Interactive Concept of Operations Narrative Simulators

Andre R. Denham[1]
The University of Alabama, Tuscaloosa, AL, 35487

This paper reports on an exploratory design and development project. Specifically this paper discusses the design and development of Interactive Concept of Operations Narrative Simulators (ICONS) as a means of enhancing the functionality of traditional Concept of Operations documents by leveraging the affordances provided by applications commonly used within the Interactive Fiction literary genre. Recommendations for an ICONS design and development methodology, along a detailed description of a practical proof-of-concept ICONS created using this approach are discussed. The report concludes with a discussion of how ICONS can be extended to the K-12 mathematics education domain and conclude with a discussion of how ICONS can be used to assist those involved with strategic planning at Marshall Space Flight Center.

I. Introduction

Concept of Operations (ConOps) documents play a vital role in the design and development of the systems NASA uses to accomplish its mission. A ConOps document provides stakeholders with a detailed description of a proposed system by explicating its attributes from an operational point of view. This quantitative and qualitative breakdown of a system's characteristics provides the reader with insight into the system's purpose and intended use. The process of creating a ConOps is important as it provides an opportunity for stakeholders to identify and specify requirements, determine design functions, and pinpoint potential user interface issues well in advance of the development of the system. The exercise of creating a ConOps also provides an opportunity to create system definition documentation and conduct a formative assessment to identify additional stakeholders who may have been overlooked initially, but through subsequent analyses deemed vital to the design, development, and implementation of the system.

While there are a variety of formats for a ConOps, NASA documentation requirements specify that a minimum, a ConOps should include the following[1]:

1) Operational goals from the viewpoint of the all stakeholders.
2) Overview of the System of interest, including supporting systems.
3) Intended use of the system during all life-cycle phases of the program/project, including but not limited to:
    a. Manufacturing and assembly
    b. Integration and test.
    c. Transportation and storage.
    d. Ground operations/launch integration.
    e. Launch Operations – launch, deployment, on-orbit checkout.
    f. Maintenance and disposal.
4) Operational timelines
5) Command and data architecture.
6) End-to-end communication strategy.
7) Integrated logistic support (resupply, maintenance, assembly).
8) Operational facilities.
9) Contingency and off-nominal operations.

[1] Assistant Professor of Instructional Technology, Department of Educational Leadership, Policy and Technology

ConOps are usually developed through a series of brainstorming sessions. These brainstorming session are held in a traditional manner, where a facilitator leads out in the sharing of ideas out loud and a scribe is assigned to capture all that is discussed. This approach to ConOps development lends itself to being dominated by vocal individuals and the pace of discussions can make it challenging for the assigned scribe to capture all of the concepts and ideas put forth in the session. Furthermore, those who were not present during the ConOps development phase will have to envision the system being described in the document, which may lead to misinterpretations of the system and the glossing over of vital information. Increasing the interactivity of ConOps may serve as a means of amplifying the functionality of these important strategic documents.

The overarching goal of the summer faculty fellowship being reported on in this paper was to develop a means of reimaging ConOps by creating a new form of these documents called Interactive Concept of Operations Narrative Simulator (ICONS). The purpose of ICONS is to widening the interactivity of ConOps beyond static text and images as a means of encouraging active reading. This would be accomplished by leveraging the affordances of Inform 7, a design system used for the development of interactive fiction via natural language coding, Twine, an open-source application for the development and delivery of interactive nonlinear stories, existing NASA and web-based resources, and design guidelines gleaned from a variety of fields. This report will discuss the catalyst for this project, the design and development of a proof-of-concept narrative simulation based on existing International Space Station procedures, a proof-of-concept application of ICONS for mathematics word problems, propose a workflow process for developing ICONS, and provide recommendations for future work which will also include a discussion of the broader impacts of ICONS inside and outside of NASA.

II. Background

The genesis of ICONS stemmed from one of the project collaborator's experience in the Space 2100 project. During the ConOps development process for the "Massless Exploration Space 2100 Sprint", the collaborator felt ConOps were being developed inefficiently. Development was spurred by brainstorming sessions in which one individual served as the facilitator, and one to two other persons served as scribes. The remaining attendees were tasked with presenting ideas to the large group for vetting and subsequent inclusion within the ConOps. This resulted in brainstorming sessions where a few individuals dominated the conversation and the scribe(s) struggled with the task of capturing all of the concepts discussed during each session.

The collaborator also realized that technological advances provided an opportunity to enhance ConOps. While the current text based format of ConOps have proven to be effective, going beyond static text and images could potentially enhance their functionality and readability. What was needed was the identification of technology or technologies that would allow for the leveraging of existing digital and web-based resources. In the attempt to pinpoint the tool(s) necessary to create the next generation of ConOps, it became apparent there was a greater need to focus on increasing the interactivity of the scenarios presented within ConOps as a means of encouraging active reading.

Schilit, Golovchinsky, and Price[2] define active reading as not simply the act of look at words on paper, but a combining of critical thinking, reading, and learning which are all crucial elements of education and knowledge work. Victor[3] defines an active reader as someone who:

> ask questions, considers alternatives, questions assumptions, and even questions the trustworthiness of the author. An active reader tries to generalize specific examples, and devise specific examples for generalities. An active reader doesn't passively sponge up information, but uses the author's argument as a springboard for critical thought and deep understanding.

