AZ-SMART Architecture Draft

Center for Urban Simulation and Policy Analysis
University of Washington

March 26, 2007

1 Introduction

This document develops an initial architecture plan for the AZ-SMART project. It is intended to provide an overview of the architecture from multiple perspectives: system, software, user interface, data and models. Design choices are generally provided as recommendations, but in some cases two or more alternatives are presented for further discussion. The intent is to update this document as design choices are made so that it can serve as an ongoing documentation effort. It is very unlikely that the final AZ-SMART application will implement this plan exactly. Rather, it provides a point of departure, and a general map for the intended application, which will be adapted continuously as the project participants develop, use, test, and refine components and their interactions and interfaces. Note that this document proposes an architecture for the full AZ-SMART project, including models and data structures that are anticipated for implementation after Phase 1. Clarification on the intended phasing is provided in the body of the document.

2 Functional Overview

The functional overview of the AZ-SMART system is depicted in Figure 1, showing multiple activities connected in a sequential but also bi-directional workflow. One can begin with the top-right of the diagram and work counter-clockwise to understand the intended workflow pattern. An AZ-SMART user will begin with the GIS tasks of compiling a database in the geodatabase that will contain all the needed information for the AZ-SMART system, with the exception of the travel model and a mid-level model that will be interfaced to AZ-SMART (in a later phase, the mid-level model is likely to become a component of AZ-SMART rather than an external process).

As the database is assembled in the geodatabase, including all land use layers, environmental, planning and political boundaries, development projects, and other related inputs, the AZ-SMART user will employ a suite of diagnostic tests to examine the database for missing, erroneous, and inconsistent data, and will use editing and validation tools to repair problems, and impute values for missing and erroneous data. During this process, there would be considerable iteration between the data assembly step and the data diagnostic step, as some input data might be replaced or updated. This step would also involve the assembly of metadata concerning the data sources, vintage, and any information developed on data quality.

The next step in the workflow is the development of models that are consistent with the database and can be used to allocate land uses to small areas (we discuss the form of the data in detail in the data subsection). The models used by AZ-SMART will be configured to undertake land use allocation as described in the model subsection, and will be flexible with respect to the variables and weights (parameters) used. The models themselves will be modular and flexible to allow ongoing refinement over time by AZ-SMART users as they gain insights into alternative means of doing the allocations. This step also involves selecting the variables to use in each model component, and estimating model parameters statistically, or if desired, inputting parameters manually for some model components. The tools for model estimation will be internal to the system, allowing ease of re-estimation if the user wishes to experiment with alternative specifications. Diagnostics from model estimation, such as sensitivity to variables and correlation among variables, will be available to assess the robustness and sensitivity of each model component. The user will also be able to compare observed to predicted data over an observed period if such data is available. This model specification and calibration process may identify problems in the data, which will require moving back upstream in the workflow process to refine input data.

Once the models are estimated and the system calibrated to the users' satisfaction, the model system may be put into use. Putting the model into use requires the specification of one or more 'scenarios' that contain the input data that define the input conditions and assumptions for a run of the model system, including control totals, land development constraints, development projects, and transportation system. These scenarios may involve graphical editing in the GIS environment, to edit boundaries of designated areas or to generate buffers or distances from specified features such as access points on the transportation system. The scenario specification will also include controls for the interfacing of the travel model system (which years to run, what variables to pass, for example), and the mid-level land use model system or the control totals that have been generated by them.

The next step in the workflow is to run one or more scenarios over a forecast period defined by the scenarios. This may be done in a batch mode that automates the entire modeling process from the loading of initial data for the base year, through multiple time steps and exchanges with the travel model, and possibly the mid-level model, until completion of the entire modeling period of 30 or more years (for example). In other cases, the user may wish to run a scenario only for a portion of the intended forecast period, then stop the simulation to examine results, and possibly to make changes in the scenario assumptions that will shape the simulation in the next section of the forecast period. Yet another possibility is that the user will see patterns in the results that are not desired, and decide to back up to the scenario specification stage, or even to the model specification stage, to make refinements and resume the process.

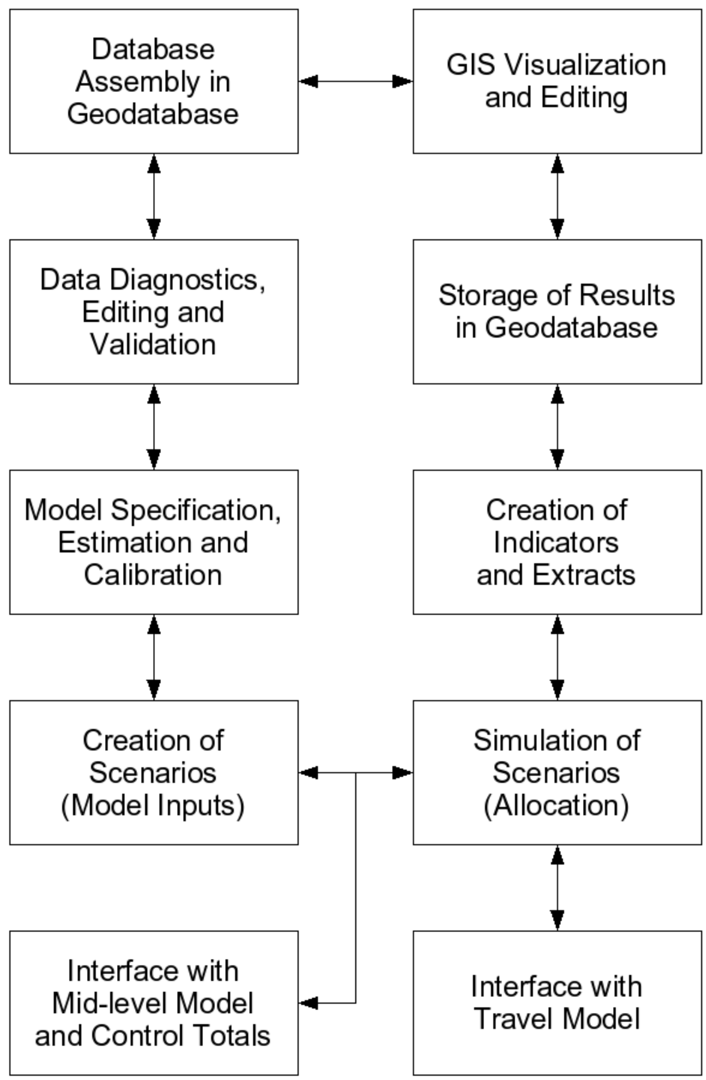

Figure 1: AZ-SMART Workflow Diagram

Typically the process of examining simulation results will involve creating indicators and extracts (or summaries) of the detailed simulation results. We shall refer broadly to these as indicators. The indicators may be simple aggregation of results, such as housing units or acres of each land use sector by zone, or RAZ. Or they may involve computations that are more complex, such as estimating the remaining development capacity within each city. The indicators may be produced as maps, or charts, or tables. They may be of three types: snapshots (looking at an indicator for a single scenario and year), changes (usually from the base year to some forecast year), and differences (usually between a specific scenario and a baseline scenario for a selected year). It is likely that many of these would be map indicators and be viewed in the GIS environment.

Simulation results may also be stored to the geodatabase. A user may wish to store either the full set of simulation results with all of the intermediate calculations needed to understand or even re-start a simulation from an intermediate year, or alternatively only a subset of specified results can be written to the geodatabase for archiving and further visual examination. The system will provide tools for both use cases.

This brings the workflow full circle to the top-right, where the data used as inputs and the data produced by the simulation may all be visually examined and edited. Refinement of forecast results would be done at this stage also. There are of course many possible variants on this workflow description, and theoretically one could draw connections among almost any pair of these task components, and the AZ-SMART system will support this kind of flexibility (but the diagram would be quite hard to interpret at that point). The next section moves to the system architecture perspective.

3 System Overview

This section provides a system perspective of AZ-SMART. The system is a distributed configuration as depicted in Figure 2, in which there are possibly different servers for the geodatabase (ArcSDE and MS SQL Server), the land use model (OPUS), the travel model, and potentially a web server (using ArcIMS or its successor). Initially we expect that OPUS and ArcGIS will be running strictly on the same machine.

The interface between ArcGIS and OPUS will be integrated through the ArcGIS user interface (described in more detail in the subsection on user interface). There may be situations in which a user wishes to run an OPUS client independently of ArcGIS, for example when focusing on model estimation, or at times when controlling simulation runs, or in the use of a fully automated batch simulation linking the land use and travel model systems. Similarly, a user might want to have multiple OPUS Clients on different machines, but share an OPUS Server on another machine, to take advantage of a fast server, for example. The interface between OPUS clients and servers would use networking infrastructure, though there may be no need for this infrastructure during the first phase of this project.

In the first phase, the OPUS client and server will be on the same machine as an ArcGIS workstation, with multiple installations, one per user (or machine). Data will be shared via the geodatabase, and if desired also through a drive that is mapped from each of the clients to share the cache directory containing simulation results.

4 User Interface

In this section we describe the principal user interface components and organization of the AZ-SMART system. AZ-SMART will include three modules (Project Manager, Data Manager, and Tool Manager) based on the AZ-SMART RFQ Appendix G. Each module's organization and implementation are described in turn in the following subsections.

4.1 Data Manager

The Data Manager will focus on the management and visualization of data within AZ-SMART. The Data Manager will be implemented within the ArcGIS framework as a dockable window, coded in VB.NET within ArcObjects, that organizes the necessary data management tools in a 'tree-view' framework. The Data Manager dockable window would be accessible within both ArcMap and ArcCatalog to provide flexibility in use, although in regular production operation it would probably make the most sense to utilize the Data Manager primarily within ArcCatalog. See Figures 3 and 4 for example screen captures of this type of dockable window implementation in both ArcCatalog and ArcMap.

