ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ONTOLOGY-BASED GENERATION OF RANDOM PATHS BETWEEN TWO POINTS

[1]AHMAD TAYYAR, [2]MOHAMMAD ALI H. ELJININI

[1]Associate Professor, Jerash University, Department of Computer Science, Amman, Jordan

[2]Associate Professor, Isra University, Department of Computer Science, Amman, Jordan

E-mail: [1]ahmad.tayyar@hotmail.com, [2]ajinini@gmail.com

ABSTRACT

In the first part of our research, a novel computer algorithm has been proposed to generate random path between two points in space. Random path consists of finite number of randomly generated adjacent points that satisfy the condition: $L(p_ip_n) < L(p_{i-1}p_n)$. Where $L(p_1p_n)$ is the length of the path between the two adjacent points p_1 and p_n. The algorithm has been coded and evaluated. Experiments showed that the randomly generated points were converged to the target point. The main importance of this method is the ability to generate paths between two points in real time, which cannot be predicted in advance. Points are generated, one at a time, where each point brings us closer to the target point. This contribution is applicable to many fields such as economics, engineering, robotics, computer science, military, and other fields of applied sciences.

In this part of our research work, we tackled the problem from different aspects. An ontology has been developed that describes the domain of discourse. The aim is two folds; firstly, to provide an optimized generation of best points that are closer to the target point. Secondly, to provide sharable, reusable ontological objects that can be deployed to other projects. The ontology is designed to guide the process of generating acceptable points based on some criteria described in the ontology. Adding intelligence to the process has provided us with a better solution, in comparison with our previous results. Ontologies have been engineered and reused in many research projects across the globe with success. Our contributed solution can attract many applications in the various fields of applied sciences. One implication of the utilization of ontologies is that it remains poorly exploited field, which may have an impact on the adoption of ontology-driven technologies outside the academic community. We underpinned our solution by the initiation of several case studies that have been designed using and extending our work, and progressing very well. Initial findings are very encouraging and leading to the applicability of the proposed solution to real life cases.

Keywords: *Ontology, Object-Oriented Paradigm, Random Paths, Mobile Robots, Graph Theory.*

1. INTRODUCTION

In the first part of our research, a computer algorithm has been proposed to generate a random path between two points in the plain. A random path consists of finite number of randomly generated points that satisfy the condition: $L(p_ip_n) < L(p_{i-1}p_n)$. Where $L(p_ip_n)$ is the length of the path between the two points p_i and p_n. The algorithm has been coded and evaluated. Experiments showed that the randomly generated points were converged to the target point [1]. The main importance of this method is the ability to generate paths between two points in real time, which cannot be predicted in advance. Points are generated, one at a time, where each point brings us closer to the target. This contribution is applicable to many fields such as economics, engineering, military, as well as other fields of applied sciences.

In this part of the research, an ontology has been developed that describes the domain of discourse. The aim is firstly to provide an optimized generation of best points that are closer to the target point. The ontology is designed to guide the process of generating acceptable points based on some criteria described in the ontology. Adding intelligence to the process has provided us with a better solution in comparison with the first part of our research work. Secondly, the ontology provides sharable, reusable ontological objects that can be deployed to other projects. We have reused and extended the ontology for other projects, which are describe in the last section of this paper. The proposed solution can attract more applications in the various fields of applied sciences.

Ontology has its roots in philosophy as the science of existence. Gruber defines an Ontology as "an explicit specification of a conceptualization" [2]. In other words, a system can be described by a standardized set of terminology and relations between entities. Today, ontology has become a popular topic in many fields that ranges from engineering and medicine to business and enterprise. Many disciplines of computing and information technology are using ontologies to solve problems in ways that are more effective and never done before. This is done by representing ontologies formally, where they can be understood, shared, and manipulated by computer programs. For example, research in information retrieval is evolving towards semantic information retrieval. Researchers are using ontologies to improve the search task by adding intelligence [3, 4, 5]. In addition, Ontology are used in database marketing process where contributions are made towards business intelligence [6]. In the tourism sector, ontology has been used in the development of intelligent tourist information services [7]. These services guide and help tourists in their travels.

As stated in the objectives, the development of ontology for the domain of random path generation is desirable to guide the system to which points should be utilized in the quest of constructing suitable paths. During the first part of this work, we have investigated many pathfinding algorithms. Most of these are concerned with finding the shortest path between two points. For example, Breadth-First Search, an algorithm for traversing points in a graph, which was invented to find shortest path in mazes [8]. It does that by exploring neighboring points before moving to the next level. It finds the shortest path between two points by measuring the number of edges. Best-First algorithm is another algorithm, where it is concerned with the most promising point to follow based on some heuristics [9]. One of the most known algorithms for finding shortest paths is Dijkstra's Algorithm [10, 11]. Many variations of the original algorithm have been developed and used in many applications. Shortest path algorithms, like these and many more, are based on comparisons between all possible paths to reach the target point. All points in the graph are predefined and the paths between the starting point and the target point are known too. In our work, all points between the starting point and the target point are dynamically generated by random function. In other words, we deploy randomness in generating paths to reach the target point. Probably, the closest work to ours is the Random-Walk model (RW), which is a mathematical object, known as a stochastic or random process [12]. It works by choosing random steps until the goal is reached. RW serve as the fundamental model for the recorded stochastic activity in many applications, in many scientific fields such as computer science, economics, physics, chemistry, biology, and ecology. The RW model is mostly concerned with the probability of reaching the target point. For example, the self-avoiding walk [13] and the reinforced random walk [14]. In ecology, various forms of the RW model have been extensively researched to simulate animal movements (i.e. bird migration) [15, 16, and 17]. While simple RW models does not guarantee the convergence to the target point with time constrains, researches extended the model with correlation and bias abilities to reach acceptable results. In our model, we used a biased approach where each generated point is tested and validated base on some constraints as described below.