Typical ConOps do not explicitly or implicitly support active reading. ConOps simply presents information and places the responsibility on the reader to decide whether they will passively or actively interact with the text. Within the field of literacy, the technological approach commonly used to increase the likelihood of active reading tend to center on providing readers with digital documents and encouraging them to use the annotation tools (highlighters, under liner, note taking, comment features, stylus, etc.) embedded within the application used to display the text[2]. While this approach has proven successful, it still places the tools for active reading extrinsic to the digital document and places the onus on the reader to choose whether to use them or not. What is needed is a technology that can be intrinsically integrated within a ConOps that implicitly requires the reader to actively interact with the text. The literary genre of interactive fiction and the recent technological advances within this community provides a potential means for intrinsically supporting active reading of ConOps.

A. Interactive Fiction

Interactive fiction (IF), also known as text adventures or text games, is a dialog system contained within a text-based computer simulation[4]. IF eschews the point-and-click and/or controller user interfaces commonly found in

most digital games and instead requires the user to employ text based, conversational commands to control the main character(s) and interact with the simulated environment. Similar to choose-your-own adventure books, the narratives within a piece of IF are non-linear and frequently contain puzzles, quests, and games for the player to navigate in order to advance the story line. The reader/player is afforded the opportunity to make decisions throughout the gameplay that will determine the outcome of the story. Modern IF differs from choose-your-own adventure books in that they can go beyond text, and also include multimedia and external web resources.

Modern IF development tools democratized the process of creating IF. These IF development tools allow for the design of interactive stories through the creation of integrated design environments that allowed authors to create "worlds" where players can use simple text commands such as "open door" to interact with the environment and control the narrative. Along with the tools needed to release either an executable application, or more commonly, provide an option to publish directly to HTML, IF so authors can share their works online. This has led to a surge in the size and productivity of the online IF community.

B. Project Goals

The availability of free, robust IF design and development applications, along with the crowd sourced resources readily available online, provides a compelling tool for supporting active reading of ConOps. For instances, creating ICONS within ConOps where readers are required to control the steps take within a simulation will require them to focus their attention, and avoid the temptation to gloss over what might be vital information.

This project sought to conduct exploratory design and development of the ICONS concept by creating the following end products:

1) A proof-of-concept narrative simulation based on existing International Space Station procedures
2) A proof-of-concept application of ICONS for word problems
3) A workflow process for developing ICONS

III. Proof-of-Concept ICONS

This portion of the report will focus on the design and development of a proof-of-concept narrative simulation based on an existing International Space Station (ISS) laboratory procedure. This discussion of the proof-of-concept ICONS will begin with an overview of Inform 7 and Twine, the IF development tools used throughout this project, and their role within the ICONS design and development process. Following that will be detailed presentation of a proof-of-concept ICONS based on a heart cell experiment which took place on the ISS. This section will conclude with additional examples of application of ICONS within the domain of K-12 mathematics education.

A. Inform 7

Inform is an integrated development environment for interactive fiction, created by Graham Nelson in 1993. The release of Inform served as a catalyst for a substantial increase in the number of people creating IF, by allowing authors to easily create story files, which are also known as Z-Code, from the source code[5]. The resulting story files can then be executed within any Z-Code interpreter, which are programs that are designed to run the Z-Code virtual machine. There are a large variety of IF platforms, so the major breakthrough for Inform was providing a means for a Z-code file to be executable on any IF platform without the need for adjustments to the source code for each platform. Inform can also handle the interpretation of source code for the Glulx virtual machine created by Andrew Plotkin to address limitation issues commonly found in older Z-virtual machines[5].

Inform 7 is the latest version of this IF design and development tool and is a free application available for use on Mac OS, Windows, and Linux. Inform 7 uses an object-oriented, procedural programming language that is adept at handling inferences between object types and their properties based on their use. For example, the statement "Sarah has a purse" creates a character named Sarah because Inform understands that only people are capable of carrying things. This statement also creates a thing called a purse, and assigns the purse as having the property of being portable, as objects that can be carried can be transported by characters. Inform 7 allows authors to manage the relationships between objects, along with the ability for authors to create their own relationships, such as jealousy or anger. This programming language is used to create story files, which are interpreted by the Inform compiler. Inform 7 also contains the Inform Library, which is used to parse the commands input by players through the text-based interface and handle the subsequent updating of the game environment. The source code for Inform 7 is distinctly readable, which lessens the slope of the learning curve for those who are new to the application. Below is an example of Inform 7 source code taken from the Inform 7 Recipe Book[6]:

"The Power of the Keys"
Afterlife is a room. "Fluffy white clouds gather round you here in the afterlife." The Pearly Gates are a door in Afterlife. "The Pearly Gates - large, white, wrought-iron and splendidly monumental - stand above you." Heaven is a room. The Gates are above the Afterlife and below Heaven.

St Peter is a man in the Afterlife. "St Peter, cheery if absent-minded, studies his celestial clipboard."

Before going through the Pearly Gates:
say "St Peter coughs disarmingly. 'If you'd read your Bible,' he says, 'you might recall Revelation 21:21 saying that the twelve gates were twelve pearls, each gate being made from a single pearl. I really don't know why people keep imagining it like the entrance to some sort of public park - oh, well. In you go.'";
end the story.

