Chapter 4: Implementation

Chapter 4 is divided into three stages consistent with the final three steps (Steps 8 through 10) in the geodatabase design method presented by Actur & Zeiler (2004), and illustrated in Table 2.5. These steps represent the physical design phase of the hybrid geodatabase design method described in Chapter 2. Step 8 consists of the implementation, prototyping, reviewing and re-designing of the model. This is followed by the creation of the standard operating procedure (Step 9) and documenting the geodatabase (Step 10).

4.1 Implement, prototype, review and refine the design

Step 8 in the geodatabase design method describes the sequential procedures that were implemented in order to create the geodatabase. The section starts by describing the data capturing process, after which it describes in detail how the geodatabase, the feature dataset and each of the feature classes were created. The section concludes by describing how the fishnet feature class, the relationship classes and topology were constructed, as well as how the network dataset was developed and implemented.

4.1.1 Data capturing

The preliminary data needed, was reference data in order to accurately align all the future data. This data would serve as the background to which all other data would be referenced. It was opted to implement a QuickBird (2008) satellite image of the campus. The QuickBird image had a pre-defined spatial reference. It employs a WGS 1984 datum, while utilizing the UTM 35S projected coordinate system. The satellite image has a resolution of 60 cm x 60 cm. The satellite image is depicted in Figure 4.1; the yellow rectangle illustrates the study area.

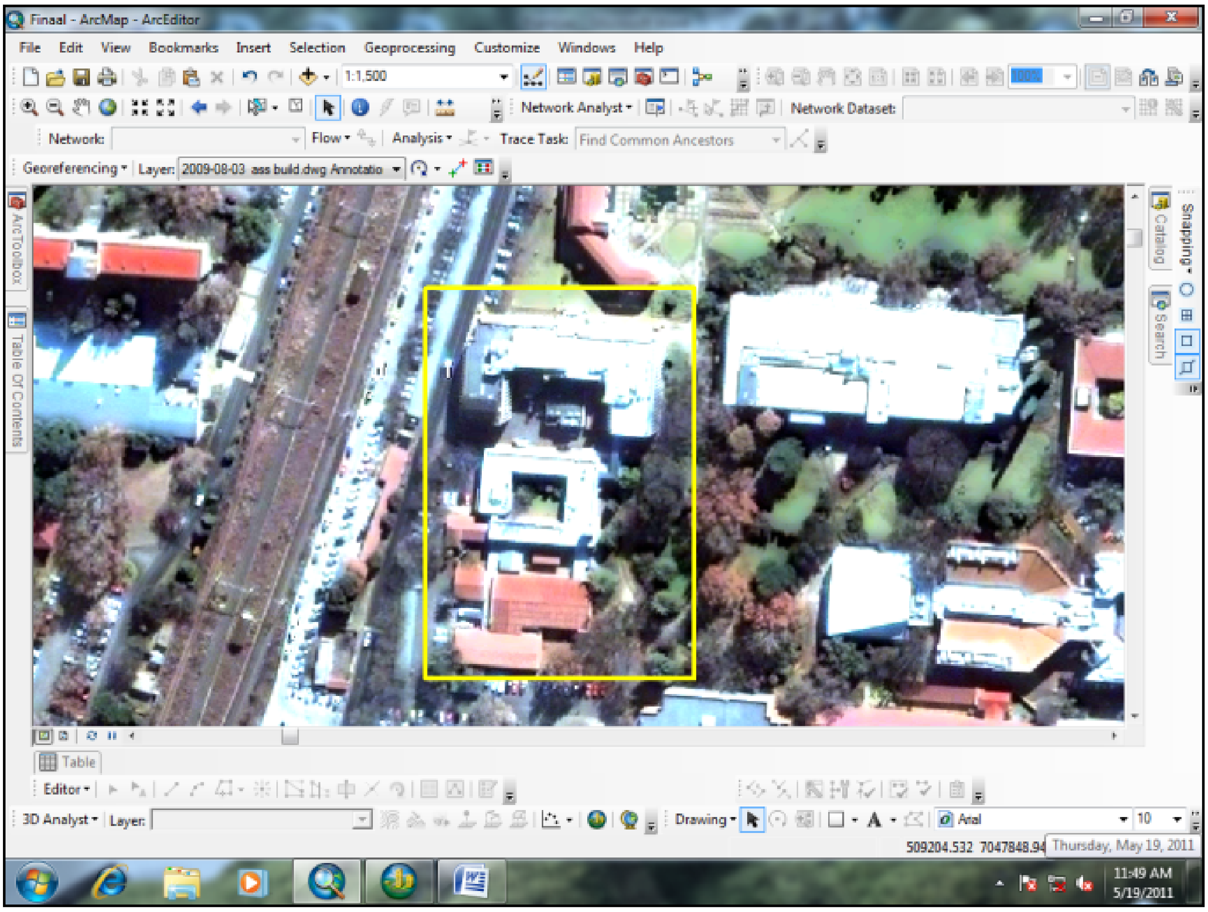

Figure 4.1: QuickBird satellite image of the study area (QuickBird, 2008)

After obtaining reference data, the next data capturing step was to obtain visual data about the campus layout, as well as data depicting the buildings and their layouts. This data was provided by the University's Department of Technical Services (2011) in the form of .dwg CAD files. The ArcGIS 10 software is able to view the CAD files. Unfortunately the CAD files were not spatially referenced. This meant that the data would have to be re-aligned correctly and changed to the accurate scale before it would become useful. The first CAD file depicted the whole campus layout, as well as its infrastructure. The campus CAD data's display properties had to be reduced (Figure 4.2) so that it only represented building outlines and streets (Figure 4.3). This made the CAD data easier to interpret. The study area is illustrated in Figure 4.3 by the red rectangle. This CAD file also depicted the network infrastructure of the campus which operates from the source to each individual building.

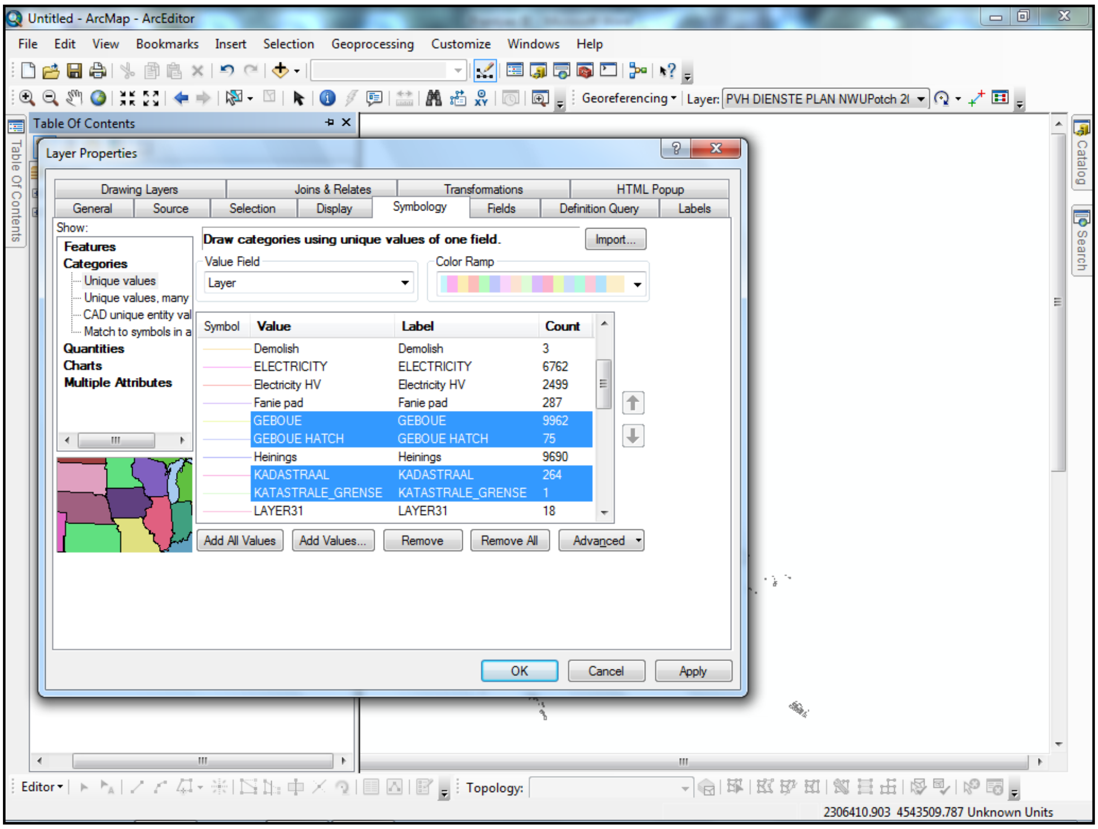

Figure 4.2: Reducing the CAD display properties (Department of Technical Services, 2011)