We have designed an Intelligence Consultation Unit (ICU) that provides guidance to the process of building random paths between the starting point and the target point. It is possible to link two specific points in by generating many points that are converged to the target point.

The importance of this method is reaching the target point using unpredictable paths. The ontology provides descriptions of which condition to consider based on the following criteria:

$L(P_iP_n)<L\ L(P_{i-1}P_n)$), For i = 1, 2, …, n-1

There is unlimited number of paths that satisfy the previous condition between p_0 and p_n. Let p_0 and p_n be two points. Initially we generate a random displacement at the point p_0, the ICU considers the end point of that displacement is p_1 if it was acceptable. This means that our ICU decides if this displacement converges to p_n. Otherwise, a new random displacement is generated. The process is repeated at point p_1 to obtain the next point p_2, and so on until a point with a specific distance from p_n is reached. At the end of the process, a set of points P_i for i= 1, 2, …, n is obtained. The set of points forms the points of the random path that links between p_0 and p_n.

It can be clearly illustrated that the generated paths using the proposed method can be useful in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

many applications such as military and economy that presents confidential pathways with various costs and different segments. For example, this method can be applied for wireless sensor networks and Ad-hoc mobile networks at schools or universities. In this manner, it is possible to generate mobile units as points between the initial mobile station and the final mobile station. Ontologies allow for the sharing of common understanding of the structure of information among these different units. Other examples include robot path planning, building and developing the roads in cities [18, 19, and 20].

The following section presents the ontology design; in addition, the principles for designing ontologies are explored. Section 3 presents the concepts, relations, structures and algorithm used in our work. In section 4, we provide a discussion on the assessment and evaluation of our work. The work is concluded in section 5 and future work is presented.

2. ONTOLOGY DESIGN

From the time of Greek Philosophers until recently, ontologies were merely a set of categories of entities that exist in our world. There were no systematic methodologies existing for the creation of ontologies, and it was only based on the contemplation of the philosophers. The utilization of ontologies in many disciplines made researchers realize the need for well-defined principles and methodologies [21, 22]. The basic methodology of designing ontologies can be defined as follows:

1. Determine the purpose and the scope of the ontology.
2. Capture the representative terms (classes) that exist in the domain of discourse to be used in the ontology.
3. Provide definitions for these classes.
4. Organize these classes in hierarchies (similar to object-oriented paradigms) by defining the relationships between these classes.
5. Transform the ontology into a formal model using one of the knowledge representation languages.
6. Provide evaluation and documentation.

We have used this methodology in the construction of our ontology, which is discussed in the next section in detail. Point 5 above is stating that ontologies are expressed in some formal knowledge representation (KR) languages. Well known ontologies, for example, the SENSUS ontology uses Loom, FrameKit, and Prolog. The GALEN ontology uses the GRAIL language that is based on description logic. The UMLS ontology uses semantic networks to represent the concept and their relationships. In general, ontology formal languages can be categorized into two types; these are logic-based and path-based. Many types and forms of logic have been developed since the fifth century B.C.; some of these are predicate logic, propositional logic, modal logic, temporal logic, fuzzy logic, and finally description logic (DL). DL allows us to construct complex concepts and roles from atomic ones. Concepts are equivalent to classes, which are used to classify the entities in the domain of discourse. Roles are equivalent to properties. Objects are called individuals in DL, which are instantiations of the concepts. Many computer-programming languages have been implemented based on DL such as Loom, Classic, OIL, DAML+OIL, and OWL (The Web Ontology Language). On the other hand, path-based languages have been developed during the seventies and eighties by the AI community. The most well known path-based languages are semantic networks, conceptual paths, and the Unified Modeling Language (UML) that is used to model object-oriented systems. Finally, the OntoUML language, a well-defined ontological language, used for ontology-driven conceptual modeling. It is an extension of the UML based on the Unified Foundational Ontology (UFO).

For ontology construction, we need ontology development environment (ODE). Several systems have been investigated and compared during this research. These systems have many features in common since they all share the same goal, which is building manageable ontologies. Some systems provide extra features while others lack some important ones. Some of these features are:

1. Graphical user interface: Most ontology systems present forms, which allow users to enter information about classes, slots, facets, and other information.
2. Application Programming Interface (API): some of these systems provide users (programmers) with a set of APIs for integration with their own applications.
3. Import and Export format: One of the most desirable features is to allow users to import and export ontologies from and to other formats. Most common formats are XML, RDF(s), and OWL.

4. Extensibility: Some systems allow users to develop and add new features that can be used from within the GUI.
5. Merging: very few systems provide facilities for margining ontologies.

Some of the systems we looked at are OilEd, OntoEdit, Protégé, UML, and OLED. OilEd has been developed by the Information Management Group and the University of Manchester in the UK [23]. It is small and simple tool based on DL that can be used to build small scale DAML+OIL ontologies. It misses some of the features that are found in full-scale ontology engineering environments such as collaboration, merging, and integration. OntoEdit is based on frames and first order logic (FOL) [24]. It is multilingual ontology engineering environment developed by AIFB, University of Karlsruhe. Protégé is being developed by the medical informatics group of Stanford University and is one of the most popular ontology development environments [25]. Protégé is based on FOL and frames, it has an interactive and simplified GUI for constructing and managing ontologies.