Test me with "enter gates".

An example of the Inform 7 integrated development environment can be found in the figure below.

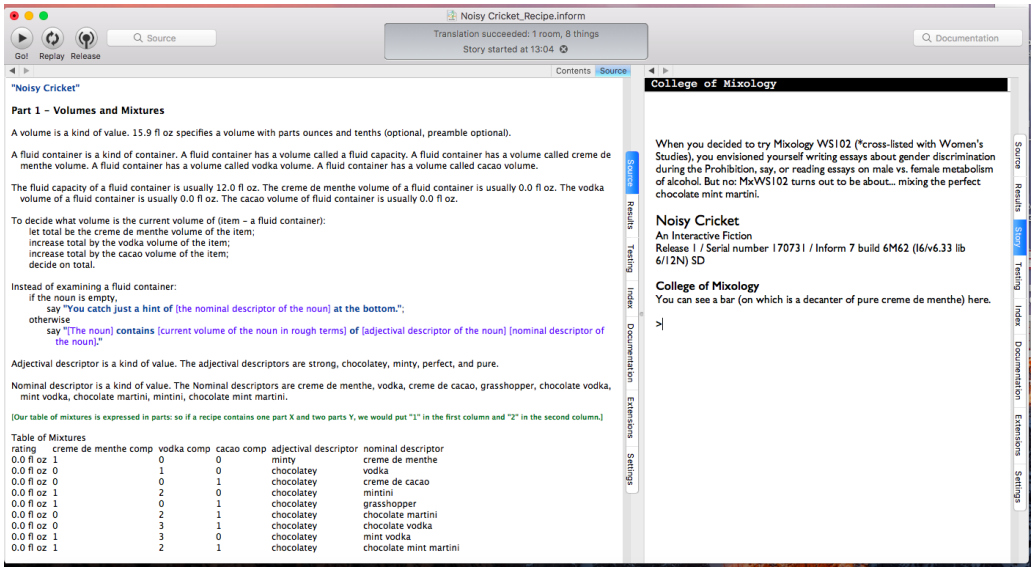

Figure 1. Inform 7 Integrated Development Environment.

B. Twine

Twine is an open-source web-based application for developing IF. Twine can also be downloaded for Mac OS, Windows, and Linux as a stand-along application, and is primarily used to develop hypertext IF and games. Twine differs from Inform 7 in that the development environment is visual rather than text-based. This negates the need for authors to learn a programming language. If you do have programming experience, Twine does allow for the extension of its capabilities through the inclusion of variables, conditional logic, images, CSS, and JavaScript[7].

Twine supports IF where the narrative is created through branching. For example one might start a Twine story by creating a passage of text that says, “You’ve parked your car and walked into the mall. You need to buy a dress for your cousin’s wedding in a few weeks and don’t know where to start”. At that point you would need to provide the reader with options on what to do next. Within the passage you would then also include the following text at the bottom of the passage:

[[Go into the formal wear section of Belk]]
[[Find a salesperson in Belk to help you find a dress]]
[[Check bank account to see how much money you have to spend on the dress]]

The brackets tell Twine to create three new passages for the story, with each branch being connected to the original passages (see Figure 2).

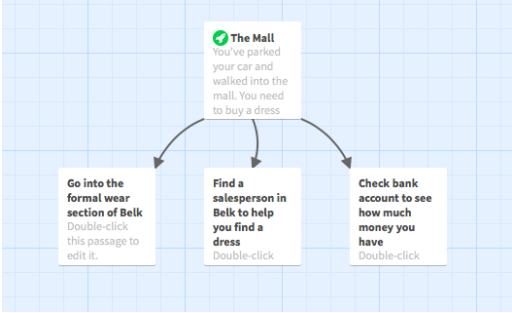

Figure 2. Twine Interactive Fiction Development Environment.

From there you can continue branching the story out by adding additional parts of the narrative to each of the three new passages, while making sure to provide the reader/player with choices within each passage. Twine publishes the story to an HTML file, which allows for the sharing of IF and a simple hypertext interface for the reader/player to control the unfolding of the narrative (see Figure 3). Twine differs from Inform 7 in that many of the additional layers of interactivity that can be added to a Twine story through the use of JavaScript, CSS, images, variables, and conditional logic are already native to Inform 7. In other words, Twine requires the utilization of additional applications to create elements of interactivity that are already included within Inform 7.

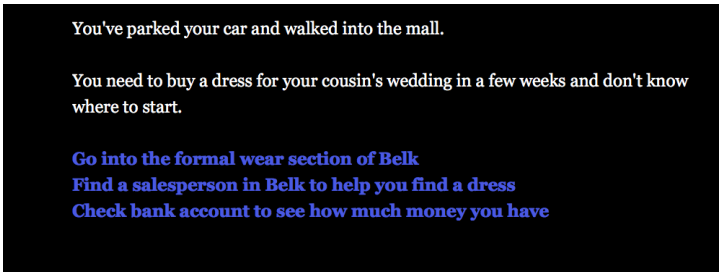

Figure 3. Example Twine Story.