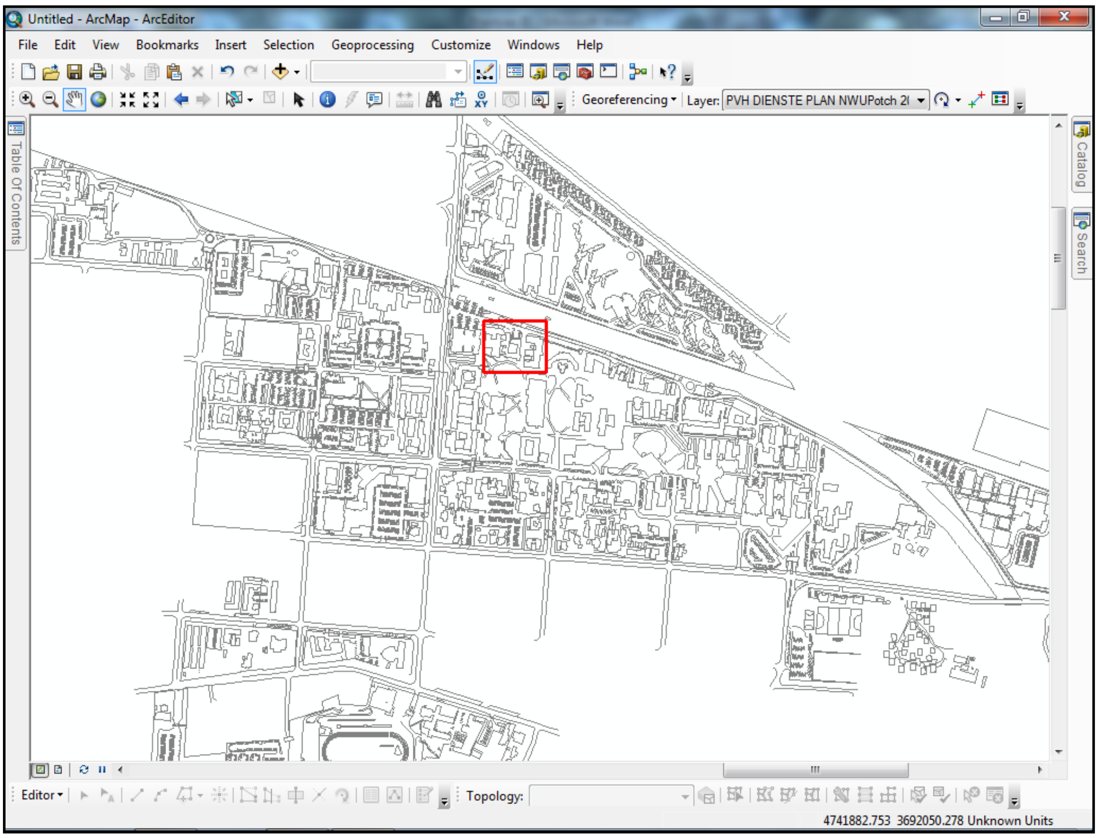

Figure 4.3: CAD depicting the campus layout (Department of Technical Services, 2011)

Unreferenced CAD files were also received for the layouts of all the floors in buildings E4 and E6. The CAD files depicted all the floors alongside one another. Figure 4.4 shows the floor layout of building E4. The ground floor can be seen on the left hand side, while the right hand side shows the first floor rotated 90 degrees to the left. Figure 4.5 shows the five floors of building E6 (basement to the third floor) alongside each other. The files displayed the room layout and the building outline of each floor.

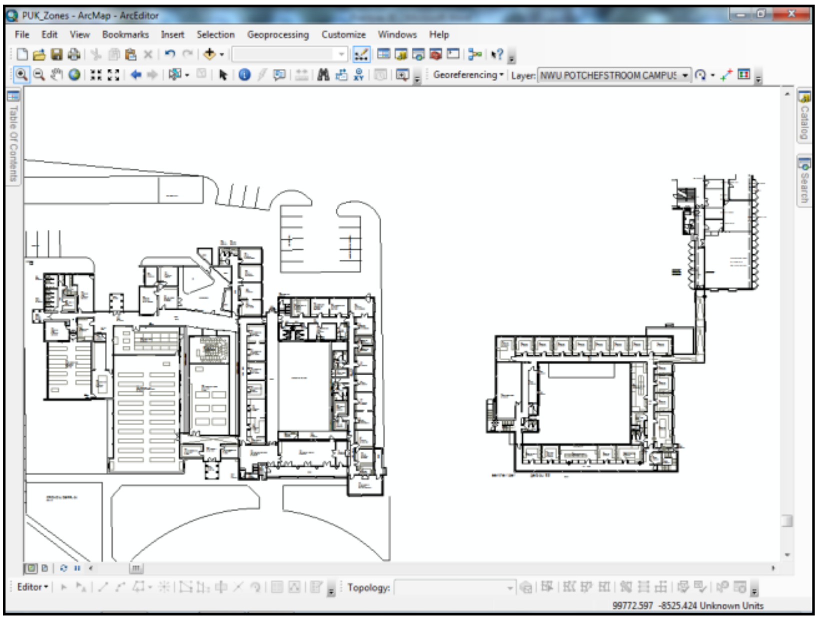

Figure 4.4: Layout of building E4 (Department of Technical Services, 2011)

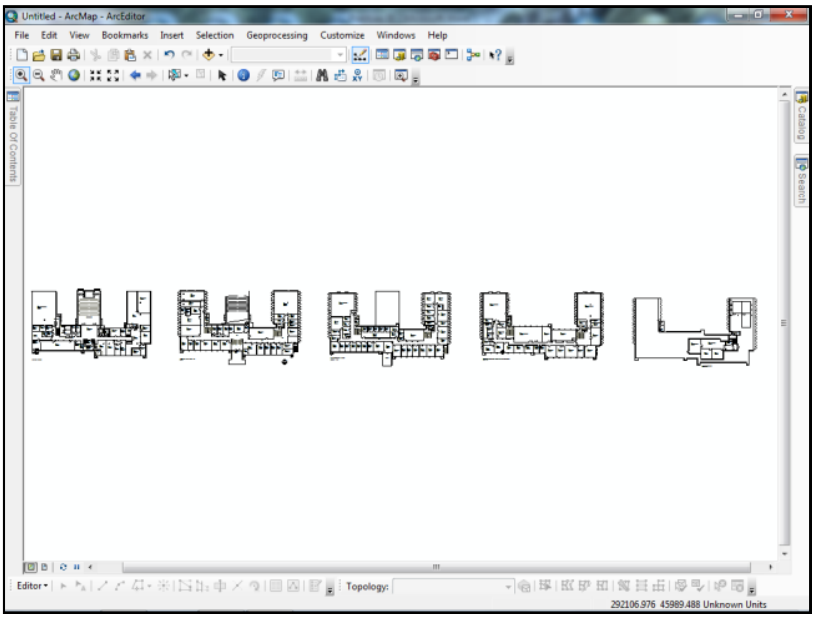

Figure 4.5: Layout of building E6 (Department of Technical Services, 2011)

The CAD files also showed the access routes (steps) between adjacent floors as well as the name of the occupant of every office (Figure 4.6: Basement floor of building E6).

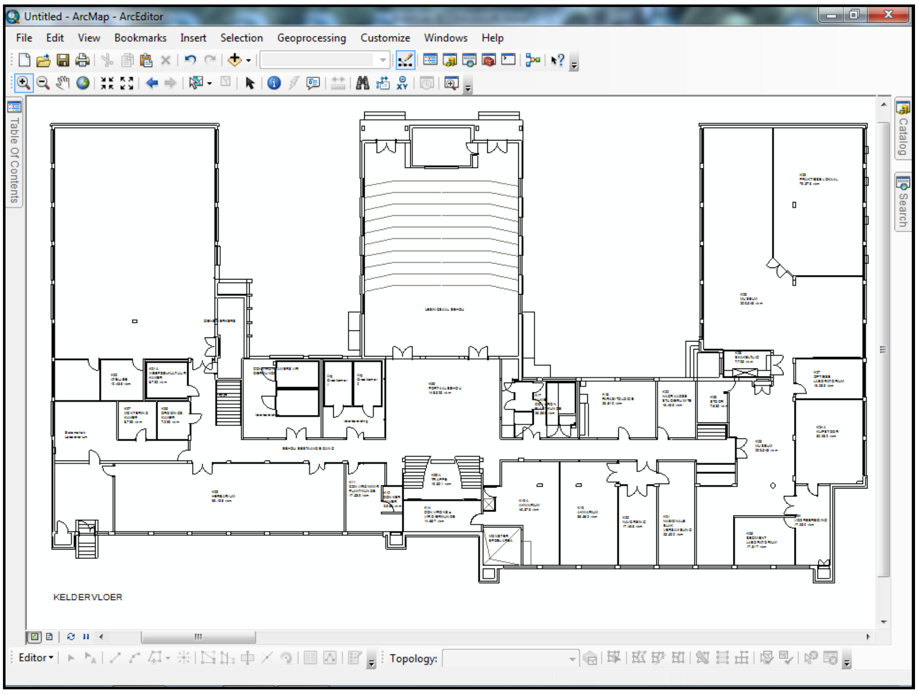

Figure 4.6: Basement floor of building E6 (Department of Technical Services, 2011)

The campus Department of Technical Services (2011) also provided CAD files which depicted the electrical infrastructure for the ground floor of building E4. This CAD data also displayed the building layout of the ground floor (Figure 4.7).