The Unified Modeling Language (UML) was developed in 1997 by the Object Management Group [26]. The main purpose of UML was to provide the computing community with a standard common design language for development and building computer applications. Its unified standard modeling notation allowed the computing community to disseminate structure and design plans of computer systems and applications. Finally, the Menthor editor, a free ontology-engineering system for the development of OntoUML models. The interested reader may find comparisons between these various systems that are more detailed in the literature, which is out of the scope of this work, for example in [27, 28].

3. ONTOLOGY FOR THE GENERATION OF RANDOM PATHS

This section presents the main contribution of this paper. Following the basic methodology for designing ontologies, which we have presented in the above section, the purpose of the ontology for the domain of random path generation is to define the various concepts used in this work, and to guide the system as to which points to be utilized in the quest of constructing random paths. One of the concept in our work is the ICU, which provides guidance to the process of building random paths. There are several concepts (classes) in our domain of discourse that need to be addressed and defined. The main concept is a point. A point is a vertex that has two attributes: x and y. These attributes represent the location of the point in x-y coordinates. Once an object of type (point) is instantiated, we use getter and setter methods to manage its location. The next concept in our model is the ICU, which contains a description of all the information needed for the construction of the random path between the initial and the final points. Its responsibility is to insure a correct displacement of random points along the path to the final point. In our implementation we used circles, which are generating random paths within circles centered in p_n: c_1, c_2, …, c_{n-1} that passes thru the points p_1, p_2, …,p_{n-1} respectively. Therefore, our next concept is the circle. Point v may approach p_n if it takes any position within circle c centered in p_n with radius pp_n. To determine the optimal solution for the task of generating and selecting random points with regular distribution in the path starting from point p, the ICU uses the following procedure:

$$x = x – L + random(2*L+1);$$

$$y = y – L + random(2*L+1);$$

As a result, this gives a random displacement within a square, where the length of its side is (2L) and its center is at the point p(x,y). Our next concept is a square. Random displacement is generated at the point $p_0(x_0,y_0)$ within the square, and point p_1 as the end point of that displacement if it was located within the circle c0. Otherwise, the procedure is reworked in the opposite case. Therefore, the generation process is repeated at p_1 and within the Circle c1 centered in p_n with radius p_1p_n to obtain the new point p_2. This iteration comes to halt when reaching a point with a specific distance from P_n. This process will result in obtaining a set of points p_1, p_2, …, p_{n-1}. The square that contains the movement is called the square of displacement. The length of its side limits the distance value between the generated points. We should note here that circles and squares are shapes. In object-oriented terminology, the relationship between a circle and a shape is the “is-a” relationship. We can say that a circle inherits the attributes and behavior of a shape. In the same sense, a square inherits the attributes and behavior of a shape too. Circles and squares are special types of shapes. Shapes are more general, and squares and circles are more specific. We should also note here that shapes are abstract concepts; we do not instantiate objects for shapes directly.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Circles and squares are used to direct the path of the generated points towards the ending point. In this sense, the paths are guaranteed to reach the target. Our next concept is the path. A path is a set of adjacent points in path $p = p_1, p_2, \ldots, p_n$. Therefore, we can say that a random path is generated from the set of acceptable points. Distance value between the adjacent points forming the random path increases by the increase of the square of displacement side length. According to the proposed method, the generated points are converged toward p_n, which are satisfying the following equality [1]:

$$L(p_{n-1}p_n) < L(p_{n-2}p_n) < \ldots < L(p_1p_n) < L(p_0p_n)$$

We should mention here that ontologies work well with logical and factual knowledge representations. It does not fit well in procedural code, because it forces us to express logical relationships as a sequence of instructions. In our work, since we do not have many rules, we decided to go with UML to describe our concepts and the relationships. UML is an excellent unified language for modelling computer systems. Figure 1 shows the UML class diagram, which presents the various classes used in our work.

