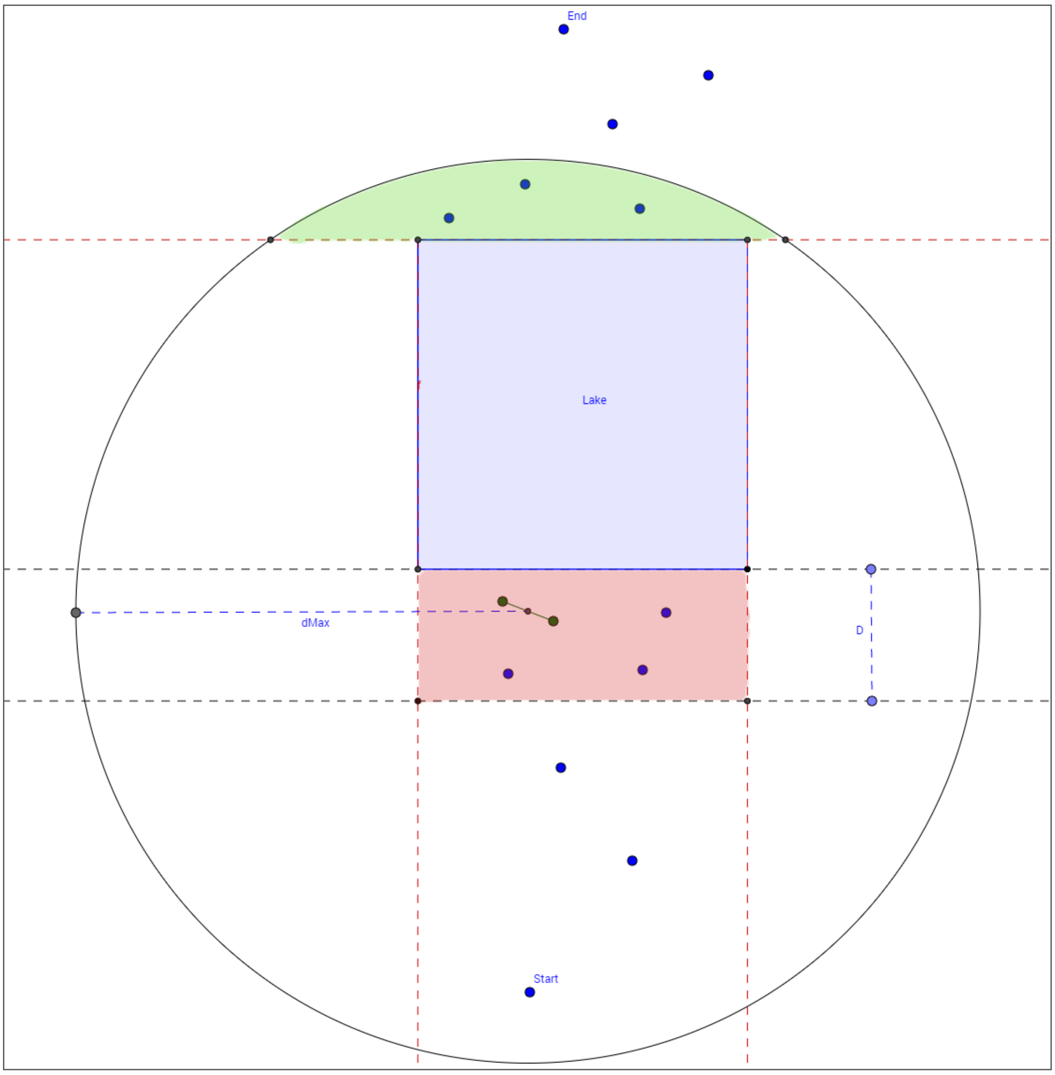


Figure 4: Beam-forming setup

The red zone is defined partly by the lake constraints and a maximal distance from shore D. The green zone is also partly defined by the constraints of the lake and by a circle with the centre as the midpoint of the chosen beam-forming pair and radius equal to the maximal transmission range of the beam-forming pair denoted $dMax$. The maximal transmission is discussed further in *2.2. Derived Equations Equation (4)*.

The distance D is calculated using the following formula:

$$D = \frac{\lambda}{4\pi} \times \sqrt{\frac{P_T}{\Delta P}} - d$$

Where D is the maximal distance from shore, the variable λ is the wavelength, d is the length of the lake, and π is a mathematical constant with the approximate value of *3.14159*. The variable P_T is the transmitter maximal power and ΔP is how much we want the second module in the beam-forming group to add to the target location in terms of power. This equation makes sure that we look for beam-forming pairs only in the zone where it makes sense. In fact, the farther from the zone we look, the less the module would matter in terms of additional power in the beam-forming transmission.

TRANSPORTATION PHASE:

The transportation phase uses the data from the setup phase. In the second phase there are in fact three path sections. The first is a traditional relay between the start node and the beam-forming group, the second stage is the actual beam-forming transmission, and finally there is another traditional relay between the beam-forming target node and the end node.

This is also reflected in the simulation, the data is first calculated for the first path then for the beam-forming and finally for the second relay path.

2. MATHEMATICAL DESIGN AND BACKGROUND

This section introduces the reader to the mathematical and physical background present in this project. It presents the equations used and explains where they come from. The physical theory used in this project is a genuine scientific endeavor as it is taken from the "Antenna Theory Analysis and Design" by Balanis (2005). The same theory for one antenna is also presented in a book by Haupt (2010). It has also been used in a number of scientific articles on the subject (Moucha, Cerny and Kubr). We can therefore see that the theory is used in practice for similar projects and is a good option to use.

2.1. INITIAL MATHEMATICAL THEORY

The initial theory used is the free-space propagation model explained in "Antenna Theory Analysis and Design" by Balanis (2005). This section presents these general physical formulas that are included in this model.

$$P_R = P_T \times G_T \times G_R \times \left(\frac{\lambda}{4\pi d}\right)^2$$

Equation (2) is a general formula to calculate the received power P_R when transmitter uses power P_T and the transmitter and receiver have an antenna with a respective gain of G_T and G_R. The variable λ is the wavelength, d is the distance between the transmitter and the receiver, and π is a mathematical constant with the approximate value of *3.14159*. This equation is in traditional units with P_R and P_T expressed in ***milliwatt (mW)***, G_T and G_R expressed as a ratio between the surface of the radiation pattern of the dipole and the surface of the antenna's radiation pattern. The variables λ and d are expressed in meters.

$$P_R = P_T + G_T + G_R + 10\log_{10}\left(\frac{\lambda}{4\pi d}\right)^2$$

Equation (3) is an equivalent equation to *equation (2)* but this time it is expressed in *decibels (dB)*, a logarithmic unit of measurement. This means that P_R and P_T are expressed in *dBm*, a decibel equivalent of power measurement. While G_T and G_R are expressed in *dBi*, a decibel measurement unit for the gain of an antenna compared with the isotropic antenna/ideal dipole.

2.2. DERIVED EQUATIONS

The initial theory general formulas were not the specific equations needed for the simulator to make the calculations. Different, more specific formulas needed to be derived from these general equations. This section presents the formulas used in the simulator and the way they were derived from the general formulas.

Equation (4) is used to determine the maximal distance of transmission between two modules of the same type. This is done by the module entity (Fig. 3) and is considered as a complex module characteristic that is derived from the user inputs. A slightly modified *equation (4)* is also used to determine the maximal transmission distance of the beam-forming group. In this modified equation the P_T is changed to P_T^2 because the group is composed of two modules.

$$P_R = P_T \times G_T \times G_R \times \left(\frac{\lambda}{4\pi d}\right)^2 \Leftrightarrow \sqrt{\frac{P_R}{P_T \times G_T \times G_R}} = \frac{\lambda}{4\pi d} \Leftrightarrow \sqrt{\frac{P_R}{P_T \times G_T \times G_R}} \times 4\pi d = \lambda$$

$$\Leftrightarrow d = \frac{\lambda}{\sqrt{\frac{P_R}{P_T \times G_T \times G_R}} \times 4\pi}$$

Equation (5) is used by the relay entity and the beam-forming entity to calculate the power that is transmitted in a second between two identical modules during transmission.

The relay entity uses this equation for all steps in the communication path as every step in the method are only two modules participating in the communication. While on the other hand, the beam-forming entity uses this only to compute the data transmission from the starting node to the beam-forming coordinator and then from the beam-forming target to the ending node. This is because the transmission done by the beam-forming group is effectuated at maximum power and does not therefore need to be computed.

$$P_R = P_T + G_T + G_R + 10\log_{10}\left(\frac{\lambda}{4\pi d}\right)^2 \Leftrightarrow P_R = P_T + G_T + G_R + 20\log_{10}\left(\frac{\lambda}{4\pi d}\right)$$

$$\Leftrightarrow P_T = P_R - G_T - G_R - 20\log_{10}\left(\frac{\lambda}{4\pi d}\right)$$

Equation (6) is used to compute the power consumption of a transmission in both relay and beam-forming entities. *PC* is the power consumption expressed in *Ampere-hour (Ah)*. *PT* is the transmitted power expressed in *watts (W)*. *t* is the duration of the transmission expressed in *seconds (s)* and *V* is the potential difference of the current in the module circuits expressed in *volt (V)*.

$$PC = \frac{PT \times t}{V \times 3600}$$

3. TESTING

This section shows the principles used during the testing of the application and its inner workings as a simulator while showing examples along the way.
To test the application, a collection of data was used. A specific topology was constructed which was fed to the simulator. Meanwhile, a series of test calculations were made by hand. The testing topology contains a series of fourteen nodes arranged so that they allow for the use of both transportation methods.

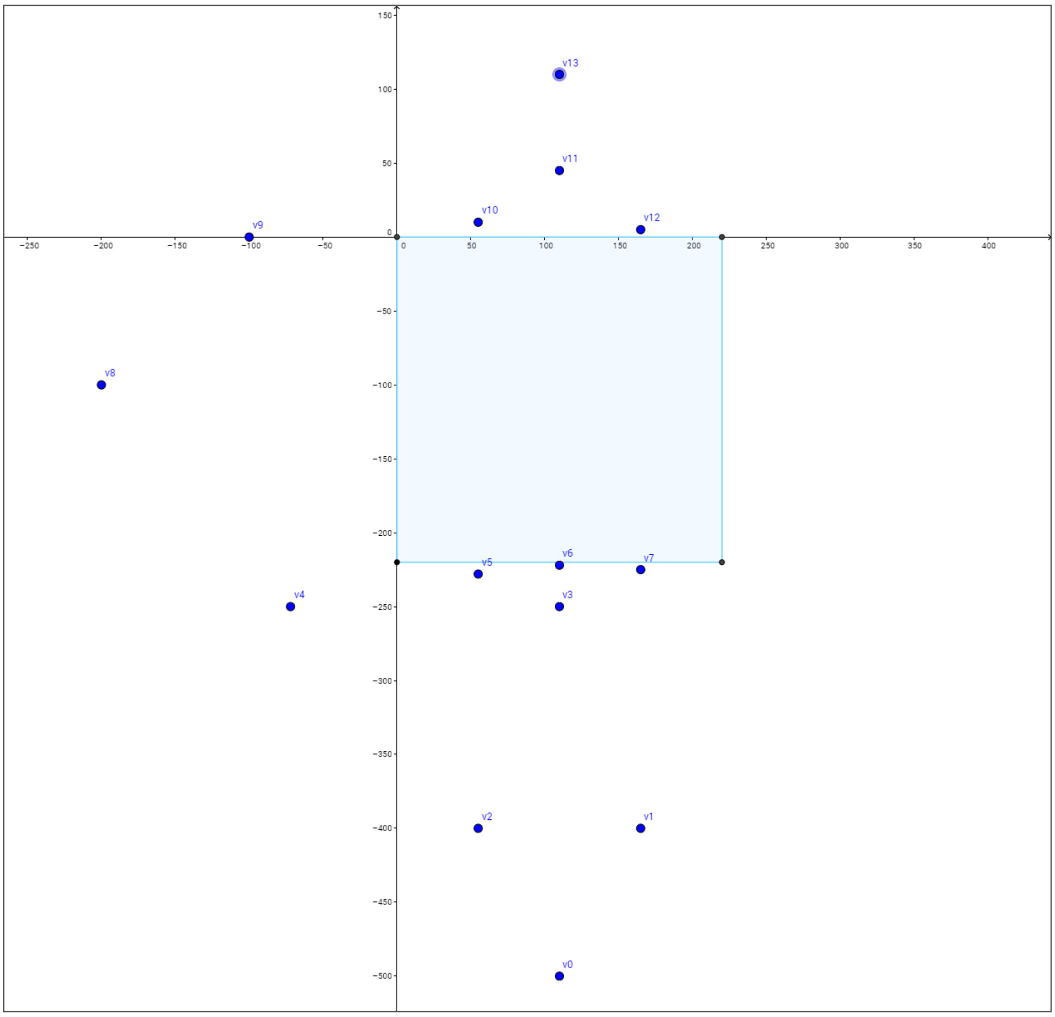


Figure 5: Testing network topology

The graph was then verified using human made calculations and visualised using Geogebra. As the equations are the same for each node couples, it is sufficient to do human made calculations for sample data and not for all of the nodes, as if we find the same results for a couple it will be consistent across the whole graph.

