Conceptual Wiki Page Simulation
A Discrete Space Agent-based Approach

Roger W. McHaney and Jonathan Mossberg
Management Information Systems, Kansas State University, 101 Calvin Hall, Manhattan, KS, U.S.A.

Keywords: Agent, Simulation, Discrete Space, Wiki, Anylogic, Agent-based Simulation.

Abstract: This paper describes the preliminary development stages of an agent-based model (ABM) used to understand and anticipate changes to Wiki pages. A discrete space approach was used to structure the model. Letters from words in the wiki were represented as agents which could be changed, deleted, or added based on rates derived from wiki page histories. A C# pre-processor, called Wiki-Hist-Heist, was developed to facilitate analysis of existing wiki page histories and provide model inputs based on detected patterns and resulting distributions. The conceptual version of the Wiki Page ABM was built using AnyLogic. It provided a framework for user-friendly features which allow easy changes to inputs so a variety of pages and scenarios can be modelled. Additionally, this project illustrated the usefulness of ABM in this domain. Limitations and future study directions are included.

1 INTRODUCTION

The overarching purpose of this research was to develop an ABM (Heath et al., 2009) to help understand, anticipate and ultimately manage wiki operations more efficiently and effectively. "Wiki" is derived from the Hawaiian word for "quick." This term is meant to imply a technology which facilitates the rapid changes of web pages resulting from a range of contributors. In a modern sense, a wiki is a web site technology used to facilitate mass collaborative authoring, editing, and sharing (Mader, 2008). In its purest form, Wikis are open to all users and permit participants to create, edit, delete, and link to pages using a standard web browser without the need for specialized training (McHaney et al., 2013). Wikis serve many practical purposes. For instance, they may offer a venue for displaying useful content or become an organization's information repository. Additionally, wikis may provide convenient ways of enabling collaborative synergies through asynchronous interaction.

Wikis support a variety of media including images, audio files, videos, text, hyperlinks, embedded widgets, and other standard Web page features. Wikis can be open access or controlled with permissions to carefully regulate user privileges (García, 2012). The most popular example of active wiki technology is Wikipedia which has become the world's de facto virtual encyclopedia. Wikipedia was developed by participants from across the globe who contributed to a variety of informational subjects by: sharing research, editing, translating, updating, correcting, and maintaining source pages.

Wikis are not without drawbacks. For instance, the use of a wiki requires careful oversight to ensure contributions are consistent with a page theme. Since most wikis are developed as venues open for contributions, they have an inherent vulnerability to a variety of misuses. Vandalism, spam, trolling, commercial hijacking and other issues can emerge as major problems (The Computer Language Company, 2011).

Depending on characteristics such as level of oversight, size, scope, structure, and importance, intentional vandalism and unintentional errors may not readily be detected. This problem, in part, has prompted the current investigation. Our research premise becomes: *Can insight into wiki page editing patterns be gained through development and analysis of an agent-based model (ABM)?* Further, this study describes a generic method whereby users can easily change model inputs so their specific interests regarding a wiki can be examined. The remainder of this study provides a look at wikis and describes the process used to develop a user-driven model of wiki page changes.

McHaney R. and Mossberg J..
Conceptual Wiki Page Simulation - A Discrete Space Agent-based Approach.
DOI: 10.5220/0005506003930400
In *Proceedings of the 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2015), pages 393-400
ISBN: 978-989-758-120-5

2 WIKI OVERVIEW

As stated earlier, without careful oversight, changes to Wiki pages may go unnoticed. Although most wiki software, including the leading software system—MediaWiki---provides tools for monitoring and reporting changes, a long-term commitment to maintaining oversight becomes a time commitment that is difficult to anticipate and manage. Based on experiences with wikis at Kansas State University, we often asked, "How much time is needed to properly oversee a wiki?" The webmaster or more appropriately phrased, wiki keeper, has emerged as a person with key roles for conducting this oversight (McHaney, 2012). The wiki keeper must spend his or her time reviewing changes then take corrective action to undo any problematic entries (Sutton, 2006). Historically, wiki keepers tend to be non-confrontational to encourage user contribution and interaction rather than provoke retaliatory responses. Wiki keepers often use a soft security approach to protect the wiki, its users, and preserve informational integrity from injurious actions (Meatballwiki, 2011).