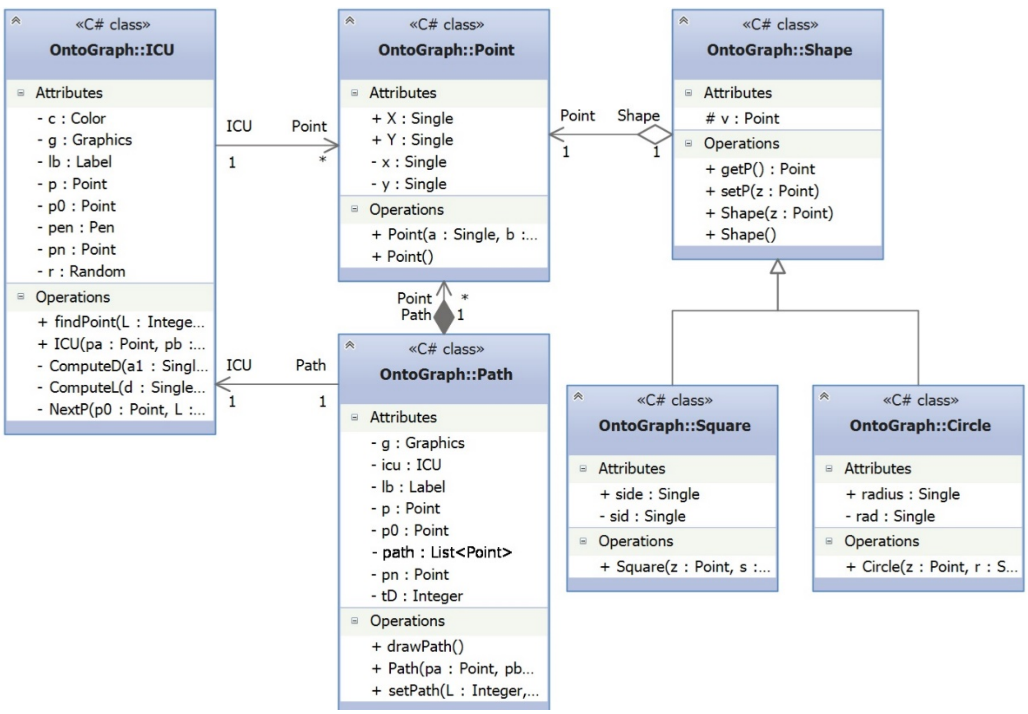

Figure 1: UML Class Diagram

Note that the point is in the central and connects all other classes. For example, a path is made of a list of adjacent points. Each acceptable point must pass a test defined in the ICU unit. The central point in a shape is a point that presents its location. In reality, a shape cannot be instantiated and it is not a concreate class. It is used for inheritance only. It is too general to instantiate objects. A shape must be shaped to a circle or square via inheritance first before creating the object. We have shown previously that circles and squares are shapes. Therefore, when we instantiate a circle, for example, it contains a point that represents its location which is inherited from shape.

Figure 2 presents our extended model, which uses object-oriented paradigms in the quest of generating random paths. After reading the initial values, an instantiation of an empty path and an ICU is performed. Every time a new point is generated, the ICU is consulted for accepting the new point to be added to the constructed path. The process ends when the path reaches the target point. The output of this process is the set of the adjacent points that connect the starting point with the target

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

point, which are generated dynamically by random function.

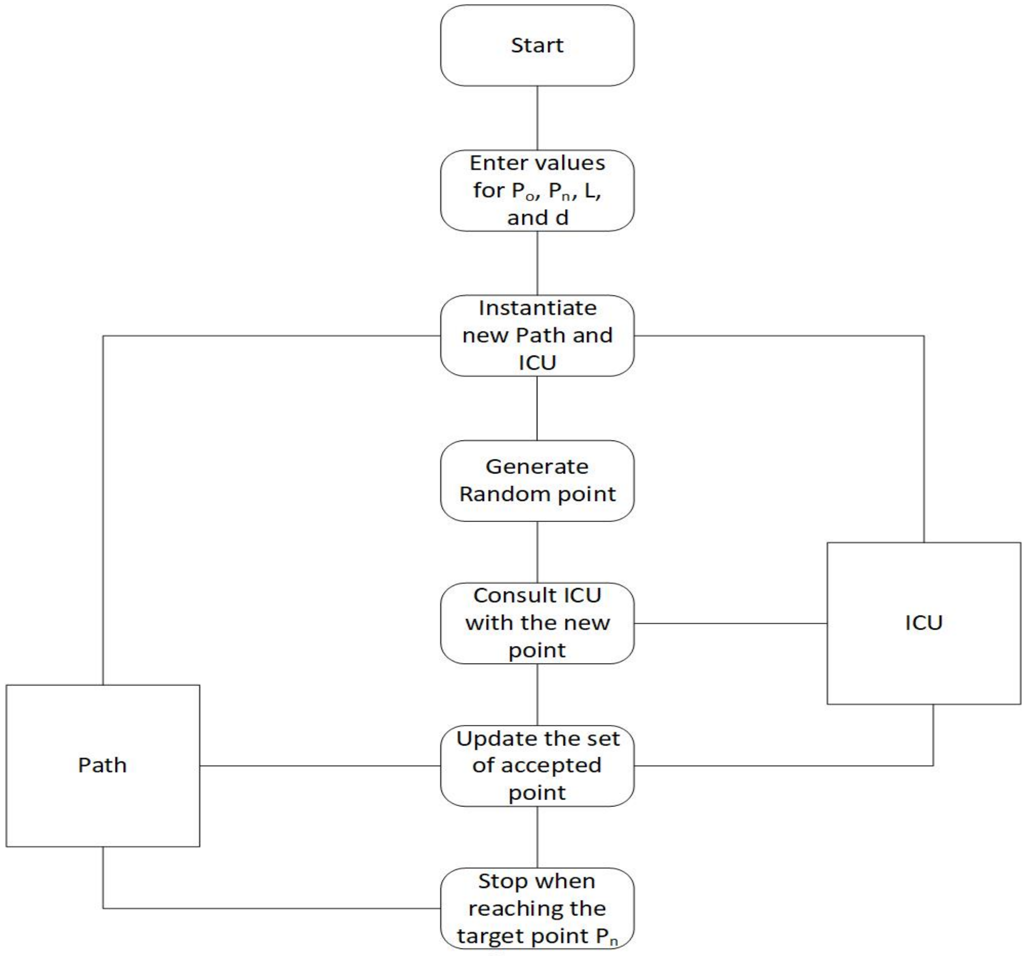

Figure 2: Random Generation Model

Our ICU is a rule-base engine that is used to decide which points to accept based on certain thresholds that is explained previously.

In our experiment, a set of initial values are needed:

Let $X_0 = 40$, $X_n = 542$, $Y_0 = 348$, and $Y_n = 348$.