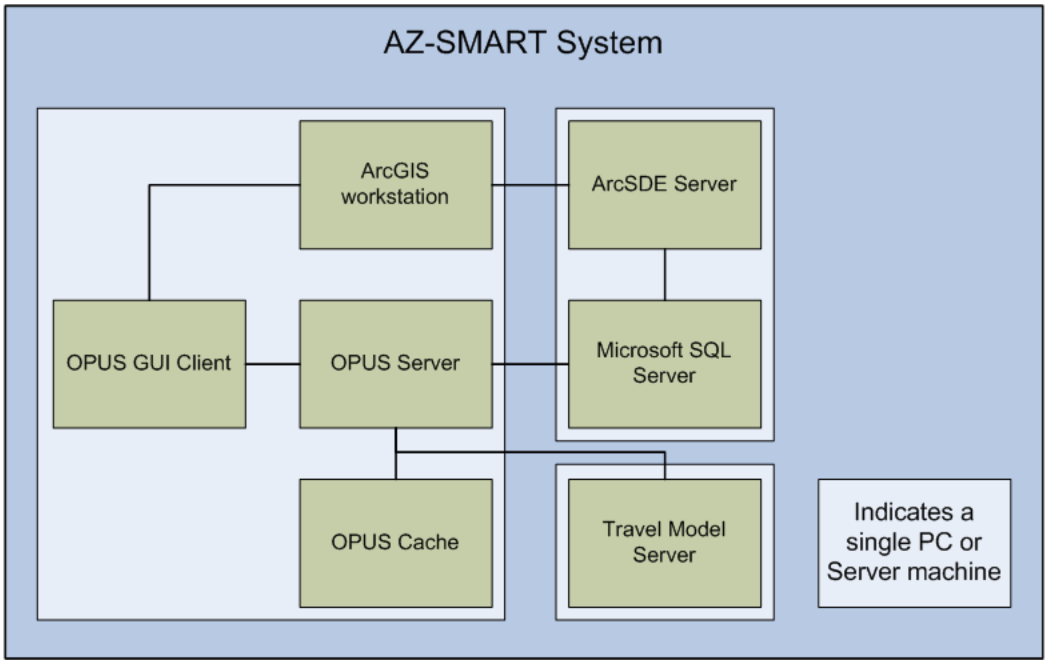

Figure 2: AZ-SMART System Architecture

The Data Manager will focus on metadata maintenance and management, data archiving, and the movement of data between OPUS and the geodatabase. To the greatest extent possible, existing ArcCatalog functionality will be utilized to manage the geodatabase, including managing and maintaining geographic datasets and creating and tracking relationships.

4.2 Tool Manager

The Tool Manager will focus on the management and execution of the wide variety of tools to be developed for AZ-SMART. The ArcToolbox/ModelBuilder framework will be used for the organization, management, indexing, and execution of tools. Specifically, a new AZ-SMART toolbox will be populated with the required tools, which will then be organized into toolsets based on common functionality. To the greatest extent possible new functionality for AZ-SMART will be developed as Python geoprocessing tools within these toolsets, while minimizing custom GUI development within the ArcMap and ArcCatalog applications. If the required functionality cannot be implemented in Python through the existing ESRI geoprocessing object and/or other add-on Python libraries, CUSPA will develop custom geoprocessing tools using VB.NET and ArcObjects [1].

4.3 Project Manager

The Project Manager will focus on managing the model components of the system. The Project Manager will leverage and manage modeling infrastructure embedded in the OPUS system, and will take advantage of the run management capabilities of OPUS such as model specification, estimation, execution, and generation of indicators. The Project Manager user interface would be accessible from within ArcGIS, and will open a windowed application which we refer to as the OPUS GUI. The OPUS GUI, developed in Python and using the Envisage tools from Enthought, could also be used outside of ArcGIS, providing flexibility for a range of use cases. The Project Manager will be the central application where simulation runs are configured,

[1] When SAM-IM was developed in the ArcView environment, there was considerable functionality not present in the underlying GIS platform that needed to be developed in SAM-IM, such as editing of polygons. In the development of AZ-SMART on the ArcGIS 9.x platform, there is no need to develop such functionality, since editing tools for spatial data are built-in. The initial plan for AZ-SMART is to maximize the use of built-in functionality and minimize the amount of custom-code development which will need to be maintained and synchronized with evolving ArcGIS functionality and interfaces

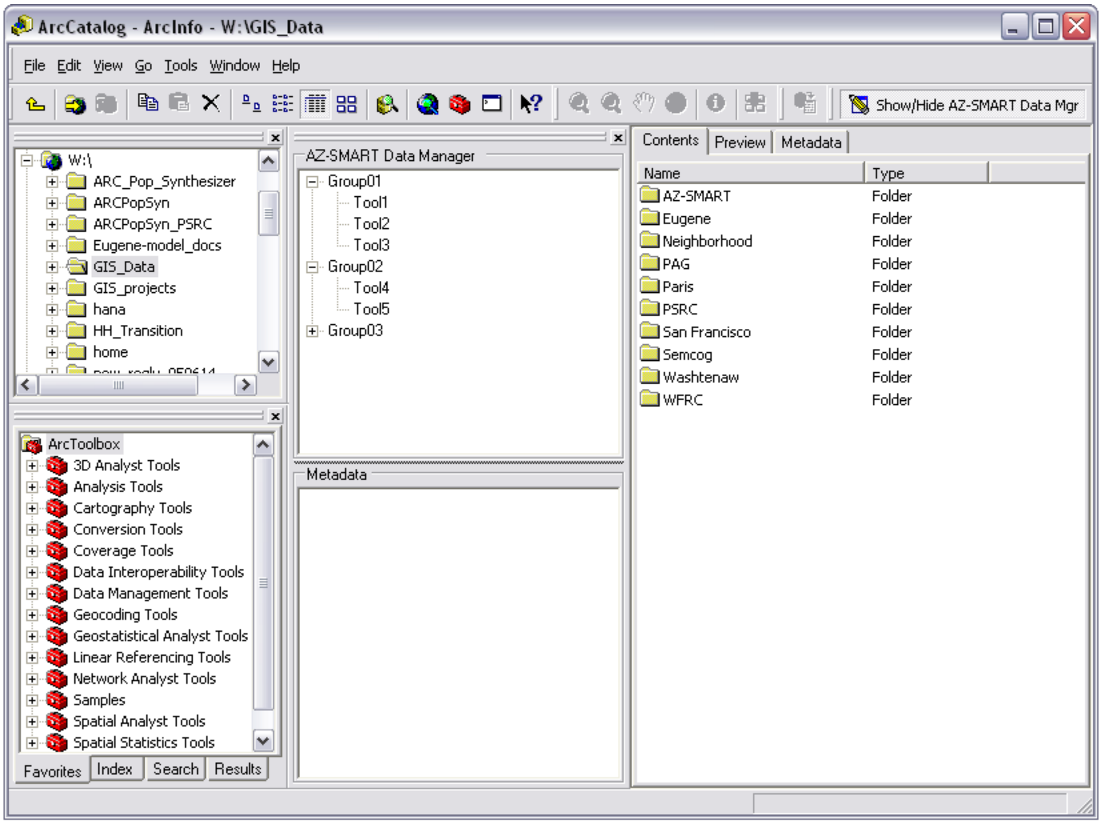

Figure 3: AZ-SMART Data Manager dockable window in ArcCatalog

managed, started and stopped, and will interoperate seamlessly with the Data Manager and Tool Manager components of AZ-SMART.

Several of the items listed in the Project Manager requirements in AZ-SMART RFQ Appendix G are very focused on control of model operations: controlling model execution, accessing the status of a model while executing, and accessing execution logs and error logs associated with a model run. Compared to an approach of using only native ArcGIS GUI tools to code the Project Manager, the approach of coding a native OPUS GUI will make it more feasible to implement these requirements. It would need to be implemented in a way that this component of the user interface inter-operates transparently with other tools in the Tool Manager and Data Manager components, and can be launched from within ArcGIS. Some initial tests of this approach should be developed early in the project to flesh out this aspect of the user interface.

5 System Security

There may be situations in which there is a need to restrict access to data or functions to different users and/or machines. We propose to exploit existing infrastructure wherever possible to address security requirements. Several security points are available using existing system functionality:

- Database Access: Access to the geodatabase may be protected at the user or group level, and can be applied to databases or possibly to the table level, for managing read and write access to data. This would provide a lightweight mechanism for managing user access, for example by restricting which users or groups have access to a database containing the database containing the core data for the AZ-SMART application. More fine-grained control would also be possibe, for example protecting employment data from read access outside a set of users within the agency that have been cleared for this access.
- Login Access: Each machine that has AZ-SMART installed will have login access managament, which can be used as a simple but effective way to prevent unauthorized access to the AZ-SMART system.