The input data are as follows:

$$Data\ rate = 115\,kbps, battery\ capacity = 200\,mAh, transmitter\ power = 63\,mW,$$
$$transmitter\ frequency = 2450\,MhZ, receiver\ sensitivity = -64\,dBm, antenna\ gain = 2.15\,dBi,$$
$$packet\ size = 128\,B, voltage = 5\,V$$

$$dMax = \frac{\lambda}{\sqrt{\frac{P_R}{P_T \times G_T \times G_R}} \times 4\pi}$$

As we know from *equation (4)* the *2.2. Derived equations*. P_R is the sensitivity of the transmitter equal to *-64 dBm*.

$$\lambda = \frac{c}{f} = \frac{2.998 \times 10^8}{2450 \times 10^6} = 0.1223\,(m)$$

Where c is the speed of light in vacuum and f is the frequency of the light-wave.
We need additional conversions as the units must correspond one with each other.

$$P_R = -64\,dBm = 3.981 \times 10^{-7}\,mW$$
$$G_T = G_R = 2.15\,dBi = 10^{2.15/10} = 1.6405\,(ratio)$$

This conversion needs to take place as we cannot use decibels and we must convert the gain in decibels in a ratio that does not have a unit. It becomes a constant multiplier. This was further explained in the *Section 2.1. Initial Mathematical Theory*.

This gives us

$$dMax = \frac{0.1223}{\sqrt{\frac{3.981 \times 10^{-7}}{63 \times 1.6405^2}} \times 4\pi} = 200.9\,(m)$$

while the simulator gives us a value of *200.9675…* without cutting the value short.

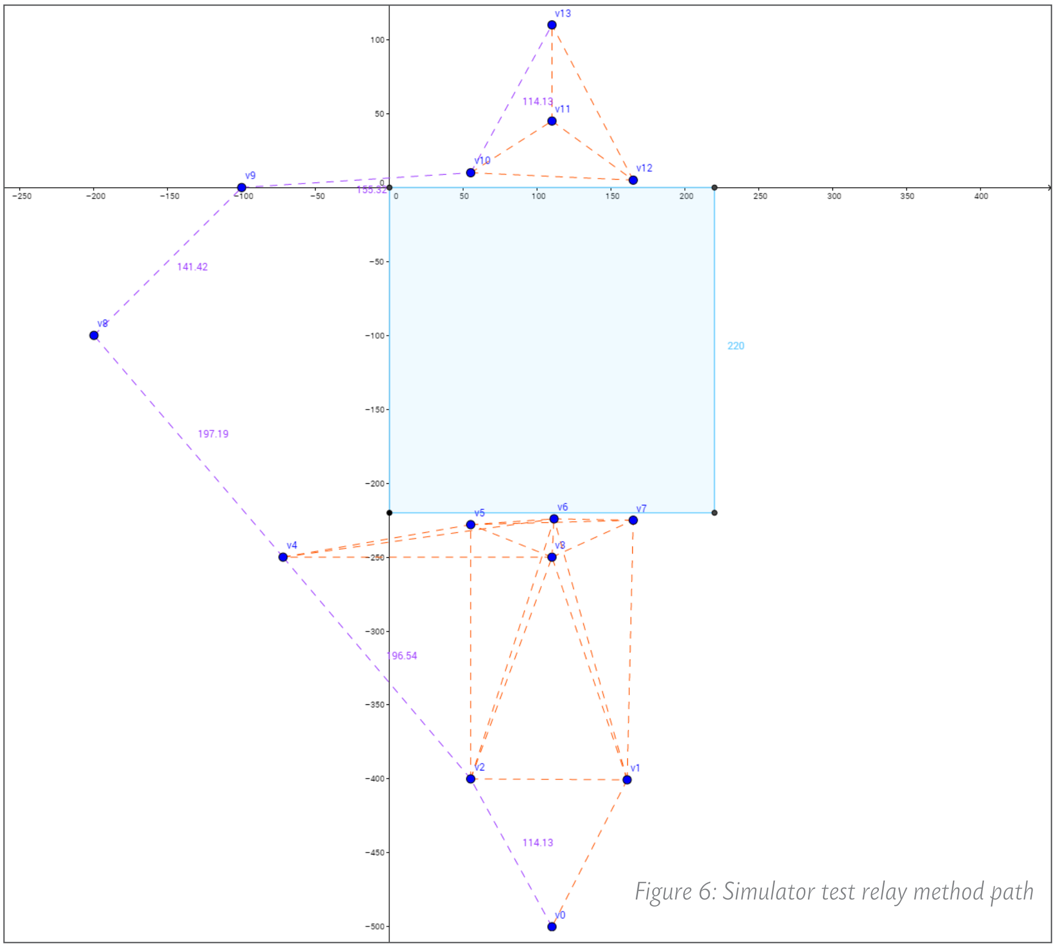


Figure 6: Simulator test relay method path

And for the

$$dBFMax = \frac{0.1223}{\sqrt{\frac{3.981\times10^{-7}}{63^2\times1.6405^2}}\times4\pi} = 284.2\,(m)$$

while the simulator gives us a value of *284.2110*.. without cutting the value short.
We can therefore see that the testing was conclusively correct.

When the graph construction was deemed correct, the testing proceeded to the next phase, the testing of the power consumption. This was done in the same spirit as the graph testing i.e. the power consumption was calculated for a pair of nodes and compared to the simulator data. In this section, the testing found a lot of inconsistency and problems. In fact, the values differed wildly from the human made calculations. After recalculating the results to know there was not a fault in the human factor, it was sure that there must be a problem in the simulation. A step-by-step debugging process began where the simulation calculation needed to be divided in the smallest steps possible to see where the inconsistency was. In fact, there was not only one inconsistency found, but several, including a missing bracket and an incorrect unit conversion.

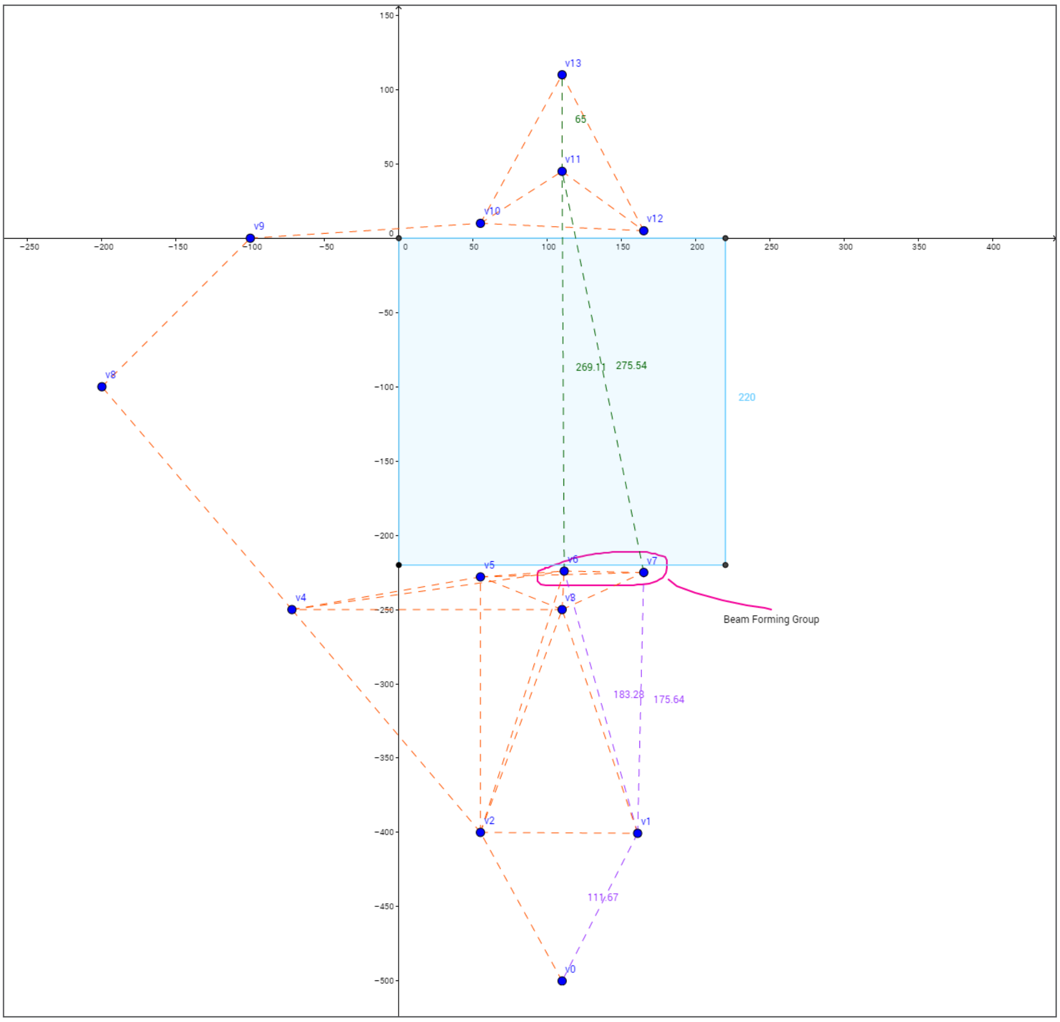


Figure 7: Simulator test beam-forming method path

In fact, it was very challenging to write the quite complex equations used in the programming language. This was because the equations needed to be written inline using only brackets. In this way even quite simple to understand equations become very unreadable by the human eye. After the debugging process was done, the power consumption testing results were compared to the newly simulated data and deemed correct.

$P_T = P_R - G_T - G_R - 20\log_{10}\left(\frac{\lambda}{4\pi d}\right)$, see *equation (5)* in *2.2.Derived Equations*.

We calculate using this, the power constant for a specific couple of nodes.

$$P_T = -64 - 2.15 - 2.15 - 20\log_{10}\left(\frac{0.12236}{4\pi 114.127}\right) = -63.8 - 20\log_{10}\left(\frac{0.12236}{1434.1621}\right) = 13.074\,(dBm)$$

$$P_T = 0.0203\,(W)$$

Now the final derived equation is to be used to calculate the battery capacity used during a transmission at a time that the simulator does the same calculation. The value of *t* was chosen *14262 seconds*, the first value in the table with real transmigration time of *6957 seconds*.

$PC\,[Ah] = \frac{PT \times t}{V \times 3600}$, see *equation (6)* in *2.2.Derived Equations*.

$$PC\,[mAh] = \frac{0.0203 \times 6957}{5 \times 3.600} = 7.845\,(mAh)$$

Now the only thing left to do is to substract this value from the battery capacity of the module.

$BC - PC = 200 - 7.845 = 192.155\,(mAh)$ which sufficiently compares to the simulator value of *192.147(mAh)* as we have a precision at the tenth of a *mAh*.