Then we calculated the distance between (X_0,Y_0) and (X_n,Y_n) which was equal to 502, using the following equation:

$$Distance\ (P_0, P_n) = \sqrt{(X_n - X_0)^2 + (Y_n - Y_0)^2}$$

We ran the program several times using the same initial values, and every time we obtained a new group consists of a set of several paths. Figure 3 shows 4 groups where each group has 5 randomly generated paths obtained from running the algorithm using the C-Sharp programming language.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

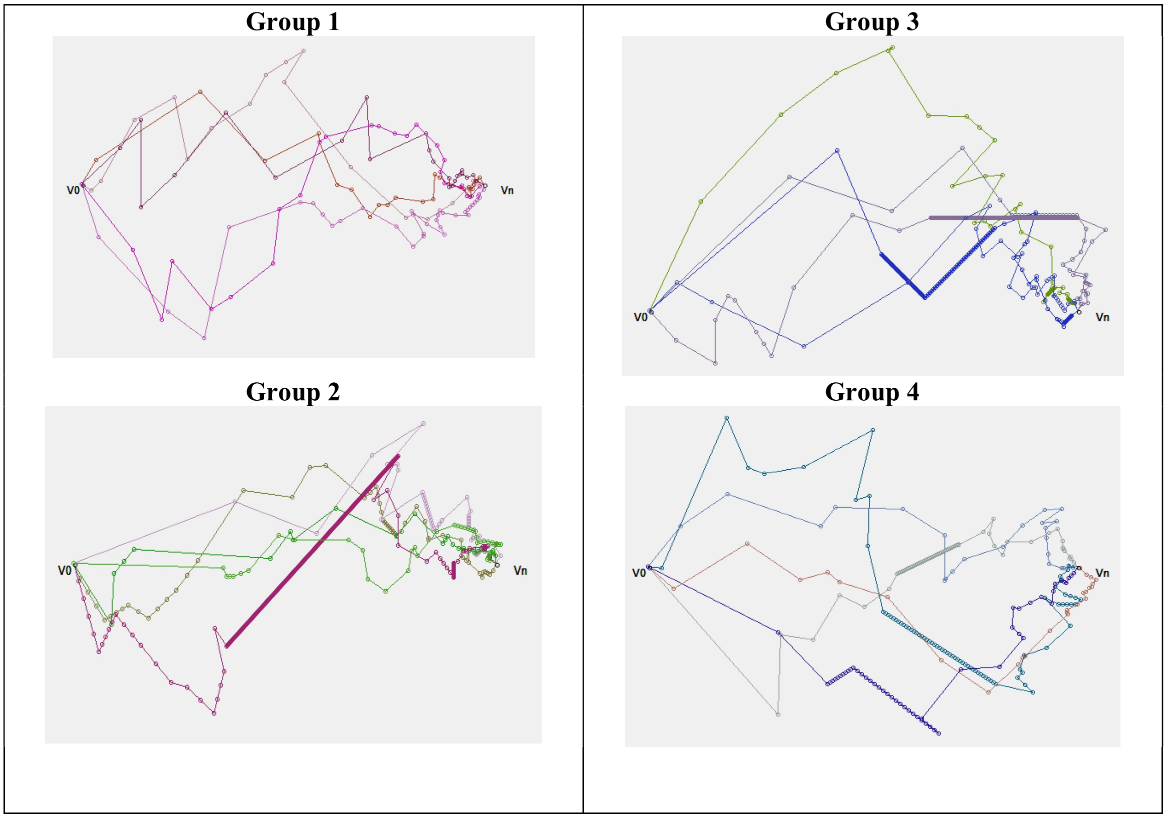

Figure 3: Four Groups of Random Paths

Table 1: Length of each path in each of the 4 groups

	Group 1	Group 2	Group 3	Group 4
Path 1	1043	1191	1439	1497
Path 2	876	1029	805	1039
Path 3	994	810	910	958
Path 4	1285	1401	1377	1036
Path 5	1020	1010	1001	1029
Average lengths	1043.6	1088.2	1106.4	1111.8

We calculated the length of every edge, and computed the length of each path. We repeated this process for each group. The results are shown in table 1. The maximum path length shown in the table is closed to 1500, and the minimum is closed to 805, where the average is approximately 1085.

We tested the algorithm again where the distance between two points is 200. This time we randomly generated paths in groups of 1, 10, 100, 1000, 10000, and 100000 iterations. We computed the length of each random path, in addition to the average, the minimum, and the maximum distance for each group as shown in table 2.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: length of each path with various iterations

Number of Iterations	Length of path	Average	Max	Min
1	200	233	233	233
10	200	272	306	228
100	200	289	401	210
1000	200	297	491	208
10000	200	298	543	202
100000	200	297	617	200

From this table, we can see that the average stayed below 300, while the minimum distance converged to 200, which is the distance of the straight path between the start point and the end point. The plotted path in figure 4 presents our findings, where we can see the average is bounded within the 200 and 300 columns, and the minimum distance, with the larger number of iterations is converging towards the 200 column.