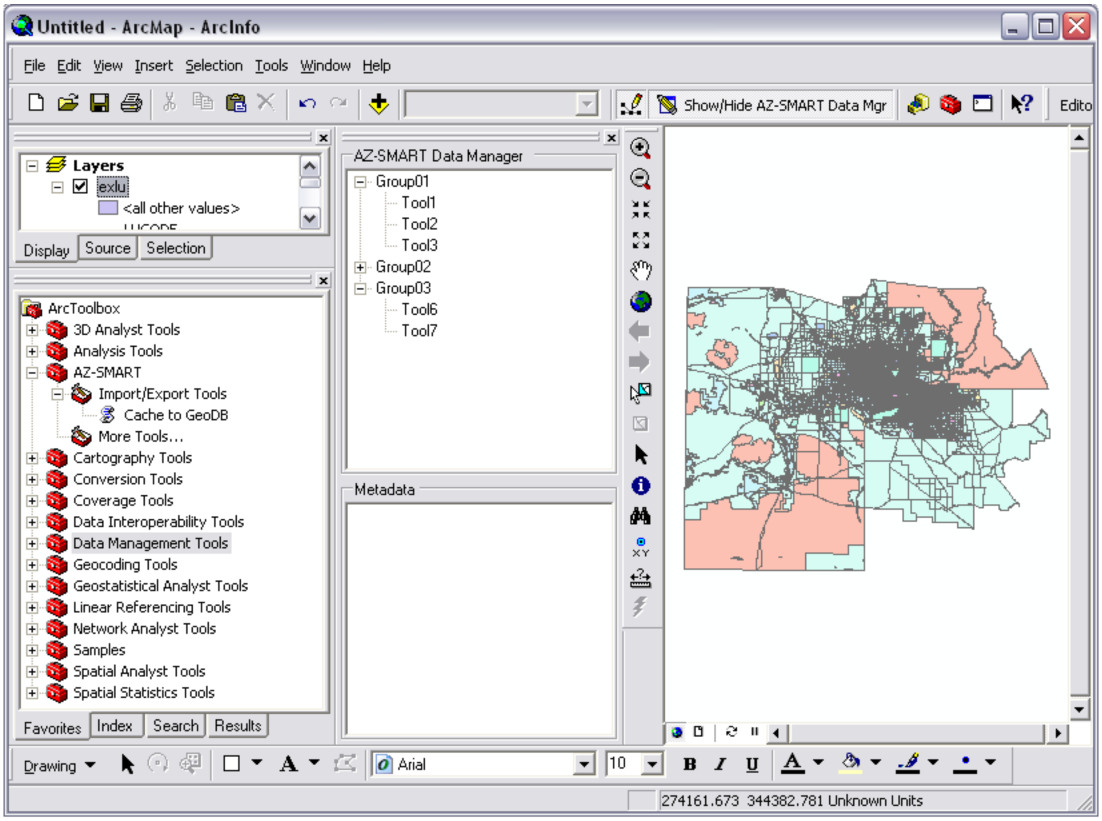

Figure 4: AZ-SMART Data Manager dockable window in ArcMap

- Machine Access: A machine may be enabled for access by setting environment variables that can store username and password information, and this can be loaded by the az-smart system as needed for security enforcement. A generic az-smart user could be defined, or a group account, or an individual user account and a password assigned to each. These usernames and passwords could be stored in an az-smart-security database that only an AZ-SMART Administrator is enabled to access and maintain.

6 Data

6.1 Data Models

The data used in the land use model system can be described as an integrated Data Model. We describe in this section the data model used by SAM, and the data model proposed for AZ-SMART, focusing on the first phase of the project.

6.1.1 SAM Data Model

The model presented in Figure 5 is developed from the data documentation of SAM-IM. The key tables in the current SAM system are the exlu, plan, and developments attribute tables and contents tables (which have a many to one relationship with the attribute tables to allow mixed use representation). There are no direct primary key - foreign key relationships that allow a database join among these three attribute tables. Instead, they are spatially cross-referenced by the conversion to grids. The various geographies used in SAM and their relationships are included in the top part of the E-R diagram, including census blocks, block groups, tracts, taz, raz, mpa, county, and state geographies.

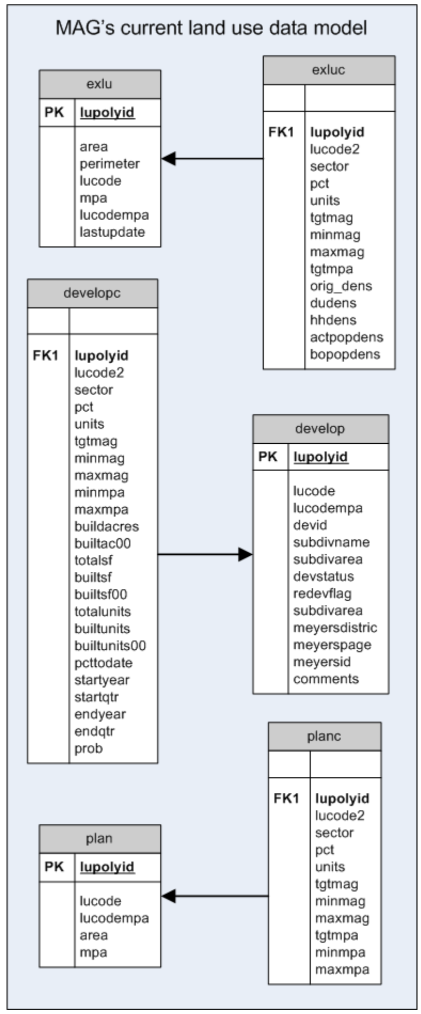

Figure 5: SAM-IM Data Model

6.1.2 AZ-SMART Data Model Using Land Use Polygons

The model presented in Figure 6 presents the proposed data model for the AZ-SMART project, focusing on an initial phase 1 implementation, making maximum use of existing data. It uses existing land use, planned developments, and land use plans just as does the SAM model. There are some major differences from the SAM data model, however. First, we propose to handle buildings explicitly in the data model. In the initial implementation, these buildings would be imputed from the existing data, based on the land use and density of the existing land use content records. Land use polygons would contain attributes for geographic ids for higher level features such as taz, raz, mpa, city, and county, and other attributes such as land area, and attributes to be used in the scoring functions, which we will refer to as the development probability functions.

We propose to use the *Development Site* as a data object that initially would be equivalent to a land use polygon, but would allow the data to also reflect a collection of land use polygons into one development site. This will be particularly valuable when we move to parcels rather than land use polygons. Development sites will have comprehensive plan assignments, so it is possible to interpret for each development site what development restrictions apply to it.

The comprehensive plan is linked to a set of *Development Constraints*, which indicate the types and densities of development that are allowed to be developed on a particular Development Site. The constraints are assumed not to apply to any Scheduled Developments that are provided by the user as known events. If a plan type is mixed use, then the constraint would be configured to reflect this by a mixture of buildings and densities that would be permissible for development.

Development Templates reflect prototypical kinds of building development, with their associated land area. We propose using a broad set of building types, drawing on the SAM classification initially, and using the FAR and Units per Acre density measures as a means of differentiating among different intensities of development within a building type. These templates can be created by examining recent development projects and creating a set of templates that would cover the range of projects being built or potentially under consideration. Development Constraints can be interpreted in terms of which Development Templates would be permitted for development on each Development Site, based on the Plan Type, and other locational characteristics of the site. To represent mixe-use development, we structure the Development Template as having one or more *Building Components*, which characterize a building of a particular type and density.

Development Proposals are Development Templates that can be considered for a specific Development Site. These are the Development Templates that are permitted by the Development Constraints, and that fit within the land area of the Development Site. The land area and other attributes of the site may be drawn from the land use polygon(s) that comprise the Development Site.

Both Scheduled Developments and Development Proposals have a start year, a number of years to be fully developed, and a reference to a *Development Velocity Function*. These velocity functions can be stored in a separate table for lookup.

Jobs and Households each are represented by their own tables, with one record per job and one per household. Jobs have an id and a sector, and are assigned to a building record, which indicates that they are on a particular land use polygon and in a particular type of building. Households have any needed socio-demographic characteristics and also are assigned to a building, which is itself assigned to a land use polygon. Control totals for population and employment are each represented in a table, with year, sector and employment for the employment control totals, and RAZ to accommodate control totals specified at this level of geography (this could be modified as needed). Similarly, population control totals would be specified year, RAZ and type. These control totals could be from previous runs of the mid-level model, or produced by interfacing to the model.

Flexible aggregation of data to TAZ, RAZ, City, County, and gridcell would be accommodated by this data model. For most of these geographies, the data could be directly aggregated using the foreign key relationships in the data model. For gridcell representation, anything linked to the land use polygon could be assigned to gridcells via a land use polygon to gridcell mapping table that contains the fractional allocation of each land use polygon across the gridcells it overlaps with. This mechanism could also be used to query spatially and assign results to the land use polygon, building, job or household.