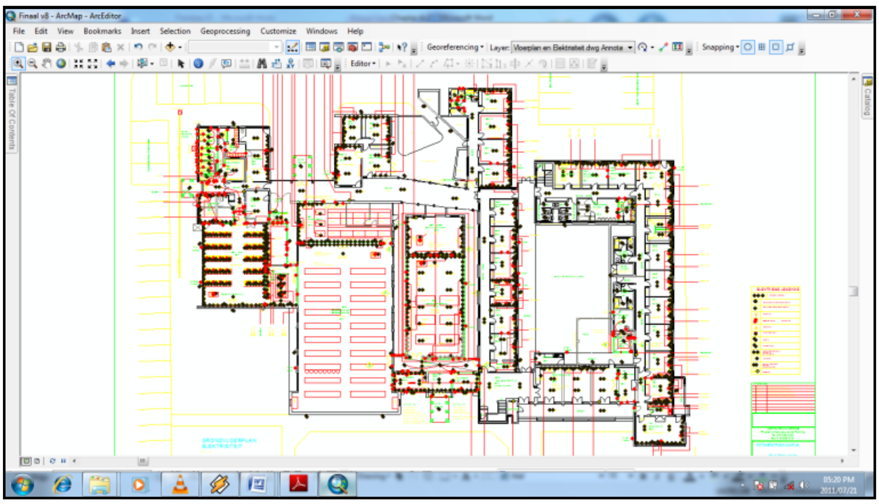

Figure 4.7: Electrical CAD data for ground floor of building E4 (Department of Technical Services, 2011)

Mr. L. Venter (2011), the building manager for both of buildings E4 and E6 provided CAD files in the form of PDF files, which illustrated the location of the computer network infrastructure inside the buildings (Figure 4.8). The PDF files were printed out as hardcopy A3 maps, one map for each floor. The maps were utilized to define the route of each cable in respect to the building layout, as well as the location of each point feature in the building. These maps served as the source of information when the interior parts of the utility feature classes were digitized.

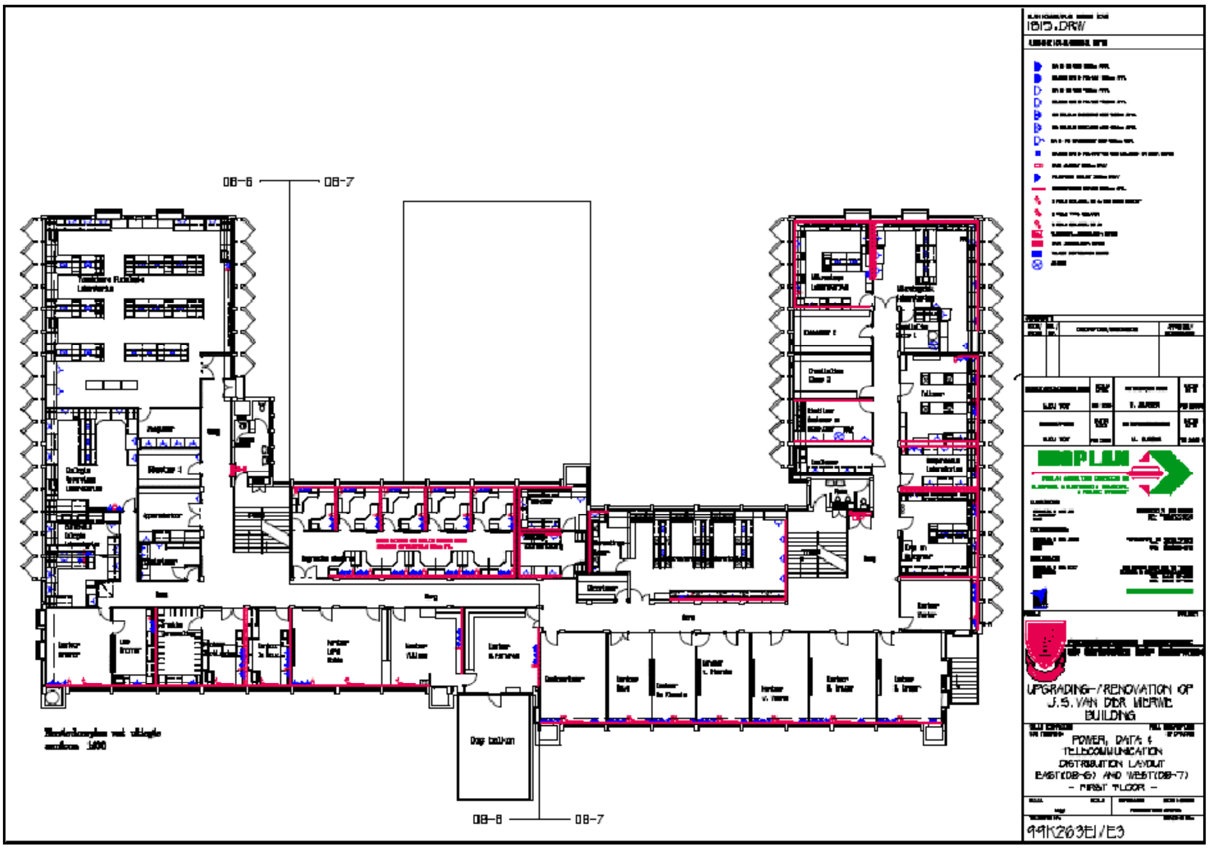

Figure 4.8: PDF of the power, data and telecommunications layout of the 1st floor of building E6 (Venter, 2011)

Descriptive data for the utilities were the final data type to be captured. This information was provided by Mr. P. Buys (2010), of the university's Information and Communication Technology department, during an interview. The information provided by Mr. Buys described properties of each utility, as well as values of these attributes for each utility subtype. This information lead to deciding which fields, subtypes and domains would be implemented for each individual utility feature class.

4.1.2 Creating a file geodatabase and importing referenced data

The first step to create a geodatabase is to designate a folder which will contain the geodatabase as well as serve as a central storage point for all the created data. This folder was named Campus IT Data. After a folder has been appointed, ArcCatalog is used to create a new geodatabase. The geodatabase created for this data model was named PUK Geodatabase. A feature dataset was also created inside the geodatabase, named: PUK_IT. This feature dataset would contain the entire collection of feature classes associated with the computer network utilities data model. The abbreviation PUK represents the Potchefstroom university campus. This abbreviation was chosen to be part of the geodatabase and feature dataset name in order to provide future opportunities to create and store data models for other utility types on the campus, such as electrical- and water distribution systems, in the same geodatabase.

Another folder was created inside of the original Campus IT Data folder to contain a collection of diverse data to be used to enhance the data model. This folder was named Miscellaneous and contains data such as the shortcuts to the Arc Map (2D) and Arc Scene (3D) displays of the data model.

Before the feature classes could be created, it was important to store the QuickBird (2008) image inside the geodatabase. The satellite image would be the only raster data type utilized in the data model, and could therefore be stored in the geodatabase as a standalone raster dataset. The QuickBird (2008) image is imported into the geodatabase by right clicking on the geodatabase icon in Arc Catalog; selecting Import; then selecting *Raster Dataset* (Figure 4.9), after which the designer navigates to the stored location of the raster.

It was also important to create the domains to be used throughout the geodatabase right away. The six domains which were described in section 3.5.3 of this dissertation were created by navigating to the *Domains* tab in the PUK_Geodatabase properties (Figure 4.10). The domains were all coded type domains as well as long integer data types. All the domains created in the geodatabase are available to any table or feature class stored in the geodatabase.