4. RESULTS AND CONCLUSIONS

There are many ways that the simulator result could be analysed in order to answer the initial question that the simulator should answer. The question remains, which method is more efficient?

One way to look at it is to look at the evolution of the energy of the whole system, not only the individual modules. Another way would be to examine the modules, see which are depleted first, and what would that means for the network; will it still be able to relay information or not?

For the purpose of our project, the most interesting analysis method is the first one that explains the network system in its entirety. This method itself can be done in multiple ways, either we can consider a positive value that would represent all the energy of all the modules in the network and see how this energy is depleted over time, the second would be to consider a negative value that would represent only the depletion of the energy over time. The second method is a better approach in the case of this work because the first method would include in the energy levels all the nodes in the network. This would mean that the energy level would include even nodes that do not participate at the transmissions. That is why the second method is much more appropriate and representative of the actual situation in the network.

To apply this method, each value level of energy of each module must be added together in the same time frame while for each we subtract the maximal battery capacity of the module. We get the following formula: $E_{loss}=(E_{N0}-BC)+(E_{N1}-BC)+\ldots+(E_{Nn}-BC)$ where E_{loss} is the overall energy loss of the whole system, E_{Nn} is the current energy level of a module with number *n*, and *BC* is the maximal battery capacity of the module. If we do this for both briefs that the simulator gives us, we get the following results.

Time [s]	v0	v2	v4	v8	v9	v10	Overall energy use
0	200	200	200	200	200	200	
14262	192.14727	176.7107	176.557003	187.9421013	185.45516	192.1473035	-89.0404167169
28524	184.29455	153.4215	153.113977	175.8841718	170.91028	184.2945769	-178.0809914962
42786	176.44182	130.1322	129.67095	163.8262423	156.3654	176.4418503	-267.1215662755
57048	168.58909	106.8429	106.227924	151.7683128	141.82053	168.5891237	-356.1621410548
71310	160.73637	83.55359	82.7848971	139.7103832	127.27565	160.7363971	-445.2027158342
85572	152.88364	60.2643	59.3418705	127.6524537	112.73077	152.8836806	-534.2432805628
99834	145.03092	36.97501	35.8988739	115.5945242	98.185913	145.0309439	-623.2838067215
114096	137.1782	13.68573	12.4558474	103.5365947	83.641036	137.1782173	-712.3243815008
128358	129.32547	-9.603562	-10.9871792	91.47866515	69.096158	129.3254907	-801.3649562801
142620	121.47274	-32.89285	-34.4302058	79.42073562	54.551281	121.4727641	-890.4055310594

Figure 8: Relay Brief Overall Energy Level Analysis

These tests were made with the same parameters as the testing in the section 4. Testing of this paper. As we can see in both Figures 8 and 9, the overall energy use linearly increases.

This is predictable as the path is invariable and the conditions stay the same during the whole simulation. We can see that if we compare the energy use in the relay method in Figure 8 and the energy use of the beam-forming method in Figure 9 the overall energy use of the beam-forming system is smaller than the energy use of the relay system. This conclusively shows that the beam-forming method is more efficient in this network than the relay method.

Time [s]	v0	v1	v7/v6	v11	Overall Energy Use
0	200	200	200	200	
14262	192.1472733973	179.0740929901	175.6502416696	197.452762388	-80.0253878855
28524	184.2945467946	158.1481591971	151.3004833391	194.905524776	-160.050802554
42786	176.4418201919	137.2222254042	126.9507250087	192.358287164	-240.0762172226
57048	168.5890935892	116.2962916112	102.6009666783	189.811049552	-320.1016318912
71310	160.7363669865	95.3703578182	78.2512083478	187.2638119399	-400.1270465597
85572	152.8836403838	74.4444240253	53.9014500174	184.7165743279	-480.1524612283
99834	145.0309238317	53.5184902323	29.5517228522	182.1693399761	-560.1778002555
114096	137.178197229	32.5925564393	5.2019645217	179.6221023641	-640.203214924
128358	129.3254706263	11.6666226464	-19.1477938087	177.0748647521	-720.2286295926
142620	121.4727440236	-9.2593111466	-43.4975521391	174.5276271401	-800.2540442611

Figure 9: BF Brief Overall Energy Level Analysis

It is worth mentioning though that the difference is significant but not enormous. We can also use look at the evolution of energy of the individual modules if we use a chart to visualise the data from the relay brief and the beam-forming brief.

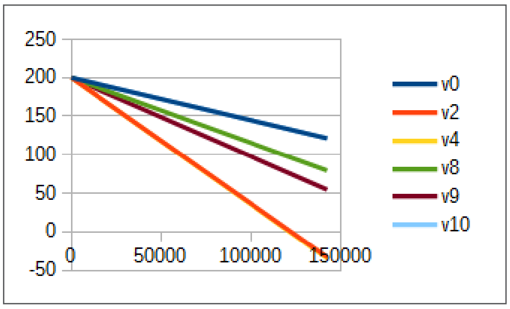


Figure 10: Relay node comparison

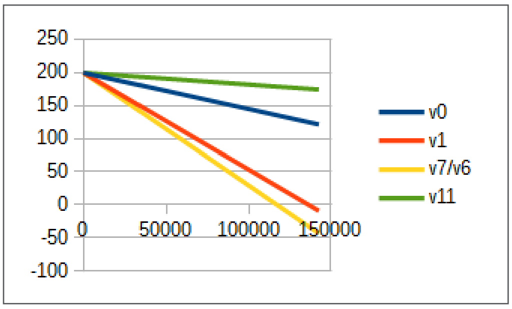


Figure 11: Beam-forming node comparison

As we know from the overall energy loss analysis, the beam-forming method is more efficient than the relay method, but if we look at Figures 10 and 11 we can see that the beam-forming group constituted from node v7 and v6 do most of the "heavy lifting" which means that even though they save quite a lot of energy in the system, these nodes in particular are depleted faster than any nodes in the relay method. This fact results in a sooner severing of the connection through this path in the beam-forming method use. This means that the use of beam-forming is definitely a useful and more efficient tool but should be used in particular situations and not as a general rule of thumb.

In further inquiries there is the potential to examine the situation of beam-forming vs. relaying for bigger beam-forming groups. In fact, using these bigger beam-forming groups would probably not change the overall efficiency but could push the severing of the network beyond the relay method as there would be more nodes participating at the same beam-forming transmission. But with this comes more inherent problems such as interference of the beam-forming group with the rest of the network, etc. which would be very difficult to predict without real experimental testing. ■

REFERENCES

Cisco. (2003) *OSPF Incremental SPF.* [Online]. Available at: http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/ospfispf.html (Accessed 17 November 2015).

Constantine A. Balanis, John Wiley and Sons. (2005) *Antenna Theory Analysis and Design.* 3rd edn., pp. 962-977.

Gupta, P. and Kumar, P. (2000) ' The capacity of wireless networks', *IEEE Transactions on Information Theory*, 46(2), pp. 388-404.

Haupt, Randy L. (2010) *Antenna Arrays*. Hoboken, N.J.: Wiley-IEEE Press, pp. 17-19.

Moucha, A., Cerny, V. and J Kubr, J. (2012) ' Distributed Phase-Shift Beamformed Power Balancing In Ad-Hoc And Sensor Networks', *The Telecommunications Systems Journal*.

Newman, M.E.J. (no date) ' The Structure and Function Of Complex Networks', *Department of Physics*, University of Michigan.

Weisstein, Eric W. (2016) "Graph." From *MathWorld*--A Wolfram Web Resource. Available at: Mathworld.wolfram.com. (Accessed 9 January 2016).

DEVELOPING
WELL
ANALYZING
APPLICATION
INCREMENTAL
NEED
SENSE
METHODOLOGIES
SOLUTION
SOLUTIONS
USED
DOCUMENTATION
ORDERS
POTENTIAL
PLAN
CODING
EMBEDDED
ENGINEERING
PRODUCTS
TERM
INCLUDES
INVOLVED
CASE
RESEARCH
DESCRIBING
SIGNIFICANT
BEST
STAGE
ADVANTAGES
STAGES
EXTREME
EVOLVE
PROCESS
TESTING
BUSINESS
DESIGN
PRODUCT
GATHERING
CUSTOM
IMPLEMENTATION
ENSURE
APPROACH
CYCLES
USING
TYPE
CODE
SOFTWARE
INCLUDE
VARIETY
VARIOUS
TAKE
SPENT
IDEALLY
BUG-FREE
COMMON
ACTIVITIES
DEVELOPMENT
FIXING
LESS
FINAL
OFTEN
TIME
DIMENSIONS
PROBLEM
STRUCTURED
MAINTENANCE
PLANNING
BASED
WATERFALL
APPROACHES
MEET
DIFFERENT
DEVELOPED
STRUCTURE
SOURCE
LIFECYCLE
USUALLY
SPECIFIC
PLANNED
SEVERAL
WORK
BROAD
MAINTAINING
CONSISTENCY

MYETL: A JAVA SOFTWARE TOOL TO EXTRACT, TRANSFORM, & LOAD YOUR BUSINESS

MICHELE NUOVO

The project follows the development of a Java Software Tool that extracts data from Flat File (Fixed Length Record Type), CSV (Comma Separated Values), and XLS (Microsoft Excel 97-2003 Worksheet file), apply transformation to those sources, and finally load the data into the end target RDBMS. The software refers to a process known as ETL (Extract Transform and Load). Those kinds of systems are called ETL systems.

The analysis involved research on the theory behind the ETL process as well as the theory behind the various phases of the applied methodology. Also an in-depth look at the design and architecture of the software has been made. To create a complete design needed to be used for the implementation, different techniques and diagrams where used to visualise and refine ideas: UML class diagrams, System Architecture Diagrams, Physical Data Model, and Project Timeline.

The implementation of the project involved the translation of the system architecture into working software using the Extreme Programming Methodology and the Java programming language. A mapping algorithm module and design patterns have been used in the implementation phase. A transformation syntax has been defined to achieve data transformation.

The testing of the software was done in the form of a unit test. A formal test plan was prepared to ensure that the main features of the system worked as defined. An error handling code implementation has been developed to avoid an unexpected crash of the system and to communicate to the user problems or errors.

1. INTRODUCTION

The Extract Transform and Load (ETL) system is the bottom line of the data warehouse. Before those systems, the business application typically had their own database that was supporting their activities. No other systems had access to this database, so it becomes an information island (Dictionary.com, 2013). The more a business was growing, the more the information island was growly rapidly because of more departments using those applications.

When a business automated those systems, more data become available, and the analytical value of the available data was soon discovered. However, to analyse that data was very complicated due to the incompatibilities among the different systems. The infrastructure created the need of collecting, analysing, and exchanging all the data and it should provide a unified view of the enterprise data was the data warehouse (gravic.com). To manage and solve the problem of the initial load of this data into the data warehouse and keep them updated, Extract Transform and Load (ETL) utilities were developed. The purpose of those utilities is to extract the data from different sources, transform them into a common format and load them into a data warehouse (Oracle®, 11 Overview of Extraction, Transformation, and Loading).

The backbone of the data warehouse architecture is constituted by ETL processes. However, ETL is not useful only for the refreshment of data warehouses. In fact, new applications have emerged with the advent of Web 2.0. Those applications integrate data which are dynamically obtained via web-service invocations to more than one source into an integrated environment. Google Maps (http://maps.google.com/), a web mapping service application and technology provided by Google, or Yahoo Pipes (http://pipes.yahoo.com/), an interactive feed aggregator and manipulator, are two examples. Under the hood, the philosophy for their operation is`pure' ETL. Furthermore, with the evolution of the technology, interest is moving to types of data that do not necessarily follow the traditional relation format, as XML, biomedical, multimedia data, and so on (Vassiliadis and Simitsis, 2007).

Although the ETL processes are well known in the computer science field, various issues still remain open. The most important problem is the standardisation: in the market, there exists several tools that provide ETL functionality but each of these tools follows a different approach for the modelling and representation of the different steps. To create a globally accepted paradigm of thinking on this topic is an issue for the academic community (Vassiliadis and Simitsis, 2007).

