

# **How2QnD: Design and construction of a game-style, environmental simulation engine and interface using UML, XML and Java.**

**Gregory A Kiker**

Dept of Agricultural and Biological Engineering, University of Florida

**Rohit Thummalapalli**

Summer Science Training Program, University of Florida

## **Abstract**

Within wicked environmental challenges, problems that exist in the nexus of environmental science and environmental values, neatly and elegantly optimized solutions are difficult to find and rarely accepted by stakeholders. Different role players must explore the challenge adaptively and through viewpoints to contribute to their understanding of the situation and to learn about the dynamics and values of other relevant stakeholders. The *Questions and Decisions*™ (QnD™) system [17] was created to provide an effective and efficient tool to integrate ecosystem, management, economics and socio-political factors into a user-friendly game/model framework. QnD is written in object-oriented Java and can be deployed in stand-alone or web-based (browser-accessed) modes. The QnD model links spatial components within geographic information system (GIS) files to the abiotic (climatic) and biotic interactions that exist in an environmental system. QnD can be used in a rigorous modeling role to mimic system elements

obtained from scientific data or it can be used to create a “cartoon” style depiction of the system to promote greater learning and discussion from decision participants. Elephant and vegetation dynamics in Africa provide an excellent example of a wicked environmental challenge as conservation objectives and societal values (both local and international) often have conflicting goals concerning appropriate elephant densities and population control options in protected areas. In attempting to capture many dynamic aspects of elephant-vegetation relationships, previous models depicting the savanna ecosystem of the Kruger National Park (KNP), South Africa can become quite complex and demanding in terms of detailed parameter inputs. Therefore the purpose of this modeling project was to create a simplified, management-focused, visual simulation of the KNP in order to chart future elephant, tree, and grass scenarios. QnD:EleSim has been designed to spatially simulate elephant-vegetation dynamics in 195 areas at 10 km resolution at a monthly time-step. As the effects of elephant populations on the tree-grass equilibrium of the savanna are documented, future management decisions can be advised after analysis of potential scenarios.

## **1 Introduction**

As a matter of recent history, environmental decision-makers are increasingly facing a set of problems that appear to have no easy solution. These “wicked” problems [28] exist “at the intersection of science and values” ([36] pg. 2) and defy neat, optimized, numerical solutions. These wicked problems require an integration of scientific information, uncertainty estimation, and social/cultural valuation for environmental decision-making.

In reaction to such intellectually and emotionally complex challenges, decision-makers and scientists have increasingly turned to the use of computationally complex systems models

that attempt to incorporate multiple system dynamics at very fine spatial and temporal resolutions to match the complexity challenge with computational “shock and awe.” However, simple, pragmatic models that require fewer parameters than complex models can be surprisingly useful in ecological studies [14] [31]. This simple-model approach was useful in highlighting selected management issues within the river ecosystems [33], where a suite of simple models at multiple scales of time and space were used to assist scientists and managers. The concept of managing environmental systems as being a game involving different role players and options, has revealed important general patterns of system behavior [6]. Their Non-Point simulation program described a simple model of ecosystem management from the perspective of selected role players. This model served to show the interaction between fast and slow variables (multiple time scales), and illustrated the point that continual learning was fundamental for adaptive and resilient systems.

Often management decisions must be made in the absence of adequate data, which is where modeling becomes a useful management tool. Thus a model's development may be driven by the objectives of the management program, rather than the available data [31]. Scenario modeling is a useful tool for envisaging future situations in an unknown future [31]. Models help to expose gaps in data and understanding, and help to screen policy options, especially under conditions where time is limited and systems are sensitive [33].

### **1.1 Conceptual Background: Learning through games**

Increasingly, the effort of building a model and its associated execution for exploring system/management dynamics are seen within an *adaptive learning* context. Aldritch [1] describes three fundamental and intersecting elements of successful educational simulation: Simulation elements, game elements and pedagogy. Simulation elements systematically represent reality into computer science-based structures (i.e.

objects, stochastic elements and temporal/spatial databases) and allow concepts to be judged, altered and reformed in an iterative fashion to provide discovery and experimentation within curriculum. Gaming elements provide the recognizable and entertaining aspects to content although their interaction with simulation components must be systematically managed to support the sustainability of educational content. Aldritch [1] points out the challenges of scale representation of gaming elements which is also mirrored in the environmental simulation research [26] [35] [13] [34]. Pedagogical elements provide the practical educational management of the simulation and gaming content into specific and monitored outcomes. The interaction of these three elements is adaptive and thus is managed both at the curriculum design (strategic) level as well as within the classroom (tactical) level.

### ***1.2 QnD: A game-style simulation for adaptive learning and decision-making***

The *Questions and Decisions* (QnD) model system is a problem-exploration tool that increases understanding of potential ecosystem behaviors and management options of a particular socio-ecological system [17][18] [15]. QnD is written in object-oriented Java and can be deployed in stand-alone or web-based (browser-accessed) modes [27]. The QnD model links spatial components within Geographic Information System (GIS) files to the abiotic (climatic) and biotic interactions that exist in an environmental system. The QnD system is divided into two primary elements: the *SimulationEngine* and the *GameView* as shown in Figure 1 (adapted from [17]). Through the user-friendly graphical interface (*GameView*), stakeholders can “play” their system by manipulating institutional and ecological components of interest. Results generated by the *SimulationEngine* element differ from the combinations of various environmental and economic drivers and the player’s response to them via management options implemented in different spatial areas and over different time pe-

riods. As various scenarios are played, the interactions between institutional and ecological parameters are exposed and future possibilities of the system can be envisioned. The QnD system not only helps stakeholders deepen their understanding of a particular system's components and dynamics but also acts as a device to bring cohesion to a stakeholder community. Throughout its iterative design process, a QnD system engages stakeholders to accomplish multiple tasks: facilitating initial agreement on key forces and themes, broadening understandings of others' interests, developing scenarios and analyzing various policy options. QnD was created as a technical tool that is complemented by and works in tandem with an iterative exploration process [17]. The system's design process was intended to be compatible with several social science methodologies, such as SSM, scenario planning [32], cognitive mapping or mental modeling [24].

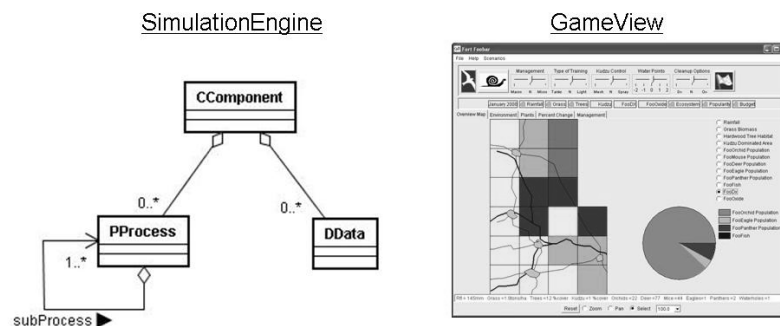


Fig. 1. QnD's two primary elements; the SimulationEngine and GameView (adapted from Kiker *et al.*, 2006).

As the fundamental philosophy and background of the QnD system was reviewed and documented in [17] and [18], ***the primary objective of this chapter is to provide a technical design overview in Unified Modeling Language and illustrate these designs with an application toward on-going modeling efforts to simulate elephant population and savanna vegetation dynamics in the Kruger National Park, South Africa.*** Thus, the chapter is divided into two sections;

(1) a technical design overview which covers the object design of the GameView and SimulationEngine elements as well as the Use Case designs and (2) a specific application of the design elements using the elephant-vegetation version of the model.

## **2 QnD Design Overview: designing from ideas to a playable game**

This section provides an overview of the object design using Unified Modeling Language (UML). The entire QnD system is coded in the Java language and is a combination of original code and open source libraries/application programming interfaces (API's) [27].

The overall design philosophy covers several steps including generation of systems designs and the translation of these concepts into actual object implementations.

### **2.1 Game View Design**

The Game View constitutes the primary user interface for most users. A user sees data results and reacts with the various management options in the player's world. QnD utilizes a standardized game format with object details and implementation are configured through the input XML (eXtensible Markup Language) files which detail which specific data objects (DData) are rendered in several graphic forms. DData objects that are spatially explicit can be rendered into both collective maps (selected by radio buttons) or line charts.

Figure 2 highlights the basic object design GameView is made of a map viewer (GeoToolsLite API:

<http://geotools.codehaus.org/>), scrolling time series charts

(Chart2D API: <http://chart2d.sourceforge.net/index.php>),

warning lights and management selection widgets. By presenting the outputs in a selectable form, the QnD system al-

allows users to choose how they want to see their output, including the following output options as listed below:

- Geographic Information System (GIS) maps that are updated on each simulated time step;
- Mouse-activated charts and text for individual spatial areas (pie charts and text line descriptions);
- Warning lights that change at user-selected critical levels;
- Scrolling time-series charts (listed on user-defined, tabbed pages);
- User-defined, text output files in comma separated format.