C. Development Process

A major goal for this project was to propose a process for developing ICONS. This process needed to be replicable and provide guidance to those who may be interested in developing ICONS. Using the IF development tools Twine and Inform 7, the following development process was created:

1. Rapid prototype the ICONS using Twine
2. Develop the ICONS within Inform 7
3. Evaluate and Implement the ICONS

1. Rapid Prototype the ICONS

During the first stage of ICONS design and development, the goal is to begin the process of transforming a traditional ConOps scenario into an ICONS. It is recommended that Twine be used to accomplish this as it facilitates rapid prototyping of scenarios, due to its ease of use. During the designing of an ICONS within Twine, the developer should focus not only on creating a simulation of a system, operation, and/or procedure, but the crafting of additional narratives. For example, these additional narratives could highlight steps in a procedure when an error can occur, or to show the interconnectedness of various parts of a system.

To help the development of additional narratives, it is recommended that ICONS developers make use of one of the common patterns in choice-based games based on the size of the scenario. For large-scale ICONS development, developers should make use of standard design patterns such as, but not limited to, Time Cave, Gauntlet, Branch and Bottleneck, Quest, and Loop and Grow[8]. The Time Cave design pattern is considered the oldest of the choice-based design patterns and contains many branches. The choices that players make throughout the game all carry equal weight and once a player begins going down a branch they will not be able to merge with another branch.

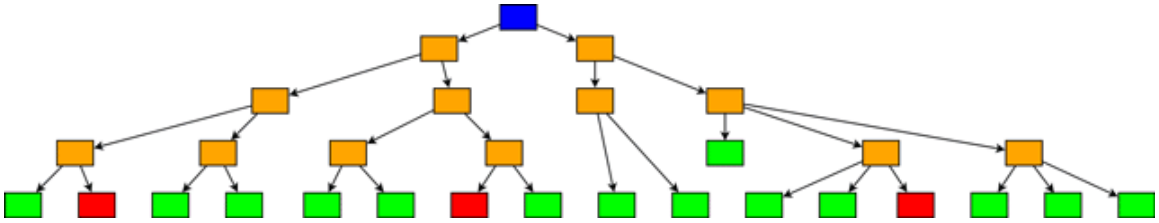

Figure 4. Time Cave Design Pattern.

While the Time Cave design pattern is broad, the Gauntlet design pattern is long. Gauntlets have a central linear thread, with the branches either containing, dead ends, opportunities to go back a step, or the ability to rejoin the central thread. This design pattern makes it quickly evident to players that they are on a prescribed path and the world they're interacting with is tightly constrained. That being said, the Gauntlet design pattern makes sure that readers/players must interact with all of the important content within a story. The same cannot be said for the Time Cave design pattern.

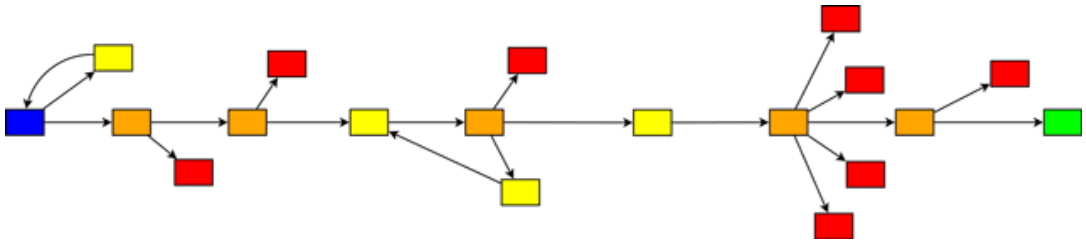

Figure 5. Gauntlet Design Pattern.

Within the branch and bottleneck design pattern, the story will frequently branch, but all branches will at some point rejoin. This rejoining of branches will take place around an element or event that makes sense for all versions of the narrative (which requires constant state-tracking). Branch and bottleneck is well suited for ICONS where the passage of time is important to the scenario.

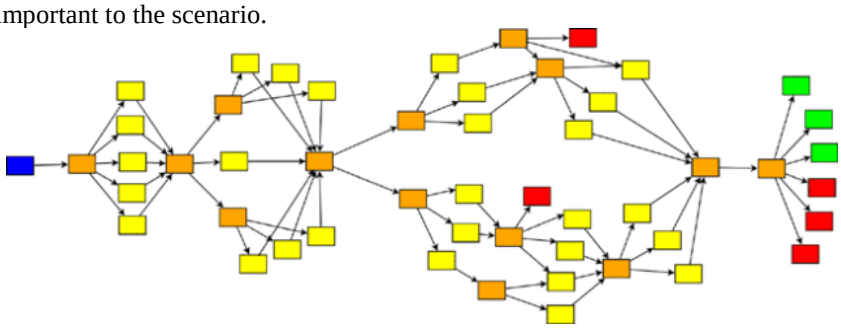

Figure 6. Branch and Bottleneck Design Pattern.

IF created using the quest design pattern contains branches that are unconnected but can rejoin at a few winning nodes. Quest design patterns are best suited for ICONS in which the goal is to support the exploration of a particular setting or where geography is of importance.