The aim of this project is to build a working prototype of Java Software which allows the user to extract data from the defined sources, apply the defined transformation on those data and finally load them into a target Teradata data mart that will store the data for Business Intelligence (BI) purpose. Examples of BI tools are MicroStrategy, IBM Cognos, or Informatica which are used to produce business reports on a data mart (bi-tools.org). Various phases has been involved including research and analysis of the theory behind the ETL process, design of the System Architecture and Software Graphical Unit Interface (GUI), implementation in Java programming language of the defined design using the chosen methodology and testing of the implemented code. Finally a User and Maintenance Documentation has been created to give assistance and to describe the practice overview to the final users of the developed system.

2. METHODOLOGY

Agile software development is a group of software development methods. They are based on iterative and incremental development. In this development requirements and solutions evolve with the collaboration between developers and functional teams. The Agile Methodology defines the iterative approach, evolutionary development and delivery, and inspires flexible and rapid response to changes. The Agile Manifesto (Beck, et al., 2001) is a formal announcement of four key values and 12 principles for approaching software development in an iterative way. The chosen method for implementing the MyETL Java Software Tool is Extreme Programming (XP) in order to better address the problems of project risk and because XP is set up for small groups of programmers.

A. EXTREME PROGRAMMING (XP)

Extreme Programming is an Agile Method where the customer and the development team are highly involved between them. The customer drives the development creating user stories. A user story is a high-level definition of a requirement that contains the necessary information for the developers to estimate the effort to implement it. The development team delivers in an iteratively way the user stories through continuous programming, testing, and planning. The software is delivered very frequently, usually from 1 to 3 weeks. In Figure 1 a typical XP project flow is represented.

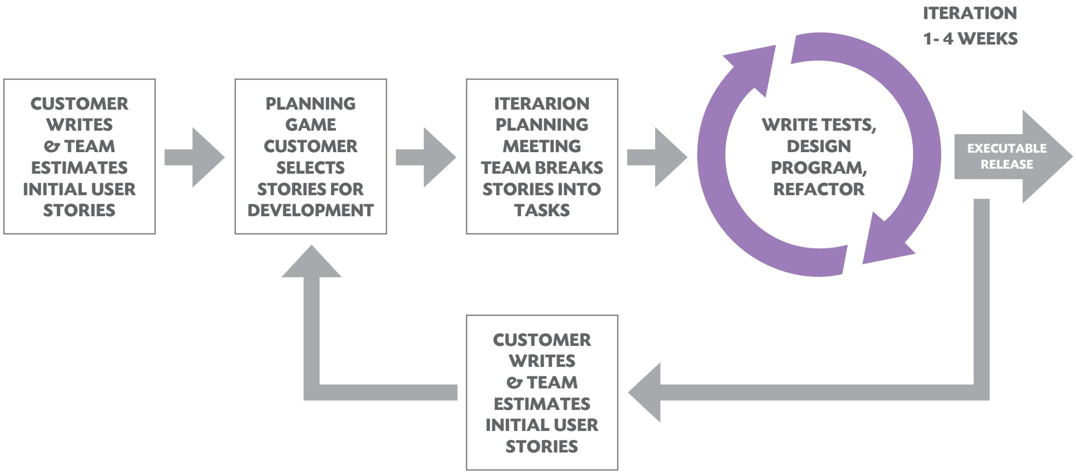

ITERATION
1- 4 WEEKS
CUSTOMER WRITES & TEAM ESTIMATES INITIAL USER STORIES
PLANNING GAME CUSTOMER SELECTS STORIES FOR DEVELOPMENT
ITERARION PLANNING MEETING TEAM BREAKS STORIES INTO TASKS
WRITE TESTS, DESIGN PROGRAM, REFACTOR
EXECUTABLE RELEASE
CUSTOMER WRITES & TEAM ESTIMATES INITIAL USER STORIES

Figure 1: The highly iterative XP project flow

XP is suited best for smaller development teams. It is a fast, aggressive delivery model. It requires high collaboration and minimal documentation. For these reasons it has been chosen for the development of the MyETL Java Software Tool. Agile methodologies are an alternative to waterfall or sequential methodology for project management.

3. SOFTWARE DESIGN

Software design is the process to understand constraints and business goals, customer needs, and technologies useful for planning a software solution and create business value. The main task of the design stage is to produce the plans necessary for software production to proceed. The figure that is responsible to produce this plan is the software designer (Budgen, 2003). To produce the MyETL software design, different communication channels where used to understand business goals. The theory behind the ETL process has been studied, and it will be part of the domain knowledge. The elaboration of these communication channels produces the System Architecture that is shown in the MyETL system architecture section.

This architecture contains the following design concepts:

- *Modularity:* The software is divided into independent components called modules. Each module has its own behaviour and purpose. Those modules are able to communicate with each other when the different application layer of the architecture needs to exchange information between them.
- *Abstraction:* Each module reduces the information content in order to keep in possession only information that is appropriate for its purpose.
- *Data Structure:* A logical relation between individual modules and data.
- *Information Hiding:* Modules are designed in a way that the information inside them is not accessible to other modules that do not need them (Pressman, 2009).

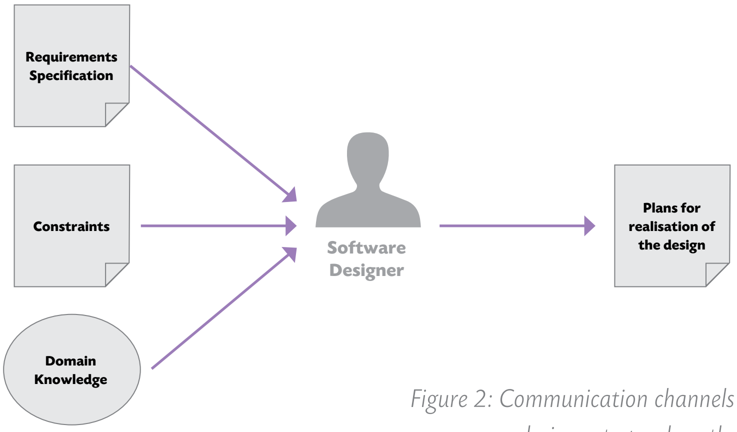

Requirements Specification
Constraints
Domain Knowledge
Software Designer
Plans for realisation of the design

Figure 2: Communication channels for the software designer to produce the software design

A. MYETL SYSTEM ARCHITECTURE

MyETL System Architecture is composed of 3 Layers: Presentation Layer, Application Layer, and Database Layer. The purpose of those layers is to separate how the information is represented from the user interaction with it.

Using the architecture in Figure 3, the three layers can communicate in both directions using specific modules to retrieve data from the database layer, apply the business rules/transformation on the extracted data, and finally display the data in a GUI representation. Once the data is displayed, the user can easily manipulate and store them again, after modifications are applied, in the database layer.

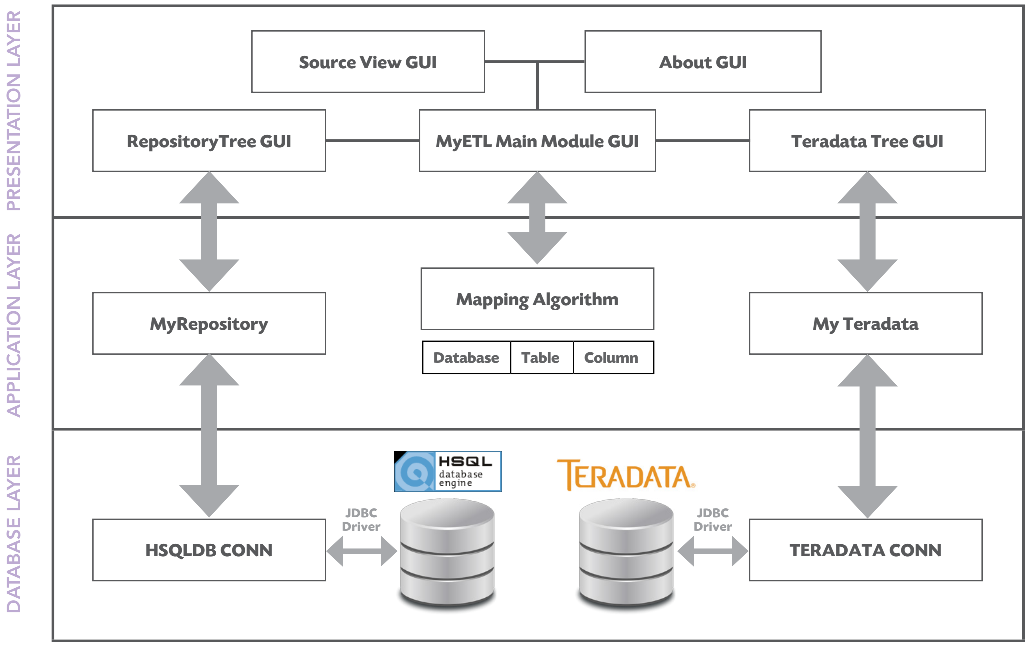


Figure 3: MyETL system architecture diagram

User interface logic changes more frequently than business logic. If a new user interface page or layout needs to be added or changed, with the architecture proposed, redistribution of the application is not necessary. Using another type of architecture, for example one in which Presentation code and business logic are embedded in a single object, will from one side decrease the code lines number, but on the other side, each time is needed to change how the data will be displayed, and the developers need to change the logic behind the modules and test it again. This operation can require a lot of effort and separate the layers that will be more efficient and less time consuming for the development of new enhancements.

Next, the 3 layers of the MyETL system architecture will be discussed in more detail using a modelling language named: UML – Unified Modelling Language (Hamilton and Miles, 2006).

B. PRESENTATION LAYER

The presentation layer is where the data is presented to the final users. This layer is composed of Graphics Unit Interface (GUI).

The presentation layer is responsible for delivering and formatting information from the application layer for further processing or display. It contains the main graphic interface of the MyETL software, which manages all the graphics units available and is accessible for the final user.

C. APPLICATION LAYER

The application layer is where the business rules are applied to the data. This layer is composed of modules that have the purpose to elaborate or transform the data retrieved from the database layer.

The Application Layer is the core of the software. Most of its functions are visible to the user through the presentation layer. For example, when the available list of tables in the repository are shown in the repository tree interface, the data is retrieved from the database layer and formatted in order to be visible adapted to fill a tree graphical representation.

D. DATABASE LAYER

The database layer is where the data is stored. This layer is composed of modules that are responsible for database connections and makes data available and reachable for the application layer.

E. DESIGN PATTERNS

The layers previously reported are designed using design patterns. A design pattern is a documented best practice or core of a solution that has been applied successfully in multiple environments to solve a problem that recurs in a specific set of situations. It can be seen as an encapsulation of a reusable solution that has been applied successfully to solve a common design problem.

The different modules in the 3 layers are designed to respond to the following design patterns:

- *Private Methods:* The purpose of this design is to provide a way of designing module behaviour so that external modules are not permitted to access to the data/operations that is meant only for the internal use. This design is used in the presentation layer to initialise the GUI components and elements.
- *Accessor Methods:* The purpose of this design is to provide a way of accessing an object's state using specific methods. This design facilitates information hiding and the module result to be more maintainable. This design is used in the application layer to provide consistent data to the presentation layer.
- *Singleton:* The purpose of this design is to provide one and only one instance of a given module during the lifetime of an application. This design is used in the database layer to be sure to have just one connection object in the application for each database.

4. IMPLEMENTATION

Most of the effort in the implementation phase went into the application layer, which is considered the core of the software. Also the presentation layer takes a lot of effort to make visible to the user all the operations behind the process and make them easily manageable through the graphic user interface.

The software was implemented using Java programme language and Eclipse IDE for Java Developers version Juno Service Release 1. There were certain criteria that were considered before to finalise this decision. First of all the language should be easy and fast to develop. In fact, the relatively short project time requires a faster development. Second, it needs to have graphical capability, because a graphic unit interface is required for the user to manage and interact with the software. Furthermore, it has to be object oriented and portable. Object oriented because of the reusability of the objects inside the programme, and portable in order to be used independently from the platform by the maximum number of users.

