Executive Summary

Given the importance of models and simulations in public policy making, and the need to improve their effectiveness, the governmental and non-governmental model and simulation building communities should be striving to explore and build on other existing model-building practices. Some of the most

Ben Sawyer
Digitalmill, Inc.

Ben Sawyer is the cofounder of Digitalmill, where he is in charge of strategy, technology, and business development. Located in Portland, Maine, Digitalmill is a technology development firm with clients worldwide. It has worked on a wide variety of projects dealing with interactive game development, including support for The Sloan Foundation's *Virtual U* game project. The company has produced two books on game development, numerous articles about developing games, and several market research reports on the gaming industry. Currently Digitalmill is working on *Virtual U 2.0*, and consulting on other projects that integrate gaming, education, and training. Sawyer has authored or co-authored more than 10 computer trade books as well as numerous articles on a wide range of technology areas including e-commerce, interactive game development, software marketing, and computer graphics. Publications include *The Ultimate Game Developer's Sourcebook*, published in 1996 by Coriolis Group Books.

To find out more about the author, please visit: www.dmill.com

Use of this Module

This module is intended to serve as an education resource tool and is free to be reproduced and disseminated so long as credit is attributed to the author. If you intend to reproduce any part of this document, please let us know at the Foresight & Governance Project. We would be interested in hearing how you intend to use this module and in connecting with others who are interested in game-based simulation.

To find out more about the author, please visit: www.dmill.com
Contact the author: bsawyer@dmill.com
Contact the Foresight & Governance Project at the Woodrow Wilson Center:
rejeskiDW@wwic.si.edu

Enhancing Simulations, Models and Their Impact Using Interactive Game Design and Development Practices and Technology

In today's public policy environment, computer simulations have become important modern-age tools used to affect the education, debate, and implementation process for a variety of initiatives. Whether they are developed with supercomputers in the national university management economic strategies. Although training is the most natural outgrowth of real-world needs and computer gaming, there are other inspirations and technologies that the greater model and simulation building communities can take from the gaming community. Not only is the game development community at the forefront of PC-based visualization programming, it is also a leading developer of applied AI, overall interface design, persistent worlds, network interaction, and other needs for next-generation modeling and simulation programs.

The Importance of Game-Based Modeling

The best way to make the case for the importance of game-based modeling is that it has already been successfully put to use. Products such as SimHealth or Virtual U were developed through the cooperation of commercial computer game developers and public policy groups. Private businesses as well are looking increasingly at game-based simulations for effective training methods. And it is the philosophy of game developers that no matter how simple or complex the underlying model or simulation is, it needs to strive for suspension-of-disbelief, be holistic, balanced, and packaged in an accessible interface. This makes game-based simulation a user-friendly training and educational option.

As computers have become more powerful, developers have pursued with even more fervor the "suspension of disbelief" principle in which a game world is so believably presented that players become totally immersed in it. Furthermore, the market goals of game developers is to sell their games as wide an audience as possible. These are not necessarily the key principals that vigorously guide models produced outside of the commercial entertainment industry. This difference is an important one because when models and simulations create closed-loop systems, and are made easily accessible, the result is more discussion, more feedback, and potentially more significant impact. Additionally, game developers with unique approaches and perspectives can potentially improve the accuracy of models – especially ones based more on empirical methods vs. data-driven models.

Ease-of-use is particularly important if a model is being used to simulate or forecast derivative issues and fallout. For example, a model that allows people to simulate decisions made in an economic climate that has resulted from a steep climb in energy prices might be created so one can examine the reactions and strategies of different managers or constituencies. If that model is difficult to use or is poorly presented, the ability to conduct further derivative simulations and surveys with it is hampered. As an overriding principal, however, game designers and developers are especially familiar with these types of constraints and the need to satisfy a specific audience or audiences.