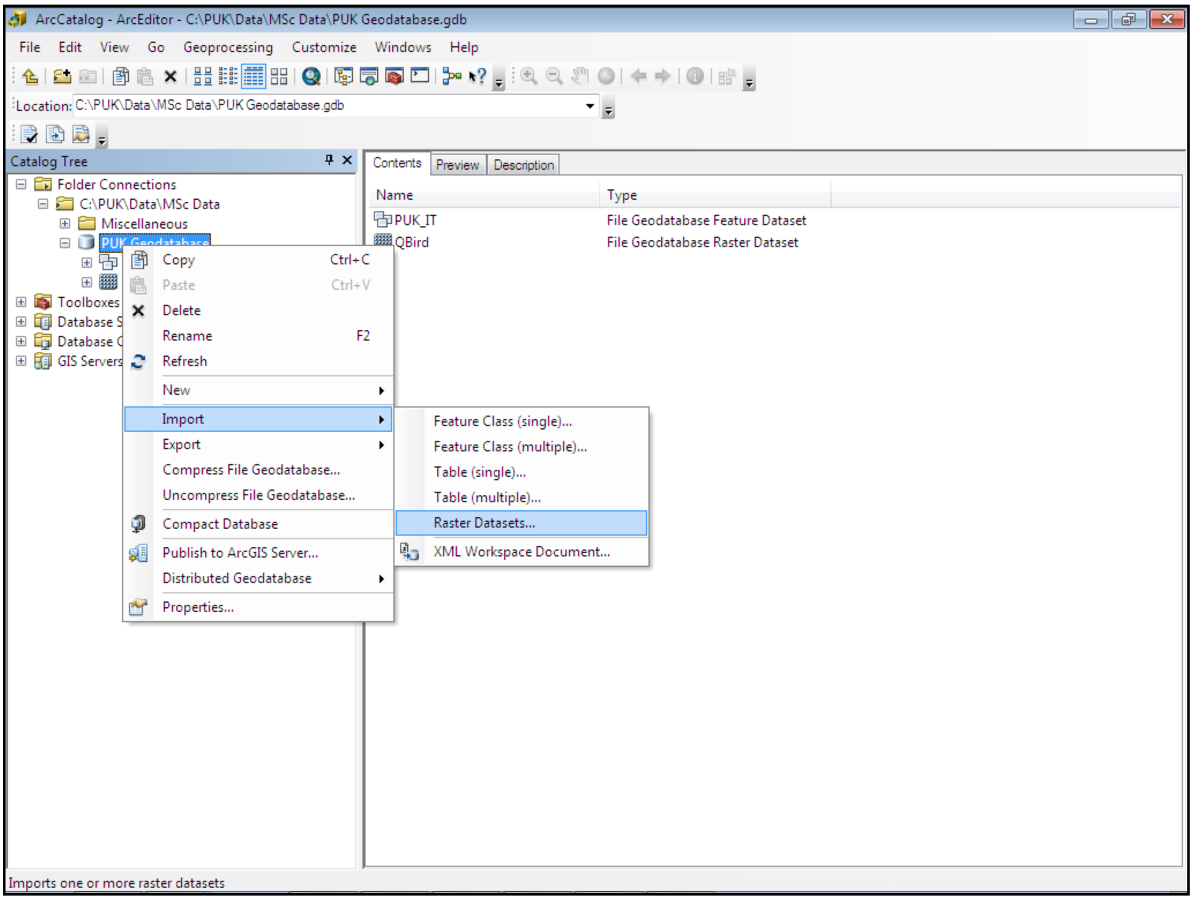

Figure 4.9: Importing the QuickBird (2008) data

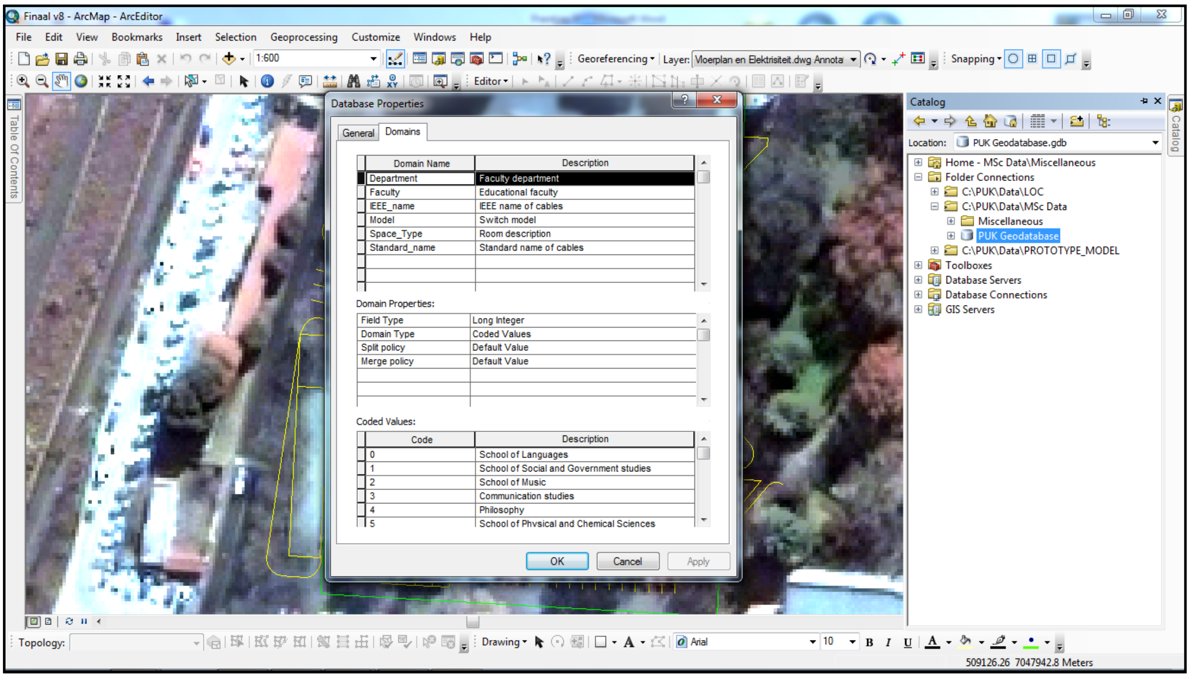

Figure 4.10: Creating domains for the PUK_Geodatabase

4.1.3 Aligning CAD data in GIS

The next phase of creating the data model consisted of aligning the CAD data in such a way, that it corresponds to the referenced QuickBird (2008) data. CAD files can be integrated into a GIS in a variety of ways.

One of the common problems encountered when using CAD data in a GIS environment, is that it has no spatial references. Although the design might be perfect, without aligning and referencing the CAD data, it is useless in a GIS environment. When CAD data is dragged into a map with referenced data, the CAD data appears to be on a large scale or very far away. The following technique describes how to scale, align and provide CAD data with spatial reference.

The first technique involves transforming the CAD data. Mattix (2006) describes the process as a sequence of steps. The first step is a pre-work checklist, to organize the data that will be used. The checklist consists of a number of questions such as:

- Do you have referenced data? Something for the CAD data to align with.
- Which CAD file do you work with? CAD files each have a number of layers including polyline, point, annotation, etc. Normally the polyline layer is used to align CAD data.
- Does the CAD file include projection information (a .prj file)? Sometimes CAD developers create CAD data, which contains information about the data's geographical and projected information. If not, the CAD data has to import a coordinate system in Arc Catalog, preferably the same coordinate system as the referenced data.
- Do you have a world file? A world file is text document which contains 2 sets of to-from x and y coordinates. These coordinates are used to align the CAD data with the referenced data.

Developing a world file is an essential part of transforming CAD data. A world file is a text document which contains from and to coordinates for two points on the CAD as well as the referenced data. A *from point* is the x,y coordinates of a point on the CAD data. A *to point*, is the x,y coordinates on the referenced data of the same point. In other words, a world file is used to transform CAD data from a certain scale and location, to another scale and location according to the referenced data. When developing a new world file, the first step is to start a blank map in ArcMap, and adding the CAD data to the data frame. The next step is to change the data frame's map units to match that of the CAD layer, after which a new data frame is inserted in order to add the referenced data (such as a satellite photo, existing data or an orhto photo). In order to create a functioning world file, certain criteria first need to be met:

- The CAD file may have no spaces in its name.
- The referenced data must have locations which can be found on the CAD data.
- The referenced data and the CAD data must have the same coordinate system.
- The referenced data must have a projection defined (Mattix, 2005).

Mattix (2005) describes the next step as deciding on two points to use as reference points on the data. The points must be visible on both CAD and referenced data. Zoom in on one of the points on the CAD layer as far as possible. On the point, create a new marker using the drawing toolbar. Double click on the marker to view its properties, then click on the size and position tab to view its x,y coordinates.

The next step is to create a new text document using notepad (WordPad tends to alter the characters). Copy the x-coordinate (not the space or the abbreviation, only the number) and paste it in the text document followed by a comma. Next paste the y-coordinate in the same fashion, followed by a space. Do the same for the second point, and paste the coordinates on the second line. The next step is to activate the new data frame. Find the first point on the referenced data, zoom in on it, create a marker and copy its x-coordinate to the text document after the CAD coordinates (with a space in between) for the same point, followed by a comma. Do the same for the y-coordinate. In the same fashion copy the referenced coordinates of the second point to the text document. If CADx1 is the x-coordinate of the first point on the CAD data, CADx2 is the x-coordinate of the second point on the CAD data, CADy1 is the y-coordinate of the first point on the CAD data, CADy2 is the y-coordinate of the second point on the CAD data, REFx1 is the x-coordinate of the first point on the referenced data, REFx2 is the x-coordinate of the second point on the referenced data, REFy1 is the y-coordinate of the first point on the referenced data and REFy2 is the y-coordinate of the second point on the referenced data, the text file should look like:

CADx1,CADy1 REFx1,REFy1

CADx2,CADy2 REFx1,REFy2 (see Figure 4.11).

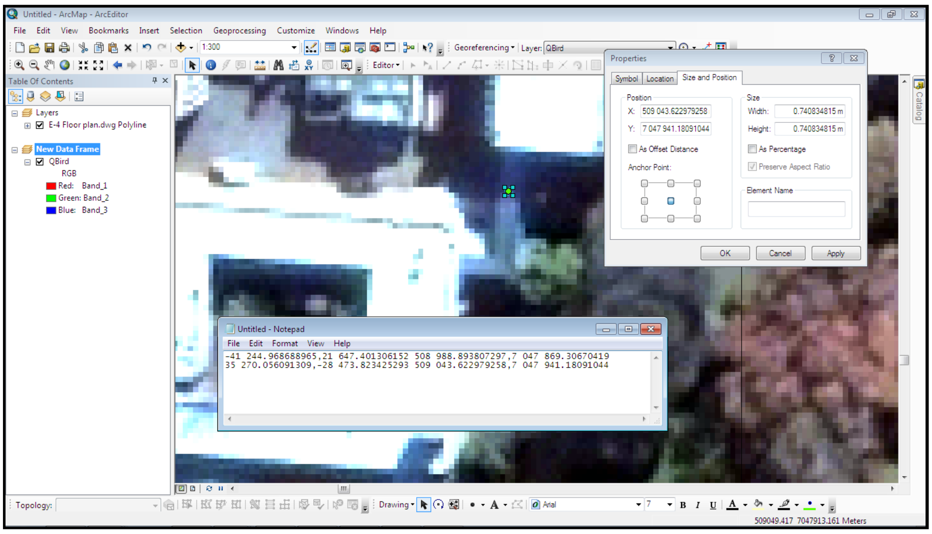

Figure 4.11: Creating a world file with Notepad

The penultimate step is to save the text file under the exact same name as the CAD file and with a .wld extension (Mattix, 2005).

The final step of aligning CAD data is to transform the CAD data. Open the CAD layer's properties box and select the transformation tab. In the transformation tab, enable transformations, choose a world file, browse for the data's world file and transform the data (Mattix, 2006).