These defenses must tread the fine line between offering protection and preventing legitimate users from unnecessary constraint. Many wiki keepers gradually ramp up action as trolling and spam becomes more troublesome. Many wiki platforms, such as MediaWiki, offer features that make undoing damage easy by offering change rollback. Often, vandals will attempt to test a wiki's oversight practices by making subtle changes and then gauging the response. If no response is detected, then the wiki might be used for posting larger amounts of information related to the vandal's agenda. Ignoring vandalism generally is not an option because contributor time is wasted and inaccurate information can ruin the reputation of the wiki (McHaney, 2012).

2.1 Wiki Keepers

A wiki keeper's primary role, particularly in small, open wikis, is that of a content editor. As new content is added by users, grammar, style, word choice, and formatting may be inconsistent or substandard. It falls to the wiki keeper to encourage other users to take on editorial roles. If this does not occur, the wiki keeper must personally make required edits. Even if others do become editors, their work must be reviewed periodically to ensure consistency. Depending on contribution quality, a substantial time commitment may be required. Higher numbers of users beget more changes. This leads to another task: the wiki keeper will need to view terms used as indexing tags and update the ontology (Hai-Jew and McHaney, 2010). It is in the best interest of a wiki to have tags with both a "conceptual consistency" and a "syntactic consistency." (Hepp et al., 2007, p. 55).

2.2 Wiki Editing

In most wiki systems, including MediaWiki-based implementations, pages are created as plain text with basic formatting symbols or strings. Usually, wikis track change histories of a document reflecting both the time of change and the magnitude of the changes in terms of a character count. Each time a collaborator makes changes to a wiki page, the newly revised page becomes the current version. Older versions of the document can be reviewed, compared side-by-side with the current or older versions, and inappropriate edits can be "rolled back." This is convenient because the current project relies on the use of history documents to provide inputs for the Wiki Page ABM. Figure 1 provides an example of a page history from a Wikipedia page titled "Wilson Sawyer."

3 MODEL DEVELOPMENT

The model created for this project is loosely coupled to Wikipedia. Its page histories provide a data source that drives model inputs. Although this article uses Wikipedia sources, it should be noted that the model pre-processor can derive input data from page histories for any MediaWiki-based wiki site. Minor modifications to the pre-processor's code could extend this functionality to nearly any other wiki site.

3.1 Model Pre-processor

A model pre-processor, called Wiki-Hist-Heist, was custom developed using C# in the Visual Studio development environment from Microsoft. The software prompts a user to enter the name of the Wikipedia page and then processes the history to derive data representing frequency and magnitude of changes to a page over time. Outputs are written to a CSV file where the data can be further analysed with spreadsheet functions and statistical tools. Figure 2 provides an image of the user interface for the program.

Currently, the software includes all changes to the history page regardless of source or nature. Future revisions are planned to enable filtering on items included in the data set. For instance, talk page changes will have the option of being excluded.

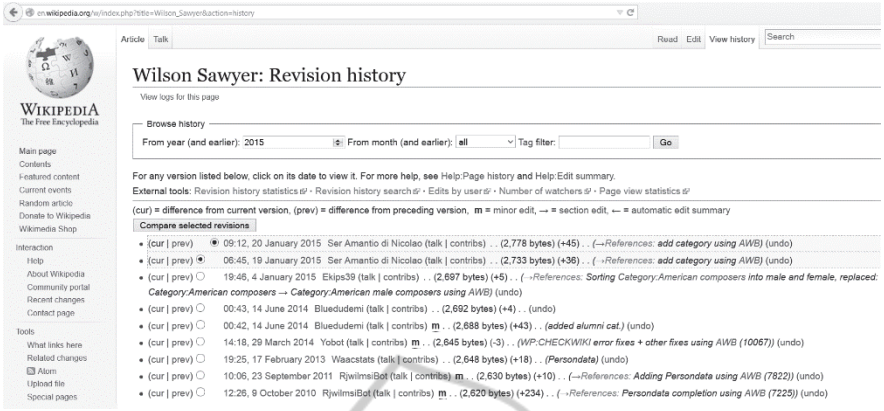


Figure 1: Example of wiki page history which can be pre-processed for ABM inputs.