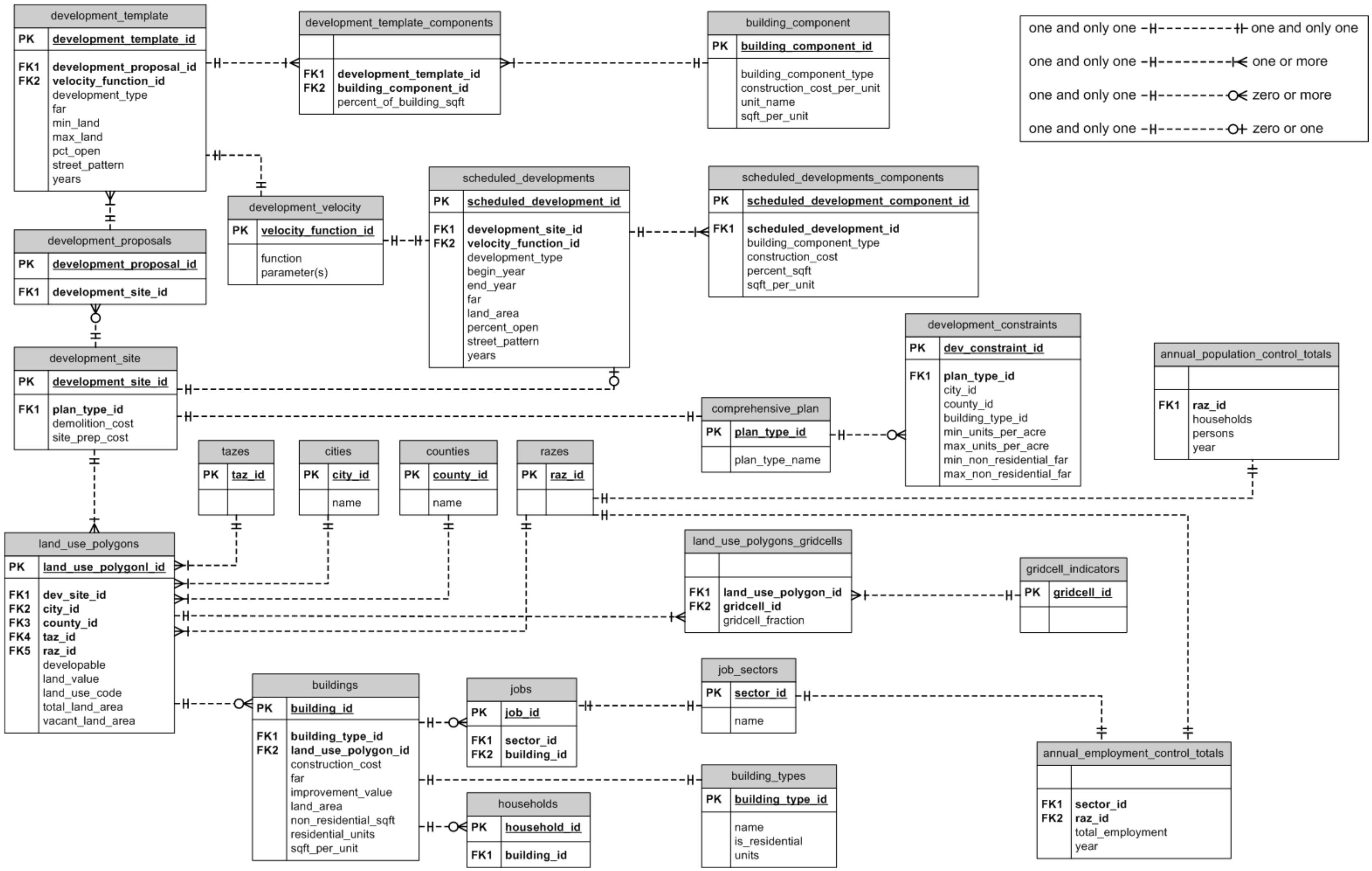

Figure 6: AZ-SMART Data Model Using Land Use Polygons

6.1.3 AZ-SMART Data Model Using Parcels

Once the model system is created and tested using land use polygons, and for those counties that have parcel data and for which AZ-SMART users prefer to do so, parcels could be substituted into the data model in place of land use polygons. This variant of the AZ-SMART data model is shown in Figure 7. Using parcel data provides more capacity to reflect spatial detail and information in the assessor database, but conceptually and in terms of data model representation is almost identical to the preceding data model based on land use polygons.

We recommend exploring the potential to transition to parcel-level in Maricopa and Pima County once the model system is operational at a land use polygon level. Since the data model was designed with this potential in mind, the data structures allow easy substitution of parcels, without significantly affecting models and other code.

The main difference between the land use polygon and parcel versions of the data model is that the source of data would be quite different, and the parcel data could require additional tools for data checking and validation.

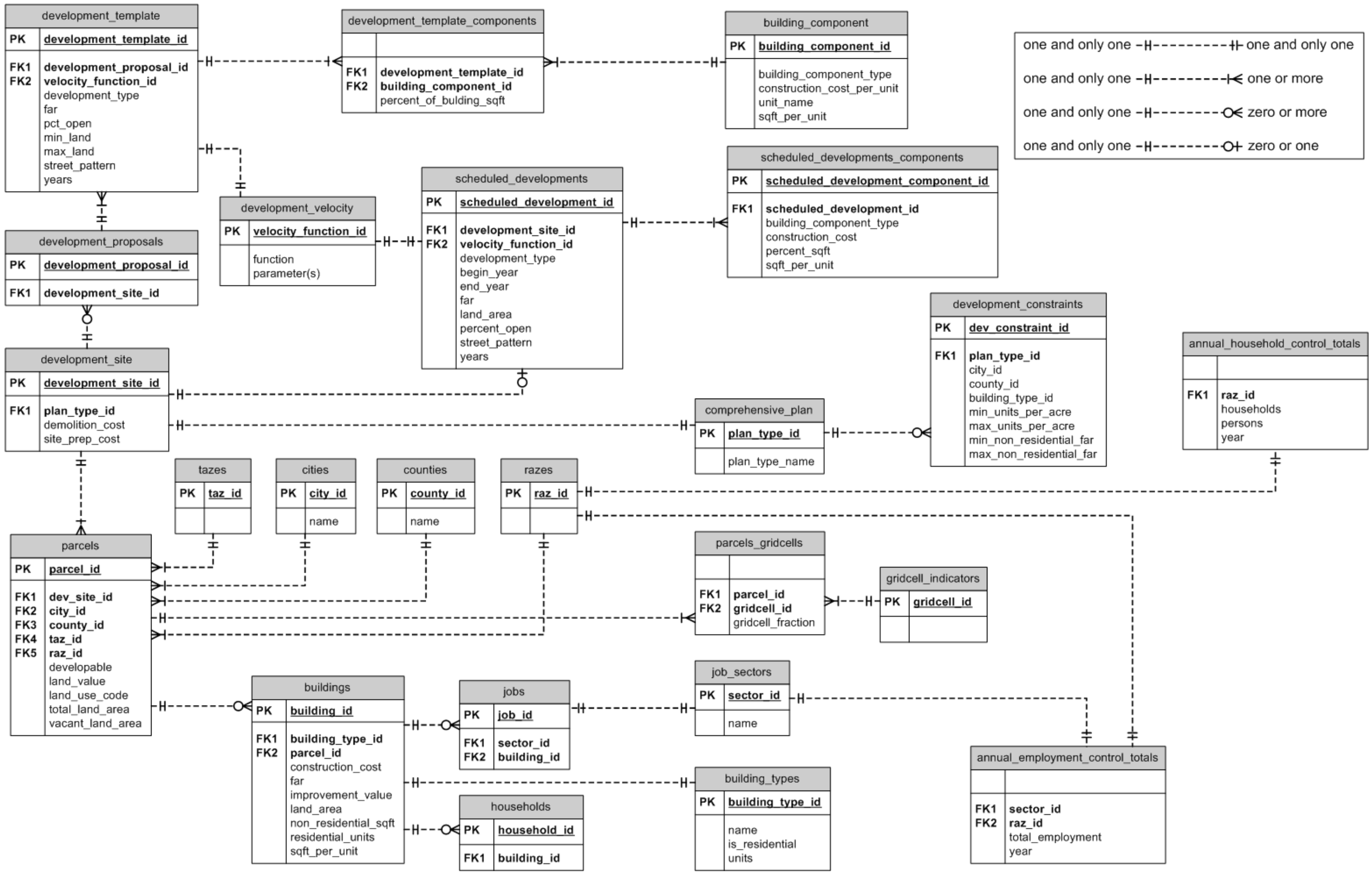

Figure 7: AZ-SMART Data Model Using Parcels

6.2 Storage

Data used in AZ-SMART will exist in a variety of storage formats and the system must be able to deal with data in these formats. At a minimum, these include the following:

- Geodatabase (MS SQL Server, SDE)
- ESRI Shape File (DBF)
- Excel Spreadsheet
- ASCII, Tab and Comma-delimited
- Fixed-format
- OPUS

ArcGIS provides built-in capacity to read and write to these formats (excluding OPUS). The AZ-SMART system will add a capacity to integrate OPUS and ArcGIS by providing read and write tools for each of these formats, leveraging ArcGIS built-in functionality and OPUS interfaces.

6.3 Diagnosing and Correcting Erroneous and Missing Data

Developing the data for AZ-SMART involves the compilation of multiple themes in the GIS and also ancillary files that are external to the GIS. A significant component of the SAM software is the availability of tools to diagnose errors in the input data and to edit these data to correct the problems identified. AZ-SMART will contain a ToolSet within ArcGIS that consolidates data diagnostics into a suite of checks designed to identify problems that could impact model results.

Data imputation tools will also be developed to fill missing data, or create data to match target distributions, or to use rules to correct errors in the data. There is some potential to use statistical datamining tools and algorithms to provide more robust means to refine data, but not likely during Phase 1 of the project.

7 Models

The model architecture is adapted from the functional design of SAM, and also informed by the development of OPUS and UrbanSim. The functionality in SAM focuses on the allocation of land use by sector to grid cells, from aggregate information at a mid-level geography such as RAZs or MPAs. By reference to the UrbanSim model system, this functionality is approximately equivalent in purpose to the real estate development model component of UrbanSim, with some key differences. UrbanSim attempts to include a complete representation of the real estate market, with occupants (consumers: households and jobs), buildings and land (suppliers: developers and property owners), and prices (markets: hedonic regressions representing the interaction of suppliers and consumers).

The architecture for the model system proposed for AZ-SMART is based on a 3-year plan, and incorporates the complete market representation as in UrbanSim, and a multi-level geography and model system. We describe this in the Full Model System subsection below, and then focus on a Phase 1 Model System in the subsequent section.

7.1 Full Model System

The full model system proposed for AZ-SMART involves some hybridization and extension of elements of UrbanSim and SAM-IM. Below we itemize the core elements of the full model system architecture. These are depicted in Figure 8, which shows the model components, their sequence of operation, and the data objects on which the models operate. Note that the data are hierarchically organized in the center of the figure, with aggregation relationships moving from bottom to top.

- *Land-Structure-Occupant Accounting*: The full market representation and explicit representation of and accounting of Land-building-occupant objects and their relationship is proposed for the full model architecture.