Transformation of the CAD data is a very effective technique, although not 100% accurate (Figure 4.12). The first attempt at developing a world file is an lengthy process and one error or wrong space can lead to failure, however the technique gets easier and more accurate with practice. It is an easy but effective technique to implement. Another advantage is that a world file can be saved, so when the CAD data is used on a different occasion, it can be transformed easily. A negative aspect about creating a world file in this fashion is that it requires a lot of concentration and time.

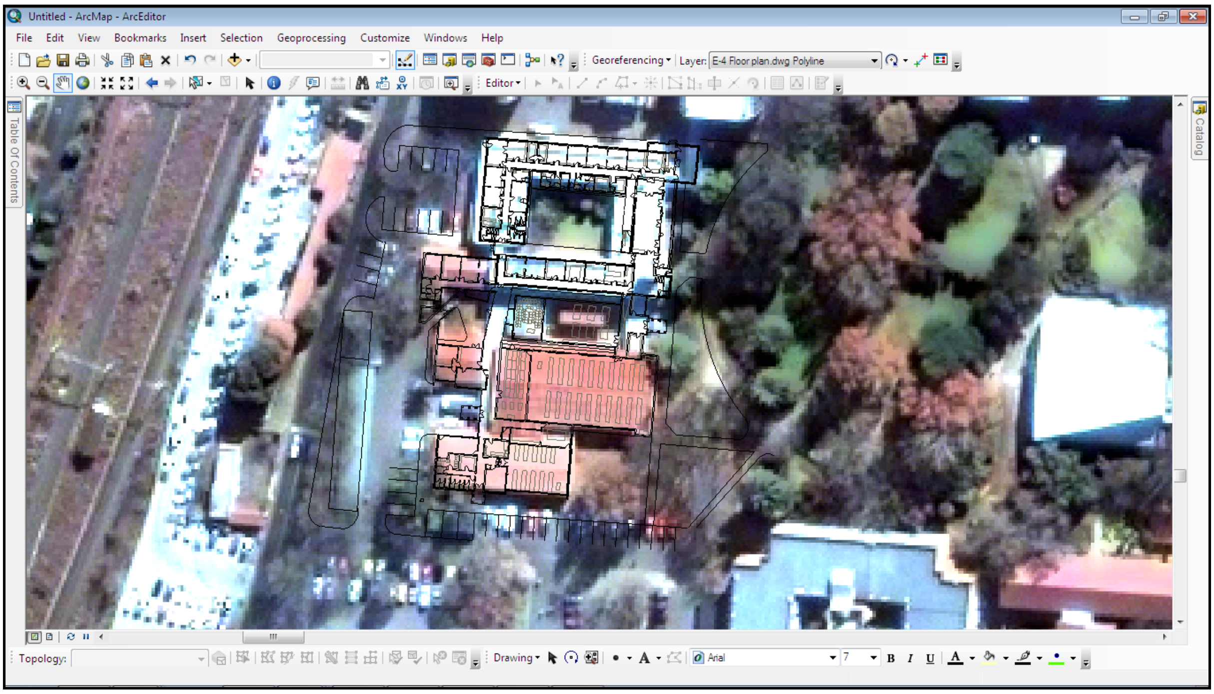

Figure 4.12: Aligned and scaled CAD data after technique 1 was applied

The second technique studied in this section which aligns CAD layers with GIS data is the georeferencing tool. This toolbar and its features as described by ESRI (2007), supports both raster as well as CAD layers. CAD transformation applies to the entire CAD dataset. If other data from the same CAD dataset is later added, all transformational alterations will be relevant for the new data. Georeferencing offers automatic transformation tools such as creating world files, but also offer interactive CAD altering tools using the mouse. Similar to the automatic transformation method discussed in the first technique, georeferencing uses a two point control for scaling and rotation. Some of the commands offered by georeferencing include: update georeferencing (commits to the transformation and creates a world file); fit to display (offers a quick and easy way to move a CAD layer into your area of focus); and flip or rotate either 90 degrees or according to manually selected degrees. Interactive commands include shift (shifting the CAD layers by dragging the mouse), rotate and scale (dragging the mouse down will reduce the size, and vice versa for enlarging the layer). Finally georeferencing tools offer an easy way of creating and saving world files by using a combination of the add control point and link table tools. The add control point tool is used to link at least two CAD points to their respective referenced data points. The link table tool depicts the control points and saves them to a world file and automatically transforms the data, as seen in Figure 4.13. Future transformations can be done easily by accessing the world file (ESRI, 2007).

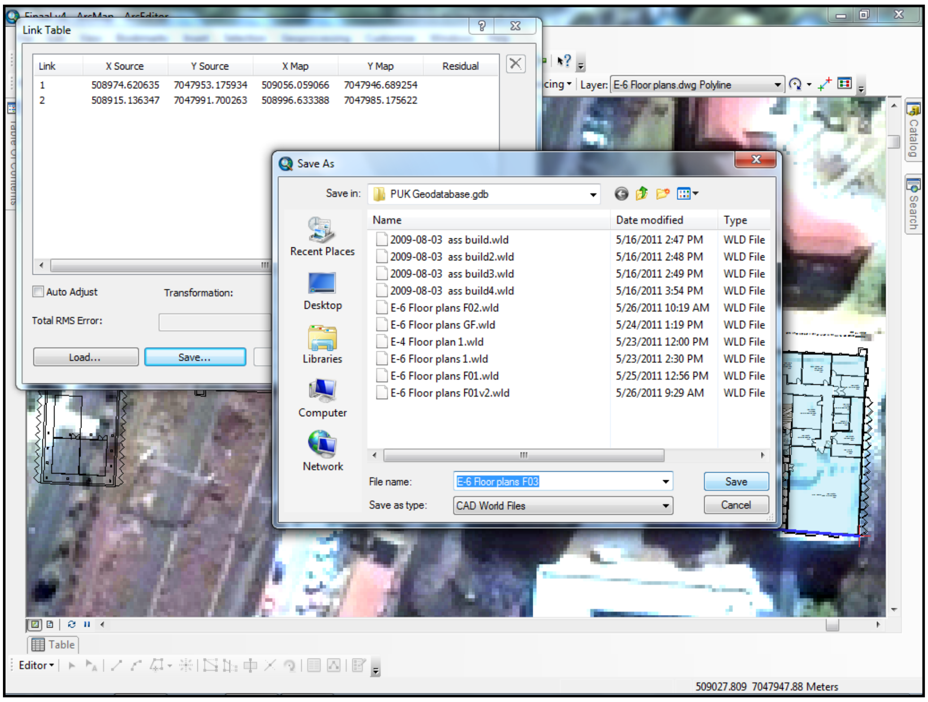

Figure 4.13: Aligning CAD data and saving a world file by using the georeferencing toolbar

The disadvantage to the georeferencing technique is that it is only applicable to CAD or raster data and cannot be implemented after CAD data has been converted or exported to GIS data. Georeferencing does however offer a variety of tools to align CAD data to referenced data, either automatically or interactively. The main advantage of georeferencing is the fact that it offers a very easy and accurate way to transform CAD data by creating, using and saving world files automatically. Even after transformation, if data doesn't align properly, additional tools are on available to tweak the data (rotation, scale and shift).

The final technique to align CAD data with referenced GIS data is the spatial adjustment method. According to the National Park Service (NPS) Northeast region GIS (2005) the spatial adjustment tool is used to edit the data after it has been exported as a shapefile. This method starts by adding the CAD data (either the polyline, point or polygon layer, whichever is going to be used) as well as referenced data to a blank Arc Map. It is essential that the CAD data and the data frame have the same coordinate system as the referenced data. It is also important to make sure that the spatial adjustment extension is loaded into Arc Map. The first step is to export the CAD layer as a shapefile because the spatial adjustment tool only handles shapefiles or feature classes. This is done by right-clicking on the CAD layer, pointing to “*Data*”, and clicking on “*Export data…*”. A new shapefile is created, as seen in Figure 4.14, which is identical to the CAD data.