A. USER STORIES

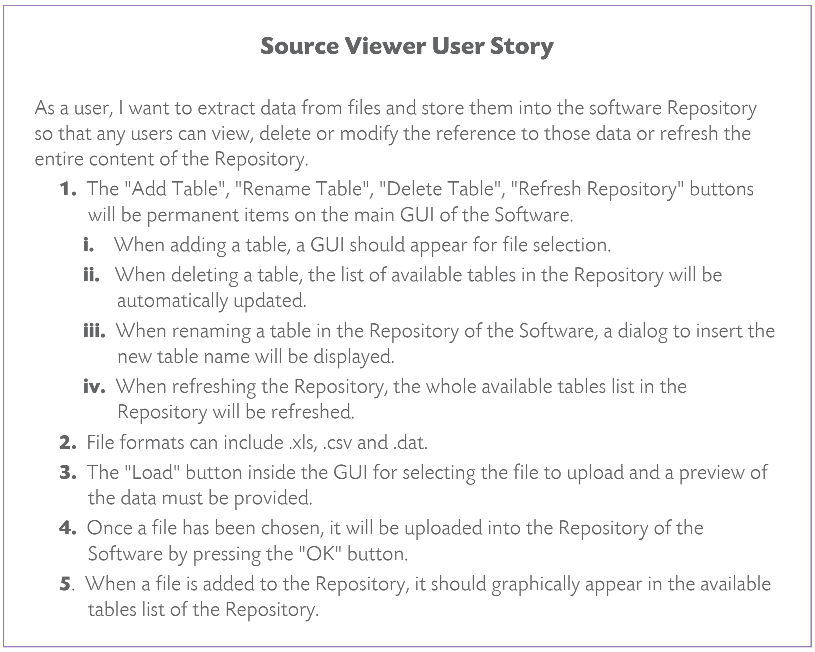
A user story captures the ' who, what, and why' of a requirement. It is used instead of a large requirements document and they are written by the customer.

A user story is in the format of about three sentences of text in a customer terminology, which means without technical-syntax. It also drives the creation of the acceptance tests that are used to verify if the user story has been correctly implemented. The difference between the user story and the traditional requirements specification are mainly two:

- *The level of detail:* The user story should only provide detail to make a sensibly low risk estimation of how long the story needs to be implemented. When the story will be implemented, developers will go to the customer and receive a detailed description of the requirements face to face.
- *Focus on user needs:* Specific technology, algorithms, and data layout should be avoided. The story has to be focused on user needs and benefits instead of specific GUI layouts.

Each story is estimated by the developers to determine how long it will need for the implementation. Usually a story will get 1, 2, or 3 weeks in an ' Ideal Development Time'. With the phrase ' Ideal Development Time', it is meant that in the development there will be no distractions, no other assignments, and the developer know exactly what to do.

When the user story estimation is up to 3 weeks, it means that the story needs to be broken into multiple stories, while when a story is under 1 week it means that it has to be integrated in another user story. To have a good release plan there should be around 80 stories with a margin of 20 stories plus or less (ExtremeProgramming.org, User Stories). The source viewer module is responsible for the selection and preview of the data inside the sources files. The estimation for the user story in Figure 4 is 3 weeks and it involves one developer.



Source Viewer User Story

As a user, I want to extract data from files and store them into the software Repository so that any users can view, delete or modify the reference to those data or refresh the entire content of the Repository.

1. The "Add Table", "Rename Table", "Delete Table", "Refresh Repository" buttons will be permanent items on the main GUI of the Software.
   - **i.** When adding a table, a GUI should appear for file selection.
   - **ii.** When deleting a table, the list of available tables in the Repository will be automatically updated.
   - **iii.** When renaming a table in the Repository of the Software, a dialog to insert the new table name will be displayed.
   - **iv.** When refreshing the Repository, the whole available tables list in the Repository will be refreshed.
2. File formats can include .xls, .csv and .dat.
3. The "Load" button inside the GUI for selecting the file to upload and a preview of the data must be provided.
4. Once a file has been chosen, it will be uploaded into the Repository of the Software by pressing the "OK" button.
5. When a file is added to the Repository, it should graphically appear in the available tables list of the Repository.

Figure 4: Source viewer user story

B. MAPPING ALGORITHM MODULE

In the application layer is an implemented mapping algorithm module. This module is responsible for auto mapping columns between the source and the target tables. The auto mapping is based on the similarity of the two strings. The similarity of two strings is defined as the minimum number of single-character edits required to change one word into the other. For the auto mapping module the Levenshtein distance algorithm has been implemented.

C. LEVENSHTEIN DISTANCE

The Levenshtein distance algorithm gives high quality string matching. The algorithm is also referred as edit distance algorithm. It calculates the minimum number of changes that are necessary to modify one given string in another given string. The way used to calculate this changes is a matrix with the size (L1+1) x (L2+1), where L1 and L2 are the length of the first and second given string. The matrix is filled from the upper left to the lower right and each horizontally or vertically jump corresponds to a change. The result number in the lower right corner is the Levenshtein distance between the given strings.

In Figure 5 is reported the entire matrix calculation for the comparison between the following strings: "meilenstein" and "levenshtein".

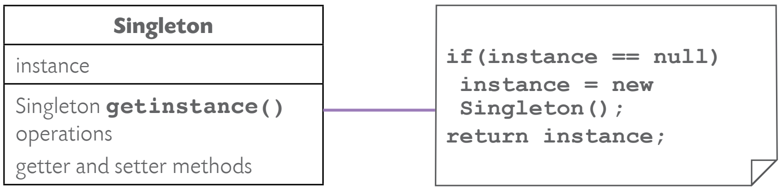
		M	E	I	L	E	N	S	T	E	I	N
	0	1	2	3	4	5	6	7	8	9	10	11
L	1	1	2	3	3	4	5	6	7	8	9	10
E	2	2	1	2	3	3	4	5	6	7	8	9
V	3	3	2	2	3	4	4	5	6	7	8	9
E	4	4	3	3	3	3	4	5	6	6	7	8
N	5	5	4	4	4	4	3	4	5	6	7	7
S	6	6	5	5	5	5	4	3	4	5	6	7
H	7	7	6	6	6	6	5	4	4	5	6	7
T	8	8	7	7	7	7	6	5	4	5	6	7
E	9	9	8	8	8	7	7	6	5	4	5	6
I	10	10	9	8	9	8	8	7	6	5	4	5
N	11	11	10	9	9	9	8	8	7	6	5	4

Figure 5: An example of how the algorithm works in the comparison of "meilenstein" and "levenshtein" made with an Excel file

The similarity between the given strings is 4, in fact 4 changes need to be applied to let the two strings be the same (Carsten). Different design patters were used during the software design.

D. JAVA SINGLETON

When it is necessary to have exactly one instance of a class and this instance is required to be accessed from different points from different classes, the Singleton design pattern is used.



```
if(instance == null)
 instance = new
 Singleton();
return instance;
```

Figure 6: Singleton class diagram

A Singleton class maintains a private static reference to themselves and returns this reference from a static instance () method. The singleton instance is created just when the getInstance() method is called for the first time. In this way it is ensured that the instance is created just when it is needed.

The following classes in the MyETL software have been implemented using the Singleton pattern:

- *HSQLDB_Connection:* It is responsible for providing one and only one connection for the software to the HSQLDB Repository.
- *TERADATA_Connection:* It is responsible for providing one and only one connection for the software to the target Teradata RDBMS.
- *MyIcons:* It is responsible for providing one and only one access to the software's icons. This class instantiate all the used icons for buttons, background image, and so on (Geary, 2003).

5. EVALUATION

A. LIMITATION

The MyETL Java software presents some limitations. The first limitation is related to the connection to the target RDBMS. A login form to input the connection parameters is not implemented. This limits the user to connect only to the embedded Teradata schema. Another limitation is related to the repository connection. The repository is embedded in the software and only one user exists. In the case of multiple users, they will share the same repository's data. The next limitation is related to the mapping frame. Currently, it is not possible to save the mapping for the source table, and it forces the user to create the mapping every time he is using the software. Furthermore, is not possible to join between them two source tables. This limitation creates a 1 to 1 relation through the sources and the target tables. The last limitation is related to the target tables in the Teradata database. All of the target tables must be empty since a FastLoad mode has been used in order to improve loading performance time.

B. FURTHER DEVELOPMENT

On the further development of the MyETL software tool the following enhancements are planned:

- Create a login form to let the software connect to different Teradata Data warehouse. Currently, it is just possible to connect to the embedded define Teradata RDBMS data warehouse (Version 1.1).
- The possibility to eliminate duplicates rows inside the data from the software repository. A button "CLEAR SOURCE" will be added to the toolbar of the MyETL frame (Version 1.1).
- Increment the number of available sources, adding the possibility to import also Extensible Markup Language (XML) (Version 1.2).
- Manage users and different workspaces in the software repository environment. In this way different users will have their own workspace on which work on (Version 1.3).
- Have a possibility to save the created mapping frame object, where transformation and mapping column are showed (Version 1.4).
- Have a possibility to apply the mapping joining multiples source table (Version 1.5).
- Create a splash screen image that appears while the programme is loading (Version 1.6).
- Have a possibility to export the created mapping into a XLS file. (Version 1.7).
- Have a possibility to show all the data table contents and not just a sample set as currently possible (Version 1.8).
- Change the load phase in the target database using UPSERT, a combination of an UPDATE and an INSERT, instead of INSERT (Version 1.9).
- Have a possibility to connect also to the Oracle database as Target RDBMS (Version 2.0). Have a possibility to connect to MySQL database as Target RDBMS (Version 3.0).

6. CONCLUSION

It is clear therefore the importance of the Extract Transform and Load (ETL) systems in the business environment. The business applications need an infrastructure to manage all of the data among the different systems. The ETL system is the answer to this need. Researching and collect information and knowledge about the theory behind an ETL process (section III) was a challenging and interesting part. It helps me to learn how to identify trustable sources and gives me a practical example about the importance to well understand the topic before to start any implementation. The importance to follow a methodology in the software development was faster clear from the beginning of the development. The Extreme Programming (XP) methodology is used because it is fast, suited for smaller teams, and it has an aggressive delivery model. This is exactly what this project needed due to the development time and the size of the team composed only of me.

Creating the design (section IV), the test plan (section VI), and plan a roadmap to increase the capabilities of the software was an interesting challenge that made me aware about all the aspects of a software development. The Java programming language used for the implementation of the software (section V) was in my plan since the beginning of the project due to its powerful and object-oriented language. Implementing the software using Java gave me the opportunity to improve my knowledge about the Java design patterns practically while applying some of them. Defining limitations (section VII) taught me how to set a perimeter for the application's domain and how to expand it.

In conclusion, the project aim was to build a fully working prototype of Java software reflecting the ETL infrastructure system. The software can extract data from Flat File, Comma Separated Values and Microsoft Excel. String Transformation, Mathematical Transformation, Aggregation Transformation and Arithmetic Transformation can be applied to the extracted data which can be loaded into a final data mart on a relation database management system.

Due to these software features and the results of the tests, the MyETL Java Tool meets all of these goals and define a plan to extend those objectives.

A. FURTHER CONSIDERATION

Regarding personal objectives developing and integrating all the required elements to build MyETL Java software was a very involved process. A careful approach has been taken in this phase to optimise performance, integrate components, acquiring the knowledge to use all of them and facing integration problems. It may be interesting to investigate and research more in others useful components to integrate in the software and in the application of additional design pattern to improve performance in the execution of the code. ■

REFERENCES

Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., and Thomas, D. (2001) *Manifesto for Agile Software Development.* Available at: http://agilemanifesto.org/ (Accessed: 24 January 2013).

bi-tools.org. (n.d.) *Business Intelligence software tools.* Available at: http://bi-tools.org/ (Accessed: 1 May 2013).

Budgen, D. (2003) *SOFTWARE DESIGN* (second edn). Edinburgh: Pearson Education Limited.