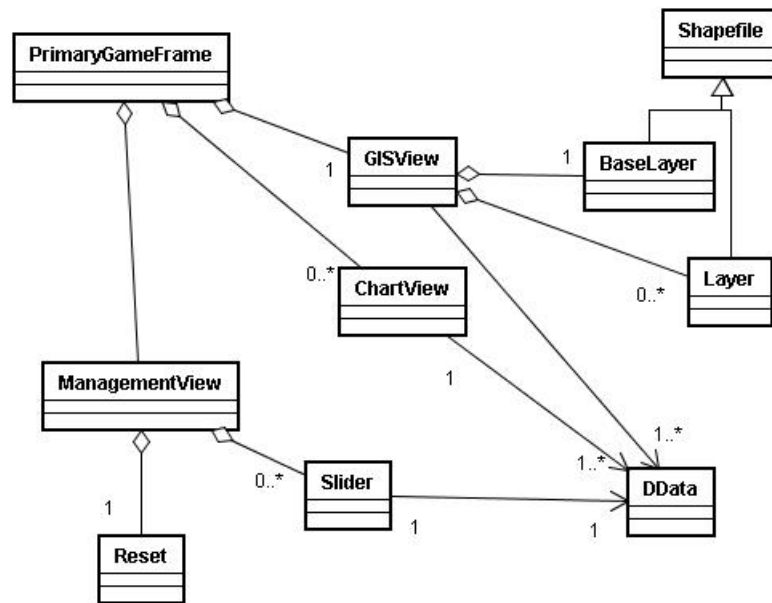


Fig. 2. Basic layout of GameView elements.

Figure 3 illustrates the basic parts of the QnD GameView interface for the QnD:EleSim example (detailed in later sections). A variety of maps can be viewed by selecting the desired radio button. Time series charts are accessed by selecting the tabbed panes above the map. As the user

simulates each one month or six month time step, each of the various graphic objects is refreshed with new data values. When the reset (white flag) button is selected, all values return to their original settings and the game is ready for another session.

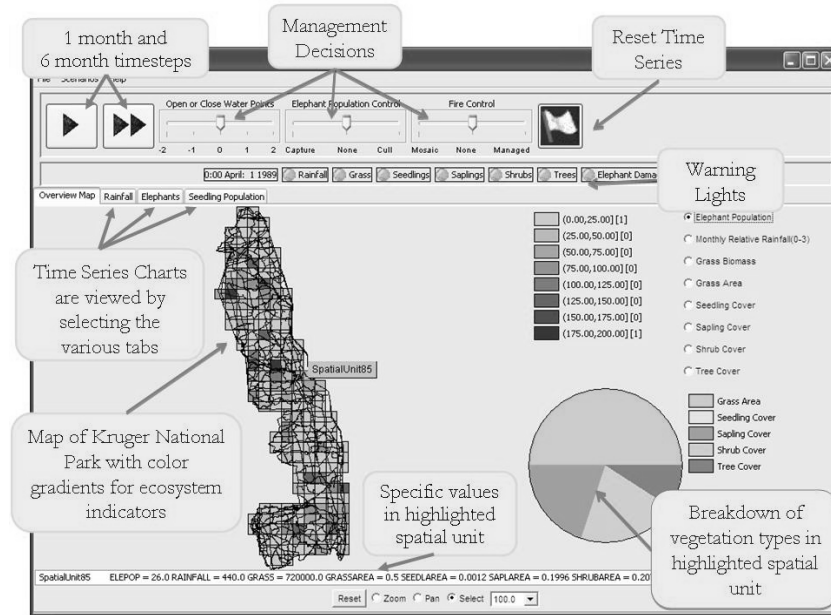


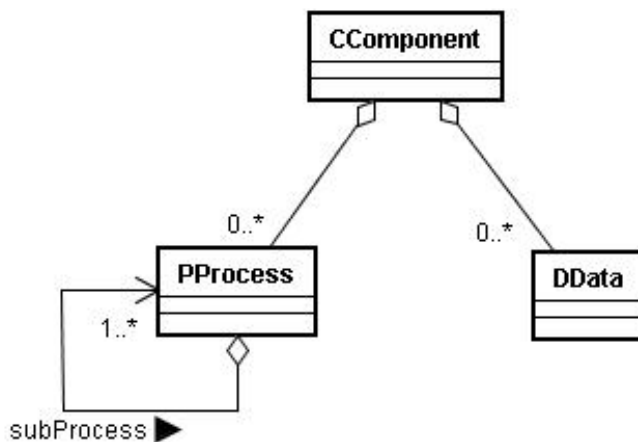
Fig. 3. An instantiation of the GameView element from QnD:EleSim.

## 2.2 Simulation Engine Design

The primary elements for creating a simulation engine is through the deployment of *component*, *process* and *data* objects illustrated in Figure 4 [17] [18]. For clarification within QnD designs and labeling, a “C” prefixes Components, a “P” prefixes Processes, and a “D” prefixes Data objects. *CComponent* objects form the basic items of interest within a simulation. *PProcess* objects provide the action and changes from one state to another. *DData* objects provide the necessary description of various attributes. These same objects have



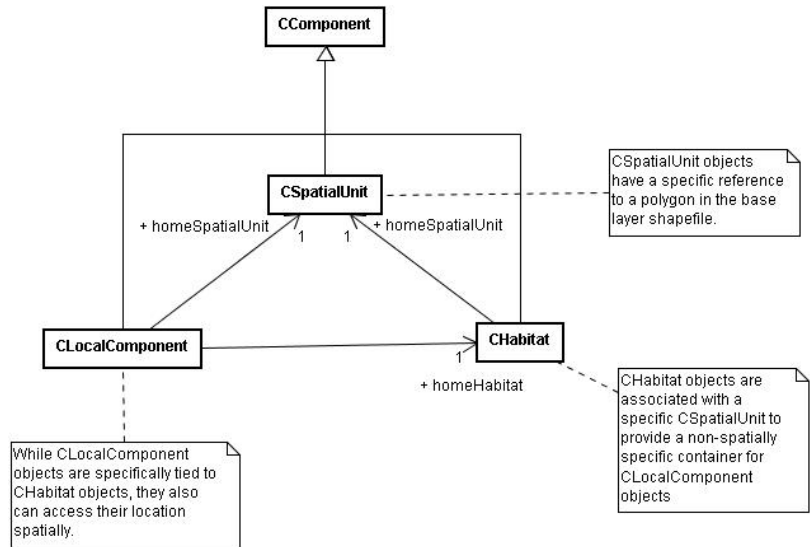
analogies in nouns, verbs and adjective/adverbs. Through various systems-envisioning methodologies such as Soft Systems Methodology [7] or Mental Modeling [24] configurations of interacting system elements can be envisioned and rendered in rich pictures and then into UML designs. The construction of various design diagrams in UML can be converted into XML-based input files for efficient instantiation as Java objects.



**Fig. 4.** Primary elements of the QnD simulation engine are components, processes and data.

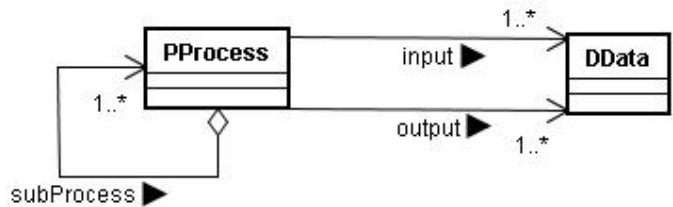
Spatially-explicit areas and non-spatially-explicit areas in specifically represented through two primary CComponent objects, CSpatialArea and CHabitat, as shown in Figure 5. A CSpatialUnit is the basic spatial entity of the QnD system. CSpatialUnits can be linked to one another and have a specific location. A CSpatialUnit can have either zero or any number of CSpatialUnits connected to them. One or more CHabitat objects exist within a CSpatialUnit and are not spatially defined, except via the relationship with the “homeSpatialUnit”. A CHabitat can hold any number of local instantiations of CComponent objects (CLocalComponents). These CLocalComponents have both relationships with both “home” CHabitat and CSpatialUnit. With this basic QnD object archi-

ecture, both simple and complex designs are possible with both spatial and non-spatial elements.



**Fig. 5.** Component objects are arranged in spatially explicit and non-explicit configurations.

*PProcess* objects provide all state changes and action within QnD. *PProcess* objects use *DData* objects as inputs, provide a calculation or series of calculations and then write the resulting products into output *DData* objects, as illustrated in Figure 6. *PProcesses* can used individually as described in Figure 6. or can be designed with constituent sub-processes within them to create a series of processes for more complex interactions. Table 1 shows the different types of processes that can be bound together in series within QnD.