The basic criticism of game-based simulations and models is that absolute accuracy is sacrificed in the name of balance, fun, and accessibility. Furthermore, the need for them to run on hardware that is far less powerful than a state-of-the-art workstation means that game-based simulations have not had the same level of detail or horsepower available in other styles of simulation and model development. These criticisms are valid enough to have become accepted as fact and have stigmatized entertainment-born simulations as far less worthy than larger, spreadsheet-focused statistical systems or models built on heftier

Design Advantages

Any casual observer who has seen someone’s total attention captured by a computer or video game, let alone a good game of chess or cards, can easily understand how games can quickly captivate their audience. With their exciting visual and audio power, computer and video games take the competitive and fun nature of games to an entirely new level. Combining simulation, strategy, and the ability to play alone (as playing partners are not always available) electronic gaming builds on basic instincts for competition, interaction, and imagination that are instinctive in so many people. By combining these elements with instructive materials, or wrapping important content in a gaming package, the hope is to utilize the strength of gaming to elevate learning and especially strategic learning among players. This has been the key goal of many non-entertainment applications of gaming technology and methods.

Unlike the entire population of simulations and models, games are designed to be, or often are, inherently…

- Challenging.
- Entertaining.
- Educational.
- Played repeatedly.
- Multiplayer - a special skill unto itself.
- Designed so no two games are perfectly alike.
- Technology Advantages

One has to be careful to categorize the technology advantages game developers can bring to a particular simulation or model. This is because in many cases the best technologies game developers create are essentially shoe-fitting other cutting-edge ideas and programming practices into the lower-end hardware, which makes up the majority of PC and console gaming systems. That isn’t to say game developers are not technology pioneers -- in areas like applied artificial intelligence or artificial behaviors, game developers have done a great deal of quality work. In terms of interface design and fast 3D engines, game developers have also pushed the frontier. As PCs become more powerful, the gap between what is pioneered on high-end systems and what is implemented on lower-end consumer platforms is shrinking.

Whether the advantage is from applied or pioneered technology advances, there are several key areas where developers can help existing models:

Interfaces

Since game software is intended to ship as a commercial product to a broad audience both geographically and demographically, critical interactive premises and elements are a crucial aspect of the design phase. Interface design is not an easy process, and indeed a key element that slows the very distribution impact of many models produced outside the game development industry is that they are completely lacking an interface. It can also be detrimental if the interface has an overall academic, scientific, or engineering-like interface that makes it much harder to comprehend, process, or otherwise modify. Quality

Model Development Advantages

There is no doubt that game developers have a wealth of model building experience. However, the models developers build will defer from models and simulations built for other purposes than entertainment and mass-market appeal. Furthermore, many types of models and simulations built by game developers are fictional in nature. Whether it's a model of a medieval castle, an interstellar spaceship, or running your own Caribbean island dictatorship, many games are fanciful in their settings. These can be seen as weakness or strength depending on how you look at it – we of course see potential strengths.

Exploring the relationship between relative accuracy and absolute accuracy is one of the defining issues concerning application of game-based development techniques for model and simulation building. Many games, be they real-world or fantasy-world based, do not always build models focused on purely accurate premises. While many games do strive to use a wide and deep range of accurate algorithms and scientific principles (e.g. basic aerodynamic theory for flight simulations, or basic economic theory for a nation-building game) the nature of games tends to require a mixture of accurate and vetted formulas

Dungeon’s and Dragons, the entire world of middle-earth created by author J.R.R. Tolkein, or the visual worlds of George Lucas, game developers are inspired by the idea of immersing themselves in worlds both real and imagined. Coupled with their technical interest in simulations and programming this has spawned some of the most highly successful mass-market computer models ever created. This is in contrast to academic- and scientific-based motivations that dominate other model- and simulation-building communities. These communities are more rooted in the scientific method and principles -- to strive for complete understanding of a subject, prove a specific theory, or create an accurate prediction of existing systems. While the inspirations and guidelines for what constitutes a successfully produced simulation and model are somewhat different, it should be considered that neither has to be mutually exclusive -- both as a purpose and as a community.

Thus the advantage of game developers working on a model or simulation is not just in their technical ability to render them but also in their creative underpinnings. The creative force that drives most game developers is what makes them natural “out-of-box” thinkers, as well as excellent observers of the world around them. This sort of “world-building” motivation is especially useful for creating models of complex ecosystems.