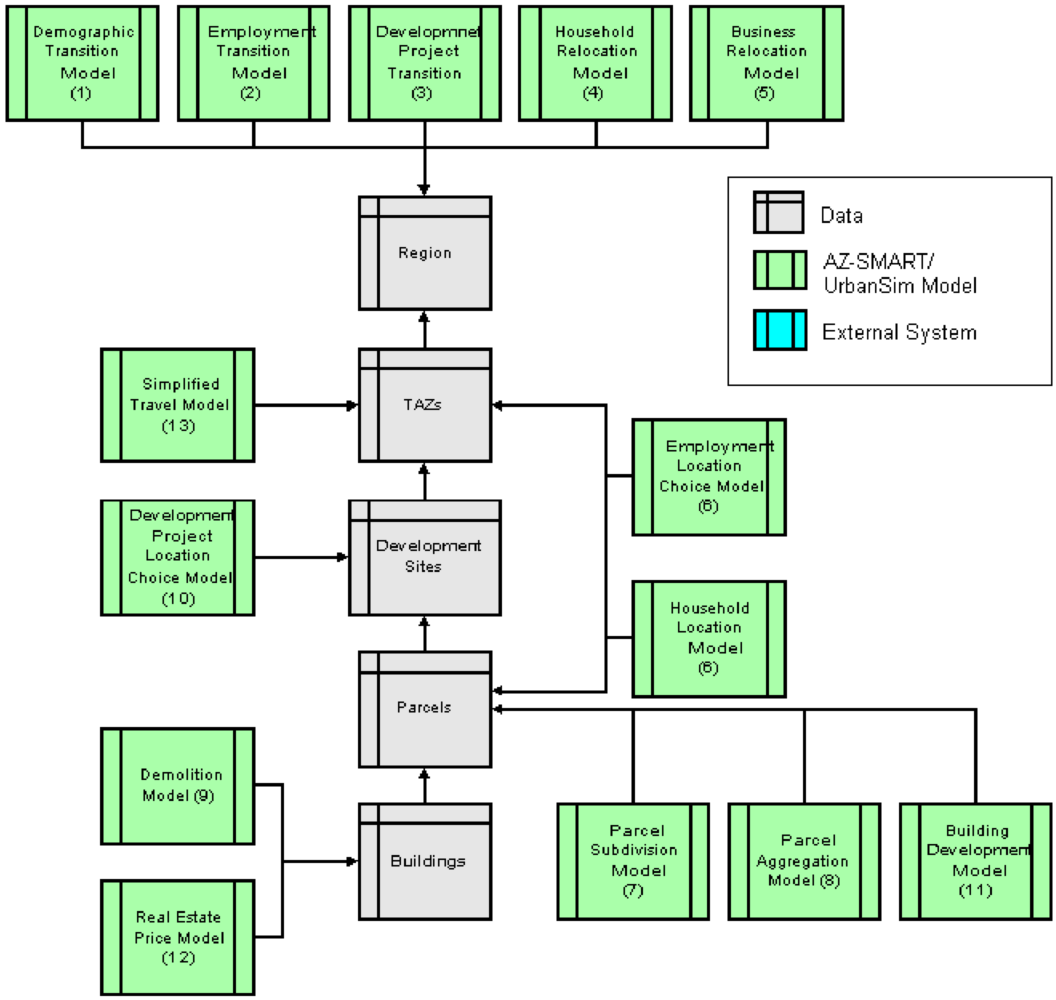

Figure 8: AZ-SMART Full Model Architecture

- *Parcels and Buildings*: Land could be represented by parcels, land use polygons, or cells, but it is expected that the parcel concept would be used as the principal representation of land, and that in areas that do not have parcel data available, land use polygons could be used in a way that treats them as equivalent to (possibly large) parcels. Note that there are many to one relationships from buildings to land and from occupants to building. That is, a building may contain multiple occupants, and a parcel (or land use polygon) may contain multiple buildings. In the event that building data is not available, it could be imputed, to preserve a consistent data model.

- *Development Projects, Sites and Templates*: Development Projects are proposed as a higher-level construct to represent one or more parcels that form a coherent single development project, such as single-family housing subdivision, or a shopping center complex. These development projects will deal both with known *development projects* which the user wishes to incorporate into the simulation, and also development projects predicted by the simulation and assigned to *development sites*. For predicted development projects, a set of pre-defined *development templates* provide a set of configurations of development that include at a minimum the mix of building types (land uses), density, size and timetable for development.

- *Multi-level Model*: The full model system would use a multi-level approach, incorporating parcels as the lowest level (buildings are linked to parcels), and for the forseeable future, one higher level geography to represent an intermediate geography between the county and the parcel. Traffic Analysis Zones (TAZ) are proposed as the basis for this mid-level geography in the full model, simplifying the interface with the travel model system. There are behavioral and practical reasons for using a two-level geography in the model system. Behaviorally, it is based on the expectation that consumers looking for a location (e.g. a household searching for a house) compare neighborhoods, and select properties to examine based on their assessment of the neighborhoods. In practical terms, the two-level geography provides a more convenient way to interface models in a modular way, for example to interface the travel model system, or to run the model system for corridor or area studies where more detail is needed in a subset of the planning region and less is needed outside of this focus area.

- *Microsimulation*: The proposed architecture is based on explicit representation of the agents and objects being modeled at a microscopic level. Parcels, buildings, businesses (or jobs), households, and eventually, persons (for supporting activity-based travel models and workplace choice models and individual-level accessibility calculations).

- *Temporal Dynamics*: The model system would be able to use a specified time interval, such as 5-year or 1-year steps, between which it would simulate changes to the state of each object and agent in the system (construction of new buildings or conversion of existing ones, movement of households and businesses from one location to another, creation of new households or businesses).

- *Models and Interface*: A set of models will be interfaced through a common data store, and managed by a Model Coordinator that controls their execution and implements events (changes to the data) proposed by models. The individual models would be the following:

  - Demographic Transition (Region): Reconciles the control totals of population (by household type) with the database - adding households to the database or removing them if a household type is declining.
  - Employment Transition (Region): Reconciles the control totals of employment (by sector) with the database - adding businesses (jobs) to the database or removing them if a sector is declining.
  - Development Project Transition (Region): Reconciles the total demand for real estate by type, including the results of the Demographic and Employment Transition models, with the existing stock of real estate, by generating proposed Development Projects until vacancy rates reach long-term structural levels.
  - Household Relocation (Region): Predicts whether a household will move from their existing residence during the next time step.
  - Business Relocation (Region): Predicts whether a business (job) will move from their existing location during the next time step.
  - Household Location (TAZ and Parcel): Predicts the building that a new or moving household will choose from among the set of buildings with a vacanct unit.
  - Business Location (TAZ and Parcel): Predicts the building that a new or moving business (job) will choose from among the set of buildings with sufficient vacanct space.
  - Parcel Subdivision (Parcel): Predicts the number and size of parcels to create from a large parcel that is to be subdivided to create a development project. Depending on whether the project is known or predicted this will use information provided in the development project description or drawn from a development template.
  - Parcel Aggregation (Parcel): Combines adjacent parcels to create a *development site* suitable to place a development project.
  - Demolition (Building): Removes existing buildings based on age and other characteristics that would indicate a high probability to convert to another use or be abandoned.
  - Development Project Location (Development Site): Predicts the development site chosen to locate a development project.
  - Building Development Model (Parcel): This would be based on the Development Velocity Model, and predicts the construction of individual buildings on parcels within a development project.
  - Real Estate Price Model (Building): This will predict the price per unit (or sqft) for each type of real estate at each location (parcel).

– Simplified Travel Model (TAZ): An abbreviated travel model with just a.m. peak, and other simplifications to provide a relatively high-speed regional travel model to use in intermediate years preceding the target year for the regional transportation plan.

The models proposed for implementation in Phase 1 of this project are described in greater detail in the following section.

7.2 Phase 1 Model System

Phase 1 of the AZ-SMART project focuses on the allocation of land use sectors, essentially the real estate development process. The plan for Phase 1 is to focus on the real estate development components of the full model system described in the preceding section, and to suppress or use only simple 'stub' models for the remaining models in the full system. The objective is to provide the functionality that is provided now by SAM, but in an implementation that is a very big step towards a fully integrated market simulation model system such as UrbanSim, and incorporating valuable innovations such as the management of known development projects, and the use of an intermediate level geography such as RAZ as the basis for control totals for allocation to parcels or land use polygons.

In the following description of the model system architecture, we a design option of using parcels or land use polygons. For most purposes this would not affect the formulation of the models. We propose initially to implement the model using land use polygons, and once this is operational to explore the parcel option.

The following models would be the focus of model development in Phase 1:

- Real Estate Price Model (Building)
- Development Project Proposal Choice Model (Development Site)
- Land Subdivision (Parcel or Land Use Polygon)
- Land Aggregation (Parcel or Land Use Polygon)
- Travel Model Interface (current emme/2 or new TransCad model)

The following models would be implemented for completeness, but would use only the simplest specification. For example, the household location choice model could use an empty specification, which would randomly allocate the household control total for a RAZ to the housing units that have been developed on parcels in the RAZ.

For the sake of brevity, we will refer to parcels below, but please keep in mind that this would initially be done with land use polygons, and that land use polygons would continue to be used in counties without parcel data. Please refer to figure 7 for details on the data model that these models would be operating on.

- Demographic Transition (RAZ)
- Employment Transition (RAZ)
- Development Project Transition (RAZ)
- Household Location (Parcel)
- Business Location (Parcel)

The components of the full model system would be deferred until after Phase 1:

- Demolition (Building)
- Simplified Travel Model (TAZ)

The proposed model architecture has the following steps, using a 5-year time interval.

7.2.1 Determine Quantity of Development Needed

The first task is to translate mid-level model predictions of population and employment by RAZ (or other mid-level geography) into predicted demands for housing units and land area of non-residential uses by type. The *Demographic Transition Model* will compare the sum of the household population to the control target for a specified year, and if the existing population is less than the control total, it will add the needed number of households to the households table, without a location being specified. If the current population of a particular type is greater than the control total target, the model will remove households of this type randomly until the control total is met. The Economic Transition Model does the same with employment, comparing the sum of employment by industry sector (the sectors from Dram/Empal) to the employment control totals, and adds new jobs if the existing employment is less than the control total for a sector. If the existing employment is greater, it removes jobs of this sector randomly from the jobs table until the control target it met.

7.2.2 Determine Development from Scheduled (Active) Projects

In this step, we compute the quantity of development expected in a RAZ (or other mid-level geography) for each land use sector based on the velocity within Scheduled (Active) Development projects, and assigns the development generated by Active Development Projects to the parcels (or polygons) within those developments. These tasks are handled by an *Model Coordinator* that implements events that have been scheduled and applies them to the database. Scheduled Projects would be stored in the database, and the model coordinator at the beginning of a simulation period would load any events that overlap this time frame. For all scheduled development projects (user specified projects, and also projects predicted by the model in a previous simulation period), the model coordinator will look up the velocity function for the project, and add the predicted number of buildings to the buildings dataset.