**Fig. 6.** Each *PProcess* object has specific *DData* input and output relationships.

**Table 1.** PProcess objects for the QnD model. All PProcess objects take “input” DData objects, perform an operation and then write the results to the “output” DData object.

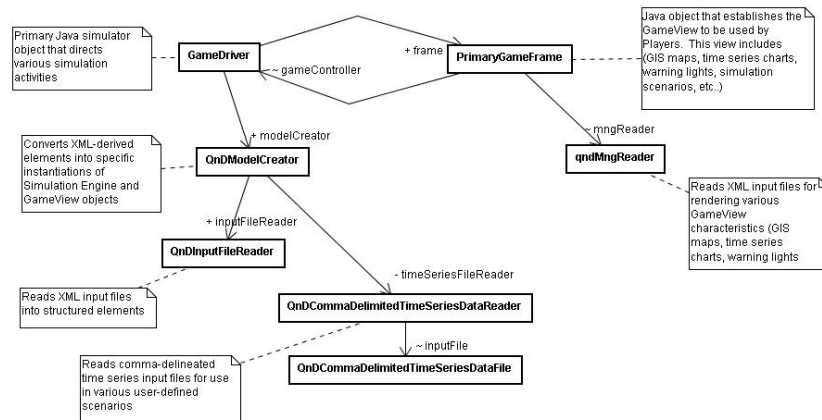
Process Type	Definition/Purpose
<b>PProcesses for Calculation</b>	
PAddValue	$\text{Input1} + \text{Input2} + \text{Input3} \dots + \text{Input}_n = \text{Output(s)}$
PSubtractValue	$\text{Input1} - \text{Input2} - \dots - \text{Input}_n = \text{Output(s)}$
PMultiplyValue	$\text{Input1} \times \text{Input2} \times \text{Input3} \dots \text{Input}_n = \text{Output(s)}$
PDivideValue	$\text{Input1} / \text{Input2} / \dots \text{Input}_n = \text{Output(s)}$
PExponentialValue	$\text{Output} = e^{(\text{input})}$
PSetValue	$\text{Output} = \text{Input}$
PMeanValue	$\text{Output} = (\text{Input1} + \text{Input2} + \text{Input3} \dots + \text{Input}_n) / n$
PTemporalMeanValue	$\text{Output} = \Sigma(\text{Input1}_t + \text{Input1}_{t+1} + \text{Input1}_{t+2} \dots + \text{Input1}_{t+n}) / t$
PTemporalRunningAverage-Value	$\text{Output} = (\text{Input1}_t + \text{Input1}_{t-1} + \text{Input1}_{t-2} \dots + \text{Input1}_{\text{Input2}}) / \text{Input2}$
PCalculateCurrentValue	Within a given sub process list, this object calculates the all the sub processes above it to get an updated and current DData object value.
<b>Specialty PProcesses</b>	
PTransfer *	$(\text{Input} - \text{TransferAmount}) \& (\text{Output(s)} + \text{TransferAmount})$
PRelationship**	Two dimensional input/cause (x axis) is used to interpolate an output/effect (y axis) value.
PSimpleLookUpTable***	Uses two input data values to choose another value from user-defined table and assign it to an output
<b>Logical Processes</b>	
PIfEquals	(If input = output) is not true, then it stops executing any further sub-processes and jumps to the next process in the list
PIfGreaterThan	(If input > output) is not true, then it stops executing any further sub-processes and jumps to the next process in the list
PIfLessThan	(If input < output) is not true, then it stops executing any further sub-processes and jumps to the next process in the list

\* This process requires Inputs, Outputs and a “TransferAmount” data object. In addition, if the TransferAmount object causes the Input value to be negative, then TransferAmount is altered so that input will be zero. This process is mainly intended for mass balance style transactions.

\*\* PRelationship requires inputs and outputs as the other Process objects but also requires XY data points defining a specific function shape.

**DData** objects store all the relevant information for a specific QnD simulation. All DData objects are created from the input

XML, GIS data files or time series files and represent a composite variable storing a set of double values. Each DData has several attribute variables that allow for various calculations. All available attributes are not always used for each DData as some data objects may use other attribute features while others do not. For example, a DData object that is linked with a time series file (through its DriverLink attribute) may constantly change current values over time while another may represent a static variable in the simulation and may not use any other attributes besides a single parameter value. In addition to the primary SimulationEngine-related objects, several packages exist for various housekeeping and organization functionality. QnD *Control* objects are used mostly in the background and thus do not have the “C,P,D” typology of the SimulationEngine objects. The GameDriver object acts as a main simulator object to coordinate both the GameView and SimulationEngine. The PrimaryGameFrame object provides the main GameView frame. Both of these control objects utilize various factory-style objects (QnDModelCreator and qndMngReader) to read XML input and time series files and to create the various constituent objects.



**Fig. 7.** Various control objects that implement the Simulation Engine and Game View elements.

### **2.3 QnD Use Case Designs: Three actors, many roles**

The basic actors within QnD simulations and software development fall into three general roles; *Players*, *Developers* and *Coders*. A primary operating philosophy and basic interactions amongst the different actors have been documented earlier in [17]. Figures 8 through 10 show the Use Case diagrams for each of the primary QnD actors.

*Players* (Figure 8) interact mostly with the Game View while playing and exploring the system, potential management responses and trade-offs. Players can be stakeholders but can also be anyone who has an interest in the game. They see the simulated world as a larger, integrated ecosystem and have broad, varying interests. While players may have some interest in technical simulation details, they mostly interact with the Game View elements via the map, charts and management options. Players provide an important reality check to the overall design and function of the QnD system. Thus, they can provide feedback to other actors concerning the functionality as well as the “look and feel” of the management system.

*Developers* (Figure 9) design and implement the game view and simulation engine objects using the XML input files. While some developers might share a player role as well, their primary role is to translate the broader ideas of the players into functional object designs that are represented in the input files. Another fundamental role of developers is to provide any formalized calibration or validation of the simulation engine/game view. This confidence-building aspect is an important function in developing trust and interest into any simulation results that are seen by the overall group as critical. Developers decide how the game should be deployed to players, either through a website as a self-contained program or as stand-alone program on local computers for specific output file analysis.

*Coders* (Figure 10) interact mostly with the java source code and concern themselves with the overall applicability and expansion of the GameView and SimulationEngine parts as well

as the functional deployment of the QnD models. Coders have control and responsibility of the overall design and evolution of the QnD system for all groups of players and developers. Coders may take specialized suggestions from players and developers and implement them at a broader more abstract level within the source code to take advantage of new developments in the Java language, computer science concepts or internet technologies.

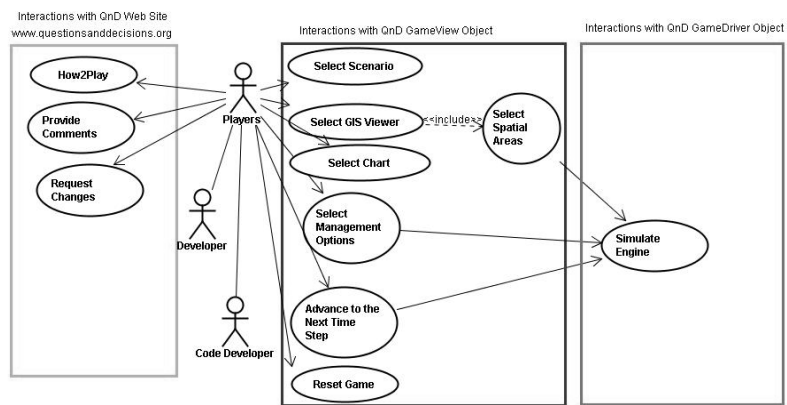


Fig. 8. Use Case diagram of QnD players.

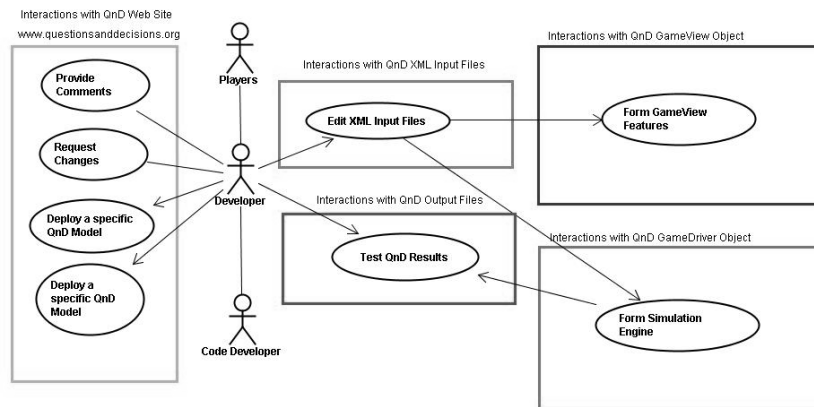


Fig. 9. Use Case diagram of QnD developers.