The Key Advantage

Most models are built for small audiences, and their less-than-interactive nature results in an impact that is only widely felt when a news organization relays the results of a model to a larger audience. The report of the outcomes however is never too deep since very few people actually use and interact with the underlying math, science and other information contained in the simulation in the first place. Thus, the impact for many developed models is fairly limited, and even if widely reported, than not necessarily deeply understood.

By packaging (or repackaging) models covering critical issues, developers can attempt to use the widespread appeal of games to reach a wider audience. By applying all the right tenants of interactivity and game design to a model, its author can seek to provide a far better understanding of the relative factors and their interrelationships to the simulation’s users. If that impact is also education, then through gaming many sophisticated models and simulations can become extremely effective training tools as well.

Gaming is by no means a replacement to existing model and simulation building processes and practices but it has tangible advantages that ultimately could result in wider, more flexible, and more versatile, products. To ignore these contributions wholesale will directly affect the ability for any simulation or model to reach its full potential.

The Goal: Models as games and game-styled models

There are two distinct approaches to injecting game industry technology and practices into the wider field of models and simulations:

- Creating an actual game-oriented simulation or model.
- Improving a model or simulation by applying game development techniques and technology to it.

Each approach has specific goals and requirements.

Creating an actual game-oriented simulation or model

The most typical approach is to package a model or simulation as a game. This requires the model to be interactive and to provide the key elements of a game. In order to qualify as a game, a simulation or model must adhere to the following principles:

- The player must be able to tangibly affect the outcome of the game.
- There must be an overriding goal/challenge as well as sub-goals and challenges to the player with positive and negative outcomes based on their actions.
- It must require mental or physical skill.
- The outcome must be uncertain at the outset.
- It must require the player to develop strategies in order to win or succeed. Those strategies needn't be apparent at the outset; in fact the discovery element of gaming is one of its most important strengths.
- It must offer multiple paths to success. Linear games tend to take the form of puzzles, which, while useful and entertaining are primarily about figuring out a specific question and not necessarily about formulating strategies.
- Players must be able to ultimately overcome most obstacles in the game. Only under certain circumstances does it make sense to provide a game that isn't at some point "winnable."
- It must be interesting and fun (relevant to its audience) to inspire repeated play.
- It can also be educational in nature, especially in the context of simulations and models.

This approach is best suited for applying models to educational and training situations. Whether the goal is to educate a constituency or train staff and managers, using game-based models for learning is an excellent application and one that is being used more. Not only do these types of training applications help focus people on strategic thinking, game-based models can also be utilized for competitive training, team-based play, and for appealing to younger people who are familiar with interactive entertainment. The entertainment aspect of games can also lead to repeated usage and help encourage otherwise unmotivated or disinterested learners.

Improving a model or simulation by applying game development techniques and technology to it

Models and simulations needn’t be actual games to benefit from some of the model-building principals and software development techniques used to construct computer games. The desired outcome here is to create an enhanced interactive model utilizing much of the technological prowess and interface skills of game developers. It is important that scientifically tested models are enhanced, not compromised, by the assistance of game designers, developers, and game-oriented technology. Specific skills and technology that can be applied include:

- Providing a completely graphical and intuitive interface that allows people to rapidly and easily interact with the simulation and model.
- Utilizing various artificial intelligence techniques to create realistic inhabitants in a

offering a few recommendations to improve the landscape, the panel missed some points entirely due to its high-level composition. Some of the obstacles also identified have been positively affected by structural changes in the industry since 1996. The following is a point-by-point review of the major barriers presented in that paper and how, if applicable, these barriers have since been addressed.

". Cost and profit structures

The council highlighted the discrepancy between the military model of payment (cost plus allowable profit) and the much higher profit margins of the game industry where the most popular titles can earn profits of more than 100 percent. While the Department of Defense model may be a bit different from other governmental and non-governmental organizations, there is no question that work outside the game industry will be contract-style work, with little or no upside other than the potential for further contract or maintenance work. The result is that successful developers and larger studios and publishers (i.e. public companies like Sega, Nintendo, Electronic Arts, and Activision) will have little or no interest in direct development work of this nature.