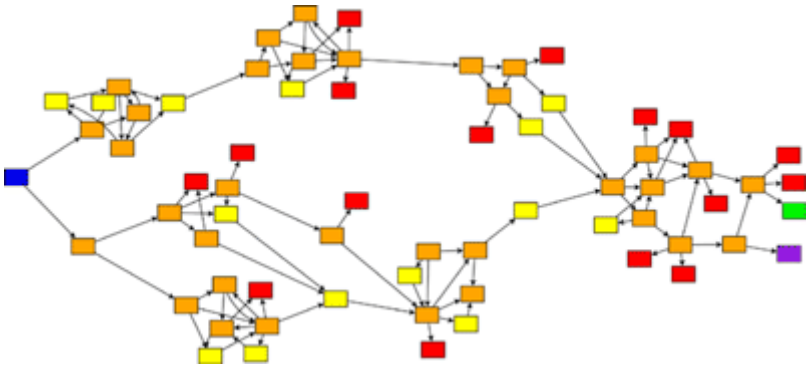

Figure 7. Quest Design Pattern.

Finally the Loop and Grow design pattern loops a central thread/narrative, but uses state-tracking to alter the options presented to the reader/player each turn around the loop. This design pattern is best suited for ICONS where there is a need to warrant the repetition of portions of a narrative. Loop and Grow is typically used within another design pattern when the reader/player is required to perform a routine task, travel through time, and/or complete a set of tasks to some level of abstraction[8].

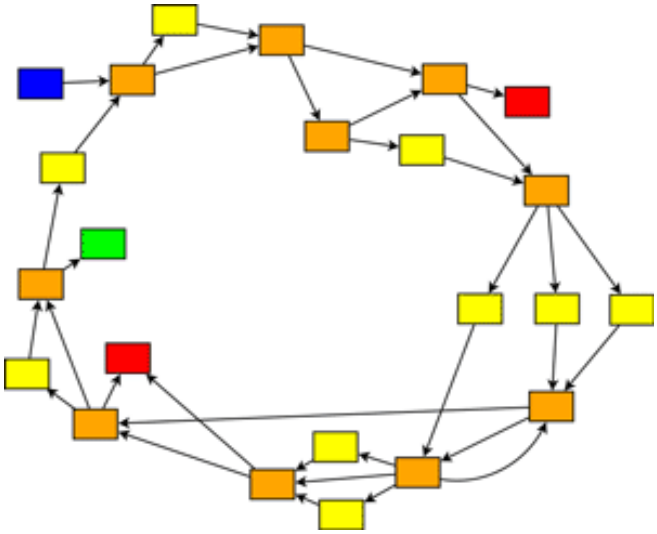

Figure 8. Loop and Grow Design Pattern.

For small-scale ICONS development, developers should make use of standard design patterns such as, but not limited to, Confirmation-required, Track Switching, Sorting Hat, and Endgame Time Cave[9]. The Confirmation-required design pattern allows readers/players to engage in increasingly risky behavior by making them opt in before each risky decision. Track switching is a variation of Confirmation-required where the player is allowed to reverse course after a series of decisions have been made. Sorting hat design patterns are usually applied within the first chapter of IF. A decision that a player makes at the beginning of the story places them one of several linear paths. Finally, the Endgame Time Cave design pattern is the opposite of the Sorting Hat and is used to conclude a game. Examples of each design pattern can be found below in their respective order.

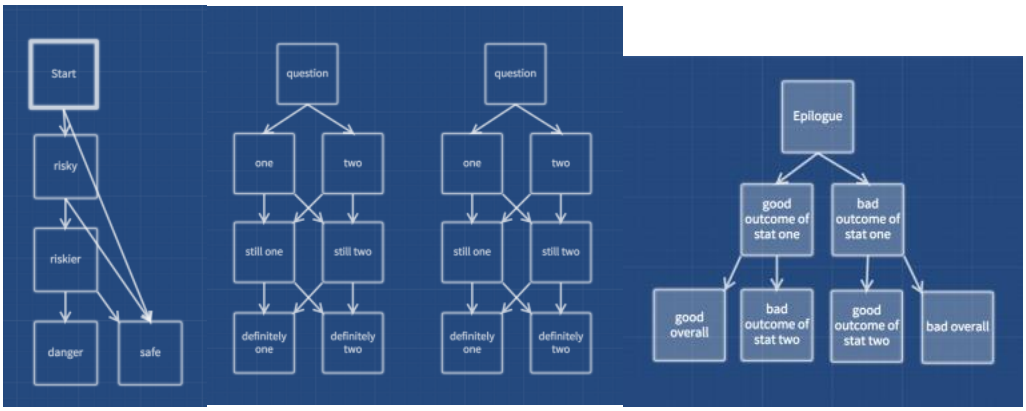

Figure 8. Small Scale Design Patterns.

2. Develop the ICONS within Inform 7

Once ICONS have been designed using Twine and one of the prescribed design patterns, development of the ICONS should commence in Inform 7. As stated earlier, Inform 7 provides access to a host of native features with the most important being the text parser. In addition, Twine allows for the inclusion of extensions. Extensions are source text shared by other IF authors which help to expedite the development process. An extension is supported by the original author, and typically contains documentation and examples of its intended use. Extensions are available that allow for the controlling of time, management of non-player characters, use of mathematical equations, manipulation of typography and layout, inclusion of multimedia, and the integration of physics to name a few.