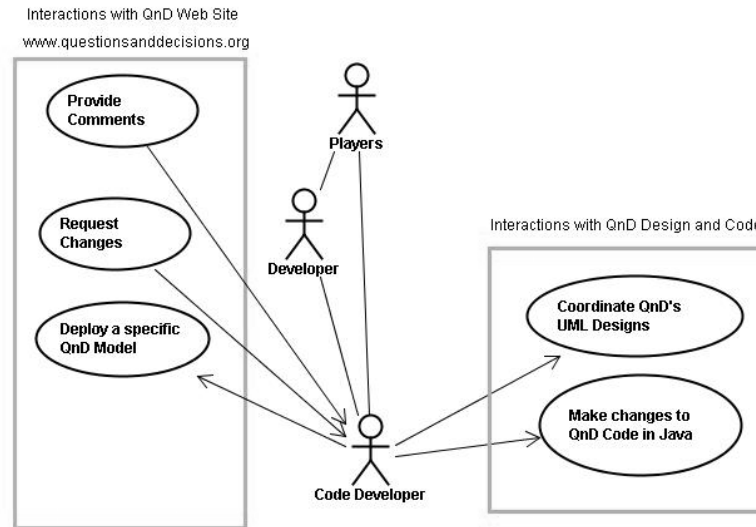


Fig. 10. Use Case diagram of QnD coders.

### 3 Questions and Decisions about Elephant/Vegetation dynamics in the Kruger National Park, South Africa

The two following sections describe how overall the QnD concepts are translated into functional objects for simulation in a game-style format. The first section describes interactions with Kruger National Park, South Africa and other scientists concerning elephant interactions with vegetation management. The latter section provides a description of how these various goals were designed and implemented with an initial version of the QnD system.

The African savanna biome is characterized by the existence of both woody trees and grasses in the same ecosystem; thus, this tree-grass coexistence is of much importance and its preservation is a goal of most conservation authorities [3]. However, there are major concerns in these areas as higher levels of elephant densities could lead to repression of biodiversity which would lead to reduction of levels of specific

woody trees and a decrease overall patch density signaling an increase in homogeneity of ecosystems. The major forces acting upon this coexistence are elephant herbivory, water, nutrients, and stochastic environmental variables such as fire and rainfall [3].

The African elephant (*Loxodonta Africana*) is a major environmental force in this ecosystem because of its behavior, including nutrient cycling, dispersal of plant offspring in seeds, and allowing for new plants to germinate in adequate space [22]. An important behavior that characterizes these populations is that during the summer (dry season), the elephant prefers to graze, or feed on grasses, while this behavior seemingly switches during the wet season, when the elephant turns to browsing leaves off of taller, woody trees [3] [5]. This browsing during the wet season has had a significant effect on woody trees, in that the habitat modification resulting from browsing has been ascribed to loss of canopy trees and an eventual transition to bushland dominated by shorter grasses and shrubs [9]. Thus, high elephant densities could irreversibly alter the structure and physiology of African savanna ecosystems [13]. Significant impacts from elephants upon tree-grass coexistence have been observed in the areas of South Africa, particularly in the northeast portion of the Republic of South Africa, within the Kruger National Park (KNP) [11].

### **3.1 KNP Elephant Model Development Strategies**

The Kruger National Park Elephant Modeling Group (KNP-EMG) is a collection of international modeling researchers working to develop a collection of models to address issues concerning elephant and vegetation dynamics in the park. The KNP-EMG has provided the overall objectives for various modeling studies by issuing various objectives and goals for teams of cooperative elephant modelers including the following texts:

*For the area or the Kruger National Park (~20 000 km<sup>2</sup>), at a spatial resolution of 10 x 10 km, can the*



*model reconstruct the numbers and distribution of elephants over the period 1900 to 2000? Can the same model predict the density distribution 2000-2007 (for which data will be withheld for testing purposes).*

*For the basaltic landscape of Central KNP, what is the shape of the long-term (50 year) tradeoff curve between elephant biomass density and tree cover percentage, for tall trees (>6 m), short trees (2-6m) and shrubs (0-2 m) of the following species: Sclerocarya birrea, Acacia nigrescens and Combretum imberbe?*

Thus, the focus of the KNP-EMG is to create a grid-based model of the KNP, and also formulate a scenario-based simulation to track and predict the progression of these ecological systems. The study groups were directed to develop and test a new elephant-ecosystem models that incorporate important environmental variables and to effectively manage the future populations of the Kruger National Park.

Further informal conversations with various South African scientists and wildlife managers at the KNP Science Networking Meeting 2007 (Skukuza, South Africa) compiled the following tactical ideas for QnD modeling:

1. *Build the initial simulation engine from the "Baxter" model [2][3][4];*
2. *Simply the Baxter model where practical and possible;*
3. *Be clear and transparent as to what assumptions are being made or changed from the original Baxter model design and execution;*

Thus, the purpose of this modeling group is to stimulate and support the development and testing of predictive models of elephant-ecosystem interactions in the KNP. The aim is to help improve the management and monitoring of savanna biodiversity levels, especially that of tree-grass coexistence, in reference to stochastic variables like fire and rainfall, and, most importantly, elephant grazing and browsing. KNP environmental data will be incorporated into the KNP-EMG effort including (*inter alia*) rainfall, elephant densities, woody tree/grass populations, rainfall totals, soil composition and fire occurrence. Thus, the modeling effort will provide a visual

analysis of the interrelationships between selected biotic and abiotic factors in the Kruger National Park.

At this initial model development phase, management options within the KNP are generally limited to those discussed in [17] and articulated within the KNP strategic planning [21]. The options include:

1. *Fire Management – via management fires or fire suppression*
2. *Elephant Population Management - via capture/off-site removal or culling*
3. *Surface Water Management - via opening or closing permanent water points (bore holes)*

### **3.2 *Design2Game: translating systems designs and previous modeling efforts into QnD SimulationEngine and GameView implementations***

The QnD model that has been developed for the initial version (QnD:EleSim) is a grid-based design based on simulation models developed by Baxter & Getz [2, 3, 4]. Collectively, these articles are further referenced simply as the “Baxter” model. According to the QnD modeling group objectives, the Baxter model was simplified in terms of algorithms to provide an initial simulation of elephant-vegetation dynamics over the entire 195,000 km<sup>2</sup> KNP area. This section provides an overview of the initial QnD:EleSim model design with occasional reference to the Technical Appendix at the end of this chapter. Thus, model elements can be viewed in conceptual and mathematical form in the main chapter with selected object representations available for further study in the Technical Appendix.

#### **3.2.1 QnDEleSim SimulationEngine: Setting Spatial and Temporal Execution**

The original Baxter model simulated 1 square kilometer of an African savanna ecosystem by dividing the area into 100, 1 hectare grids and analyzed the effect of elephant herbivory on the tree and fire dynamics against one generic savanna tree species. The QnD:EleSim initial version will simulate the entire KNP with 195, 10 kilometer grid cells (CSpatialUnits). Temporally, the original Baxter model simulated at 6-month intervals which correspond to wet (t, t+2...etc.) seasons, and dry (t+1, t+3...etc.) seasons. To allow for greater control of management activities at the beginning and ending of the wet/dry seasons, the simulation time step in QnD:EleSim was set to one month. All seasonal algorithms were subsequently disaggregated to monthly processes with cognizance to the general wet/dry seasonality.

### 3.2.2 QnDEleSim SimulationEngine: Setting Input Drivers and Scenarios

In terms of environmental factors, the Baxter model assumes that the moisture and nutrient availability are uniform throughout the entire grid system of 1 km<sup>2</sup>. However, in this study, GIS and temporal datasets from the Kruger National Park will be used to incorporate precipitation and the east/west divide between granite-based and basalt-based soils as well as the north/south rainfall gradient [21]. Table 2 provides a summary of spatial and temporal input datasets that are used by QnD for simulation.

**Table 2.** Summary of input information. All maps are 10km grid resolution while all time series files include monthly data. All area maps are fractional coverage.

Input	Format
Rainfall (mm/month)	Time Series Files/Maps/Stochastic Relationships
Fire (intensity, % burnt)	Time Series Files/Maps/Stochastic Relationships
Soils (general class)	Spatial Input Map
Potential Grass Biomass (kg/ha)	Spatial Input Map
Initial Grass Biomass (kg/ha)	Spatial Input Map
Initial Grass Area (fraction)	Spatial Input Map
Initial Woody Seedling Population (#/ha)	Spatial Input Map

Initial Woody Sapling Population (#/ha)	Spatial Input Map
Initial Woody Shrub Population (#/ha)	Spatial Input Map
Initial Woody Tree Population (#/ha)	Spatial Input Map
Initial Woody Seedling Area	Spatial Input Map
Initial Woody Sapling Area	Spatial Input Map
Initial Woody Shrub Area	Spatial Input Map
Initial Woody Tree Area	Spatial Input Map
Initial Elephant Population (#/grid)	Spatial Input Map

### 3.2.3 QnDEleSim SimulationEngine: Setting CLocalComponents, DData and PProcesses

This section describes the implementation of simplified Baxter model concepts into the XML-based, object-oriented structure of the QnD model. As described above, both temporal and spatial scales have been modified from the original concepts to allow for more active management interventions over time and space. The following paragraphs highlight selected elements for further detailed discussion. A more expansive and detailed description of object designs and calculation algorithms is currently being constructed for the more extensive peer review, testing and iteration.