Carsten, S. (n.d.) *The Levenshtein-Algorithm.* Available at: http://www.levenshtein.net (accessed: 15 February 2013).

Dictionary.com. (2013) *information island.* Available at: http://dictionary.reference.com/browse/information+island (Accessed: 1 May 2013).

ExtremeProgramming.org. (n.d.) *User Stories.* Available at: http://www.extremeprogramming.org/rules/userstories.html (Accessed: 13 February 2013).

Geary, D. (2003) *Simply Singleton*. Available at: http://www.javaworld.com/javaworld/jw-04-2003/jw-0425-designpatterns.html (Accessed: 15 February 2013).

gravic.com. (n.d.) *Part 3 - The History of Business Intelligence.* Available at: http://www.gravic.com/shadowbase/uses/historyofbusinessintelligence.html (Accessed: 1 May 2013).

Hamilton, K. and Miles, R. (2006) *Learning UML 2.0* O'Reilly Media.

How to write meaningful User Stories. (2010) Available at: http://www.subcide.com/articles/how-to-write-meaningful-user-stories/ (Accessed: 13 February 2013).

Oracle®. (n.d.) *11 Overview of Extraction, Transformation, and Loading.* Available at: http://docs.oracle.com/cd/B19306_01/server.102/b14223/ettover.htm (Accessed: 1 May 2013).

Pressman, R. S. (2009) *Software Engineering: A Practitioner's Approach.* McGraw-Hill.

Vassiliadis, P. and Simitsis, A. (2007) *EXTRACTION, TRANSFORMATION, AND LOADING.* Ioannina: Department of Computer Science. [Online]. Available at: http://www.cs.uoi.gr/~pvassil/downloads/ETL/SHORT_DESCR/08SpringerEncyclopedia_draft.pdf (Accessed: 1 May 2013).

OPTIMIZATION
WWW
MANAGEMENT
GOAL
CONTENT
DATA
STRATEGY
ONLINE
UPDATE
CREATE
WEBSITE
HTML
CMS
CONTENT MANAGEMENT SYSTEM

ONTOLOGY TRANSLATION: THE SEMIOTIC ENGINEERING OF CONTENT MANAGEMENT SYSTEMS

ALEJANDRO VILLAMARIN M.

The present paper proposes the application of Semiotic Engineering theory to Content Management Systems (CMS) focusing on the analysis of how the use of different ontologies can affect the user's efficiency when performing tasks in a CMS. The analysis is performed using the theoretical semiotic model Web-Semiotic Interface Design Evaluation (W-SIDE model).

The Ontology Translation concept is proposed as a practical enhancement for the W-SIDE model. The enhanced model is validated by using a proof of concept based on test cases of a fictitious scenario. The validation process is carried out modifying an open source CMS. The testing of the modified CMS if performed by a volunteer tester that fits both the Ontology Translation concept and the fictitious scenario. Finally, the results of the experiment are critically evaluated establishing limitations, conclusions, and recommendations for future studies.

1. INTRODUCTION

It is argued by Villamarin (2015) that content management systems must be seen not just as commercial products than can be traded. They should be considered as complex interactive communication tools. According to Armour (2000), software is a medium to store knowledge and its value is not in the code itself, but what it does. At the same time, De Souza et al. (2001) points out that despite all the theoretical frameworks that have been used to analyse the human factor dimension in artefacts, simple questions still remain without a satisfactory answer. For instance, how can a theoretical framework support the design of user interface languages?

In this context, the present research applies semiotic theory to the content management interface of a CMS. This is in order to answer the following question: How can a CMS be guaranteed to be "easy to use" if rather than being an ad-hoc tool it is a general purpose tool?

2. PRELIMINARIES

2.1 SEMIOTIC ENGINEERING - THE DIGITAL COMMUNICATION PROCESS

De Souza (1993) argues that computational systems have two intrinsic communicative roles: they can send and receive messages at the immediate interface level, but at the same time they can be seen as messages sent from a user interface designer to users using the computational medium. Therefore, it is argued that systems can be considered as metacommunication artefacts. Semiotic Engineering constitutes a theoretical background that helps to study the Human-Computer Interaction where systems are considered metacommunication artefacts.

The concept of semiotic engineering is used by Valtolina et al. (2012) to define a digital communication process model, which is composed of the following elements:

- Two communicants: an end-user (I1) and an interactive system designer (I2). I2 is not presented during the process of interaction between I1 and the software system involved in the communication process.
- A software system, the decoding process performed by users is carried out through interaction with the software system developed by I2 (the designer).
- A message α, which is exchanged by means of the computer hardware. Consequently, it is virtual, transient, dynamic, and multi-modal. The persistence of the message is: (a) perpetual for its internal form and (b) transient for its external form (i.e. its materialisation exists as long as a software program interprets and materialises it).

2.2 W-SIDE FRAMEWORK

The Web-Semitic Interface Design Evaluation Framework (W-SIDE Framework) is proposed by Speroni (2006) as a set of conceptual utilities for analysing the semantics of web interface elements. Its main purpose is the evaluation of the correspondence between the knowledge presupposed by web interfaces and the one owned by critical users. It is argued by Speroni (2006) that the role of interface designers is to understand what the website should talk about and how the user (based on their experience) can interact with it. The core of the W-SIDE Framework is the W-SIDE Model. This model is composed of seven concepts:

I. *Web Semiotic Unit* – It is a sign (group of strongly interrelated signs) used to compose a meaningful and functional message to the user.

II. *Conceptual Semantics* – It links the semiotic unit to pre-existing knowledge of the user regarding the topic the website talks about, i.e. to understand the sign "New Content" in a configuration page of a CMS, the user needs to have a previous idea of what a user is.

III. *Procedural Semantics* – It links the semiotic unit, including its meaning, to the interactive dialogue and the context in which it is being used. It takes into account all the different pragmatic meanings that users should correctly guess in order to interpret a sign in the way intended by the designer.

IV. *Referential Content* – It represents the real world concepts that a sign refers to. For instance, the sign "New Banner" could refer to the real world concept of a banner and its meaning.

V. *Interactive Function* – It represents how the user interacts with a web sign and the purpose of the sign. For instance, a text which changes colour when a user hovers the mouse pointer over it; it is a navigational sign that represents a link that the user can click to display additional content.

VI. *Dialogic Function* – It is the navigational-dialogic function of a sign in the context of the dialogue being performed between a user and web interface. For instance, the dialogic function of a button "Back" could refer to a content that has not been visited in the website by the user or it could refer to content that has been already visited.

VII. *Ontologies* – It is the set of concepts and skills that the user should own for understanding web semiotic units and what they want to communicate.

Speroni (2006) places the concept of ontology in the process of communication between web user and designer/website and argues the following: from the web user (receiver) perspective, an ontology is the "corpus" of knowledge that needs to be mastered in order to correctly interpret the meaning of web semiotic units. From the designer/website (sender) perspective, it can be asserted that an ontology is the "corpus" of knowledge, which is presupposed and pointed out by semiotic units.

The W-SIDE model defines the following ontologies:

- *Interlocutor/Institution Ontology (ILO)* – It refers to the knowledge related to the institution/corporation/organisation on behalf of which a website talks. It also includes the knowledge related to the generic sector to which the institution/corporation/organisation belongs.
- *Topic Ontology (TO)* – It represents the knowledge related to the particular topic/subject the interlocutor talks about.
- *Context Ontology (CO)* – It represents the knowledge that is not directly related to the website's topic but relevant for making the dialogue possible and comprehensible.
- *Website Ontology (WO)* – It represents the knowledge or conventions that are generated in the website itself.
- *Internet Ontology (IO)* – It represents all the knowledge related to concepts, skills and conventions shared among people that are familiar with web browsing in general.
- *Web Domain Ontology (WDO)* – It represents the knowledge that is shared among websites that belong to the same domain or "business sector".
- *Common Sense Ontology (CSO)* – It represents the set of concepts that belong to the common sense (common and everyday terms that users need to master in order to correctly interpret semiotic units).

Table 1 defines the levels of familiarity of users with ontologies. Speroni (2006) states that these values were obtained using empirical validation.

Level of Familiarity	Value
High	9
Medium	6
Low	3

Table 1: Level of familiarity of user profiles with ontologies

This section explains how the Ontology Translation expands the W-SIDE Model defining a process that can be applied to the content management interface of a CMS.

To formalise the Ontology Translation process, assumptions regarding user profiles and how the content is structured in the CMS are needed. Subsequently, these assumptions are used for establishing conditions when the concept is viable and for its validation.

3. SEMIOTICS AND CMS

3.1 ASSUMPTIONS

User Profiles Using the context of what a CMS is, two types of user profiles are identified: technical users and non-technical users. These two profiles are formulated as follows: technical users are in charge of the technicalities of the CMS, and they are well trained to setup and configure software. Non-technical users are in charge of managing the content that is published in a website, and they have to possess a high level of understanding of the website's topic. In order to formalise the expanded model, these two profiles need to be defined in terms of the W-SIDE Model. This formal definition lets establish the level of knowledge that each profile has regarding the ontologies that are being used in the CMS's content management interface. The formalised user profiles are defined in the following list using the evaluation scale of user familiarity with ontologies described in Table 1.

- *System Administrator* - It is the user who is in charge of the installation, configuration, security and updated of the CMS. Table 2 describes the *System Administrator* profile in terms of its familiarity with the ontologies defined in Section 2.2.

Ontology	Level of Familiarity	Value
Interlocutor	Medium	6
Topic	Low	3
Context	Medium	6
Website	High	9
Internet	High	9
Web Domain	Low	3
Common Sense	High	9

Table 2: System Administrator Profile

- *Website Content Manager* – It is the user who is in charge of maintaining, creating and updating website content. Table 3 describes the *Website Content Manager* profile in terms of its familiarity with the ontologies defined in Section 2.2.

Ontology	Level of Familiarity	Value
Interlocutor	High	9
Topic	High	9
Context	Low	3
Website	Low	3
Internet	Low	3
Web Domain	Low	3
Common Sense	High	9

Table 3: Website Content Manager Profile

Contrasting the description of both the System Administrator profile and the Website Content Manager, the following facts can be highlighted: While the System Administrator has a high level of familiarity with the ontologies that are directly related to the "world" of the Internet and CMSs, their level of familiarity with the ontologies related to the topic of the website that is being administrated by the CMS is low. (b) Website Content Manager has a high level of familiarity with the ontologies related to the topic that the website talks about. However, their level of familiarity with the ontologies related to the Internet and CMSs is low. Based on facts (a) and (b) the following can be predicted: (I) Managing the content of the website is not a trivial task neither from the perspective of the System Administrator nor from the perspective of the Website Content Manager. (II) It is unlikely that the Website Content Manager will be able to manage in a straightforward manner the website content without help/training from the System Administrator. (III) Even after training, the Website Content Manager may be subject to confusion and misunderstandings. Especially given the fact that it is highly probable that the language of the semiotic units in the CMS administration interface are incompatible with the topic that the website is talking about.

CMS Content Structure It is assumed that the CMS structures content in the following manner: A website is composed of sections. Sections may contain subsections; sections also may have content. Subsections may contain other subsections and also each subsection may have content. Subsections are recursive entities, which means a subsection of a subsection can have another subsection. Finally, content can be composed of text, images, links, forms, flash files, flash videos, JavaScript carousels, etc.

3.2 EXPANDED MODEL - ONTOLOGY TRANSLATION CONCEPT

W-SIDE Model is composed of seven concepts: semiotic unit, conceptual semantics, procedural semantics, referential content, interactive function, dialogic function, and ontology. The proposed expanded model incorporates an extra concept: "Ontology Translation".

In the context of CMS, Dirgahayu, Setiani and Zukhri (2014) argue that organisations must define *information requirements* before publishing content on their websites. The information requirements constitute the information to be published and its organisation. It must be pointed out that *information organisation* does not refer to either how web pages are linked to each other or how web pages should look like. It defines in which web page an information item should be presented (in relation with other information items) taking into account the levels of importance of all the information items that need to be published on the website.