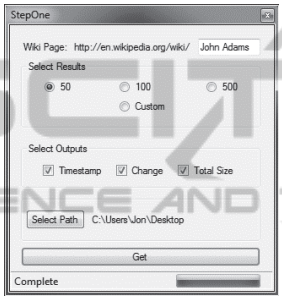


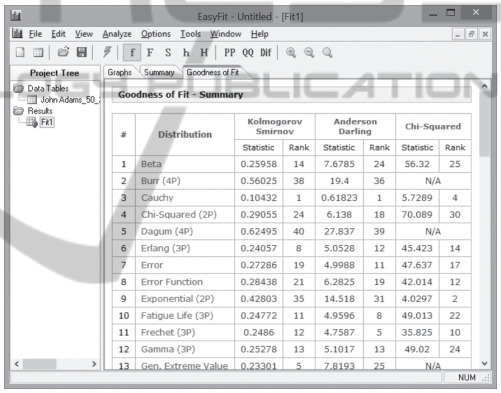
Figure 2: Wiki-Hist-Heist user interface.

Figure 3 provides an image of the CVS file containing the data generated by Wiki-Hist-Heist. For this project, the contents of the output CVS files were analysed with EasyFit Professional from MathWave Technologies to provide change frequency and magnitude distributions for use as model inputs.

	A	B	C	D
1	TIME	DATE	CHANGE	SIZE
2	4:17	20 Februa	6	110417
3	1:34	15 Februa	-7	110411
4	1:15	15 Februa	7	110418
5	0:01	15 Februa	-4	110411
6	13:25	14 Februa	90	110415
7	16:21	12 Februa	-58	110325
8	4:17	6 Februar	-20	110383
9	4:17	6 Februar	-5	110403
10	4:16	6 Februar	0	110408
11	4:15	6 Februar	-12	110408
12	4:14	6 Februar	32	110420
13	21:55	3 Februar	22	110388
14	21:54	3 Februar	22	110366

Figure 3: CVS file contents derived from a Wikipedia history page by Wiki-Hist-Heist.

These distributions were used to drive agent activities within the Wiki Page simulation. Figure 4 provides a representative view of candidate distributions derived from the character change magnitudes per wiki edit. The output distributions were generated and input into the ABM. The character change magnitude data from the "John Adams" wiki page, as shown, was found to most closely resemble a Cauchy distribution (p=.10) in this example.



#	Distribution	Kolmogorov Smirnov		Anderson Darling		Chi-Squared	
		Statistic	Rank	Statistic	Rank	Statistic	Rank
1	Beta	0.25958	14	7.6785	24	56.32	25
2	Burr (4P)	0.56025	38	19.4	36	N/A	
3	Cauchy	0.10432	1	0.61823	1	5.7289	4
4	Chi-Squared (2P)	0.29055	24	6.138	18	70.089	30
5	Dagum (4P)	0.62495	40	27.837	39	N/A	
6	Erlang (3P)	0.24057	8	5.0528	12	45.423	14
7	Error	0.27286	19	4.9988	11	47.637	17
8	Error Function	0.28438	21	6.2825	19	42.014	12
9	Exponential (2P)	0.42803	35	14.518	31	4.0297	2
10	Fatigue Life (3P)	0.24772	11	4.9596	8	49.013	22
11	Frechet (3P)	0.2486	12	4.7587	5	35.825	10
12	Gamma (3P)	0.25278	13	5.1017	13	49.02	24
13	Gen. Extreme Value	0.23301	5	7.8193	25	N/A	

Figure 4: Representative output from EasyFit Professional from MathWave Technologies.