#### 3.2.3.1 Climatic Inputs

Two options are used to the control rainfall inputs throughout the QnD:EleSim program. One option can be used by configuring internal stochastic generator objects using Java objects developed by [11] and applying concepts by [23]. These objects can generate stochastic values according to various user-input statistical properties. The second option can utilize historical KNP climate datasets to directly input desired inputs. Given the modular structure of various CScenario objects, users can combine both time-series datasets and stochastic ge-

erator objects to construct rich and varied scenarios for further simulation.

In the Baxter model, six month rainfall values were normalized with the long term mean value to create a relative rainfall for use in spatial simulations. Given the revised monthly time-step in QnD:EleSim, the monthly relative rainfall (RRfl) was calculated for each spatial area with the following equation:

$$RRfl(t) = \frac{Rfl(t)}{Rfl_{mean}}$$

where

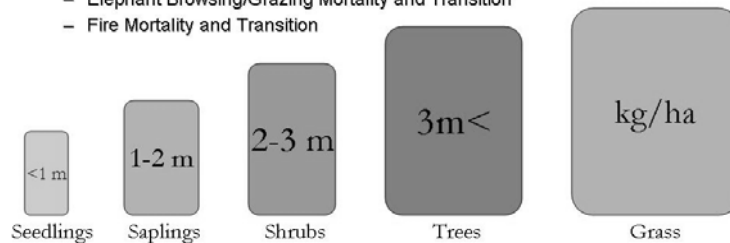
Rfl(t) = rainfall (mm/month)

Rfl<sub>mean</sub> = 30 year monthly average (mm/month)

### 3.2.3.2 Simulating Woody Plant Layer Growth

The original Baxter model incorporated one generic savanna tree species into nine different size classes (i = 1...9) ranging from Seedlings (< 15 cm height) to Woody trees (> 5m height). The tallest woody tree groups were considered fire-resistant. In order to simplify the Baxter model for initial testing with KNP datasets and woody species, QnD:EleSim uses 4 different woody height classes were constructed from the original nine functional groups. As with the original Baxter model, the grass layer is represented with one class. This object layout is illustrated in Figure 11.

- Woody Plants – 4 CLocalComponents based on height
- Grasses – 1 CLocalComponent based on biomass
- Processes
  - Woody Plant Growth and Mortality (Area and Population)
  - Grass Growth and Mortality (Area and Biomass)
  - Elephant Browsing/Grazing Mortality and Transition
  - Fire Mortality and Transition



**Fig. 11. Conceptual diagram of vegetation-related CLocalComponent objects present in each of the 195 CSpatialUnit/CHabitat objects.**

This simplified structure was designed to allow linkage with limited KNP woody vegetation datasets as well as the inclusion of additional woody species in future model iterations. If additional sub-divisions are required, QnD's modular structure can quickly be re-expanded to simulate all 9 original Baxter size classes.

### 3.2.3.3 Wet and Dry Season Dynamics

Basic interaction diagrams for both wet and dry seasons are presented in Figures 12 and 13. All plant growth occurs in the wet season (October – April) with fires less prevalent. A season's growth is closely related to rainfall and soil conditions. As the wet season draws to a close in April, overall senescence begins within grass layers and woody layers tend to end active growth. Woody plant mortality in each size class is assumed to come through a base mortality level, fires and elephants. As woody plants eventually progress into the dry season, the Baxter model proposes that lower levels of rainfall are said to induce the death of some woody plants. Similarly, the prevalence of fires during the dry season leads to an average height decrease during these summer months ([10] p 301). Also, elephant grass grazing occurs during the dry season, resulting in deaths and biomass decreases throughout.

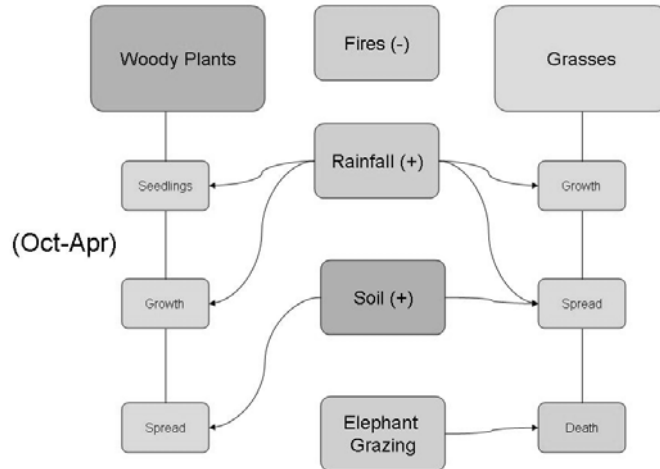


Fig. 12. Conceptual diagram of wet season dynamics as adapted from the Baxter model.

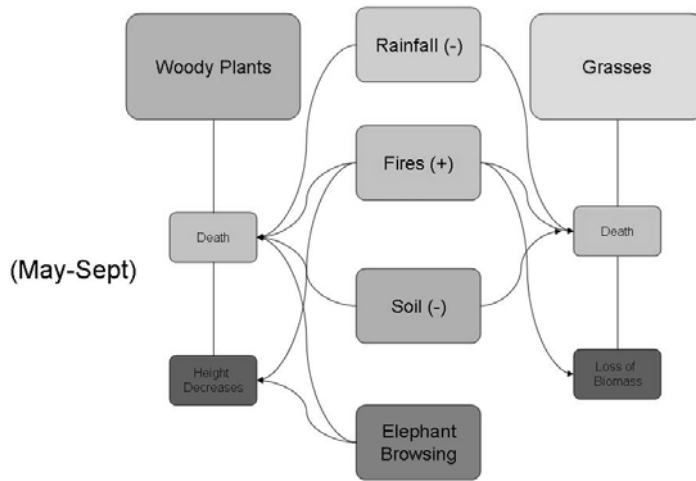


Fig. 13. Conceptual diagram of dry season dynamics as adapted from the Baxter model.

A simplification of concepts from the Baxter model was instituted in the accounting of the relationships between the four woody tree metaclasses (Figure 14). It is accepted that three primary forces: elephant effects, fire effects, and growth ef-

fects, can dramatically change the dynamics of the various populations of trees. Therefore, a system of nine “transition factors” has been set up to account for addition and loss of biomass to individual groups. For example, Growth Transition Factor 1 (GTF1) adds a certain amount of population and cover yearly to the sapling metaclass (because of upward growth) while subtracting that amount of population and cover from the seedling metaclass. Similarly, Elephant Transition Factor 1 (ETF1) subtracts a certain amount of population and cover from the sapling metaclass (because of browsing’s effects on reducing average tree heights) and adds that amount of population and cover to the seedling class. Finally, for example, Fire Transition Factors (FTFs) behave the same as ETFs in that they subtract population and cover from the “higher” metaclasses and add that amount of population and cover to the “lower” metaclasses. Figure 14 illustrates the conceptual design in which transitions between woody size classes are simulated. Transitions from smaller groups to larger groups are simulated through woody plant biomass and population transitions while effects of fire and elephant browsing create transitions from larger to smaller size groups. The following section provides a more detailed description of selected algorithms used in the QnD SimulationEngine. Woody seedling establishment was simulated with the following equation:

$$Rc_{seedl}(t) = RRfl(t) \bullet RcPot_{seedl}$$

where

RRfl(t) = Monthly Relative Rainfall (unitless)

RcPot<sub>seedl</sub> = potential seedlings (mm/month)

Area-based competition ( $Comp_i(t)$ ) from existing woody plants within the same size group or higher is included in the equation:

$$Comp_i(t) = 1 - \sum_i^{i=4} RCvr_i(t)$$

where

RCvr(t) = relative area coverage (fraction) of woody species (seedlings, saplings and shrubs). *Note: In calculating the competition coefficient for the smallest woody size category (CSeedlings, i=1), the relative area cover by grass is included in the summation.*



For saplings, shrubs, and trees, the adjusted growth rate (GTF) is described from

$$GTF_i(t) = RRfl(t) \bullet GTFbase_i(t) \bullet Comp_i(t) \bullet Pop_i(t)$$

where

RRfl(t) = Monthly Relative Rainfall (unitless)

GTFbase<sub>i</sub> = base growth transition rate for each woody size class (i=1..4)

Comp<sub>i</sub> = area-based competition from existing woody plants

Pop<sub>i</sub>(t) = the current population of woody plants in size class i

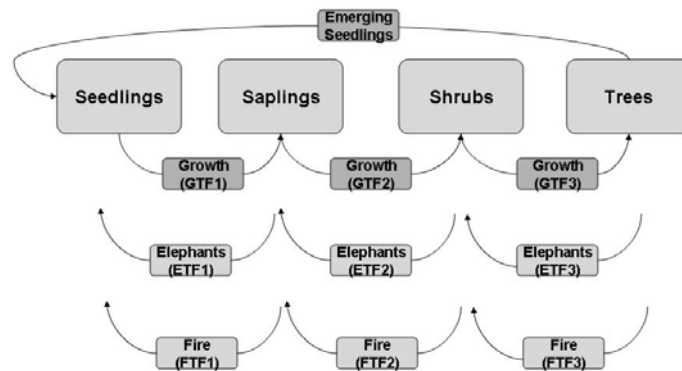


Fig.11 .