7.2.3 Generate Development Proposals

The *Development Project Proposal Choice Model* will begin by evaluating for each development site the applicable development constraints, and generating a set of proposed development projects, from the set of development templates that would fit on the site and are consistent with land use regulations. Note that more than one project that might be allowed on a site, and there may be sites that would not allow any projects to be developed. The result of this step would be a list of proposed development projects, from which the most likely projects should be selected, while imposing the constraint that once a project is selected for a site, the site is excluded from further consideration for other projects, and that once the project committments reach a stage that at completion they would cause the vacancy rate in a given property type (land use) to exceed a long-term stable rate (a break-even rate for developers), a project will not be accepted for development. This is equivalent to the role of the financial sector providing capital to developers for financing construction projects. When vacancy rates exceed a long-term level that is seen as a high-risk threshold, the likelihood of being able to secure financing for a development project should fall quickly.

7.2.4 Development Scoring: Estimated Return on Investment

The evaluation of which development proposals are most likely to be implemented requires computing some kind of a scoring function, to use the term as applied in the SAM model. In SAM, each land use sector has its own scoring function, and the user specifies a pre-determined order to apply the model to the set of land use sectors. Since the scoring functions cannot be compared across sectors, there is no way to decide which land use sector would outbid or out-compete for a particular site. We propose to improve on this approach by using a common scoring mechanism: estimated return on investment. This would allow comparisons to be made among all development proposals, and economic logic to be more directly used in making the predictions of what kind of development will occur at different locations.

As noted earlier, the proposed approach is consistent with a view of the decision-making agent as a financial agent who is evaluating which development proposals to provide financing for. The probability of being successful in securing financing is likely to be proportional to the return on investment on a project. Simply put, the ratio of the expected profit to the amount invested is a reasonable measure of the relative probability of one project being developed as compared to other potential projects. Note that this approach combines and reconciles the two contrasting approaches used in real estate economics and finance: the site looking for a use (the landowner perspective) and the use looking for a site (the developer perspective). If we generate possibly multiple project proposals on each development site, and then compare their expected profit not only across proposals for the same site, but also across sites, then we have an approach that accounts for the competition among developers for sites, and among land owners for development. Note that this is a significant advance over the current specification of both UrbanSim and SAM-IM.

We can estimate the expected profit by a fairly straightforward method: hedonic regression. This approach is already well supported in OPUS and UrbanSim, and allows the specification and estimation of models with price as the dependent variable (or log price), and a set of structural and locational attributes on the right hand side of the model. It essentially 'decomposes' the sales transaction price (or assessed value if sales data are not readily available) into component marginal prices for each attribute, such as bedrooms, square footage, age, accessibility and other locational and structural characteristics.

$$ln(P) = \alpha + \beta X + \gamma Z \tag{1}$$

where $ln(P)$ is the natural log of the price per unit for residential, and per square foot for nonresidential buildings and land; X is a set of location characteristics, and Z is a set of characteristics of the building. Parameters β and γ would be estimated using Ordinary Least Squares, with OPUS estimation utilities. The particular variables to be used in the model specification can be determined during the model development phase, through iterative testing of alternative specifications. A relatively simple model specification with good explanatory power is preferred over model complex specifications.

Using the Real Estate Price Models for each property type, we can estimate the resulting final value of any development project by predicting the valuation of the component parts of a project and summing them over the components. If we subtract the value of the site prior to development, and subtract the cost of the construction (and possibly the cost of demolishing existing buildings on the site), then we have a working estimate of the expected return on investment from the development.

$$ROI^* = \frac{\sum e^{ln(P^*)}Q^* - \sum e^{ln(P^0)}Q^0 - C_c - C_s - C_d - C_f}{\sum e^{ln(P^0)}Q^0 + C_c + C_s + C_d + C_f} \tag{2}$$

where ROI^* is the estimated return on investment on a proposed project, P^* is the estimated market value per unit (sqft) of the building after completion, Q^* is the quantity or size of the development project (number of units, number of square feet), P^0 is the estimated pre-development value per unit of the existing property, and Q^0 is the pre-development quantity of development. C_c is the construction cost of the development, C_s is the cost of site preparation, including provision of infrastructure to the site, C_d is the cost of demolition of any existing buildings on the site, and C_f is the financing cost over the duration of time that the project is under development and on the market.

The costs in the model could initially be estimated using simple constants, but would be available for refinement using more sophisticated estimation algorithms over time.

7.2.5 Predict Development Project Proposal Probabilities

Once the development project proposals have been evaluated and the expected return on investment from each has been estimated, we can use a simple multinomial logit model to generate the probabilities of each being developed. The principle term in the utility function is the expected ROI, and the probability of development will be proportional to the relative profit of the proposed project.

$$Pr(i) = \frac{e^{ROI_i}}{\sum_{j \in J} e^{ROI_j}} \tag{3}$$

7.2.6 Sample Development Projects Until Demand Met

Once we have generated the probability distribution across projects, we may use that to sample from. We draw samples from the probability distribution and add them to the list of committed projects, at each step updating the amount of committed space by type, and evaluating whether we have reached a cutoff threshold for any property types. Once the committed projects reach a point that the next project adds enought space to exceed the structural vacancy rate, then the algorithm will reject any additional proposals that would further increase the vacancy rate in that property type. The drawing process proceeds until the threshold vacancy rate has beeen exceeded in all property types or a predefined number of iterations has been reached.

7.2.7 Iterate over RAZs (or other geography) and Allocate Unplaced Development Projects

Once allocation of all development in a RAZ or other mid-level geography is completed, process the next RAZ. If not all development could be allocated in a RAZ, accumulate unmet demand within a higher level geography to be processed in a final iteration. If needed, allocate unmet demand from higher level (MPA for example), to Development Sites as above.

7.2.8 Data Implications

The implications for data requirements of the proposed Phase 1 model design are intended to be relatively modest, allowing the effort to focus on developing the software infrastructure for the AZ-SMART project. However, there are some new data requirements implied by the model design. One of these is the development templates. These templates can be drawn from information about the known developments, so may not involve any new data collection effort, but rather just additional analyis of data that is already in hand. Another data requirement of the proposed approach is information on real estate values. These could come from the county assessors office, or from a real estate research source that records sales transactions. An alternative to sales transactions is the use of assessed values, which are available for all properties, but may have some limitations depending on how these assessments are made by the assessors' offices.

Parcel data may eventually be desirable to use, but initially, as noted earlier, we propose to rely principally on the existing land use data used in SAM, to reduce the data preparation effort and focus more on functional model and software development.

7.3 Parcel Subdivision

One of the new models that will be needed in Phase 1 is a parcel subdivision model. As an initial approach, we propose to avoid the problem of operating on the topology of the parcel polygon, and instead just create new parcel records that become children of the original parcel, containing all the relevant attributes including area, and respecting the total area of the original parcel, but not being depicted by polygons. This approach would manage risk and avoid spending too much of the development effort on this one aspect. We can also explore options for algorithms to cartographically subdivide parcels.

If this is not sufficiently detailed for MAG or PAG, then options for developing a topological subdivision of polygons will be explored in greater depth, so that alternative approaches can be assessed for feasibility and level of effort.

7.4 Parcel Aggregation

A parcel aggregation model will also be needed in Phase 1. We propose to initially develop this model using user-specified redevelopment area polygons, that can be treated as development sites. We will define the spatial relationship of parcels within each redevelopment area to the development area polygon, and assign the polygon to the set of development sites. At this point, the cost of development on one of these sites would include the cost of acquiring the parcels contained in the site and demolishing any existing development on them. This will reduce their expected profit as compared to greenfield development, but is economically realistic.

In a later phase, we may be able to generate redevelopment districts through an algorithm that finds parcels meeting certain conditions, and then attempts to 'grow' the development site by finding neighboring parcels that when combined, would generate a profitable development site. Our assessment is that this would be beyond the scope of Phase 1, however.

8 Use Cases and Tools

In this section we describe several possible entry points to the AZ-SMART system, corresponding roughly to the different components of the workflow described in Figure 1. This is an attempt to identify the different uses of the system and to identify functionality that will be needed. Each of the following subsections provides a brief narrative and also a list of tools that could be utilized.

8.1 Data Preparation: Diagnostics, Exploration, Refinement

In the initial phase of model development, or during an update phase, an AZ-SMART user will focus on assembling data, assessing data, and refining data for use in the model. Data will likely come in a variety of formats, including disk files and database formats, but the data integration process envisioned for AZ-SMART will produce an integrated geodatabase. Many tools to facilitate this process are existing in ArcGIS, and do not need to be further extended. Some tools that will need to be developed or customized for AZ-SMART include the following (see Appendix A for a description of sample Data Preparation tools, each of which will correspond to a tool entry in a DataPreparation ToolSet under AZ-SMART):

- Data Import and Export
- Data Queries
- Data Summaries
- Data Visualization
- Data Diagnostics
- Data Imputation
- Data Editing
- Data Model Development

8.2 Model Development: Specification, Estimation, Diagnosis

Once a user is satisfied that the database is usable for model development, the focus shifts to model specification and estimation. At this point, the following tools will be the focus of attention, and will be clustered within an OPUS GUI that can be launched from within ArcGIS or independently (e.g. from a batch file, icon, or Start Menu item):

- Model Configuration
- Model Specification
- Model Estimation
- Model Diagnostics
- Model System Configuration
- DataSet Storage Configuration
- DataSet Statistical Profile

8.3 Scenario Creation

Once the model specification and estimation process has been completed to the satisfaction of the user, the user will focus on the process of configuring a scenario to run the model on, meaning that the particular inputs that will differentiate one run from another need to be specified. This will likely involve combinations of different control totals, transportation model networks, land use plans, development projects, or other inputs that would reflect policies or assumptions. This activity will again be focused within the ArcGIS interface, within a Scenario Editing ToolSet consisting of tools such as:

- Control Totals (Load, Edit, Save)
- Travel Model (Configure)
- Land Use Plans (Load, Edit, Save)
- Development Projects

- Scenario Management (New, Load, Edit, Save, Save As, Copy, Delete)

We expect that scenario configurations will need to be archived in a central repository, potentially with version management. If so, this would be done either with a database or a version control repository such as Subversion.