3.2 ABM Approach

An ABM worldview was used to construct the basic units of activity for this simulation project (Bruch and Atwell, 2013; Macal and North, 2010; Taylor, 2014). The primary agent population, wikiletters, was modelled using a discrete space approach where the letter agents were represented in the form of a rectangular grid of cells. Each cell was one letter on a wiki page and became a wikiletter agent with unique identities, parameters and system states. The grid was designed to be variable in size. This allows modelling wiki pages of different sizes. A second agent type was also built into the model. This agent was the wiki editor, represented by a single entity with the power to affect changes on the wikipage. In other words, the editor agent, driven by data from the Wiki-Hist-Heist program, periodically interacted with the grid based on derived interarrival frequencies, and made changes

to the wikipage impacting a discrete number of letters. The number of letters changed per edit was based on the letter change magnitude distribution derived from the Wiki-Hist-Heist output data from wiki page histories. The editor's actions directed particular letter agents to become obsolete and eventually be deleted from the page. New letters were added or changed as agents were recycled or added. This reflected the real world activities of a person entering the wiki and adding, deleting or changing a letter, word, or sentence.

It is important to note that the structure of this model's agent hierarchy was influenced by several goals (Helbing and Balietti, 2011). First, the real world system of how wikis are developed and edited had to be distilled into its components. Second, the model was developed to make it flexible in terms of size. Third, the model was devised in a way that would provide a meaningful visual component. And finally, the model was developed to allow the addition of more details and constraints as the system matured and became more sophisticated.

3.3 AnyLogic

AnyLogic simulation software was used to create the model for this project. AnyLogic contains elements that provide support for mixed modelling including discrete event, system dynamics, and agent-based modeling. AnyLogic facilitates prototyping models, detailing system design, and constructing user interfaces. It is powerful and flexible, and offers pre-built model constructs as well as a Java environment for custom coding. It approaches software and model development from an object-oriented perspective and includes facilities for implementing models based on UML conventions such as statecharts, inheritance, and transition diagrams (Borshchev, 2013). AnyLogic has been used in a variety of ABMs and has achieved industry-wide acceptance as a robust, flexible tool. The professional version of AnyLogic version 7.1 was used for this project.

3.4 Wiki Letters

As stated previously, the model was created using an ABM discrete space approach (Mustafee and Bischoff, 2013, p. 479). Agents were graphically displayed in a variable-sized grid with each cell representing a particular letter. Figure 5 provides the initial grid appearance. Letters in each cell were derived from a function based on expected distributions in English language writing. The distributions used are publically available on the data-compression website (Data-compression, 2015). Figure 6 shows a portion of the custom Java code used in the function as represented in AnyLogic. Figure 7 provides the custom distribution used to drive the GetLetter function for selecting a letter for display.

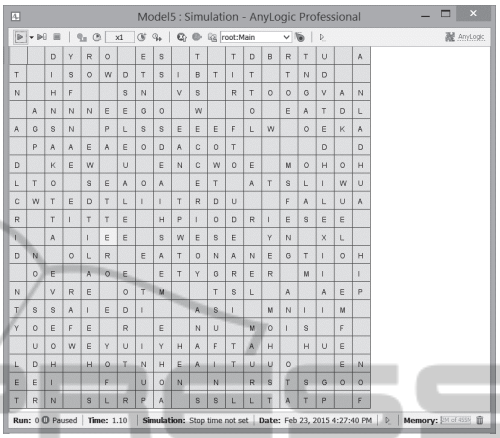


Figure 5: Initial discrete space grid for the Wiki Page ABM.