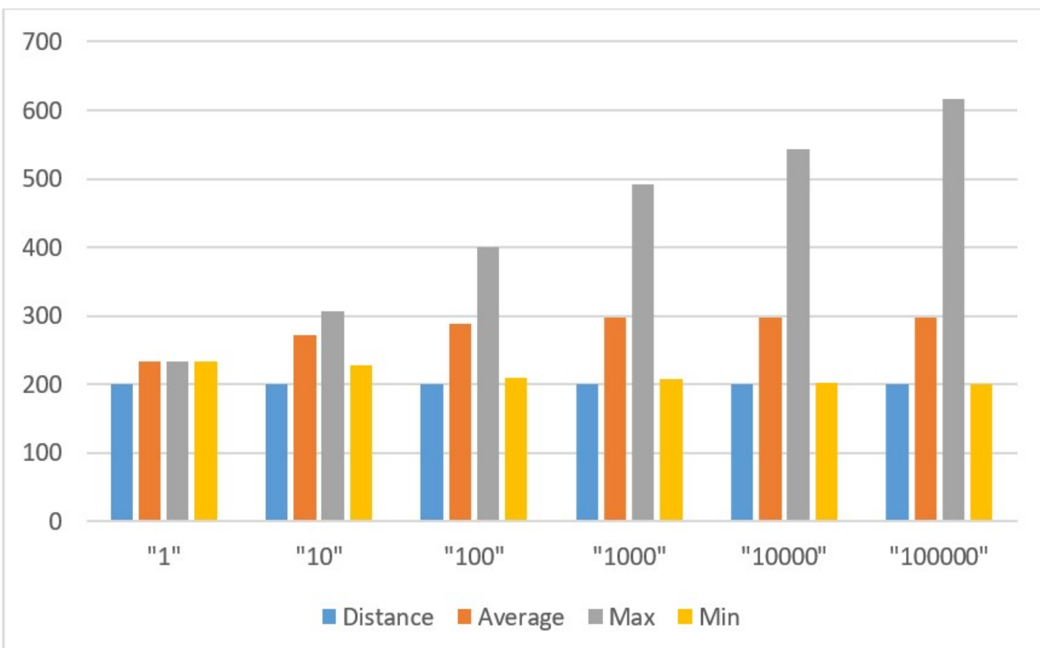

Figure 4: Results

4. DISCUSSION AND EVALUATION

The main objective of this work is the creation and utilization of the ontology. The purpose of the ontology is firstly to guide the process of generating acceptable points that converge to the target point. Secondly, the ontology provides us with sharable and reusable objects that can be shared and reused in other projects. Thousands of ontologies have been created and used in various projects. Many of them are reported in the literature. How can we evaluate the quality of these ontologies including ours? While there are no standard approach that has been adopted by everyone, most evaluation methodologies in the literature are clustered around two concepts; these are verification and validation of the ontology.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Verification is about how well the ontology is structured. Does it cover all related concepts in the domain of discourse? How these concepts are relate to each other's?

Validation is about how well the ontology is used in ontology-enabled applications. How well does the ontology serve its purpose? It is concerned about the usability of the ontology.

To answer questions about ontology verification, we have formed a set of quality criteria that are used with the state of the art in ontology evaluation. We have consulted several experts in the domain of discourse. Since "Shape" is an abstract concept on the top level of our ontology, we have searched the literature for ontologies about shapes. In [29], there are two types of shapes; geometric shapes, such as circles and squares, which have precise mathematical formalization. The other type is physical shapes (or organic shapes), such as trees and clouds which have irregular shapes. In [30], dimensionality are taken into consideration, for example, a circle is 2-D Shape, and a sphere is 3-D shape. In this work, we are mostly concerned with 2-D geometrical shapes like circles and squares, which are used in the generation of random paths. Other types have been taken into consideration with our future work. For example, in the Drone Navigation System, which presented in section bellow, some 3-D shapes and physical shapes are considered obstacles and must be defined in the ontology.

With these concepts in mind, now we can present the quality criteria as follows:

1. Clarity: It states in the literature that an ontology should effectively communicate the intended meaning of defined terms. Since our ontology consist of shapes, which are mathematical objects with well-defined meaning, this does not leave room for ambiguity.
2. Extendibility: This is another measure of an ontology with good quality. The ontology should allow for extendibility. In our work, we are using pure object-oriented paradigms (OOP). Inheritance and composition are two concepts in OOP that are used for software reuse and extendibility, which our model is based on.
3. Expandability: This refers to the effort required to add new definitions to an ontology and more knowledge to its definitions without altering the set of well-defined properties already guaranteed. One of the main benefits of using OOP is that we can add new concepts in the hierarchy. New concepts can be specialized by adding new properties.
4. Consistency: A given definition is consistent if and only if the individual definition is consistent and no contradictory sentences can be inferred from other definitions and axioms. Natural Language ontologies that are mainly used with the semantic web are the most suffering from consistency, where contradictions can be easily found. However, in domains such as ours, most concepts are mathematical objects with well-formed definition.
5. Completeness: Incompleteness is a fundamental problem in ontologies. Our ontology is currently incomplete in the sense that it does not have definitions for all known concepts of shapes. On the other hand, it is complete in the sense that it has definitions for all the shape concepts that are used in the problem of generating random paths between two points.
6. Conciseness: The ontology utilized all concepts, and with no redundancies between concepts.

We have chosen these quality criteria that fits with our type of ontology, other criteria are more suited towards other types like domain ontologies and top-level ontologies.

The other concept used with ontology evaluation is validation. It is about how well the ontology is used in our application and other domain-specific applications. In Section 3, we have performed extensive testing with thousands of iterations. Our empirical results have shown that the generation of random paths are guaranteed to reach the target point. We also compared the new results with previous results, which shown that using the ontology gave us better control over the results. This has proved the usability of the ontology in the application of generating random paths between two points.