8.4 Run Management

Run management involves the configuration of runs on one or more scenarios. It includes setting the start and end year for the simulation, and tracking the progress of a simulation, stopping and restarting a simulation if needed. These activities will be integrated closely with the OPUS GUI, and can be launched from the ArcGIS interface as well as independently.

- Run Configuration (Load, Edit, Save, Save As, Delete)
- Start Run
- Monitor Run
- Stop Run
- Restart Run

8.5 Indicators: Production, Visualization, Reporting

An OPUS indicator could be defined as any data in the OPUS system that a user may want to visualize or examine, whether it be prior to, during, or after a model run, or even in other stages such as model estimation. A few examples of indicators are population densities (overall or by socioeconomic class), travel times, job densities (overall or by sector), and amount of developable land. One may also want to examine other indicators such as probability of development.

- Configure Indicator Computation (Scenario(s), Year(s), Comparison, Expression)
- Configure Indicator Output (Map, Chart, Table)
- Configure Indicator Set
- Generate Indicator
- Generate Indicator Set

8.6 Run Refinement

Run refinement is a user defined process to refine simulation results after a run has been completed and reviewed. We anticipate that the predominant approach to refinement of results will be to modify one or more components of the model inputs and re-running a scenario. These modifications could include user-specified events to incorporate into the model, in addition to data edits, such as revision of the control totals. These revisions would be documented through the metadata process, so that revised results could be fully documented. At this point, the process would involve moving back to an earlier point in the processing, and re-using the interfaces already available there. If there is a need to develop an alternative approach that involves more direct overriding of results, the specifications for this will need to be developed collaboratively. Our recommendation is to wait until the model system is operational and producing results before doing much more design work on this aspect.

9 Metadata

In this section we develop an approach to managing metadata throughout the AZ-SMART system. We envision three places where one would want to track metadata. The first place is in a pre-model data processing step where the user is bringing raw input data into the system to build a baseyear database. It would be useful to track data on geoprocessing steps, SQL queries that were run, and perhaps OPUS tools that were used in the creation of the baseyear database. It is likely that this would not be done very often. The second place for metadata tracking would be Recording the details of the simulation run (e.g. models, specifications, configurations, etc.). Lastly, we envision recording metadata for any post-model processing that was done. For instance, any refinement to model results or indicators computed would be candidates for recording metadata about.

There are several built-in tools in ArcGIS to create and manage metadata within the FGDC framework. ArcCatalog provides forms for maintaining metadata about GIS datasets. At version 9.2 ArcGIS also automatically tracks geoprocessing steps performed to create data, storing them in several locations, including in the XML metadata document associated with the dataset. We propose to leverege these built in tools to track metadata about GIS datasets. While the FGDC framework may be a bit cumbersome, it does provide a solid standard upon which to build metadata documentation. We anticipate that AZ-SMART users will need to define the amount and exact nature of the metadata they wish to document for each of the input data sources. The system should be able to preserve geoprocessing metadata and pass forward any initial metadata concerning inputs. Similarly, we anticipate a need to generate and manage metadata for models and scenarios, and to be able to couple these with GIS and geoprocessing metadata. A run of the model system should store the metadata that describes the run configuration and all data inputs to it. Furthermore, this metadata should be stored in a way that can be queried later as documentation of a run and its inputs, perhaps in a metadata database. By extension, indicators and other postprocessing should also generate metadata and this should be available to the end user. In short, we anticipate three types of metadata:

- Input Data
- Model and Run Configuration
- Indicator and Postprocessing

10 Appendix A - Tool List

Data Import/Export Tools:

OPUS to Geodatabase
Purpose: Convert attributes from OPUS to Geodatabase
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Geodatabase to OPUS
Purpose: Convert all or part of an attribute table from a Geodatabase to OPUS
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

OPUS to Delimited ASCII table
Purpose: Convert OPUS attributes to ASCII table on disk (e.g. .csv, .tab, etc.)
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Geodatabase to ASCII table
Purpose: Convert all or part of an attribute table from a Geodatabase to an ASCII table
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

OPUS to DBASE table
Purpose: Convert OPUS attributes to a DBASE table on disk
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Geodatabase to DBASE table

Purpose: Convert Geodatabase table to DBASE table
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

OPUS to Excel table
Purpose: Convert OPUS attributes to Excel table
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Geodatabae to Excel table
Purpose: Convert Geodatabase table to Excel table
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Excel table to Geodatabase
Purpose: Convert Excel table to Geodatabase table
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Excel table to OPUS
Purpose: Convert Excel table to OPUS
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Indicator Tools:

Generate Spatial Indicator(s)
Purpose: To read OPUS data and generate a spatial indicator for viewing in ArcMap
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Generate Spatial Indicator Map
Purpose: To read OPUS data and generate a complete map based on a map template
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Generate Indicator Graph/Chart
Purpose: To generate a graph or chart based on an indicator over specified years
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Generate Indicator Report
Purpose: To generate a document that consists of Spatial Indicator Maps and Graphs
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Aggregation/Disaggregation Tools:

Aggregate to Geography
Purpose: Aggregate a theme to a larger polygon geography (overlaps based on percent area)
Implementation: Standard ArcToolbox GUI

Disaggregate to Geography
Purpose: Disaggregate a theme to a smaller polygon geography (based on percent area)
Implementation: Standard ArcToolbox GUI

Model Estimation Tools
Estimate Regression Model
Purpose: Tool to estimate a standard regression model
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Estimate Logit Model
Purpose: Tool to estimate a logit model
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Run Configuration Tools

Data Diagnostic Tools

Descriptive Statistics
Purpose: Reports descriptive statistics for attributes in a data table(s) (OPUS or geodatabase)
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Variable Distribution
Purpose: Produces graphics of the distribution of variables (OPUS or geodatabase)
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Data Diagnostic Report
Purpose: Produces a document that consists of multiple Descriptive Statistics and Variable Distributions
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Data Imputation Tools

Impute average/most-frequent value
Purpose: Fills in missing values with the average (continuous) or most frequent (discrete)
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Impute Random Value
Purpose: Fills in missing values with random values (within a specified or computed range)
Implemementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

Impute via Model
Purpose: Fills in missing table values based on a specified model
Implementation: Traits/UI dialog box run from AZ-SMART Toolbox or OPUS GUI

11 Appendix B: AZ-SMART Architecture Functional Requirements

This Appendix includes content from Appendix G of the AZ-SMART RFQ, along with with CUSPA's design recommendations for AZ-SMART core modules and including some questions that remain. The purpose of this section is to begin to reconcile the design, implementation, and feature ideas presented in Appendix G with functionality within ArcGIS and Opus, and to identify the nature of new development needed.

11.1 Data Manager

Overview

- Enhancements to ESRI ArcCatalog
- At minimum, maintain current functionality

  ArcCatalog will be used as the basis for the AZ-SMART Data Manager. An AZ-SMART directory structure would organize these data, scripts, configurations and other components.
- Access to, development, and maintenance of all data
- Create and track relationships (spatial and rule based) between datasets
- Uses tools from Tool Manager
- All data potentially used by more than one project. Examples include:
  - Land Use Codes
  - Base Year

  - Allocation Sector Names
  - Legends
  - Symbol table associated with global variables

  CUSPA proposes to create a custom AZ-SMART ArcCatalog Tree structure, that would be a dockable window containing and organizing the data and tools used in AZ-SMART.

- Metadata must be maintained for all datasets
- Security
- Consultant to recommend directory structure

  A directory structure will be recommended and implemented in the AZ-SMART ArcCatalog Tree.

- Consultant to recommend and ultimately create data archiving procedures

  Archiving procedures will be recommended once the database and all items needing to be archived are identified.

11.2 Project Manager

Overview

Projects are defined as the database and model configuration used to support running a variety of scenarios that would share data and model specifications and parameters. Scenarios are defined as a set of input data, assumptions, and run configuration parameters used for a specific run of the model system.

CUSPA proposes to implement the Project Manager user interface as an OPUS GUI (developed using Envisage) that could be launched from an AZ-SMART Tool within ArcGIS. Compared to an approach of using only native ArcGIS GUI tools, this approach will make it more feasible to implement several of the requirements listed below, particularly the last three items, which are very focused on control of model operations – including during a simulation. It would need to be implemented in a way that could inter-operate transparently with other tools in the Tool Manager and Data Manager components, as noted in the first requirement below. Some initial tests of this approach should be developed early in the project to flesh out this aspect of the user interface.

- Create new projects and scenarios, or open projects and scenarios that have been created previously for further analysis
- Links tools from Tool Manager with data from Data Manager using ESRI ModelBuilder concepts
- Selects required components and limits execution of model to tools necessary for the scenario subset.
- Accesses all data relevant to potentially more than one project via the Data Manager
- Stores all data relevant to only that one project within a project. Examples include:
  - Projection Years
  - Switches utilized in the project
  - Status of the project
  - File and database names etc.
- Controls model execution: start, stop, and restart model execution
- Access the status of a model while executing
- Access various execution logs and error logs associated with a model run

11.3 Tool Manager

Overview

- Enhancements to ESRI ToolBox and ESRI ModelBuilder

  ESRI ArcToolbox and ModelBuilder will be used as the basis for the majority of the GUI for AZ-SMART. It will be enhanced by adding tools that leverage OPUS and UrbanSim functionality, generally through the OPUS GUI).

- Will have an indexing system by which end-users can find appropriate tools relevant to their needs

  The built-in ArcToolbox indexing system will be used for this purpose.