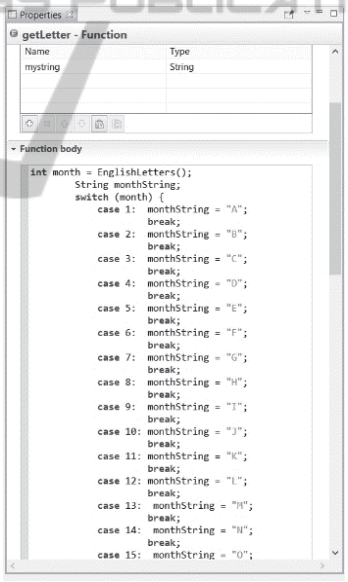


Figure 6: GetLetter function used to distribute letters with frequencies expected in typical English.

Although the letters in discrete space were not arranged according to natural words, using a representation of letters found in typical writing, gave the visual display a sense of realism and provided a more interesting user interface.

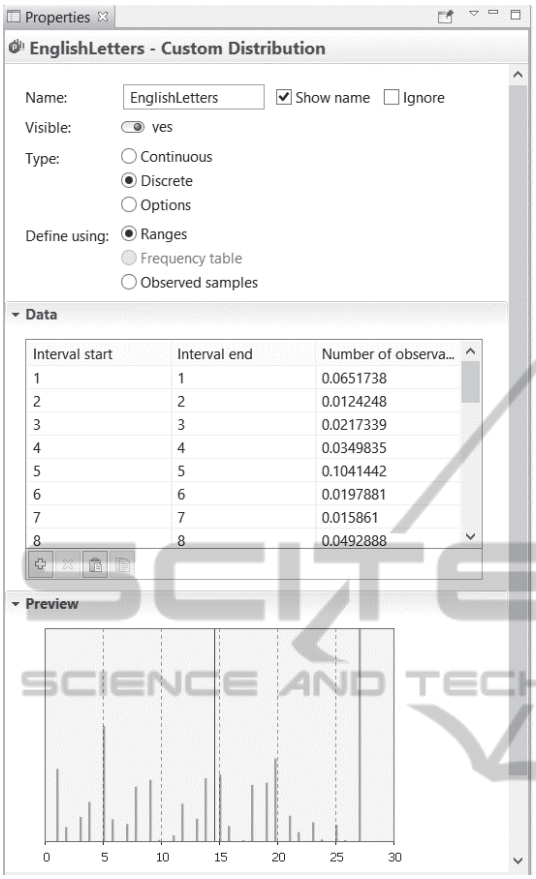


Figure 7: Custom AnyLogic distribution for typical English letter occurrence in writing.

3.5 Agent Interaction

Interaction between the wikiletter agents and the editor agent was accomplished using a combination of state charts and messages. The wikiletter agents resided in a variety of states including Fresh (just entered by an editor), ReFresh (newly changed), Approved (stable and part of the wikipage), Obsolete (marked for change or deletion by the editor), and Gone (temporarily blank). The state diagram indicated the various states as shown in Figure 8. The letters transitioned between the states based on messages sent by the editor agent. Figure 9 shows the editor agent state diagram with two states: working or resting. The transition times between working and resting were derived from data gathered with the pre-processor. The number of letters to be changed also came from that source.

The editor agent sent messages which told letters to move to the next state in their transition diagram. This essentially drove the model and the visual display.

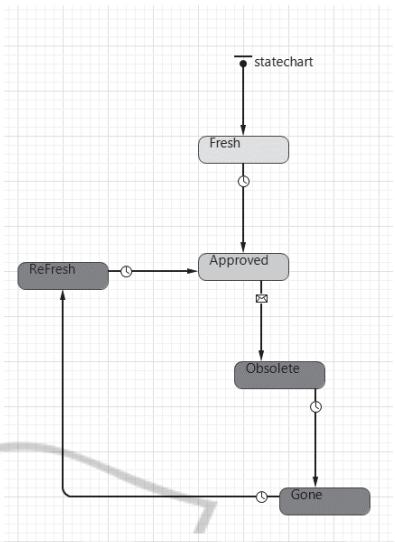


Figure 8: Wikiletter agent state diagram.