Of all the extensions available to authors, the ability to include multimedia was the one that was most important to this project. It is recommended that the Vorple extension[10] developed by Juhana Leinonen, be leveraged as a means of adding depth to ICONS. Vorple allows authors to go beyond the Z-virtual machine to allow the Inform source code to communicate with the web browser where the story is being executed. Through the use of a virtual file system, Vorple allows for the execution of JavaScript, the use of hyperlinks, the inclusion of images and sounds, modification of the notification and scoring system, and control of the command prompt. This allows for the inclusion of existing NASA web-based resources within ICONS, as well as the inclusion of other resources such as, but not limited to, embeddable videos and web pages.

Using the Twine design document as a guide, the development of the ICONS should be straightforward. It is also recommended that someone with a literary background be included in the development team to assist with the fleshing out of the various narratives/storylines. This will to make it more likely that the ICONS being developed is not only technically correct, but that it is also an engaging piece of IF.

3. Evaluate and Implement the ICONS

Once ICONS have been developed, it is imperative that it be evaluated. The first evaluations should focus on ascertaining if the ICONS is engaging and immersive, and to identify any bugs or glitches in the story/text game. This evaluation should be conducted by a small group (5-7) of stakeholders and people who are familiar with project. It is recommended that focus groups and/or talk-a-louds where the reader/player goes through the ICONS with a member of the development team present and provides on the spot evaluation of the story. ICONS developers should remain in this phase and make iterative refinements to the source code until the majority of issues have been addressed. Once the first phase of evaluation is completed, the ICONS should be released to the entire group of stakeholders. This evaluation phase should focus on making sure that the ICONS is technically correct and properly meets a pre-established criteria for simulating a scenario, system, procedure, or mission. Once again, iterative refinements informed by the feedback received from the larger body of stakeholders should be taking place during this phase of evaluation. Once stakeholders have signed off on the ICONS as being engaging, immersive, technically sound, and in fidelity to the intended system, scenario, and/or mission it represents, the development team is now free to release the ICONS for open use. It is recommended that at this point the development team seek feedback from those outside the stakeholder community in order to acquire additional insights into the effectiveness of the ICONS and to improve their skills in designing and developing future ICONS.

Heart Experiment International Space Station ICONS

In order to validate the aforementioned design process, a proof-of-concept ICONS was developed. This ICONS was developed using the procedures from an experiment that was performed on the ISS. The focus of this study,

entitled Effects of Microgravity of Stem Cell-Derived Heart Cells (Heart Cells), was to supplement our "understanding of microgravity on heart function, the improvement of heart disease modeling capabilities, and the development of appropriate methods for cell therapy for people with heart disease on Earth"[11]. It was beyond the scope of this project to develop an ICONS for this entire study, so only one phase of the study, Heart Cells Multiwell BioCell Insert into Microscope, was used to create an ICONS.

Twine was used to design this ICONS using a combination of the Confirmation-Choice, Track Switching, and Gauntlet design patterns. At various points throughout the narrative readers/players were provided with the ability to make choices that would end the story or continue moving them along. For example, once the BioCell has been removed from its habitat, it must be allowed to cool for twenty minutes before proceeding. If the player does not wait for the allotted time frame before proceeding, the game will end and provide text identifying the mistake and how that decision negatively impacted the experiment. Figure 10 provides a portion of the ICONS designed using Twine.

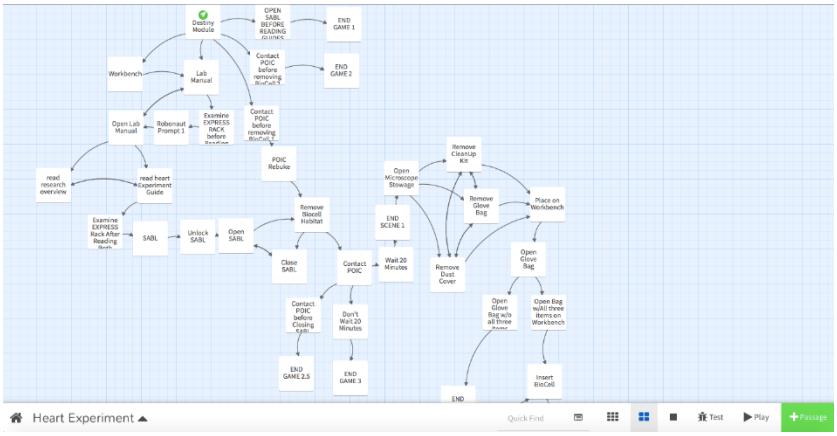

Figure 10. Twine Heart Cell Design.

Using the ICONS designed in Twine as a guide, the ICONS was fully developed within Inform 7. The suite of Vorple extensions were included within the ICONS to enhance the overall presentation and as an attempt to increase the fidelity of the procedure being simulated to the actually experiment taking place on the ISS. Images of various steps of the procedure were included along with images of the ISS, and a scoring system that added or subtracted points based on the decisions that the reader/players makes.