A conceptual diagram of transition factors involved in woody plant size classes.

### 3.2.3.4 Simulating Grass Layer Area and Biomass

The following section provides a more detailed description of selected grass growth algorithms used in SimulationEngine. An example configuration of Grass Layer PProcess and PSubProcess objects (along with concomitant input and output DData objects) is found in the Technical Appendix. As before, grass biomass and cover growth are simulated based on the original Baxter model concepts with some alterations to match up with the KNP Vegetative Composition Assessment (VCA) database on the grass layer. During the rain season (October – April), higher rainfall levels result in growth of grass populations though limited by competition with woody trees results in higher death rates.

Monthly fractional area of grass cover ( $Area_{gr}$ ) is calculated through the equation:

$$Area_{gr}(t+1) = RRfl(t) \left( 1 - \sum_{i=1}^3 RCvr(t) \right) (1 - RGraz(t)) \left( 1 - \sum_{i=2}^3 RCrowd(t) \right) \text{ where}$$

$S_{gr}^w$  = wet season survivability (fraction),  
 $RRfl(t)$  = monthly relative rainfall (fraction),  
 $RCvr(t)$  = relative area coverage (fraction) of woody species (seedlings, saplings and shrubs),  
 $RGraz(t)$  = proportion of grass in the grid cell grazed by elephants (fraction),  
 $RCrowd(t)$  = proportion of crowding from saplings and shrubs (fraction),  
 For calculating monthly wet season grass biomass (kg/grid) ( $Bio_{gr}$ ) the following equation is used:

$$Bio_{gr}(t+1) = S_{gr}^w RRfl(t) \left( 1 - \sum_{i=1}^4 RCvr_i(t) \right) (Bio_{gr}(t) + Pr_{gr} Area_{gr})$$

where

$S_{gr}^w$  = wet season survivability (fraction),  
 $RRfl(t)$  = monthly relative rainfall (fraction),  
 $RCvr_i(t)$  = relative area coverage (fraction) of woody species (seedlings, saplings, shrubs and trees),  
 $Bio_{gr}(t)$  = grass biomass (kg),  
 $Pr_{gr}$  = monthly wet season grass productivity (kg/grid) =  
 (Total wet season potential productivity (kg/ha)/ 6 months)  
 $Area_{gr}$  = total grid area covered by grass (ha)

During the dry season (May – September), grass biomass or area growth does not occur while both area and biomass can be reduced by either fire or elephant grazing.

### 3.2.3.5 Simulating Elephant Populations

In the initial version of QnD:EleSim, elephants are conceptualized in two types of CLocalComponent objects: the Elephant Herd (a meta-population class object which simulates a herd and its cell-to-cell movements) as well as the Tagged Elephant, in which one specific elephant will be “virtually tagged” and documented separately to simulate more individual reactions to ecological phenomena. Main processes undergone by

elephants are grass grazing, tree leaf browsing, as well as birth, death, and a design of cell-to-cell movements. As described above, elephant herds (CElephantHerd instances) are simulated as metapopulation classes existing temporarily in a 10km x 10km grid cell for the entire month before having the option to move to another grid cell that has more desirable qualities in terms of vegetation or water. In reference to elephant movement between the spatial units, initial process designs include the calculation of Habitat Suitability Indices (HSIs) [8] [30], for each of the 195 cells, based on factors such as woody tree biomass, grass biomass, and distance from watering holes. HSI values will provide a grid weighting for movement probabilities. Thus, elephant herd objects will be “set in motion” as they navigate through various spatial units. This design allows algorithms for elephant movement to be tested ranging from pure random walks to probability weighted directional walks.

#### 3.2.3.6 Simulating Fire

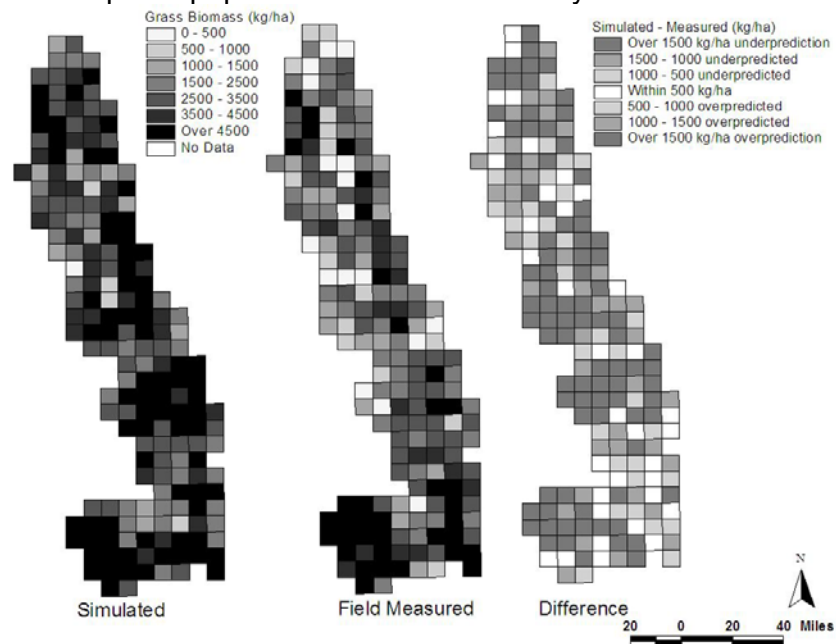
In reference to fire processes for burning vegetation, fire CLocalComponents were designed and implemented as if they were “ephemeral herbivores” consuming both area and population for woody size classes as well as area and biomass for the grass layer class. In some object designs showing a natural fires only management scenario, fire, grass and tree biomasses will be calculated and assigned to each spatial unit (0.0 to 1.0) and the fires will be randomly started within the grid. If the fire ignites in a specified cell, all linked cells will have the “opportunity” to catch fire; the adjacent cells with the higher biomass levels will have a higher probability to catch fire next. Other management scenarios call for spatially explicit fire scheduling and placement, these fires are simulated through the use of input maps and thus fires in QnD:EleSim can be stochastic or user-defined.

#### 3.2.4 QnDEleSim GameView: Setting the User Interface

As seen in Figure 3, the initial version of the QnD:EleSim, GameView includes a base map of the 10 km grid combined with selected additional GIS layers for added reference. User-selected, radio-button maps display various spatial DData values for consideration. Tabbed display panes along the top of the map allow the user to view the user interface for QnD:EleSim version 0.0 will model a simulation game-based environment. In the QnD GameView, a large map of the Kruger National Park (divided into 195 numbered 10km square grids) will present itself, and various options can be selected from for viewing by selection of various radio buttons. For example, on the left hand side, viewing of elephant densities, grass and tree biomasses, and rainfall totals will describe various phenomena using user-selected color gradients. In the lower right hand corner, a pie chart (specific to each spatial unit selection) will appear that contains a breakdown of the area cover for grasses, seedlings, saplings, shrubs, and trees in the cell. At the top-left of the screen there will be a "play" button, which signifies running of either a one-month or a six-month timestep, and thus, changes can be observed visually (in the map) or by selecting one of various X-Y charts which plots the abiotic and biotic factors of the KNP over time. As warning signs, red lights will appear next to "Rainfall," "Grasses," "Seedlings," "Saplings," "Shrubs," "Trees," and "Elephants." if any levels reach the KNP's defined levels of Thresholds of Potential Concern (TPC). Yellow lights will warn the user before TPCs, but as TPCs are attained, a red light will flash. Since the purpose of this QnD model is to create a management-based simulation of the KNP, to manage the ecosystem, toggle bars at the top allow the user to increase or decrease water point openings, decide whether to blot out or set off random fires, and eventually, whether or not to capture or cull elephants.