However, most game development studios and especially smaller development teams do not attain the levels of profit highlighted in this paper. The paper also fails to consider that with continued consolidation in the games industry, there will be more developers and skilled designers interested in work beyond that of the increasingly competitive

made publicly available for others to read. This style of publishing is popular for structural or academic purposes. However, this creates a significant barrier that can stall projects before they even start. In some cases, certain intellectual property control is out

Current Issues to Resolve For More

group. Within a development team usually only a small core part of the team design the project, with one or two being completely responsible for the finite design elements of a project.

Programming is usually also handled by a similarly small group where a typical core team has 1-4 programmers. Larger projects may have as many as 8-10 programmers but the bulk of these will be assigned to subordinate aspects of a large program, such as development tools, country localization, etc. It is usually only the art staff that balloons the size of a overall staff, and subsequently the

Who will be the designer(s); Who will be the developer(s);Who will they work with and report to?, etc.

What is the desired completion date?

What is the available funding for the project?

With a good synopsis and some background resources in hand the best place to find development teams is by using a combination of industry resource listings and agencies. Most independent studios are not easily found in the phone book, and nearly all developers don’t look for RFPs (although this could change in the years to come as more projects are launched), thus at least in the foreseeable future organizations will have to initiate the search. Gamasutra.com, the leading B2B community for the game development industry, as well as the IGDA (International Game Developers Association) provide the best resources to start a search. Gamasutra has a listing of agencies that lead a search for capable talent, and the site also offers its own list of contractors within the industry. While by no means exhaustive, the list does provide a strong starting point. When utilizing an agency or other third-party recruiters, be sure to look for agencies with a history of quality projects, especially with simulation and strategy titles. Due diligence is important as agencies are not as embedded, fully connected, or as reputable as they might be in other more historical, mainstream industries.

When potential development talent is found several steps will help create a successful evaluation process. These are outlined in Table 1.2. An evaluation should be coupled with a request (possibly funded) for a game design document, a technical document that describes how the design will be actually developed, a proposed schedule, and potentially (also funded) a prototype of the product. Prototypes could be a series of screen mockups, or a fully working component of the project. Prototyping (or demos in industry parlance) is a common publisher practice used to evaluate teams which don’t have and extensive portfolio of previously developed products to evaluate, and is also used to ensure a project team can work cooperatively to achieve the vision initiated by an outsider.

Table 1.2: Game Development Talent Evaluation Tactics

Tactic	Purpose	Judgment Points

resume	the studio.	are formed with talent possessing a deep previous experience.
Form an opinion of the team’s cohesiveness	Many small development studios, especially greener teams can fall apart. Loss of key talent or team cohesion can be a particularly problematic risk.	Find teams, however small, with long operating histories. Look for strong management, and don’t rule out

Working with a development team

With a development team on board, and a development plan in place, an organization still has much to do to ensure a project's success. Many of the responsibilities are the same tasks commercial publishers perform with their development teams. The biggest difference is that developers will especially need help working on topics and purposes that are not the staples of traditional game consumers. Thus in non-entertainment fields, developers will require more support and expert cooperation from the hiring organization.

Key tasks to manage once a development team is ready include[u8]:

Providing the key vision for the end product

Chief among an organization's responsibilities is providing from the outset a strong vision of the final project. While many game projects actually begin some of the programming work before every last design detail is completed on paper, projects falter fast if there is not an overall specification for the goal of the final product. Full agreement development. Aside from art and other media assets, no amount of funding or manpower will likely make programming and design progress any faster. Eight programmers on a project are not necessarily any faster than two, especially in game programming where programming styles and the resulting source code are unconventional. Development is an engineering process and all engineering processes, especially software development, hits snags and unintended delays.

". Flexibility is key

One of the key aspects of games is that a designer has the ability to truly control the premise, the desired outcomes, the variables, and much more to shape what they feel is the perfect game. In fact it's the ability to balance many disparate elements that can provide for the optimum playability of a game title. With games meant for audiences outside the commercial market, there is a tendency to need more flexibility to change and recalibrate a game’s initial play balance and other features. This is primarily because users of simulations, and models outside of the entertainment industry, may want to stress or skew [VGC12]them in many different ways. Thus we recommend that games developed for government, educational and other non-entertainment needs focus on providing as much flexibility to their designs and operations as possible. While it's important to ship what a developer feels is the optimum configuration of the product