5. CONCLUSION AND FUTURE WORK

We have shown the importance of generating a random path between two points (P_0, P_n). The proposed work demonstrates the generation of the proposed work demonstrates the generation of random paths in intelligent way using ontology. The ontology provided us with a description of the domain of discourse that contains reusable and shareable concepts that can be used in other projects. The original algorithm has been completely refactored into object-oriented paradigm and rewritten in the C-Sharp Programming language. The new organization allowed us to reuse the various component in new applications and case studies. Moving from the C language under the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DOS operating system to C-Sharp under windows also helped us in the processes of visualization and comparison of the random generated paths with ease.

The utilization of the ICU also can with the advantage of selecting the most desirable paths based on some criteria that can be set in advanced. This feature is becoming one of the major factors when applying to new applications.

This study is also applicable and valuable to several applications in networks such as WSN, Ad hoc mobile networks and telecommunication networks [31].

As stated above and as part of future work, this algorithm will be extended to calculate the weights of the best random path between two points to come to more optimal solution. We are also in the progress of applying our algorithm to the following cases:

a. Automated Drone Navigation Systems

Our preliminary study on this subject has shown that current technologies of the automated drone navigation systems (ADNS) is still in its infantry stages [32, 33, 34]. Most drones today use pilot-controlled navigation systems.

Due to its low cost and low risk of casualties, many countries around the world are using drones to preform various tasks ranging from taking surveillance data to sensing enemy's targets to enhance monitoring in disputed territories.

Flying drowns in straight lines would make them easy targets. On the other side, the utilization of random paths generation would give drowns the ability to maneuver intelligently with unpredictable paths.

b. Computer Simulations

Computer simulations of real-word processes are commonly used in many computer applications. For example, in computer games, simulating the movements of some entities like animals, characters, and other objects in randomly generated paths [35, 36]. The aim of such unpredictable movements is to gain a winning edge in the game. Some scholars suggest that random movements can be predictable using some mathematical methods [37, 38].

A group of our students has initiated a study to capture the regularities of such movements using derivations of Markov Chains Model.

c. Crawling the World-Wide-Web

The World-Wide-Web is considered a huge dynamic digraph. Each vertex in the graph consists of a webpage. Each webpage may contain textual data, images, and video clips. The edges of the graph are made of hyperlinks that connect webpages with other adjacent webpages. Users search the web for subjects by entering keywords using search engines.

Search engines deploy some applications, called crawlers that traverse the web collecting and processing all types of data they find on webpages, saving it in databases for later retrieval. Crawlers use breadth-first graph searching algorithm while traversing the web. In breadth-first search, the crawler starts with a webpage, as a seed, and starts the processing level-by-level using a queue.

Is it possible to crawl the web in random order? What kind of results we may get? Some scholars are using random walk theories to visualize subsets of the web and collect information in order to study web properties. Crawling the web in random order is unbiased process, while our algorithm is biased, because we must converge to the target point. Random walk models has been used in biology, physics, ecology, medicine, computer science, and other scientific disciplines [39, 40, 41, 42].

REFRENCES

[1] A. Tayyar, "Generating Random Paths between Two Points in Space: Proposed Algorithm", *Proceedings of the International Conference on Computer Science, Computer Engineering, and Social Media*, Thessaloniki, Greece, 2014.

[2] Gruber TR. "Toward principles for the design of ontologies used for knowledge sharing", *International Journal of Human-Computer Studies*, Vol. 43, 1993, pp. 907-928.

[3] L. Gondy, C. Hsinchun, "Meeting medical terminology needs-the ontology-enhanced medical concepts mapper", *IEEE Transactions on Information Technology in Biomedicine*, Vol. 5 (4), 2001, pp. 261-270.

[4] P. Srinivasan, J. Mitchell, O. Bodenreider, G. Pant, F. Menczer, "Web crawling agents for retrieving biomedical information", *Proceedings of the International Workshop on Bioinformatics and Multi-Agent Systems (BIXMAS 2002)*, Bologna, Italy, July 2002.

[5] M.A. Eljinini, "The Medical Semantic Web: Opportunities and Issues", *International Journal of Information Technology and Web Engineering (IJITWE)*, Vol. 6 (2), 2011, pp. 18-28.