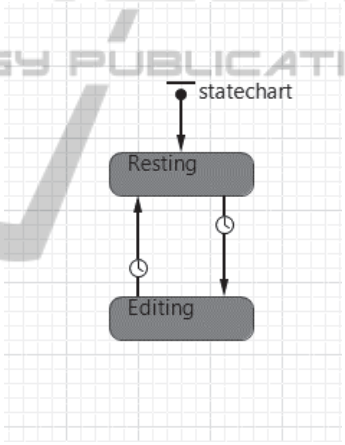


Figure 9: Editor agent states.

3.6 Model Execution and Visualization

Currently the model is in its preliminary stages of completion. A basic set of functionality has been developed and is in place but the final user dashboard has not been constructed. This means that letter change frequencies and magnitudes must be manually entered. Currently, the model provides a visual display that indicates the number of letters being changed and the current state of the wiki page. Figure 10 provides a view of small page with 400 characters as an illustrative example. The final user interface will permit easy changes to: user-specified input distributions, wiki page sizes, editing frequencies, run lengths, and output characteristics. These changes will comprise Phase 2 of this project.

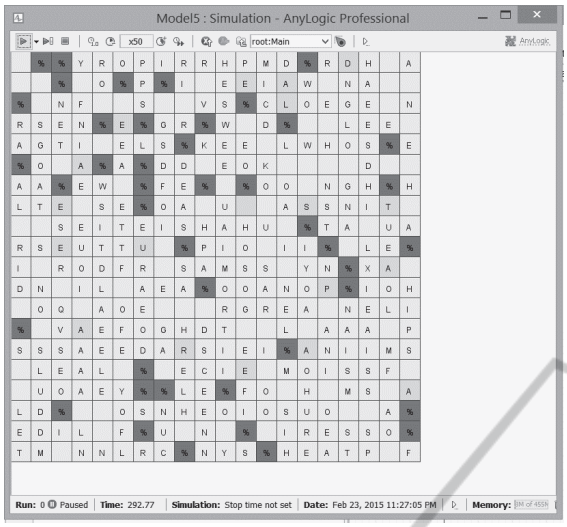


Figure 10: Model during execution. Red letters are currently available for replacement, yellow letters are obsolete but not yet deleted and pink letters are newly added.

4 DISCUSSION

The purpose of the Wiki Page ABM was to provide a tool for wiki keepers and organizational information specialists whereby their time commitments could be better understood and managed. As described in early sections of this article, wikis are widely used and have become a useful technology that provide both advantages and challenges. Among the greatest advantages is that wikis can become long-term information repositories developed and maintained through collaborative efforts. Challenges, on the other hand, require developing approaches and policies to ensure quality, consistency, and responses to intentional and unintentional changes that may not be aligned with wiki goals. The current project mitigated these challenges through providing a better way to visualize and understand edits to the text of wiki pages. The model also provided a better way to anticipate wiki changes and determine human resource requirements needed to ensure wiki quality.

Our preliminary development led us to believe that an agent-based approach was useful in this domain. The constructed, preliminary ABM provided an interesting and useful way to longitudinally examine changes made to wikis based on the reality of its history. The Wiki-Hist-Heist pre-processor mined data regarding events that occurred on a specific wiki page and facilitated creation of letter change frequency and magnitude distributions. These distributions provided a natural way to drive the model.

5 LIMITATIONS AND FUTURE

Although the Wiki Page ABM is in its preliminary stages of development (See Figure 11), we believe it offers much promise. The pre-processor written in C# is flexible and permits data collection to be customized according to a variety of specifications. We anticipate using it in two ways. One is to permit analysis of a specific wiki page. This means that distributions unique to a page of interest can be created and used to determine human oversight requirements over time. A second use is to analyse a series of wiki pages to arrive at a more universal set of change frequency and magnitude distributions. This information would provide general wiki staffing requirement information and permit long term studies to understand the impact of various strategies and policies on managing a wiki site.

Already, use of the model and pre-processor has provided insight into patterns of changes to wikis. We have experimented with fine-tuning our results with additional filtering features and by looking more deeply at history page data. For instance, it might be possible to predict future change activity based on past changes to talk pages or page views.