4. Public source or open-source publishing should be pursued

The ultimate peer-review in the entertainment industryfocuses on quality, capability, and playability of a game as reviewed by trade magazines, and the sales generated by the market. In models and simulations for other constituencies the peer-review is much more about the methods, assumptions, data, algorithms, and other techniques used to create the program. Without access to the source code that isn't as possible. With access to the source code other developers, scientists, academics, etc. can work to improve and extend the model. Game developers must recognize this and work to provide public or open-source distributions of products constructed for non-entertainment purposes. This includes well-commented code, as liberal a license agreement as possible, and support documents that help other programmers and researchers get up to speed with the source code. In cases where derivative commercial interests may exist, public source or private permission-based availability of source with a restrictive license is quite suitable. While open-source advocates will frown upon source availability that isn't pure open-source, that shouldn't discourage developers from releasing some grade of publicly available source code.

5. Platform requirements will lag the game market.

Game developers typically will write many titles with a focus on cutting-edge hardware the running of multiple instances of their software. Game developers should ensure, whenever possible that their titles:

- Can be installed over a network
- Can be run from a central file server by a local machine
- Are devoid of cumbersome piracy prevention processes such as necessitating the CD-ROM to be present in a machine in order to operate the product.[VGC15]

In terms of support, expect needs to be greater than traditional game audiences. This includes configuring such issues as sound, music, input devices, and graphics cards. Providing strong troubleshooting documentation with a product or via a Web site will be important.

Expecting that many titles developed for non-entertainment oriented uses will be offered at zero cost, developers should also anticipate widespread distribution via the Internet. This gives the opportunity to embed into the installation process electronic user registration and user surveys. Developers might also include an integrated update system for extensions and patches, as well as in-game links to online support and information systems such as a Web site.

7. Expanded, thorough and quality documentation including elements focused on user training, key evaluators, and expert users.[VGC"6]

Most game manuals are as brief as possible without leaving out information critical to playing a game. Only rarely do games provide documentation akin to a standard PC application. Since game-based simulations and models are themselves more application focused than traditional games it is necessary that the accompanying documentation be likewise skewed. Furthermore, while no one appreciates spelling and proofing errors, these errors tend to be amplified in the academic and scientific circles. Developers may find themselves working with these people when they take on game-based simulation and modeling projects.

In addition to basic operating documentation for a simulation and a model, expanded documentation areas might include:

- A syllabus and teaching guide for trainers that may utilize the product in a classroom or other learning situation. This might optionally include a test bank, assignment ideas, and tutorial information.
- Complete background on the program's structure and systems for simulation and modeling.
- A strategy or expert playing guide that couples program tips with background information on the topic on which the simulation or model is based.
- A reviewers' guide that steps non-game savvy evaluators (e.g. press, agency heads, etc.) through typical playing session.

A formal bibliography that details data sources, empirical sources, reading, and further materials that were used in constructing the simulation and model.

In many cases it is expected that many users will be familiar with the subject matter of the program. However, in cases where a key element of the project is to educate users on the product's subject matter, considerable background and tutorial information on the topic is a major component of the manuals and in-game documentation.

8. Beta-testing will be a difficult process, but also a necessary quality issue.

Game developers are accustomed to a world where internal and external testing populations are extremely savvy, not only about their hardware but also about games and especially about works in progress. Support groups and steering committees outside of the core-entertainment industry will not be as understanding of works in progress. Beta-testing games will be a very foreign concept for them and even the slightest problems or errors may cause them to stop their evaluations at the very point developers expect them to be deepening their evaluation. For example, many academics will wonder why they are being asked to find major game-play bugs or miscalculations before common grammar and spelling mistakes are fixed. Other testers may not understand how to focus only on a subset of elements that you think are working and instead critique those that a developer pre-qualified as not working in the current build. Furthermore, unfamiliarity with gaming comprehensive simulation might help people understand the impact each component of a university -- from admissions to facilities to faculty -- had on each other.

The initial plan was to create "SimCity" for universities. With Dr. Massy’s experience as a major university administrator and a seasoned economist and model builder, the foundation believed it could deliver on a vision to create new ways to educate. By packaging it as a game with the plan of fulfilling various scenarios to keep key components of the university's finances in balance, it was hoped the simulation would not only be accurate but relatively entertaining. The goal was to deliver a novel approach to teaching modern theory about university budgeting, operations, and leadership.