[6] F. Pinto, M.F. Santos, A. Marques, "Ontology based Data Mining – A contribution to Business Intelligence", *Proceedings of the 10th WSEAS Int. Conference on MATHEMATICS and COMPUTERS in BUSINESS and ECONOMICS*. Portugal, 2009.
[7] H Park, A. Yoon, H. Kwon, "Task Model and Task Ontology for Intelligent Tourist Information Service", *International Journal of u- and e- Service, Science and Technology*. Vol. 5, No. 2, June 2012.
[8] E. F. Moore, "The shortest path through a maze". *Proceedings of the International Symposium on the Theory of Switching*. Harvard University Press, pp. 285–292, 1959.
[9] J Pearl, "Heuristics: Intelligent Search Strategies for Computer Problem Solving", *Addison-Wesley*, p. 48, 1984.
[10] E.W. Dijkstra, "A note on two problems in connexion with graphs". *Numerische Mathematik*. 1: 269–271, 1959.
[11] P. Frana, "An Interview with Edsger W. Dijkstra". *Communications of the ACM*. 53 (8): 41–47, 2010.
[12] K. Pearson, "The Problem of the Random Walk", Nature, 72(1865), 1905, pp.294-294.
[13] N. Madras, G. Slade, "The Self-Avoiding Walk", *Birkhäuse*r, Boston, 1996.
[14] R. Pemantle, "A survey of random processes with reinforcement", *Probability Survey*, 4: 1–79, 2007.
[15] EA Codling, MJ Plank, S Benhamou, "Random walk models in biology", *Interface*, 5(25), 813–34, 2008.
[16] EA Fronhofer, T Hovestadt, H.J Poethke, "From random walks to informed movement", *Oikos*, 11(6), 857–866, 2013.
[17] JS Horne, EO Garton, SM Krone, JS Lewis JS, "Analyzing animal movements using Brownian bridges", *Ecology*, 88: 2354-2363, 2007.
[18] J. Barraquand, B. Langlois,, J. Latombe, "Numerical potential field techniques for robot path planning", *IEEE Trans. Syst., Man, Cybern.*, 22(2), 1992, pp. 224-241.
[19] N. Amato, Y. Wu, "A randomized roadmap method for path and manipulation planning", Proceedings *of IEEE International Conference on Robotics and Automation*. 1996.
[20] A. Roth, J. Vate, "Random Paths to Stability in Two-Sided Matching", *Econometrica*, 58(6), 1990, p.1475.
[21] N. Fridman Noy, D. Hafner, "The state of the art in ontology design", *AI Magazine*, 18(3), 1997, pp. 53-74.
[22] N. Guarino, "Some ontological principles for designing upper level lexical resources", *Proceedings of First International Conference on Language Resources and Evaluation*, Granada-Spain, 1990, pp. 527-534.
[23] S. Bechhofer, I. Horrocks, C. Goble, R. Stevens, "OilEd: A reason-able ontology editor for the semantic web". *Advances in AI, LNAI 2174*, Berlin: Springer, 2001, pp. 396-408.
[24] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke, "OntoEdit: collaborative ontology development for the semantic web", *Proceedings of the International Semantic Web Conference*, Berlin: Springer-Verlag, 2002, pp. 221-235.
[25] N. Fridman Noy, A. Chugh, W. Liu, M.A. Musen, "A Framework for Ontology Evolution in Collaborative Environments", *The 5th International Semantic Web Conference*, Athens, GA: LNCS 4273, 2006.
[26] PPD. Silva, NW. Paton, "User Interface Modelling with UML", *EJC*, pp. 203-217.
[27] M.A. Eljinini, NA. Sarhan, "An Ontology for extracting information from the World Wide Web", *Proceedings of the Second Scientific Conference, (Administrative and Strategic Thinking in changing world)*, Faculty of Administrative & Financial Sciences, Amman-Jordan, 2007.
[28] M.A. Eljinini MA, "The Role of Ontology in the Process of Extracting Information from Online Resources", *Proceedings of the 2010' Joint Scientific Symposium of Faculty of IT, Isra University, and Baghdad Economic University college*, Amman-Jordan, 2010.
[29] R. Rovetto, "The Shape of Shapes: An Ontological Exploration", *Proceedings of the 1st Interdisciplinary Workshop on SHAPES*. Karlsruhe, Germany, 2011.
[30] M. Niknam and C. Kemke, "Modeling Shapes and Graphics Concepts in an Ontology", *Proc. SHAPES 1.0 - The Shape of Things. Workshop at CONTEXT-11, CEUR-WS*, Vol. 812, 2011.
[31] S. Meguerdichian, F. Koushanfar, "Exposure in wireless Ad-Hoc sensor networks MobiCom", *7th annual international conference on Mobile computing and networking,* New York: ACM New York, pp.139-150.
[32] T. Krajn´ık, J. Faigl, V. Von´asek, K. Koˇsnar M. Kulich, L. Pˇreuˇcil, "Simple yet stable

bearing-only navigation", *Journal of Field Robotics*, Vol. 27, 2010, pp. 511–533.

[33] W. Shan Ng, E. Sharlin, "Collocated interaction with flying robots", published in: *RO-MAN*, 2011 IEEE, Atlanta, GA, USA, 2011.

[34] T. Krajn´ık, V. Vonasek, D. Fi ´ ser, and J. Faigl, "AR-drone as a platform for robotic research and education," in *Proc. Research and Education in Robotics: EUROBOT* 2011, 2011.

[35] L. Kleinrock, "Queueing Systems", John Wiley & Sons, 1975.

[36] L. Allen, G. Jackson, J. Ross, S. White, "What counts is how the game is scored: One way to increase achievement in learning mathematics", *Simulation & Games*, Vol. 9, 1978, p.p. 371-389.

[37] O. Kallenbeg, "Random Measures, Theory and Applications", *Springer International Publishing AG*, 2017.

[38] R. Kaulz, "Chaos: The Science of Predictable Random Motion", *Oxford University Press Inc.*, New York, 2011.

[39] F. Bartumeus, M.G.E. da Luz, G.M. Viswanathan, J. Catalan, "Animal search strategies: a quantitative random-walk analysis", *Ecology*, Vol. 86, 2005, p.p. 3078–3087.

[40] E.A. Codling, M.J. Plank, S. Benhamou, "Random walk models in biology", *Journal of the Royal Society Interface*, Vol. 5 (25), 2008, p.p. 813–834.

[41] P.M. Kareiva, N. Shigesada, "Analyzing insect movement as a correlated random walk", *Oecologia*. Vol. 56, 1983, p.p. 234–238.

[42] W. Alt, "Biased random walk models for chemotaxis and related diffusion approximations", *J. Math. Biol*. Vol. 9, 1980, p.p. 147–177.