### 3.3 Ongoing QnD:EleSim Calibration and Validation Activities

At this point in the design and testing process, QnD:EleSim v.0.0 has produced useful preliminary results and more advanced calibration/validation/testing is underway. Figure 12 provides an example results map of the 195 Kruger National Park spatial units, and documents “difference figures” between 1990 QnD simulation data of grass biomass and 1990 grass biomass data obtained from the Kruger National Park VCA. While the QnD:EleSim model tended to over-predict many cells’ grass densities, the effect of fires and moving elephants are not included in this figure, thus a bias towards over-prediction is expected until fire and elephant dynamics are finalized. Thus, the grass biomass local component of the QnD model has been initialized and gives plausible early results against historical KNP data. Continued testing, calibration and validation with monitored grass, woody vegetation and elephant population data are underway.



**Fig. 11.** Example map of KNP observed and initial QnD-simulated data (without fire and elephant effects) for 1990 grass biomass levels.

### **3.4 Serious Play: Playing games for systematic analysis**

Playing management games for heuristic development and exploration was the original goal of QnD development and use. Recent research efforts have expanded this role to include systematic analysis of sensitivity and uncertainty within these complex systems. Kiker et al. [19] presents a conceptual design for linking simulation, sensitivity and decision analysis tools together in service of adaptive management of transboundary water issues. The software package, SIMLAB v2.2 [29] is used in the global sensitivity and uncertainty analysis of the Okavango QnD application. SIMLAB is designed for pseudorandom number generation-based uncertainty and sensitivity analysis. SIMLAB's Statistical Pre-Processor module provides potential parameter values using user-defined on Probability Distribution Functions (PDFs) to produce a matrix of sample inputs to run the QnD model. The QnD code has been altered to allow the incorporation of SIMLAB-derived matrices for automatic simulation. The program automatically substitutes the new parameter set into the input files, runs the model, and performs the necessary post-processing tasks to obtain the selected model outputs for the analysis. The outputs from each simulation are stored in a matrix containing the same structure as the samples generated by SIMLAB. With the input and output matrices, SIMLAB is used to calculate the sensitivity indexes of the Morris [25] and extended FAST method [9] [20]. Finally the output probability distributions are constructed in SIMLAB based on the set of variance-based sensitivity run results to systematically quantify the uncertainty inherent within model parameters and its output. Given that models being applied to large complex environmental problems are often challenged by various groups to substantiate results and/or predictions, these model analysis tools are in-

creasingly being used to scrutinize various assumptions of the model and the inherent limitations of the environmental data being used to drive them.

#### **4 Conclusions**

The QnD modeling software and its associated development methodology was created to quickly and efficiently construct a management/stakeholder-relevant model that integrates both explicit scientifically-derived data and expert/anecdotal knowledge. Given QnD's object-oriented design and XML-based input files, systematic iteration with stakeholders is encouraged and promoted. New and novel ideas about the problem and potential solutions can be explored, adopted or discarded to promote greater system learning.

Development of a QnD model is undertaken within a larger context of stakeholder engagement and public participation. When eliciting information to build QnD scenarios, many different perspectives are expressed, each with its own assumptions about cause-effect relationships and beliefs about what potential interventions would constitute ecosystem improvement. The development process which involves actively working with stakeholders to build the model, play the game, and revise the model is undertaken within a soft systems approach [7]. The soft systems approach distinguishes the QnD gaming and scenario-building process from the more traditional use of models simply as system predictors. The QnD development process can accommodate both hard data, such as field-measured experiments, and soft data, such as experiential learning, impressions or general "rules of thumb". The model is used to facilitate dialogue and learning about the factors that influence the environmental system under consideration, and to explore potential management actions.

This research has designed, implemented and tested an elephant/vegetation/fire model in an attempt to assess the KNP biodiversity challenges and how managers can begin to address ecosystem management. After the calibration period, a

global sensitivity analysis using the program SIMLAB will be coupled with the QnD:EleSim model to test how sensitive the model is to changes in environmental phenomena and input parameters.

The applications for this project are widespread, in that after validation, it could potentially be used as a management tool for the Kruger National Park scientists or managers in reference to when potential Thresholds of Potential Concern will approach, and what types of management actions should take place. In the future, this QnD model can be improved upon by validating predictions of elephant and woody plant densities by comparison to the rich monitoring datasets available in the KNP. Thus, QnD:EleSim version 0.0 has been created as one of the many tools available to scientists and managers for the addressing the complex savanna biodiversity challenges in the Kruger National Park, South Africa. Continued changes in the object structure and the strategic implementation of QnD are fully expected as KNP managers adaptively learn and test various strategies to answer questions and make decisions within this complex environmental system.

## **5 Acknowledgements**

Special thanks to the following people/organizations that allowed for the development of this model to be made possible: Dr. Judith Kruger, Dr. Rina Grant and Dr Harry Biggs of the South Africa National Parks for supplying climate, soils, vegetation, animal census, fire and other KNP datasets. In addition, their guidance and friendship are truly appreciated.

The KNP Elephant Modeling Group (chaired by Prof Robert Slotow and Dr Robert Scholes) for coordination of elephant modeling research activities.

The South African Weather Bureau for provision of climate data.

Prof. David Saah of the University of San Francisco for his input and comments.



University of Florida Center for Precollegiate Education and Training for organizing and supporting this research opportunity for Mr Thummalapalli.

## 6 References

- [1] Aldritch, C. 2005. Learning by Doing: A comprehensive guide to simulations, computer games, and pedagogy in e-learning and other educational experiences. John Wiley & Sons. ISBN: 0-7879-7735-7.
- [2] Baxter, P.W.J. (2003). Modeling the Impact of the African Elephant, *Loxodonta Africana*, on Woody Vegetation in Semi-Arid Savannas. PhD Dissertation, University of California, Berkeley.
- [3] Baxter, P.W.J. & Getz, W.M. (2005). A Model-Framed Evaluation of Elephant Effects on Tree and Fire Dynamics in African Savannas. *Ecological Applications*, 15, 1331-1341.
- [4] Baxter, P.W.J. & Getz, W.M. (2006). Development and Parameterization of a Rain- and Fire-driven Model for Exploring Elephant Effects in African Savannas. *Springer Science*, 9, 2-23.
- [5] Buss, I.O., & Smith, N.S. (1966). Observations on Reproduction and Feeding Behavior of the African Elephant. *The Journal of Wildlife Management*, 14, 375-388.
- [6] Carpenter, S., Brock, W. & Hanson, P. (1999). "Ecological and social dynamics in simple models of ecosystem management." *Conservation Ecology* 3(2): 4 [online].
- [7] Checkland, P.B. (1999) Soft systems methodology in action, Toronto: John Wiley & Sons.
- [8] Cook, J. G. and L. L. Irwin. (1985). Validation and modification of a habitat suitability model for pronghorns. *The Wildlife Society Bulletin* 13: 440-448.
- [9] Cukier, R. I., H. B. Levine, and K. E. Schuler. 1978. Nonlinear sensitivity analysis of multiparameter model systems. *Journal of Computational Physics* 26: 1-42.
- [10] Du Toit, J.T., Rogers, K.H., & Biggs, H.C. (2003). *The Kruger Experience: Ecology and Management of Savanna Heterogeneity*. Washington, D.C.: Island Press.
- [11] Dorai-Raj, S. 2000. Java Random Number Generator. <http://152.3.140.5/~dsv/compbio/ppstuff/ppinteraction/RandomNumberGenerator.java> [Last Accessed Dec 4, 2007]
- [12] Estes, Richard. (2004). *The Safari Companion: A Guide to Watching African Mammals*. White River Junction, Vermont: Chelsea Green Publishing Company.

- [13] Holling, C.S., Gunderson, L.H. and Peterson, G. 2002. Sustainability and Panarchies. (Chapter in Gunderson, L.H. and Holling, C.S., (Eds) "Panarchy: Understanding Transformations in Human and Natural Systems." Island Press. Washington DC. Pp:63-102.
- [14] Holdo, R.M. (2003). Woody Plant Damage by African Elephants in Relation to Leaf Nutrients in Western Zimbabwe. *Journal of Tropical Ecology*, 19, 189-196.
- [15] Jeppesen, E. I., Iversen, T.M. (1987). "Two simple models for estimating daily mean water temperatures and diel variations in a Danish low gradient stream." *Oikos* 49: 149-155.
- [16] Justice, L.C., Kiker, G.A. and Kiker, M.K. (2007, in press). Jamaican food security in a dynamic world: collaborative methods to envision food system models. *Caribbean Journal of Geography*.
- [17] Kiker, G.A., Rivers-Moore, N.A., Kiker, M.K., & Linkov, I. (2006). QnD: A Scenario Based Gaming System for Modeling Environmental Processes and Management Decisions. *Springer Science*, 11, 151-186.
- [18] Kiker, G.A. and Linkov, I. (2006). The QnD Model/Game System: Integrating Questions and Decisions for Multiple Stressors. (Chapter in Arapis, G., Goncharova, N. and Baveye, P. (Eds) "Ecotoxicology, Ecological Risk Assessment and Multiple Stressors " Springer, Netherlands. Pp:203-225.
- [19] Kiker, G.A., Muñoz-Carpena, R., Wolski, P., Cathey, A., Gaughn, A. and Kim, J. (2007, in press). Incorporating Uncertainty into Adaptive, Transboundary Water Challenges: a conceptual design for the Okavango River Basin. *International Journal of Risk Assessment and Management*.
- [20] Koda, M., G. J. McRae, and J. H. Seinfeld. 1979. Automatic Sensitivity Analysis of Kinetic Mechanisms. *International Journal of Chemical Kinetics* 11: 427-444.
- [21] Kruger National Park (KNP). (2006). KNP Park Management Plan Version 1 (31 October 2006). [Online: [www.sanparks.org/conservation/park\\_man/kruger.pdf](http://www.sanparks.org/conservation/park_man/kruger.pdf) Last accessed Dec 5, 2007].
- [22] Laursen, L. & Bekoff, M. (1978). *Loxodonta Africana*. *Mammalian Species*, 92, 1-8.
- [23] Matsumoto, M. and Nishimura, T. 1998. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Transactions on Modeling and Computer Simulation*, Vol. 8(1), 3--30.
- [24] Morgan, M.G., Fischhoff, B., Bostrom, A. & C.J. Atman, C.J. (2002) *Risk Communication: A Mental Models Approach*. Cambridge University Press.
- [25] Morris, M. D. 1991. Factorial sampling plans for preliminary computational experiments. *Technometric* 33: 161-174.