The Development

With Dr. Massy driving the overall design and algorithm development, the foundation made $1 million available to sponsor the research, design, and development of the model. It was decided early in the process to use a seasoned development studio. Underscoring the importance of strong due diligence and the interview process, the first development team resulted in a false start due to poor development skills. The second development group, Enlight Software of Hong Kong, did prove successful. The team's strong experience with similar management strategy titles, including its well-reviewed Capitalism program, proved a good match. The key to the development partnership was that an experienced development team was married with a topic expert who further enhanced the process by bringing his own extensive programming background to the project.

Development first focused on the underlying engine and data set. The primary design was developed in tight coordination by Enlight's chief programmer and developer Trevor Chan, and Dr. Massy. The design input was enhanced by contributions from an ad-hoc steering committee of various university professionals and researchers at the University of Pennsylvania's Institute for Research in Higher Education. The Hong Kong-based Enlight staff handled all aspects of the actual game engineering, including programming, artwork, and sound. Coordination with the Dr. Massy and Sloan Foundation personnel consisted mostly of emails with occasional face-to-face meetings.

Development moved quickly at the outset, but poor planning and program development management, coupled with development team turnover within Enlight, severely hampered the schedule. Additional production management with game development knowledge came on board toward the final 20 percent of the project, but had this help been available at the beginning of the project scheduling issues might not have been such a problem. In addition, some issues concerning its design may have been better addressed. In the rush to get the project going, some decisions -- such as whether to make a version for the Macintosh, or what all the goals and challenges to the player would be -- hadn't been fully developed. The model itself was exquisitely developed but the game, which worked from that model, hadn't been as well specified. This underscores the need to have a development manager in place with an understanding of gaming and development to guide the process.

Packaging the Product

Developing the software is just the first aspect of any game or software development project. In order for a product to be successful it must be transformed into a comprehensive product and/or project. This includes it being packaged in some form, development of a distribution scheme, and in the case of many game projects used for education or training purposes, a training strategy developed in conjunction with any marketing strategy.

It was decided early in the development of Virtual U that it would be a product sold to targeted end-users. The goal was to create an initial investment by the user that would encourage them to use the product once they acquired it. Two products were created -- a low-cost game version and a higher-cost administrator version that sold for $130 – and pricing was determined by examining the common cost of graduate and doctoral textbooks. To ensure wider use and impact a free, playable demo version was available online.

For the packaged versions, two sets of manuals totaling nearly 200 pages of documentation were developed. Documentation was an important aspect of the product given its depth and breadth. A multimedia tutorial was also created to help users visually run through the game.

Distribution was via mail order and through the Internet. Users can visit the Virtual U Web site and find ordering information or the available demo for download. Anker Publishing, a book publisher targeting the higher-education professional market, was brought on as a co-marketer and fulfillment partner. Anker proved more useful than other potential partners because they were willing to be quite flexible in promoting and marketing a product of such a unique nature. It may not be apparent right away the best

Strong team produced a strong simulation

The biggest asset to Virtual U's success was that the key people working on the project code because the developers worried about both the intellectual property issue and how they might feel compelled to provide support the source code. The developers also believed they hadn't commented or structured the source code for public source use. Eventually these issues and fears were put to rest and the source code was published in the summer of 2001. To date over 200 people have downloaded the source code and helped the project fulfill a sub-goal of providing as much documentation of its model as possible.

What Went Wrong

The key problem for Virtual U was that schedule and organizational issues hampered its ability to finish strongly and on time. Schedule delays are impossible to blame on one

The complexity of the software created obstacles to wider use

Not every game is easy to use. While the argument that a game interface and motif makes it more enjoyable and easier to use than an inherently statistical simulation or model, no interface can hide the underlying complexity of the subject matter unless the overall simulation is watered down. Water down a simulation too much and it's overall realism, purpose and/or educational value may be irreparably harmed.