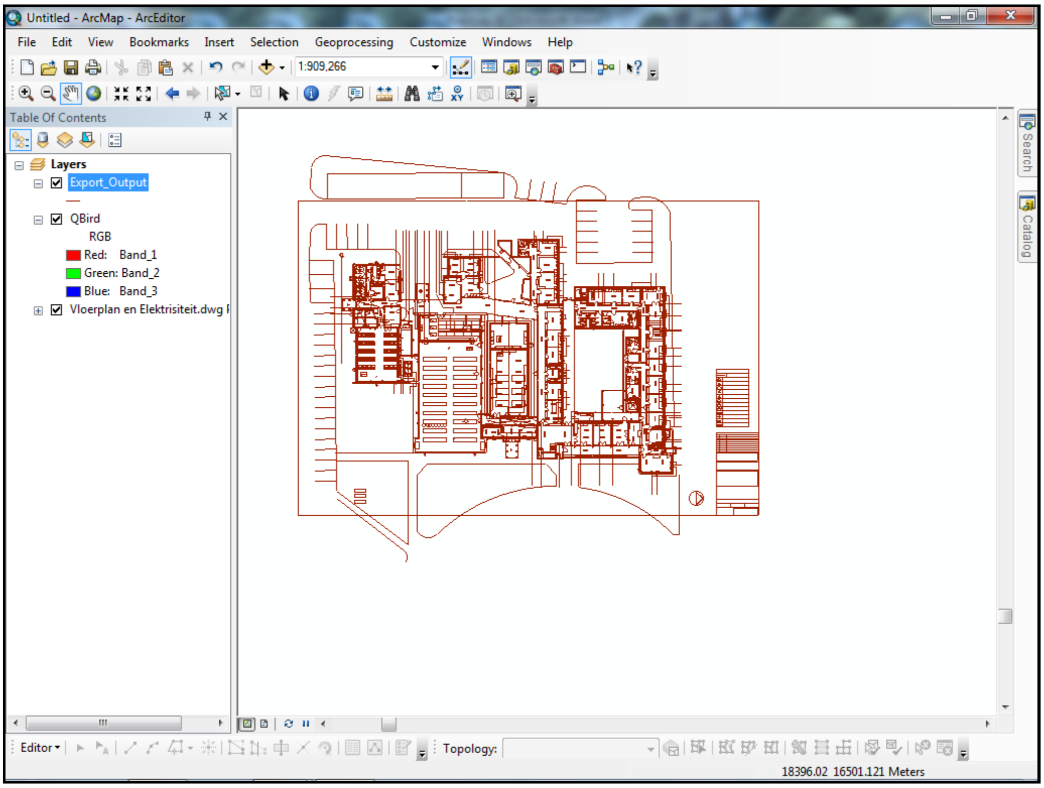

Figure 4.14: CAD data exported to a shapefile

The next step is to select the “start editing” tool on the Editor toolbar as well as selecting “set adjust data”, under the spatial adjustment dropdown box, followed by “all features in these layers”. The “New displacement link” tool on the spatial adjustment toolbar is used to create links between identical control points on the newly created shapefile and referenced data. Click on the tool to activate it, click on a control point on the shapefile (for example a street intersection or building corner), next click on the identical control point on the referenced data in order to complete the link. Repeat this process until satisfactory amount of links have been made (Figure 4.15).

The final steps are to save the link in a links file in order to use the same links in the future; set the adjust methods to either transformation similarity (maintains aspect ratio and doesn’t skew the drawing) or transformation affine (alters the drawing to fit); and click adjust on the spatial adjustment toolbar. Export the shapefile again to save its new alignment (NPS northeast region GIS, 2005).

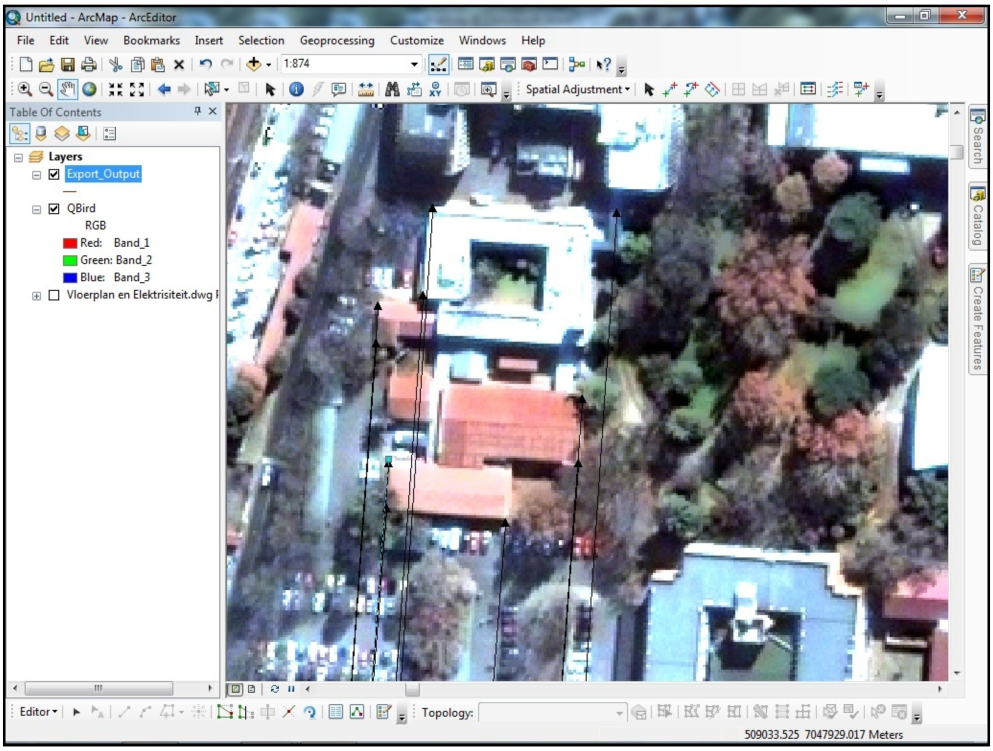

Figure 4.15: Links between the shapefile and the referenced data

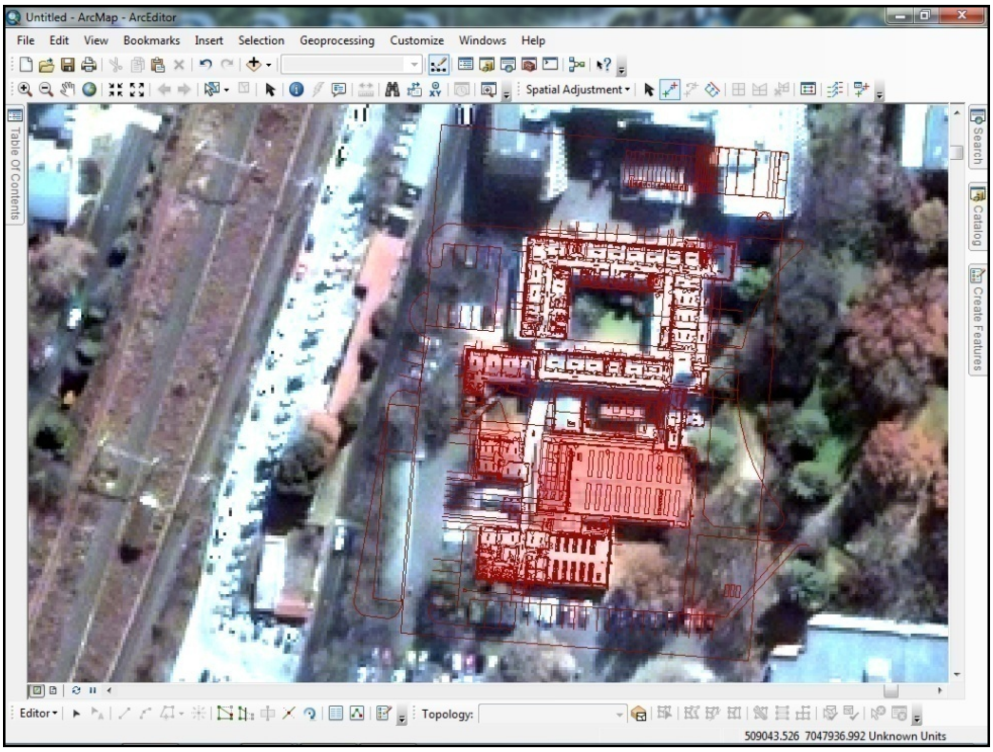

Figure 4.16: The shapefile is aligned with the referenced data

The result is that the shapefile comes into picture and is aligned with the referenced data (Figure 4.16). The advantage of using this technique is that the CAD data can be transformed into a shapefile and the changes in alignment can be saved so that the next time that the data is used it stays aligned. Spatial adjustment however is a lengthy method when a lot of links are needed. This technique is also not always very accurate. Although it can manage many control points, it lacks the interactive altering tools of the georeferencing toolbar.

All three methods were tested and it was decided to implement the georeferencing tool to align the CAD data with referenced GIS data. The transformation techniques require the user to create his/her own world file when it is absent, whereas the georeferencing technique offers a tool to create and save world files with ease. The spatial adjustment tool has a similar technique to transform the exported shapefile to the correct alignment, but lacks the interactive altering tools of the georeferencing technique. It is always possible to export the CAD data to a shapefile after it has been transformed and interactively edited.

4.1.4 Creating a feature dataset and feature classes

The first step before populating the geodatabase with data is to correctly organize the database. The aim of the project is to determine to what extent GIS software can be implemented in order to manage, analyze and visually illustrate an IT-network between buildings as well as inside of buildings on a campus. The data model offers a network as well as topologies to ensure that the data integrity is kept intact. Therefore all the feature classes which will participate in the network as well as the topology have to be stored in a central feature dataset. The feature dataset created for this data model was named: PUK_IT.

PUK_Zones feature class

The first feature class digitized in the PUK_IT feature dataset was PUK_Zones. The Zones feature class was developed by using a combination of the CAD infrastructure data of the campus (Figure 4.3) and the campus map (Figure 1.1) to depict where the zones were located. The initial step is to correctly align the CAD data, by implementing the georeferencing toolbar technique. The campus map was used as a reference, while the CAD data was utilized to digitize a new zones feature class. While the Editing toolbar initiated an editing session, it was possible to create new polygon features, while populating their respective fields in the attribute table. The PUK_Zones feature class is only implemented to provide orientation information of the study area and for that reason does not need z-values (Figure 4.17).