- [26] NRC. 1991. *Opportunities in the Hydrologic Sciences*. National Academy Press, 348 pp.
- [27] Papajorgji, P.J. and Pardalos, M. (2005). *Software Engineering Techniques Applied to Agricultural Systems: An Object-Oriented and UML Approach*. Springer-Verlag. ISBN: 0387281703; 247 pp.
- [28] Rittel, H. and M. Webber. 1973. Dilemmas in a General Theory of Planning. *Policy Sciences*. Vol.4: 155-169.
- [29] Saltelli, A., S. Tarantola, F. Campolongo, and M. Ratto. 2004. *Sensitivity Analysis in Practice: A Guide to assessing Scientific Models*. 219 pp., John Wiley & Sons, Ltd, Chichester, England.
- [30] Schamberger, M., A. H. Farmer, and J. W. Terrel. (1982). Habitat suitability index models: introduction. U.S.D.I. Fish and Wildlife Service. FWS/OBS-82/10. 2 pp.
- [31] Starfield, A. M. and Bleloch, A.L. (1991). *Building Models for Conservation and Wildlife Management*. Burgess International Group, Edina. MN, USA.
- [32] Van der Heijden, K (1996) *The Art of Strategic Conversation*. John Wiley & Sons, New York.
- [33] Walters, C., Korman, J., Stevens, L.E. & Gold, B. (2000). "Ecosystem modeling for evaluation of adaptive management policies in the Grand Canyon." *Conservation Ecology* 4(2): 1 [online].
- [34] Wu, J. and David, J.L. 2002. A Spatially Explicit Hierarchical Approach to Modeling Complex Ecological Systems: Theory and Applications. *Ecological Modelling* 153:7-26.
- [35] Wu, J. and Loucks, O.L. 1995. From Balance of Nature to Hierarchical Patch Dynamics: A Paradigm Shift in Ecology. *The Quarterly Review of Biology* 70(4):439-466.
- [36] Yoe, C. (2002). *Tradeoff Analysis Planning and Procedures Guidebook*. U. S. Army Corps of Engineers, Institute of Water Resources Report (IWR 02-R-2). Prepared for U. S. Army Corps of Engineers, Institute of Water Resources by of Planning and Management Consultants, Ltd. Contract # DACW72-00-D-0001. <http://www.iwr.usace.army.mil/iwr/pdf/tradeoff.pdf>

## 7 Technical Appendix

This section contains a sample list of objects used in the Grass CLocalComponent (CGrass) object. The CGrass object exists within each of the 195 grid cells. In addition, the four woody vegetation objects (CSeedlings, CSaplings, CShrubs and CTrees), potential fire objects (CFire) and elephant objects (CElephantHerd and CTaggedElephant) are also present in grid cells. While they have a similar DData/PProcess object structure as CGrass, they are not listed in this appendix. Overall in QnD:EleSim, approximately 7 CLocalComponent objects, 23 Process objects, 88 Sub-Process objects and hundreds of DData objects were programmed into each of the 195 CSpatialUnit/CHabitat combinations via the XML input files.

### Grass object (CGrass) DData and PProcesses

DData		
DBiomass	DProductivity	DNewAreaCovered
DAreaCover	DGrassBiomassSenesced	DBiomassPerUnitArea
DBaseProductivity	DWetSeasonSenescence	DNewGrassBiomass
DGrassValue1	DDrySeasonSenescence	DRelativeAreaCoveredBySmallWoodyPlants
DGrassValue2	DGrassCrowdingCoefficient	
DGrassValue3	DMaxBiomass	
DGrassValue4	DRelativeGrazingIntensity	
DGrassValue5	DBiomassAdded	
DGrassValue6	DAreaCovered	

### Process / SubProcesses / DData

#### PProcess: PWetSeasonProcessesCalculateGrassAreaCoveredAndBiomass

**PSubProcess:** PIfWetSeason (Type = PIfEquals)

If (Global.DWetSeason == 1) **Then** continue to next PSubProcess - **Else** Go To Next Process

**PSubProcess:** PCalcProductivity (Type = PMultiplyValue)

CGrass.DProductivity = Global.GridArea x CGrass.DBaseProductivity

**PSubProcess:** PCalcMonthlyProductivity (Type = PDivideValue)

---

$CGrass.DProductivity = CGrass.DBaseProductivity / Global.DSix$   
**PSubProcess:** PCalcAvailableAreaCover (Type = PSubtractValue)  
 $CGrass.DGrassValue1 = Global.DOne - CSeedling.DRelativeAreaCover -$   
 $Csapling.DRelativeAreaCover -$   
 $CShrub.DRelativeAreaCover$   
**PSubProcess:** PCalcInverseOfElephantGrazingIntensity (Type = PSubtractValue)  
 $CGrass.DGrassValue2 = Global.DOne - CElephan-$   
 $tHerd.DRelativeGrazingRate$   
**PSubProcess:** PCalcInverseOfCrowdingIntensity (Type = PSubtractValue)  
 $CGrass.DGrassValue3 = Global.DOne - CSapling.DCrowdingCoefficient -$   
 $Cshrub.DCrowdingCoefficient$   
**PSubProcess:** PCollectingTerms (Type = PMultiplyValue)  
 $CGrass.DAreaCovered = HomeSpatialUnit.DLocalRainfall x$   
 $CGrass.DGrassValue1 x CGrass.DGrassValue2 x$   
 $CGrass.DGrassValue3$   
**PSubProcess:** PCalculateGrassSenesced (Type = PMultiplyValue)  
 $CGrass.DGrassBiomassSenesced = CGrass.DWetSeasonSenescence x$   
 $CGrass.DGrassBiomass$

**PProcess: PCalculateDrySeasonGrassSenescence**

**PSubProcess:** PIfDrySeason (Type = PIfEquals)  
**If** (Global.DWetSeason == 0) **Then** continue to next PSubProcess - **Else** Go To  
 Next Process  
**PSubProcess:** PSetNewBiomassToZero (Type = PSetValue)  
 $CGrass.DNewGrassBiomass = Global.DZero$   
**PSubProcess:** PCalcInverseOfElephantGrazingIntensity (Type = PSubtractValue)  
 $CGrass.DGrassValue2 = Global.DOne - CElephan-$   
 $tHerd.DRelativeGrazingRate$   
**PSubProcess:** PCalculateFireIgnition (Type = PSubtractValue)  
 $CGrass.DGrassValue6 = Global.DOne - CFire.DFireIgnition$   
**PSubProcess:** PCalculateGrassFireLoss (Type = PMultiplyValue)  
 $CGrass.DNewGrassAreaCovered = CGrass.DGrassValue6 x$   
 $CGrass.DAreaCovered$   
**PSubProcess:** PCalculateDrySeasonGrassSenescence (Type = PMultiplyValue)  
 $CGrass.DGrassBiomassSenesced = CGrass.DDrySeasonSenescence x$   
 $CGrass.DGrassValue6 x CGrass.DGrassBiomass$

**PProcess: PCalculateFinalGrassBiomass**

**PSubProcess:** PCalculateFinalGrassGrowth (Type = PAddValue)  
 $CGrass.DGrassBiomass = CGrass.DGrassBiomass + CGrass.DGrassBio-$   
 $massSenesced + CGrass.DNewGrassBiomass$   
**PSubProcess:** PCalculateNewBiomassPerUnitArea (Type = PMultiplyValue)  
 $CGrass.DBiomassPerUnitArea = CGrass.DGrassBiomass / Global.DGridArea$

---