Another planned change to the Wiki Page ABM includes the addition of page controller, vandal, wiki keeper, and legitimate user agents. Currently the model utilizes letter agents and an editor agent which help simulate words within the wiki page. To make the visual interface more realistic and interesting, we plan implement changes to contiguous groups of letter agents. This will bring a greater sense of realism to the model viewers through improved visualization. We also plan to make the view screen resemble a wiki page rather than a grid. Currently, the model operates with random letters being changed by the editor agent but they are spread throughout the grid. The desired pattern of notification would be to represent letters as a contiguous string that may start on one row and continue on the next. AnyLogic offers built in functionality to make agent communication easy but these functions do not provide a default method for a partially contiguous group of agent to ‘talk’. A solution has been devised and involves adding a page controller agent which manages the letter agents according to ID parameters. This change is mostly cosmetic so it has been relegated to Phase 2 of the project. Other planned enhancements to the model in Phase 2 include adding a user interface with menu items that make it easily specify model input values

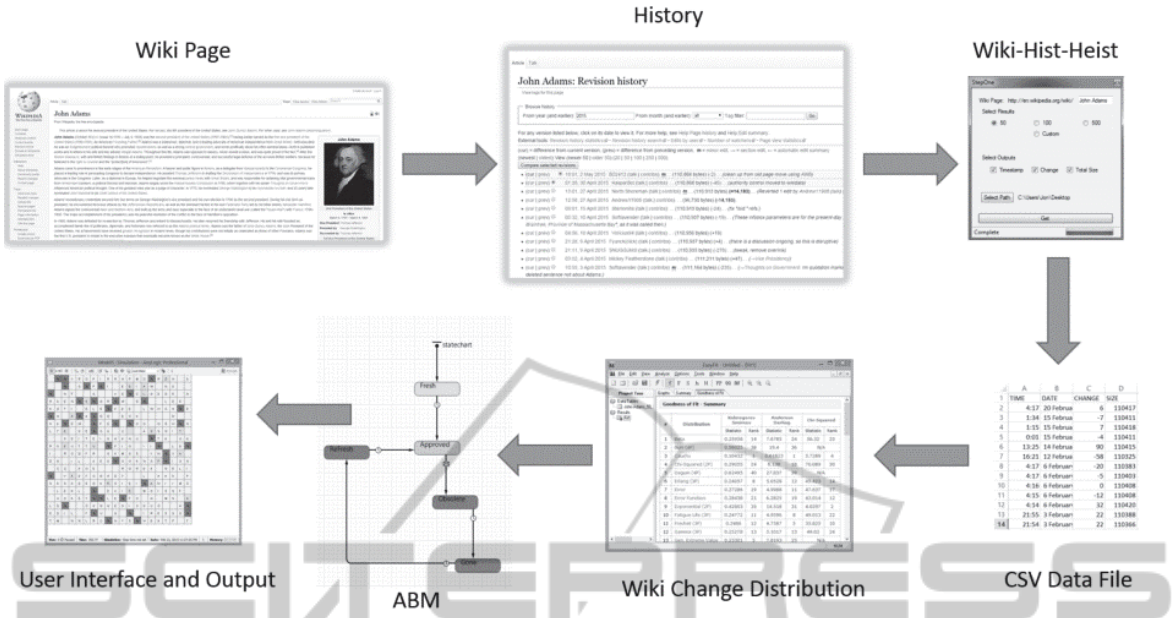


Figure 11: Overall flow of model in Phase 1.

and distributions, as well as provide custom run times and replication counts. An output report will be formatted and created to make tabulating results easier and more accessible. Extensive validation activities are also planned. Comparisons to prototype discrete event simulation (DES) and system dynamics (SD) models will also be provided (Chan et al., 2010; Tako and Robinson, 2009).