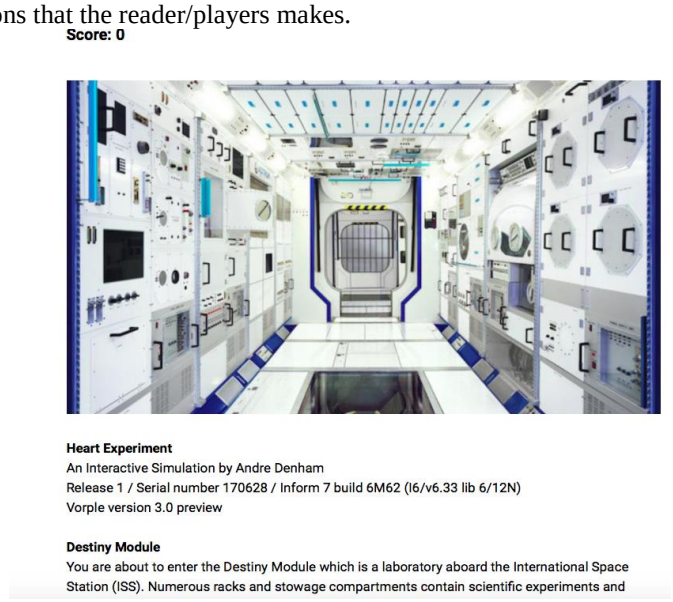

Figure 11. Heart Cell ICONS Proof-of-Concept using Inform 7.

D. K-12 Application of ICONS

Due to a series of reorganizations and subsequent schedule demands placed upon the teams that provided the Heart Cell experiment procedures, evaluation of the proof-of-concept ICONS was left to the discretion of the development team. This serendipitous turn of events enabled the project team to focus their attention on exploring, designing, and developing potential K-12 application of ICONS within the domain of mathematics. In order to properly scope the work, it was decided to focus on word problems. This idea of leveraging ICONS to assist in the creation of interactive narrative word problems for mathematics education is supported by empirical research. Gunbas[12] conducted a study found that found students who were asked to solve mathematics word problems which were embedded within the narrative of a computer-based story, significantly outperformed students who were asked to solve the same problems within a paper-based story or in the text based word problem format typically found in mathematics textbooks and assessments.

Using the design process described earlier, several small-scale text games were created to demonstrate the applicability of ICONS to the design and development of narrative based mathematics word problems. For example, an interactive word problem was created in which a reader/player entered a room in which they found a safe. Within the safe was a key that was needed to open another door within the game. In order to find the six-digit combination to the safe, the player had to find the area and circumference of three circles (rounded to the nearest whole number) that were located within the room. Another scenario had the reader/player in possession of two dice and a notebook. The reader/player was provided with instructions to roll the dice and record the sum of the resulting dice roll in their notebook. This information would then later be used to ascertain the probability of getting a particular sum on a roll of two die. The finally scenario had players sitting at a table with a deck of cards on their left hand side and a discard container on their right. The reader/player could only have four cards in their hand at a particular time. The goal of the game was to collect four of a kind, with the reader/player being allowed to discard of any card in their hand, but not be allowed to pick that card up again. The collecting and discarding of cards continued until the player had exhausted all the cards in the deck or until they had four of a kind in their possession. The challenge was that the cards were not typical playing cards, but had various representations of ratios (fraction, decimal, percent) on them.

IV. Potential Follow-On Activities

The ICONS project developed as a result of this MSFC Summer 2017 Faculty Fellowship is not intended to be a one off. Instead the time spent working on ICONS is intended to serve as the foundation for the transformation of the manner in which ConOps are designed and developed. While writing this report, the author participated in a scenario based planning meeting conducted within the MSFC Office of Strategy. Scenario planning, defined as "a disciplined method for imagining possible futures that companies have applied to a great range of issues"[13], is a tool used by those within the strategic planning community to help avoid the two most common errors committed when conducting strategic planning – overconfidence and tunnel vision.

When creating scenarios for strategic planning, it is recommended that each scenario contain the following building blocks[13]:

1. Drivers of change: enable identification of trends and uncertainties
2. Rules of interaction: Define the relationships among trends and uncertainties and enable the development of multiple scenarios.

In addition to the building blocks of scenarios necessary for strategic planning, it is also recommended that a framework be used to develop scenarios where two uncertainties are used as vertical and horizontal axes. The resulting four quadrants of uncertainty can then be used for narrative development[14]. Using these building blocks and scenario framework, it is not difficult to imagine the tools and processes used to design and develop the proof-of-concept ICONS being used in the development of scenario for strategic planning. Twine and Inform 7 could be used to create an ICONS that represents each of the four quadrants. To further illustrate this, let us use the two biggest uncertainties facing NASA or any enterprise of this size: supply and demand. By placing the supply and demand in terms of NASA's budget and compelling missions on two axes, one could create these four quadrants:

1. Bold budget and compelling mission
2. Shy budget and compelling mission
3. Bold budget and languid mission
4. Shy budge and languid mission:

In the first quadrant, the scenario could focus on the developing and deploying of space and planetary infrastructure without partners. A scenario representative of the second quadrant could be where space infrastructure delivery services are offered in order to encourage partnership development. The third quadrant could have a strategy where delivery services, propulsion systems, and operations infrastructure are offered. The fourth quadrant could have a

scenario where the focus is entirely on fostering industry and governmental partnerships, and the offering of technologies.

Using Twine to design a scenario will require strategic developers to avoid overconfidence and tunnel vision by forcing time to be spent thinking about trends and uncertainties that may arise within a given scenario. Twine could be used to create branching decision trees for each scenario, with the scenario framework being at the top level, space markets at the mid-level, and vignettes related to categories of products and services at the lower level. This branching tree would then be used to guide the development of specific scenarios within Inform 7 to describe a market as a room within the world, services as scenes, and products as objects.