> ***Based on the concept of information organisation and the generic nature of the structure of the CMS content structure, Villamarin (2014, p. 39) defines the Ontology Translation concept as follows: "Ontology Translation is the process of establishing an alternative web semiotic unit for signs that belong to a different ontology that the user needs to interpret."***

The Ontology Translation process requires that both the information organisation and the CMS content structure have been previously established. Ontology translation may be performed by the System Administrator with the help of the Website Content Manager; by combining their user profiles a new profile can be defined: *Ontology Translator*.

According to Villamarin (2014), the Ontology Translation process is composed of the following steps:

I. Definition of information organisation – It may be carried out by the business side of the organisation that owns the website. It may be performed with the help of the Website Content Manager. It requires a high knowledge of the business domain.

II. Definition of CMS content structure – It may be performed by the System Administrator (or a technician). It is required an advance level of expertise in web technologies.

III. Selection of key signs – It may be performed by an Ontology Translator or by the System Administrator with the help of the Website Content Manager. The System Administrator selects the signs that can be mapped from the CMS content structure to the information organisation. The Website Content Manager selects the signs that could lead to confusion when new content is added to the website. These signs mainly refer – but they are not limited to – the Website and Internet ontologies.

IV. Definition of alternative web semiotic units – It may be carried out by the Website Content Manager with the help of the System Administrator. Alternative web semiotic units are defined for the key signs previously selected (Step III); its main purpose is to help avoid communication breakdowns during application-user interaction. These signs mainly refer to the interlocutor or topic ontologies.

V. Application of alternative web semiotic units – It may be performed by the System Administrator; it consists of wrapping the key signs selected in Step II with the alternative web semiotic units from Step IV. This step may include altering the CMS's source code using appropriate programming techniques, i.e. JavaScript functions, PHP controllers, etc.

It is expected that after performing the Ontology Translation process communication breakdowns during application-user interaction will decrease. Figure 1 depicts user's interpretation of signs after ontology translation has been performed.

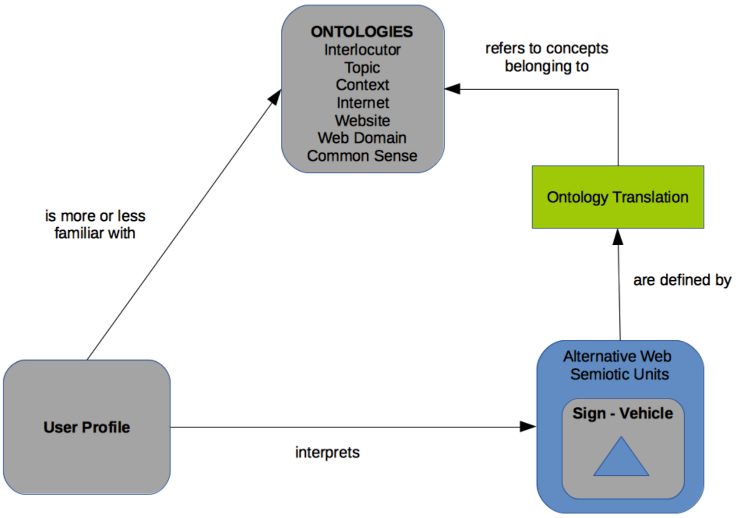


Figure 1: User sign interpretation after ontology translation

4. PROOF OF CONCEPT

An experimental approach was used to demonstrate the feasibility of the Ontology Translation concept. The website www.ferech.com, which is the personal website of a fashion designer based in Prague, was taken as a guideline to define a fictitious scenario. Using the fictitious scenario an experiment was designed. The main subject of the experiment was the Website Content Manager.

4.1 FICTITIOUS SCENARIO

As the main objective of the scenario is to manage a fictitious website that should serve as a personal catalogue for a fashion designer, both the ILO and the TO refer to the world of "fashion design".

The information organisation is designed as follows: the website is composed of *works* of the fashion designer; there are three types of works (*footwear, fashion design,* and *accessories*). *Footwear* is composed of *collections*; each *collection* is composed of *type of customers* (i.e. women, men, etc.). Each *type of customer* contains *models* and each *model* has a *description* (i.e. text, images, videos, etc.). *Fashion design* is composed of *collections*; each *collection* is composed of *types of customers* (i.e. women, men, etc.). Each *type of customer* contains *garments* and each *garment* has a *description* (i.e. text, images, videos, etc.). *Accessories* is composed of *types of accessories*. For the fictitious scenario just one type is defined: *bags*. *Bags* contains *collections*, each *collection* is composed of *types of customers* (i.e. women, men, etc.). Each *type of customer* contains *models* and each *model* has a description (i.e. text, images, videos, etc.).

The CMS content structure consists of a website that is composed of three sections, each section is composed of subsections and each subsection can be composed of more subsections. Sections and subsections may include content.

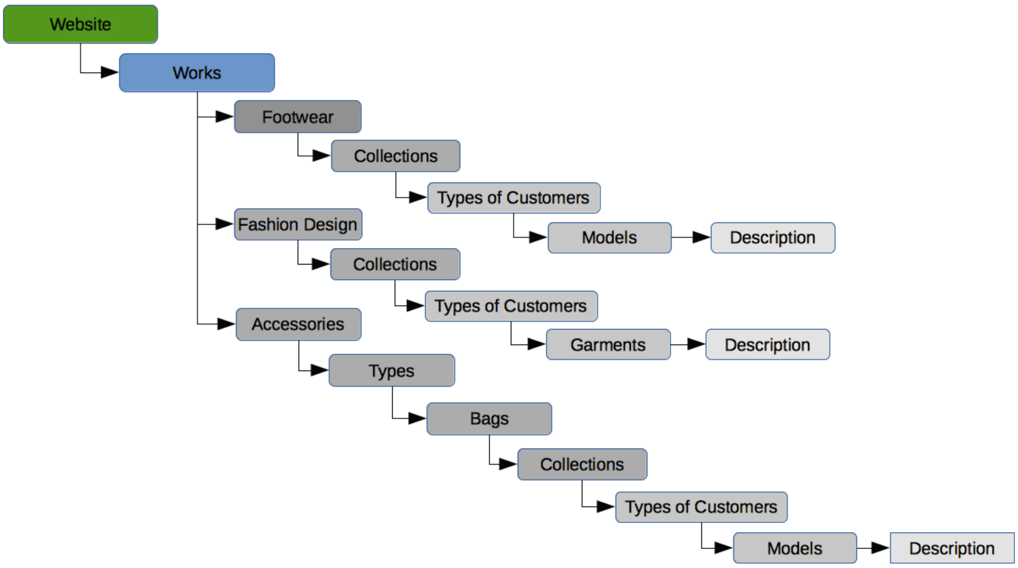


Figure 2: Information organisation fictitious scenario

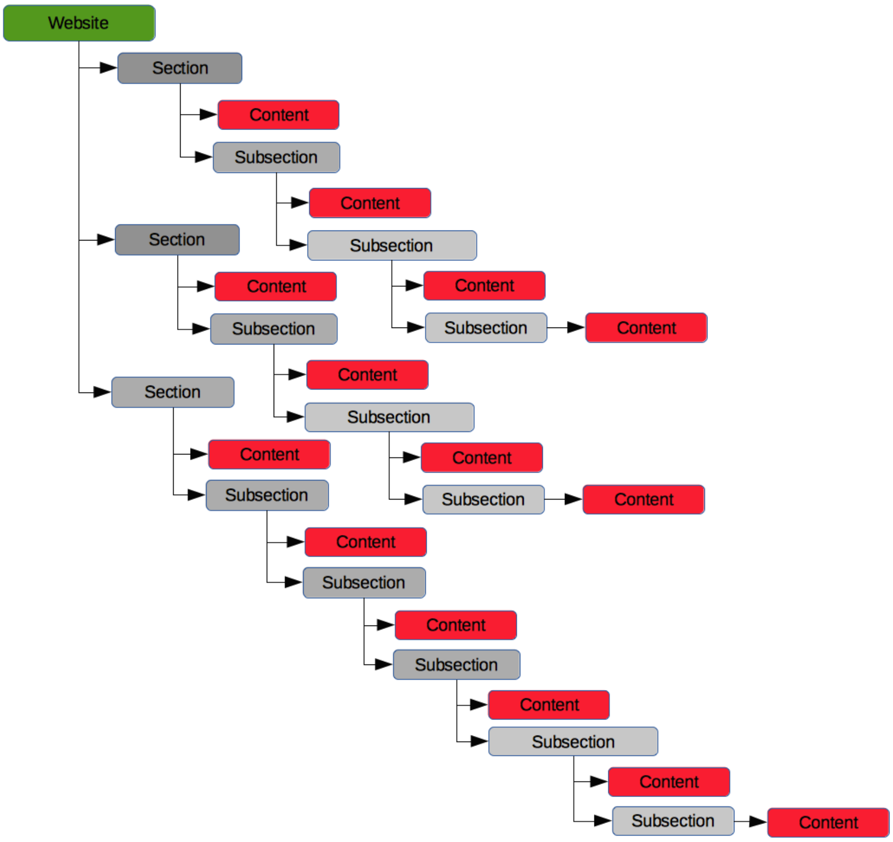


Figure 3: CMS content structure

Figure 2 and Figure 3 depict the information organisation and the CMS content structure of the fictitious scenario. By comparing both structures it is noticeable that information organisation and CMS content structure are compatible. Consequently, a mapping from *Website Ontology* and *Internet Ontology* to *Interlocutor Ontology* and *Topic Ontology* is plausible.

4.2 VALIDATION OF THE ONTOLOGY TRANSLATION CONCEPT

After defining the information organisation and the CMS content structure, steps I and II of the Ontology Translation process have been completed. The remaining steps require a working CMS in order to be performed.

Validation of the Ontology Translation concept was carried out by building a working prototype. The prototype was developed using the source code of the Open Source CMS WicaWeb available at https://github.com/mushoq/wicaweb. Mushoq (2015) points out that WicaWeb is built using *Zend Framework* – an open source framework for developing web applications using PHP programming language; being this the main reason why WicaWeb was selected from among other CMSs like Joomla or WordPress. Adding a new functionality to it, it is equivalent to create or modify a module, controller, helper, or function on a Zend Framework web application.

Top-level requirements for the prototype were established from the definition of both the Ontology Translation process and the fictitious scenario. Two top level requirements were identified: (1) Systems Administrator should be able to generate *tooltips* (alternative web semiotic units) for specific labels, buttons, and links of the CMS's content administration interface. (2) Website Content Managers should be able to visualise the *tooltips* (alternative web semiotic units) generated by the System Administrator.

The source code of the prototype is available at https://github.com/ AlexandrosV/WicaWeb. According to Mushoq (2015), the minimum software requirements to run the WicaWeb are: PHP version 5.3, Apache HTTP server 2.2, and MySql 5.5. The prototype contains a new module called *Ontology*, which fulfills top-level requirements (I) and (II). A detailed technical description of the prototype is given by Villamarin (2015) in section 3.3 of his Master's project *The Semiotic Engineering of Content Management Systems*. Figure 2 shows the *Ontology* module in the control panel of the WicaWeb CMS.

4.3 SELECTION OF KEY SIGNS & DEFINITION OF ALTERNATIVE WEB SEMIOTIC UNITS

Four generic configuration interfaces can be found in the WicaWeb's CMS module: (1) configuration interface for sections, (2) configuration interface for subsections, (3) configuration interface for contents, and (4) forms for creation of new sections, articles content and link to content.

Based on Section 4.1, Table 4 defines the selected keys and their alternative web semiotic units for the generic interface (1).

Sign	Alternative Web Semiotic Unit
New Section	New Work
Sections	Work
Section name	Work Name

Table 4: Alternative web semiotic units for generic interface (1)

Table 5 defines the selected keys and their alternative web semiotic units for the generic interface (2). Table 6 defines the selected keys and their alternative web semiotic units for the generic interface (3). Table 7 defines the selected keys and their alternative web semiotic units for the generic interface (4).