6 CONCLUSIONS

This paper has provided information related to the preliminary development of Wiki Page ABM. We used a discrete space approach to structure the model comprised of agents representing letters in the words on a wiki page. The agents moved through states representing whether the letters were changed, deleted, or added based on rates derived from wiki page histories. We discussed our custom developed C# pre-processor, called Wiki-Hist-Heist, which pulls information from wiki history pages to facilitate derivation of change frequency and magnitude distributions. These distributions provide model inputs based on a compilation of past events. The initial version of the Wiki Page ABM was built using AnyLogic 7.1 Professional. It provided a framework with user-friendly features. Overall, the initial stages of the project have been beneficial and we plan to continue adding enhancements that make the model useful to wiki keepers and information managers that need to staff and understand the behaviour of their wiki sites better. Limitations of the current preliminary implementation and ideas for future study were also described.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the help of Matthew McHaney with his insight into Java programming and his idea to create a wiki page controller agent to be included in Phase 2 of this project.

REFERENCES

Borshchev, A., 2013. *The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6*. AnyLogic North America.

Bruch, E., Atwell, J., 2013. "Agent-based models in empirical social research." *Sociological Methods & Research*. 1-36.

Chan, W., Son, Y., Macal, C., 2010. Agent-based simulation tutorial-simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation. In *Proceedings of the Winter Simulation Conference*. 135-150.

Data-Compression, 2012. "First Order Statistics for Distribution Values for Letters." Retrieved from http://www.data-compression.com/english.html.

García, G. P., 2012. *Improving Creation, Maintenance and Contribution in Wikis with Domain Specific Languages*. Diss. University of the Basque Country.

Hai-Jew, S., McHaney, R., 2010. ELATEwiki: Evolving an E-Learning Faculty Wiki. In Cases on Digital Technologies in *Higher Education: Issues and Challenges* (Rocci Luppicini and A. K. Haghi, eds.). 1-23.

Heath, B., Hill, R., Ciarallo, F., 2009. A survey of agent-based modeling practices (January 1998 to July 2008). *Journal of Artificial Societies and Social Simulation*, *12*(4), 9.

Helbing D., Balietti S., 2011. How to do agent-based simulations in the future: from modeling social mechanisms to emergent phenomena and interactive systems design, in *Technical Report 11-06-024*. Santa Fe, NM. Santa Fe Institute.

Hepp, M., Siorpaes, K., Bachlechner, D., 2007. Harvesting Wiki Consensus: Using Wikipedia Entries for Knowledge Management. Special issue on Semantic Knowledge Management, *IEEE Internet Computing*. 54–65.

Macal, C., North, M., 2010. Tutorial on agent-based modelling and simulation. *Journal of simulation*, *4*(3), 151-162.

Mader, S., 2008. *Wikipatterns: A practical guide to improving productivity and collaboration in your organization,* Wiley Pub., Indianapolis, IN.

Meatballwiki, 2011. *Softsecurity*, Retrieved from http://meatballwiki.org/wiki/SoftSecurity.

McHaney, R., 2012. "The Web 2.0 Mandate for a Transition from Webmaster to Wiki Master." *Open-Source Technologies for Maximizing the Creation, Deployment, and Use of Digital Resources and Information*. IGI Global. 193-218.

McHaney, R, Spire, L., Boggs, R., (2014). "E-LearningFacultyModules.org." *Packaging Digital Information for Enhanced Learning and Analysis: Data Visualization, Spatialization, and Multidimensionality: Data Visualization, Spatialization, and Multidimensionality*. IGI Global. 103-119.

Mustafee, N., Bischoff, E., 2013. Analysing trade-offs in container loading: combining load plan construction heuristics with agent based simulation. *International Transactions in Operational Research*, *20*(4), 471-491.

Sutton, A. (2006, September). Stop Using Wikis as Documentation. Symphonious. Retrieved from http://www.symphonious.net/2006/09/02/stop-using-wikis-as-documentation/

Tako, A., Robinson, S., 2009. Comparing discrete-event simulation and system dynamics: users' perceptions. *Journal of the Operational Research Society. 60.* 296-313.

Taylor, S. (Ed.), 2014. *Agent-based Modeling and Simulation*. Palgrave Macmillan.