Twine could also be used in to capture the results of strategy development sessions. For example, a strategy development session could be conducted where participants are divided into four groups and assigned one of four quadrants created from two axes of uncertainty. Each group would be tasked with using Twine (after a brief tutorial) to collectively create as many scenarios related to their assigned quadrant as possible within the allotted timeframe. The scenarios created during this strategic planning session could then be used to guide the development of an ICONS using Inform 7.

The work from the ICONS project can also be used to inform further research into the impact of interactive computer-based mathematics word problems. The scenarios developed during the summer could be further developed and used in empirical investigations into the viability of this approach to designing word problems. It might be that IF presents a novel means of designing and developing the next generation of formative and summative assessment for students by leveraging the theoretical benefits of anchored instruction.

The ICONS development process could also be used to create scenarios that connect the mathematics students learn in school to their practical application within common NASA systems. For instance, the Environmental Control and Life Support Systems (ECLSS) needed for space exploration have to be designed to support those manning the controls of space vehicles. These systems are responsible for maintaining oxygen levels and atmospheric pressure. Additionally, ECLSS is responsible for purifying the water astronauts need to survive, and manage the waste they generate. Finally, in the unfortunate event of a fire, ECLSS is responsible for providing fire detection and suppression. Using the ISS as a context for an ICONS and a failure in the ECLSS system as the compelling problem, one could craft a narrative where students are asked to solve problems that require them to apply their knowledge of algebra, physics, chemistry, and other STEM subjects.

Future projects could benefit from ICONS by simulating the usage of the system within the context of an ecosystem or market. Center Director Todd May outlined a strategy that includes enabling new markets through space exploration beyond Low Earth Orbit and focusing on Design, Development, Test, and Evaluation. Strategic scenario analysis could identify systems and vignettes to be implemented as ICONS. Through these ICONS, strategic analyst could simulate operations of future space infrastructure that enables future markets beyond Low Earth Orbit. Success in this endeavor will allow for the creation of the markets needed to support exploration of our galaxy and beyond. This will require going beyond traditional methodology for strategic planning to design and development processes that will facilitate the development of the next generation of ConOps. The work reported on in this paper provides a potential avenue for accomplishing this through the use of ICONS.

Acknowledgments

Andre R. Denham thanks Christopher Randall, Daniel O'Neil, the MSFC Office of Strategic Analysis and Communication, the MSFC Office of the Chief Technologist, the MSFC Office of Human Exploration and Development, and the Education Office of MSFC for the opportunity to participate in the 2017 MSFC Summer Faculty Fellowship Program.

References

[1]"Data Requirements Description: STD/SE-CONOPS" *Marshall Integrated Document Library* [online repository], URL: https://masterlist.msfc.nasa.gov/drm/drd_docs/SE-CONOPS.pdf [cited 1 August 2017].

[2]Schilit, B. N., Golovchinsky, G., and Price, M. N., "Beyond paper: supporting active reading with free form digital ink annotations," *SIGCHI conference on Human factors in computing systems*, 1998, pp. 249-256.

[3]Victor, B., "Explorable Explanations" *Worry Dream* [Blog posting], URL: http://worrydream.com/#!/ExplorableExplanations [cited 1 August 2017].

[4]Niesz, A. J., & Holland, N. N., "Interactive fiction," *Critical Inquiry*, Vol. 11, No. 1, 1984, pp. 110, 129.

[5]"Inform 7" [website], URL: https://en.wikipedia.org/wiki/Inform [cited 1 August 2017].

[6]"Preface", *Inform 7 Recipe Book*, [website], URL: http://inform7.com/learn/man/RB_1_1.html [cited 1 August 2017].

[7]"Twine", [website], URL: http://twinery.org [cited 1 August 2017].

[8]Ashwell, S. K., "Standard Patterns in Choice Based Games" *These Heterogeneous Tasks* [Blog posting], URL: https://heterogenoustasks.wordpress.com/2015/01/26/standard-patterns-in-choice-based-games/ [cited 1 August 2017].

[9]Short, E., "Small Scale Structures in CYOA" *Emily Short's Interactive Storytelling* [Blog posting], URL: https://emshort.blog/2016/11/05/small-scale-structures-in-cyoa/ [cited 1 August 2017].

[10]"Inform 7 Interactive Fiction with Vorple", [website], URL: http://vorple-if.com/doc/ [cited 1 August 2017].

[11]"Effects of Microgravity on Stem Cell-Derived Heart Cells (Heart Cells) – 07.12.17", [website], URL: https://www.nasa.gov/mission_pages/station/research/experiments/1914.html [cited 1 August 2017].

[12]Gunbas, N., "Students' mathematics word problem-solving achievement in a computer-based story", *Journal of Computer Assisted Learning*, Vol. 31, No. 1, 2015, pp. 78, 95.

[13]Schoemaker, P. J., "Scenario planning: a tool for strategic thinking", *Sloan Management Review*, Vol. 36, No. 2, 1995, pp. 25, 40.

[14]Garvin, D. A., & Levesque, L. "A note on scenario planning," HBS Case Collection, Harvard University, Cambridge, Massachusetts, 2005 (unpublished).