Sign	Alternative Web Semiotic Unit
New Section	New Collection
New Content	New Description
Subsections	Collection
Subsection name	Collection name
Has Subsections	Has Type of Customer

Table 5: Alternative web semiotic units for generic interface (2)

Sign	Alternative Web Semiotic Unit
New Content	New Description
Contents	Description
Content	Collection name

Table 6: Alternative web semiotic units for generic interface (3)

Sign	Alternative Web Semiotic Unit
New Section	* Same as for generic interface (2)
New Content	* Same as for generic interface (2)
New subsection of ...	* New Type of Customer (in the case of *Fashion Design* sub-section)

Table 7: Alternative web semiotic units for generic interface (4)

4.4 APPLICATION OF ALTERNATIVE WEB SEMIOTIC UNITS

It constitutes the last step the Ontology Translation process. It is performed by the System Administrator using the administration interface for the Ontology module.

4.5 TEST CASES

To evaluate the prototype as well as the Ontology Translation process the following two test cases are defined: (1) Without applying any alternative web semiotic units of the footwear section and subsections, the Website Content Manager must create a new model of shoes (Textura Kids) for a new collection (Textura Spring Summer 2015) for new type of customer (kids). A short description (text) of the new model should be added. (2) Alternative web semiotic units are applied to all the selected signs of the Fashion Design section and subsections. The Website Content Manager must create a new garment (radio) for a new collection (Volumetra Sprint Summer 2015) for a new type of customer (men). A short description (text) of the new garment should be added.

A short introduction (five minutes approximately) about the WicaWeb is given to the Website Content Manager before performing the test cases. After the test cases are carried out, feedback is gathered using a simple questionnaire. The questionnaire serves as an instrument to attempt to quantify the usefulness of applying alternative web semiotic units in the WicaWeb content's management interface.

4.6 QUESTIONNAIRE

It is composed of ten closed-ended questions. Answer are quantified using a matrix that contains a rating scale of five points (adverbs: very, slightly, neither, slightly, and very) and pairs of adjectives to measure attributes. Each question intents to quantify one of the following attributes: level of complexity (adjectives: easy and difficult), level of understanding (adjectives: clear and vague), level of assertion (adjectives: clear and confusing) and level of helpfulness (adjectives: helpful and useless). Table 8 shows the questions and its corresponding main purpose.

Question	Purpose
Estimate the level of difficulty of completing task 1.	Measure the level of complexity of adding new content to the website without using the Ontology Translation concept.
How understandable is the meaning of the table headers Section and Subsection when adding the required content?	Measure the user's understanding level of the default interface for managing content in the WicaWeb.
How understandable is the purpose of the buttons New Section and New Content afters reading their labels?	Measure the user's understanding level of the purpose of the default buttons for managing content in the WicaWeb.
When adding the description of the model of shoes, was it clear which button must be used to add the description?	Measure the user's assertion level when deciding which button is used for a specific action.
Estimate the level of difficulty of completing task 2.	Measure the level of complexity of adding new content to the website after applying the Ontology Translation concept.
How understandable is the meaning of the table headers Collection, Type of Customer, Garment, and Description when adding the required content?	Measure the user's understanding level of the interface for managing content in the WicaWeb after applying the Ontology Translation concept.
How understandable is the purpose of the buttons New Collection, New Type of Customer, New Garment, and New Description after reading their labels?	Measure the user's understanding level of the purpose of key buttons for managing content in the WicaWeb after applying the Ontology Translation concept.
When adding a description of a garment, was it clear which button must be used to add a description?	Measure the user's assertion level when deciding which button is used for a specific action after applying the Ontology Translation concept.
Estimate how helpful were the table headers Collection, Type of Customer, Garment, and Description when adding the required content.	Measure the level of helpfulness of applying the Ontology Translation process over table headers in the content management interface.
Estimate how helpful were the button's labels New Collection, New Type of Customer, New Garment, and New Description when adding the required content.	Measures the level of helpfulness of applying the Ontology Translation process over key buttons in the content management interface.

Table 8: Questionnaire

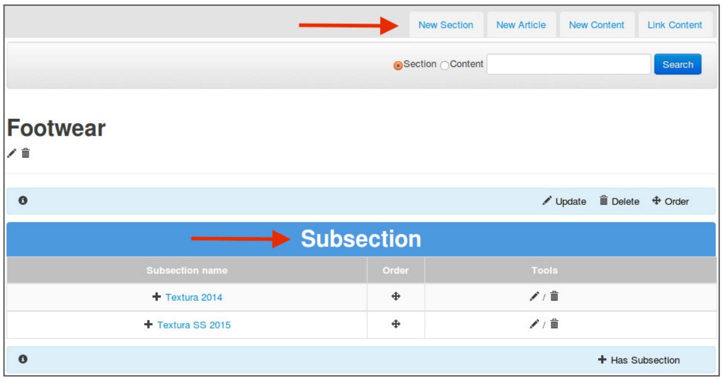
Additionally, the questionnaire includes a section titled *Notes*, *Comments*, *Remarks* that allows the Website Content Manager to express their opinion/experience regarding the two test cases.

4.7 RESULTS

In order to complete the proof of concept of the Ontology Translation process, both test cases were carried out by a fashion designer based in Prague. The fashion designer fulfils the profile of a Website Content Manager. After analysing the results of the questionnaire, the following assertions *can* be made.

For users whose level of familiarity with the Context, Website, Internet, and Web Domain Ontologies is low, managing website content may be a difficult task even if the organisation information and the CMS Content Structure have been previously defined. Table headers like: *Section* and *Subsection* in combination with button's labels like: *New Section, New Content*, etc., may cause confusion during application-user interaction.

For instance, the button's label *New Section* and the table header *Subsections* may produce communication breakdowns when adding content to the website. It is mentioned by the fashion designer that because of the table header indicates that subsections are being created, they expected to have a button called *New Subsection* to create new content. Figure 4 shows this inconsistency in the user's interface.



Subsection name	Order	Tools
+ Textura 2014	✥	✎ / 🗑
+ Textura SS 2015	✥	✎ / 🗑

Figure 4: User's interface inconsistency

Also the purpose of buttons with labels such as *New Section, New Article, New Content* may be not clear and lead to confusion when creating new content for the website. Translating signs from the Context, Website, Internet, and Web Domain Ontologies to Topic and Interlocutor Ontologies facilitates managing website content. By using specific table headers i.e. *Collection, Type of Customer, Garment,* etc. in combination with buttons labels like *New Collection, New Type of Customer, New Garment*, etc. helps to reduce communication breakdowns during application-user interaction. Overall, it can be stated that the Ontology Translation concept improves the usability of generic CMSs.

4.8 CRITICAL EVALUATION

Assumptions defined in Section 4.1 allowed the Ontology Translation process to be tested and proved. At the same time, they pose a constraint for the concept, and specifically the assumption regarding the CMS content structure. A hierarchical structure composed of sections, subsections, and content is needed in order to define which alternative web semiotic units shall be used at different levels of hierarchy in the information organisation.

> ***Feasibility of the Ontology Translation process cannot be a guarantee if applied on a different CMS content structure. A detailed analysis effort versus results is needed in order to prove the feasibility of the concept in cases where CMS structure content is handled in a different manner. It is highly probable that the complexity of designing an algorithm to carry out the translation will increase making the concept not viable.***

Due to time and logistic constraints, proof of concept of the Ontology Translation concept was limited to just one fictitious scenario. Further validation of the concept is needed in order to strictly prove its feasibility.

5. CONCLUSIONS

Based on the results of the test cases, some of Speroni (2006) can be reasserted. It was demonstrated that the more there is matching between ontologies used in a website and the one mastered by the user, the better the user's understanding and interpretation of signs contained in the interface. It can be also reasserted that to correctly evaluate web semiotic units, user interface language designers should take into account both the user's presupposed knowledge and the knowledge that is directly referred to by the semiotic unit. Not properly considering either of both can lead to misleading assessments of the web semiotic unit. It is also imperative to always keep in mind that regardless of the referential content of a semiotic unit, a sign can have interactive and dialogic functions.

The *Ontology Translation* concept applied to a CMS prototype contributes in a practical manner to the enhancement of the theoretical semiotic concept W-SIDE model. It also constitutes a practical example of how applying semiotic theory the usability of a CMS's content configuration interface can be greatly enhanced. Furthermore, the validation process of the *Ontology Translation* concept exposed an extra utility for it; alternative web semiotic units can serve as a guideline for the next action that the user should perform when managing content (this depending on how alternative web semiotic units are presented to the user i.e. marked with a different colour).

Overall, the present project constitutes a step forward to the study of how communication breakdowns may occur when using certain ontology in content management interfaces. It also contributes to the study of how to design better technical interfaces for users that lack the proper knowledge to correctly interpret its signs. Interface designers can use the prototype as a tool to evaluate the efficiency of ontologies used in content management interfaces.

Despite the fact that some important limitations were identified during the critical evaluation of the *Ontology Translation* process, these limitations should be considered as opportunities for future studies. The main limitations of this project were the size of the survey sampling and the evaluation of the social and cultural implications related to the users' origin. Also, it must be pointed out that the present project did not take into account any aesthetic aspect of the interfaces. ■

REFERENCES

Armour, P.G. (2000) 'The Business of Software: The Case for a New Business Model', *Commun.ACM*, 43(8), pp. 19-22.

De Souza, C.S. (1993) 'The semiotic engineering of user interface languages', *International Journal of Man-Machine Studies*, 39(5), pp. 753-773.

De Souza, C.S., Barbosa, S.D.J. and Prates, R.O. (2001) 'A semiotic engineering approach to user interface design', *Knowledge-Based Systems,* 14(8), pp. 461-465.

Dirgahayu, T., Setiani, N. and Zukhri, Z. (2014) 'Information requirements engineering for specific content management systems', *Open Systems (ICOS)*, 2014 IEEE Conference on. Oct., pp. 54-59.

Mushoq. (2015) *Descargas* Available at: http://www.wicaweb.com (Accessed: 11 March 2015).

Speroni, M. (2006) *Mastering the Semiotics of Information-Intensive Web Interfaces*. University of Lugano.

Valtolina, S., Barricelli, B.R. and Dittrich, Y. (2012) 'Participatory knowledge- management design: A semiotic approach', *Journal of Visual Languages & Computing*, 23(2), pp. 103.

Villamarin, M.F.A. (2015) *The Semiotic Engineering of Content Management Systems.* MSc. Teesside University.

EDITORIAL IMAGES

PAGE 3 & PAGE 15: ar130405/123RF.COM (2015) *Stock Photo - Barter concept image with business icons and copyspace* [Online]. Available at: http://www.123rf.com/photo_44898188_barter-concept-image-with-business-icons-and-copyspace.html (Accessed: 01 July 2016).

PAGE 20: rawpixel/123RF.COM (2015) *Stock Photo - Trade Stock Swap Investment Trading Concept* [Online]. Available at: http://www.123rf.com/photo_58550035_trade-stock-swap-investment-trading-concept.html (Accessed: 01 July 2016).

PAGE 33: spacedrone808/123RF.COM (2015) *Stock Photo - Inside computer blue vibrant bokeh background backdrop* [Online]. Available at: http://www.123rf.com/photo_58195719_inside-computer-blue-vibrant-bokeh-background-backdrop.html (Accessed: 01 July 2016).

PAGE 40: mindscanner/123RF.COM (2015) *Stock Photo - Word Cloud with Software Design related tags* [Online]. Available at: http://www.123rf.com/photo_26917338_word-cloud-with-software-design-related-tags.html (Accessed: 01 July 2016).

PAGE 52: vanatchanan/123RF.COM (2015) *Vector - content management system concept* [Online]. Available at: http://www.123rf.com/photo_25028393_content-management-system-concept.html (Accessed: 01 July 2016).

Prague College

PRAGUE COLLEGE
Polská 10
120 00 – Prague 2, Vinohrady
Czech Republic

Tel.: (+420) 222 101 020
(+420) 222 722 544
Fax: (+420) 222 718 813

www.praguecollege.cz
info@praguecollege.cz

ISSN 1805-5109

2015/02

CRIS