- Accesses Script Editor to code/implement tools

  The built-in editing functionality for Python will be used for this purpose.

- Uses ModelBuilder for computer-aided programming (i.e., data flow chart graphical interface)

  The built-in ModelBuilder functionality will be used for this purpose.

- Manage scripts: open, create, copy, delete, edit tools and their properties

  The built-in ArcToolbox functionality for managing scripts will be utilized here. This is accessible via context-menu selections at the toolset and tool level in ArcToolbox.

- Multiple security levels

- Multi-user system

  The system will be designed to use the ArcSDE multi-user environment with Microsoft SQL Server, which provides multi-user capabilities.

- Ability to share tool based on properties assigned

  Question for MAG: This functionality needs further clarification about what is needed. It may be possible to store tools in the Geodatabase, providing security and sharing capabilities within.

- Will perform conditionals and loops

  The built-in ArcToolbox and ModelBuilder functionality will be used to perform conditionals and loops, augmented by Python and OPUS capabilities.

- Unless specified, all tools should be capable of accepting input from user, script, output from other tools, or a database

  Tools will be constructed to meet this qualification.

- Consultant to recommend and ultimately create:

  - Tool archiving procedures

    This will require the use of a version control system such as Subversion. A Subversion repository has been set up for this project at CUSPA. An existing tool for accessing Subversion from Python could be linked from the user interface (www.pysvn.tigris.org).

  - Definition and management of production and development versions

    This could be accomplised by having two branches in the Subversion repostitory, one for production code, one for development code.

  - Prevention of the inadvertent modification and/or deletion of tools referenced elsewhere in the system

    This could be accomplished by placing relevant Toolboxes and Toolsets in a 'read-only' location.

Example Tools from MAG

11.3.1 Land Use Editor

Overview

- An ArcGIS application toolbar that provides controls specifically suited for editing and managing land use databases

  CUSPA proposes to use ArcGIS built-in editing functionality for editing of spatial features (e.g. existing land use, development projects), augmented by a toolset. If there is remaining functionality in SAM-IM that is not already addressed by built-in ArcGIS editing, then an ArcMap toolbar would be created to meet these needs.

- Maintains planar polygon topology/grid; Data model aware; Performs validation and domain checking

  CUSPA proposes to utilize built-in geodatabase functionality for maintaining topology, domain checking, etc. There are existing tools (e.g. Geodatabase designer 2) that can address these requirements.

- Advanced wizards for manipulating, validating, and assembling land use themes using interactive input and configurable rules

  CUSPA proposes to focus on developing tools and ModelBuilder models to meet these needs.

- Aggregation of parcels based on predefined rules (e.g. eliminate minor roads)

- Subdivision of parcels or polygons based on predefined rules

- Variable sized grids with associated attribute data (e.g. areas with high vs. low resolution for modeling and analysis)

- Can access data in Data Manager or Project Manager

  CUSPA proposes that all data in the geodatabase will be directly accessible in Data Manager and/or Project Manager as appropriate. Data generated during a simulation would be accessible in Data Manager and/or Project Manager through the use of tools to copy the data into the geodatabase or other tabular formats.

- Consistency checking across multiple themes

  Tools would be developed to check for consistency within and across themes as needed.

- Summary and indicator statistics

  This would be built using the Opus indicator framework.

- Completely configurable to suit any land use data model, coding scheme, and installation

  AZ-SMART will be designed to be highly modular and configurable and will use the flexibility inherent in the Opus sytesm.

- Similar to current editing capabilities in SAM-IM

  Existing ArcMap functionality will be utilized to provide this functionality.

11.3.2 Land Use and Socioeconomic Synthesizer

CUSPA proposes to develop tools for creating, manipulating, and synthesizing a base year database in the ArcToolbox/ModelBuilder environment.

- For creating and populating the base year (e.g., 2000) land use database by assembling multiple sources.

- For creating projected land use and socioeconomic datasets based on configurable rules.

  This would be done by implementing models in Opus and allowing them to be configured and run within ArcGIS.

11.3.3 Calibration and Validation

- Utilities for creating calibration data sets based on user supplied specifications
- Use 3rd party programs to perform regression analysis (e.g., ALOGIT, SPSS)
- Utilities for validating calibrated model data against observed data

  Opus includes tools for creating estimation datasets, estimating parameters of multiple regression and discrete choice models. These tools would be used for specifying and estimating models in AZ-SMART. Model validation will be supported by tools to compare predicted and observed data and visualize patterns of error in the results.

11.3.4 Analysis, Visualization, and Reporting

- A spatial calculator to perform computations on socioeconomic databases, examples:
  - Incorporate data from external sources;
  - Prorate projections to polygons based on a demographic property;
  - Drop point data into polygons (TAZ or land use polygons);
  - Perform row and column normalization and matrix balancing
  - Automatically summarize land use themes according to other polygon geographies (e.g., TAZs)
  - Calculate socioeconomic and land use statistics (population, employment, acres by type) for user defined areas based on geospatial rules.
  - Compute indicators and measures on land use or other polygon geographies (e.g., job-housing balance).

    Question to MAG: The needs detailed in this section warrant further discussion. CUSPA anticipates that some of this functionality will be handled by ArcGIS and Opus functionality.
- Capability to export tables to any file format, including custom format text files needed by travel models as well as ArcIMS; Users can define and save various file formats into a library of templates, and recall them for later exports.

  The system will provide a capacity for the user to define the particular data to be exported, variables, their sequence, the file format, and location. Export formats would include dbase, SQL server, ASCII (tab delimited, comma delimited, and fixed format). Question for MAG: Does this address the needs for ArcIMS?
- Provides methods by which end-users can define series of thematic maps to be generated automatically
- Provides methods by which end-users can define statistical tables and reports to be generated automatically

  The process of generating indicators and displaying them on a user configured base map will be automated.

11.3.5 Data Manipulation and Conversion Utilities

- Data available in a number of different file/DBMS formats: MS Excel spreadsheets, MS Access, Formatted ASCII files, Geodatabases, MySQL, etc.
- A library of utilities for accessing/converting data from one form to another so that it can be accessed directly by tools implementing models

  The system will support accessing and converting data among various data formats including but not limited to those formats listed above.

11.3.6 Accessibility

- Consultant to recommend and implement methodology or methodologies for travel times from geography to geography. Examples include:
  - Accesses travel times directly from third party systems used by MPOs (e.g., EMME/2, Cube)
  - Accesses travel times directly from modified third party systems using larger levels of geography
  - Creates travel times within AZ-SMART without using 3rd party systems

  CUSPA proposes to develop an interface to the forthcoming MAG TransCAD travel model if it is available within the timeframe of the AZ-SMART phase 1 project. It may be desirable to develop a sketch-level (fast running) travel model in later phases.

11.3.7 Submodels

CUSPA will need to work with MAG staff to develop clear specifications for sub models to be developed.

11.3.8 Site Suitability Tools

- Characterizes potential development sites throughout a region with respect to its suitability for development;
- A toolbox for portraying site characteristics from other GIS users (e.g., age and condition of structure, land value, proximity to highways, distance to developed land, residential market within 3 miles, etc.)
- Creates input datasets used in calibration
- An important component of allocation of lands during a projection, using calibrated factors

  CUSPA proposes to develop a set of tools organized within a Site Suitability Toolset to use themes for planned land use and various environmental or other features that would be used in determining suitability for each land use sector. These would be used to determine the capacity for development of each corresponding building type, such as Single Family units, Multi-family units, Office Sqft, etc, and could account for planning constraints such as minimum or maximum floor-area ratio (FAR) regulations.

11.3.9 Allocation Tool

This section describes functional requirements for a model system used to allocate land uses. CUSPA proposes to use existing functionality in OPUS to implement these models. The models and their specification would be configurable using dialog boxes or forms.

- A key tool for projecting growth in a region
- At minimum, maintain current functionality of SAM-IM

  The intent is to provide at least comparable functionality, though not using exactly the same approach.
- Process works by selecting lands, among candidates, to be built in order to absorb growth based on an evaluation of their inherent site suitability characteristics
- Features include but are not limited to:
  - Observes constraint layers that prohibit development due to environmental or policy factors
  - Observes general plan layers that designate acceptable conforming land uses and densities
  - Accepts any land use coding scheme that the user defines
  - Allocation sectors (variables of interest for projections) are user-defined
  - Sectors are allocated in a user-defined sequence.
  - Mechanism by which large development tracts are subdivided into parcels appropriate in size for the development considered
  - Ability to observe adopted land use plans and densities on a polygon/grid basis

  - Development Velocity Curve dictates the pace at which developments are built
  - Observes regional control totals of growth, or growth forecasts for subareas, as defined
  - Address "mixed use" polygons
  - Address redevelopment and demolition

- Same process can be used, with different inputs, for vacating lands due to demolition and redevelopment
- Controlled by a number of different switches and rules supplied by the user that control how the allocation process specifically works
- Driven by a set of projected control totals of population and employment change that apply to the entire region or subareas of it
- Can control subarea growth at different geographic levels
- Capability for "gravity effects" model projection mechanisms reacting to measures of accessibility, land use constraints and opportunities, growth trends, and other socioeconomic attributes
- Provides specific treatment of known developments scheduled to be underway
- Provides support for analysis of scenarios:
  - Generates alternative scenarios of land use and socioeconomic projections
  - Ability to work on complete area or revision-areas (sub-parts of complete modeling area)
  - Interactive designation of "revision areas"; Capability to manipulate both polygon and grid
  - Migrates changes in downstream years; that is, changes made to a 2010 forecast migrated automatically to subsequent years;
- Provides different ways to react:
  - When build-out conditions are reached in individual subareas
  - How active developments are treated
  - With respect to policy initiatives
  - To demolition and redevelopment
- Different applications of the Allocation procedure in the projection model stream:
  - Regular production projections
  - "Min-Max" procedure to create set of floors and ceilings to estimate reasonable growth potential
  - "Scenario Builder" enabling analysis of changes to land use and other policy variables.