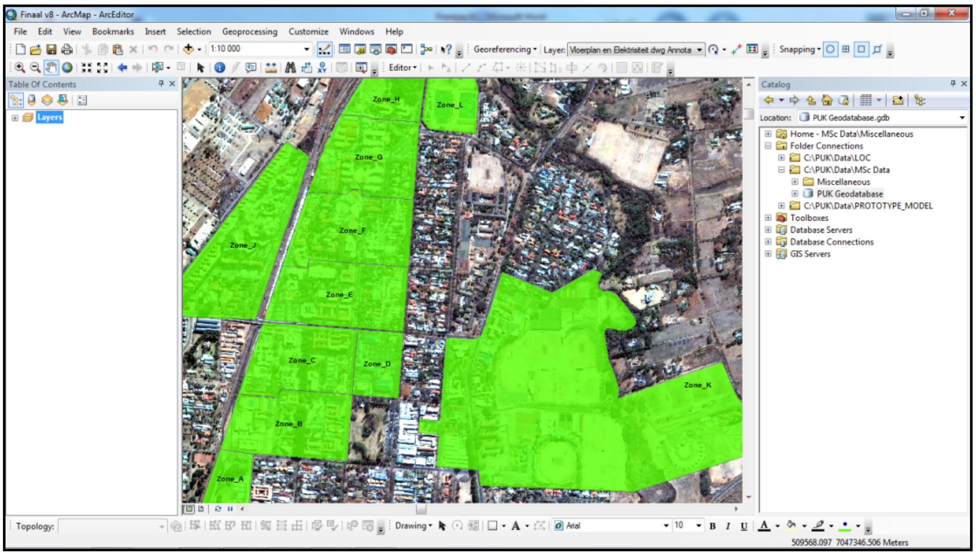

Figure 4.17: PUK_Zones feature class

PUK_Buildings feature class

The PUK_Buildings feature class represents the outline of a building. The PUK_Buildings feature class was created in the PUK_IT feature dataset. It was vital to assign the domains to the fields (where applicable) while the fields were created. The CAD data which showed the building layout was used to create the features for this feature class. The first step was to align the CAD files correctly with the referenced satellite image. The georeferencing toolbar technique was utilized to align the CAD data with the buildings.

The CAD data represents a wall as three parallel lines. It was decided to use the outer lines of the building walls as a guide to create the building outline polygon. The inner lines would later be used as the boundaries of rooms such as corridors or offices. This would ensure that the rooms inside a building will never be located outdoors and therefore be topologically incorrect. The PUK_Buildings features were created by digitizing along the outer walls, while populating the fields of the attribute table (Figure 4.18).

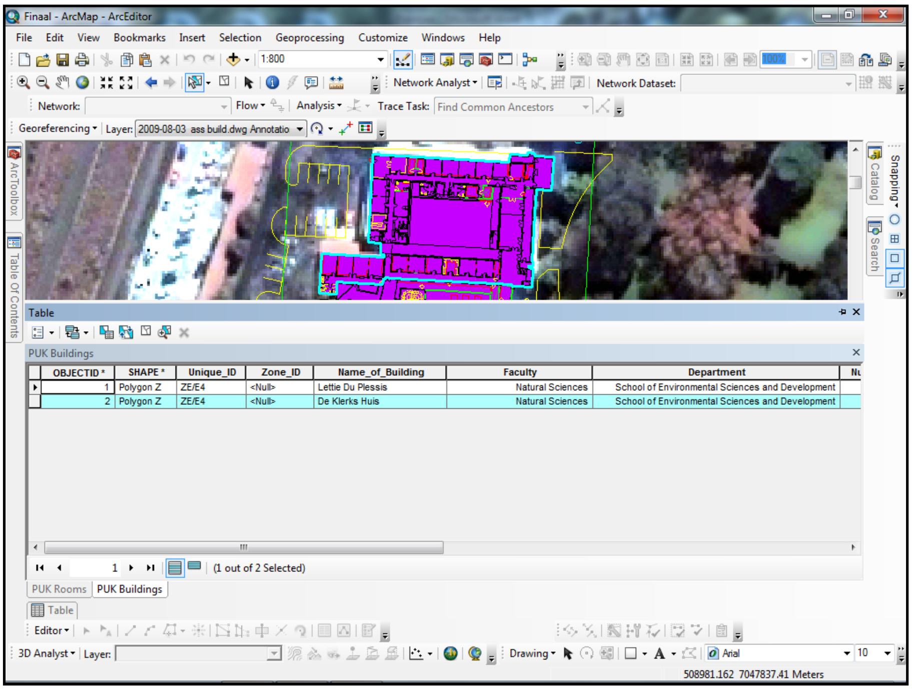

Figure 4.18: Populating the attribute table while creating the building features

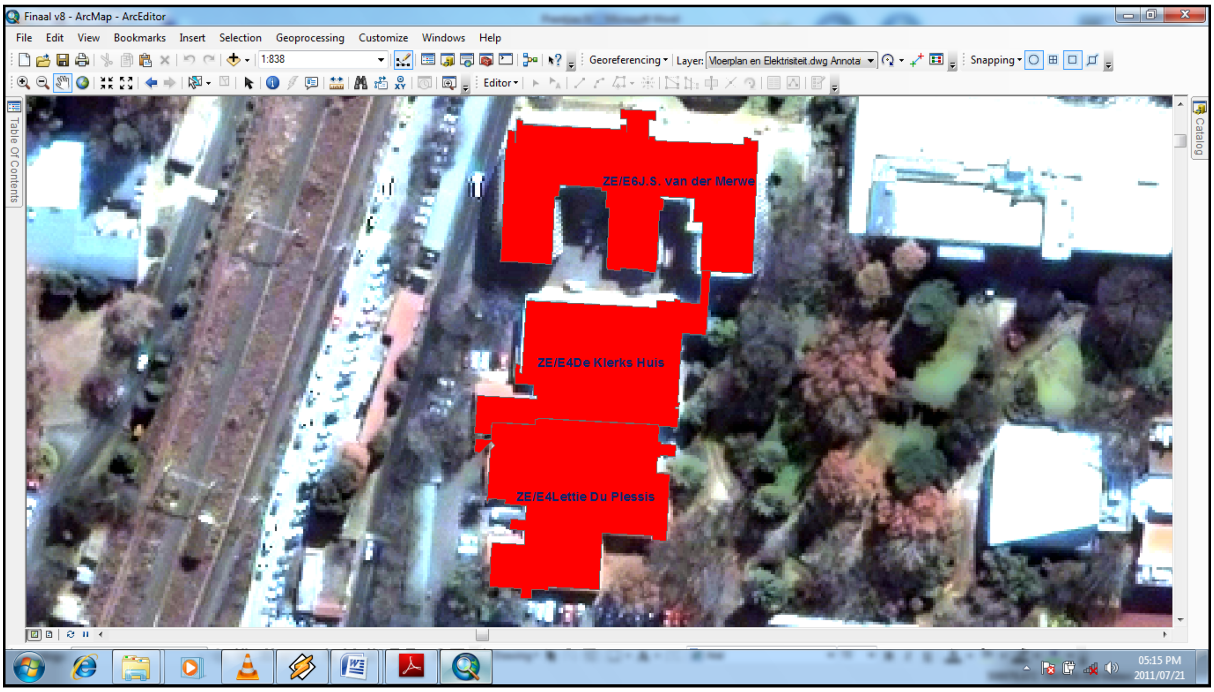

Figure 4.19: PUK_Buildings feature class

The PUK_Buildings feature class outlines the two buildings which serve as the study area for this study, as seen in Figure 4.19. Building E4 was divided into two features due to the fact that these parts have individual local names, but are still classified as E4. Both the buildings therefore share the same Unique_ID (Figure 4.18). This feature class was implemented in the topology to ensure that all indoor facilities are covered by buildings. The PUK_Buildings feature class can also be extruded in the 3D environment in order to provide a guideline as to what the building looks like as a whole.

PUK_Rooms feature class

The PUK_Rooms feature class is a polygon feature class which depicts the building layout inside a building, including offices, corridors and class rooms. The feature class is created in the PUK_IT feature dataset, and the fields are assigned their correct domains. After the feature class was created, the subtypes, as well as their pre-defined default values for each field were delineated. The features were divided into subtypes according to the floors. The building layout CAD data was then aligned with the referenced data by using the georeferencing toolbar technique. Due to the fact that the individual floors are adjacent to one another in the CAD files, the CAD files had to be aligned separately for each floor. After the rooms of a certain floor were digitized in the feature class, the CAD data was aligned again so that the following floor could be digitized. Figure 4.20 shows how the 3rd floor was aligned, after the 2nd floor was digitized for building E6. Figure 4.21 shows that the 3rd floor CAD data is properly aligned.

The CAD data was utilized to digitize the rooms. The CAD files depict the walls between rooms as three lines. It was decided to use the inner lines (lines closest to room) as the wall boundary of each room. This meant that the walls were omitted, leaving gaps between adjacent rooms. This was done to accurately display the location of a feature which is situated next to the wall. For instance if a wall was depicted as a common line between two rooms, and a utility feature was located along the wall, it would be impossible to visually establish in which room the utility is located. If the wall is omitted, there are two boundary walls, one for each floor. This makes it easier to determine in which room a utility is located.

The PUK_Rooms feature class depicts the rooms on all floors, and are thus provided with z-values. The ground floor has a z-value of zero, while the 1st floor and 1st basement floor has z-values of 5 and -5 respectively. As the floors increase in height, its z-values increase with an increment of 5. The number 5 was chosen in order to simplify the z-value increments awarded to switches, which will be explained in the next section. For example: the ground floor's z-value is 0; the 1st floor's z-value is 5; and the 2nd floor's z-value is 10. In the same way, the z-values of the basement floors decrease with an increment of 5.