While it's hard to categorize it as something completely wrong with the end result of version 1.0, Virtual U suffers from a complexity problem. In order to begin instigating more pronounced and interesting policy moves, it takes a user as much as three or four hours to properly acclimate themselves to the simulation. Even with all of its documentation the game is more suited for one that moves at a slow pace, and has a steep learning curve.

Much of VU's complexity may have to do with its concept and content rather than any outright failure of design. If anything, the design has taken a very complex subject and made it infinitely more accessible to its users. However, by categorizing VU's overall complexity as a failure the design and development team feels it can do much better. This includes improving feedback mechanisms for the player, improving the model itself, and improving documentation. Furthermore the project has begun listing trainers among VU's more experienced user base to help those who need more personal one-on-one training with the product as well as providing paid one-on-one training by the project team itself.

Initial assumption of user mix for the product was not correct

VU's initial priority target users were presidents, provosts, and deans of United States-based universities. Unfortunately, this group is also among the busiest so sales to and impact on this group has been lower than expected. Conversely exposure and impact on students, lower-level administrators, faculty, and department chairs has been higher than expected. To adjust for this the project has focused its efforts on lower-level administrators in its direct mailings, as well as professors and students of education. To keep the goal of reaching out to presidents and other upper-level managers the project has tried to arrange for demonstrations and interactive sessions at various presidential conferences, workshops, and retreats.

The lesson here is that many times models and simulations are built with the hope of educating and impacting very high-level targets. Unfortunately the combination of limited computer access, skills, and available time usually hinders the ability for personal one-on-one exposure to the software. Even the novelty and ease-of-use of a game might not be enough to overcome these obstacles. Instead, personalized demonstrations are best, especially at times that have already been set aside for exposure to new ideas such as conferences and workshops. In some cases, getting that first important exposure has resulted in personal orders for themselves and staff.

The Future of VU

Thanks to its initial levels of success, and further generous support by The Sloan Foundation, Virtual U 2.0 is under way. With so much learned by the initial experience of

Virtual U, the ability to develop a second version of the product and the continuing evolution of the project would prove exponentially beneficial to its overall goals. Learning from the mistakes and strengths of the initial product, some of the adjustments to the upcoming second version include:

- Gaming features will be amplified. Players will have more challenges and more ways to lose the game.
- More training functionality is in the product. Players can output values to a spreadsheet for analysis and assignments, plus the ability to tailor the scoring system and other key weightings in the game as an instructor (or player) sees fit in order the achieve desired goals.
- Printing was greatly improved in a 1.3-released patch that will be included in the 2.0 version.
- The model pays better attention to distance learning and public university issues, two key constituency issues that weren't as well detailed in the initial project
- The ability to turn off randomness in the model - a key component of its game capability was added to assist those who wanted to more closely examine the effects of the model without random factors present.
- Better testing of the game has been organized from the ranks of experienced 1.0 users.

Virtual U 2.0 is scheduled for release in January 2002. Subsequent to this release is a grant that will enable VU 2.0 to be available in its complete form free for download from the Internet. This will widen VU's impact and let its more than 15,000 downloaders experience all of VU without requiring a purchase. With an improved model, an improved game premise, and a burgeoning user base, Virtual U 2.0 should continue to have impact on the education and practice of university management for years to come.

Virtual U is not the first combination of public policy simulation and gaming. It may not even be the penultimate version. However, it is one of the more successful, best-documented, and more high-profile projects of its type. As it shows, there are a lot of ups and downs in creating a hybrid product from a hybrid partnership, but the results can be quite innovative and worth the effort.

Bringing it all together

By specifying the case for more involvement, and by detailing the specifics of how to create more cross-pollination it is hoped that more will occur and that the benefits from simulations on subject matter such as peacekeeping, health-care, university management, and hazardous chemical clean-up shows the potential if more aggressiveness, education, and experience in building the hybrid partnerships takes place.

The advantages are not just one way either, game developers could benefit greatly not only from a widening of their market for their talent but also by gaining access to a whole cadre of simulation and model building professionals, and data collectors.

Every day game developers reach an audience of millions of people using increasingly state of the art hardware and programming. By bringing together that strong end-user talent, with the problem solving and education cause of academia, government, science, and non-governmental organizations a new generation of models, simulations will be