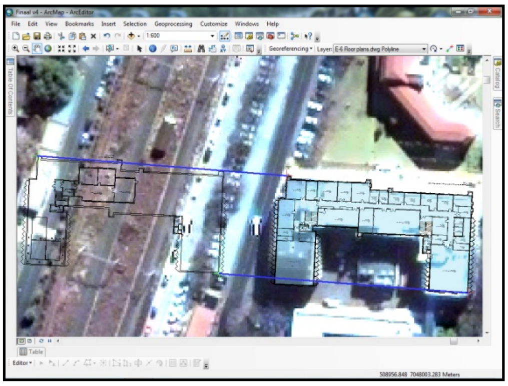

Figure 4.20: Aligning the 3rd floor of building E6

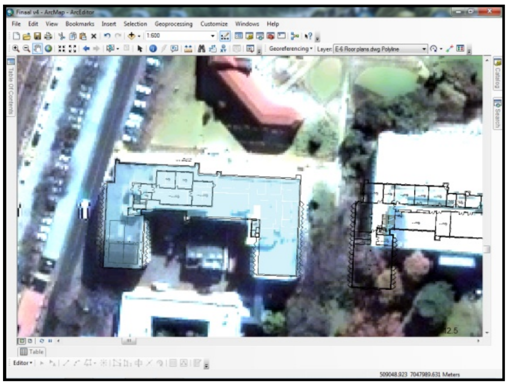

Figure 4.21: The 3rd floor of building E6 is properly aligned

The z-values are assigned to each vertex, while the feature is being created. The z-values have to be assigned before the sketch is finished. The z-values of the polygon vertexes cannot be altered after the sketch has been completed. The z-values are assigned by right-clicking on the vertex and selecting the Sketch properties option. A window opens which depicts the x and y-values of each vertex, as well as a default z-value of 0 for each vertex (Figure 4.22). The z-values for each vertex are manually altered.

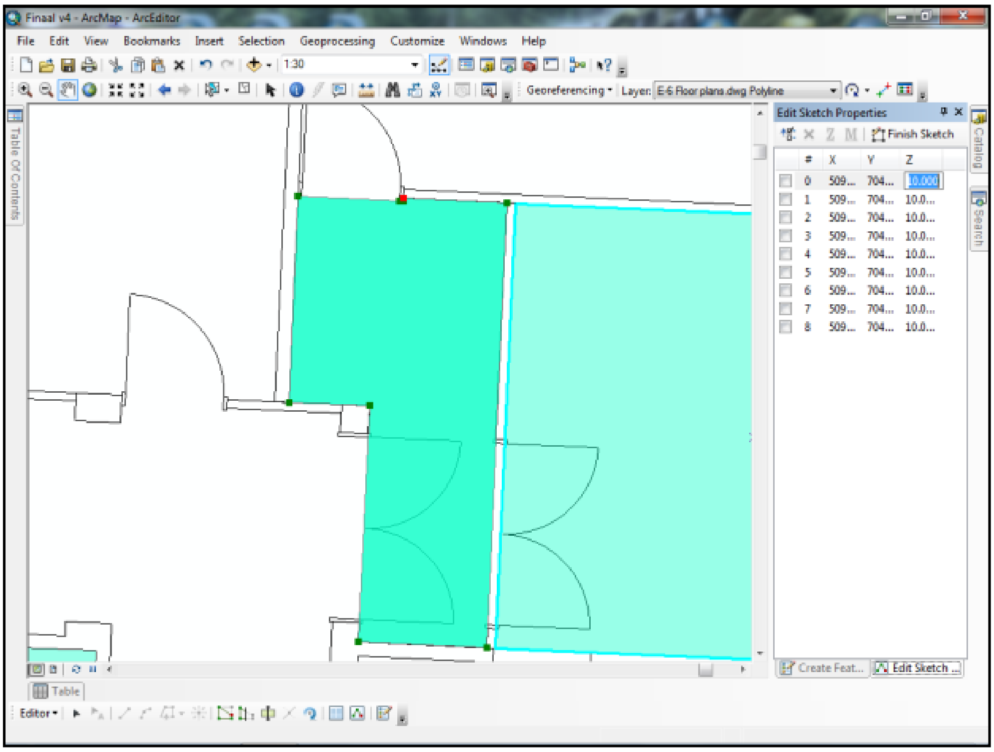

Figure 4.22: Altering the z-values of the feature vertexes

Creating steps between the individual floors acquired extreme concentration. Although in a 2D environment the steps looks like a rectangle, in a 3D environment it had to look like a polygon sloped at an angle (Figure 4.23).

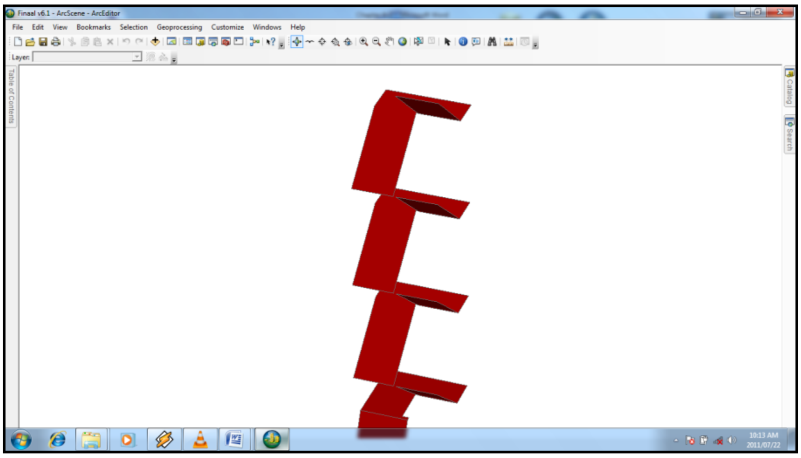

Figure 4.23: Steps are sloped in a 3D environment

All the steps between floors lead to a platform between the two adjacent floors. The steps had to be depicted in three parts. The first part is a polygon leading from the lower floor to a platform between the floors at a sloped angle. The second part was a level polygon between the floors, while the final part was another sloped polygon between the platform and the higher floor. This was achieved by assigning the vertexes of the steps polygon which is attached to the lower floor a z-value which is equal to that floor. The top vertexes were assigned z-values which are equal to that floor plus half an increment (2.5). All the vertexes of the platform are assigned a z-value equal to that of the lower floor plus half an increment. The sloped steps which lead from the platform to the higher floor would be assigned a z-value equal to that of the platform. The vertexes attached to the higher floor, was assigned z-values equal to that of the higher floor. Figure 4.24 show that, while steps appear as normal rectangles in the 2D environment, the vertexes are assigned different z-values which ensure a sloped polygon.

For example if the second floor has a z-value of 10; the bottom part of the steps would also have a z-value of 10, while the part of the steps attached to platform would have z-values of 12.5. The platform would have a z-value of 12.5 for all the vertexes. The sloped polygon leading to the higher floor would have z-values of 12.5 for the vertexes attached to the platform. The vertexes attached to the 3rd floor would have z-values of 15.

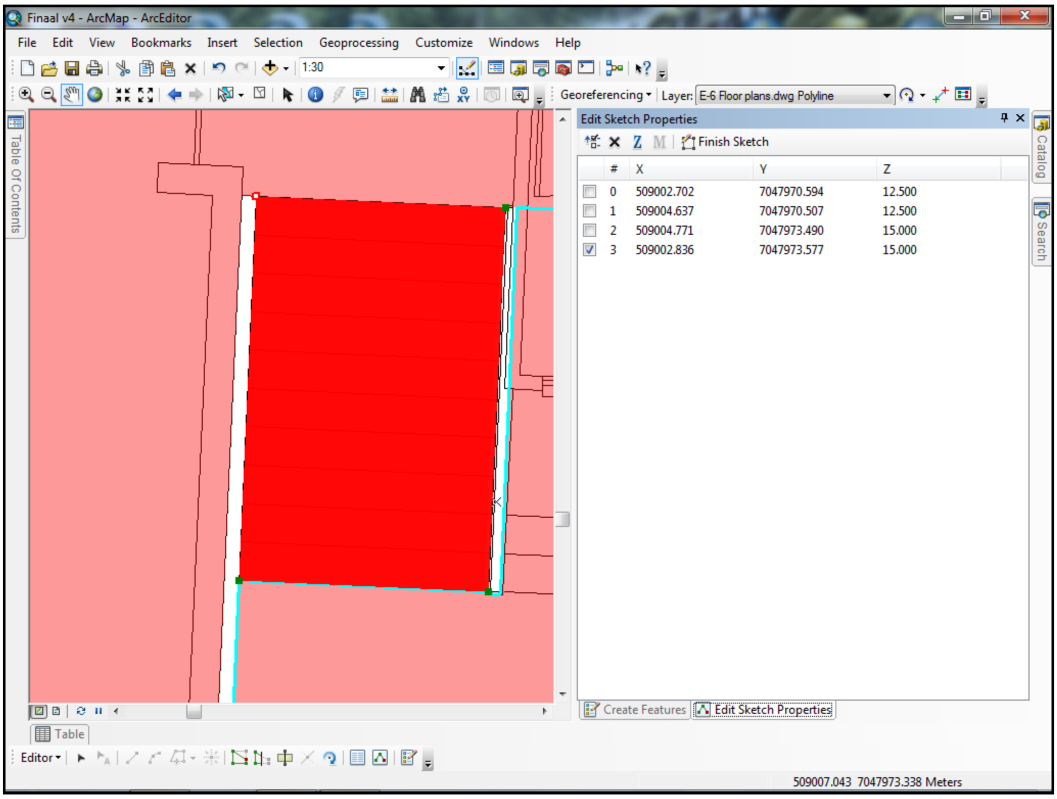

Figure 4.24: Vertexes are assigned different z-values in order to slope the polygon

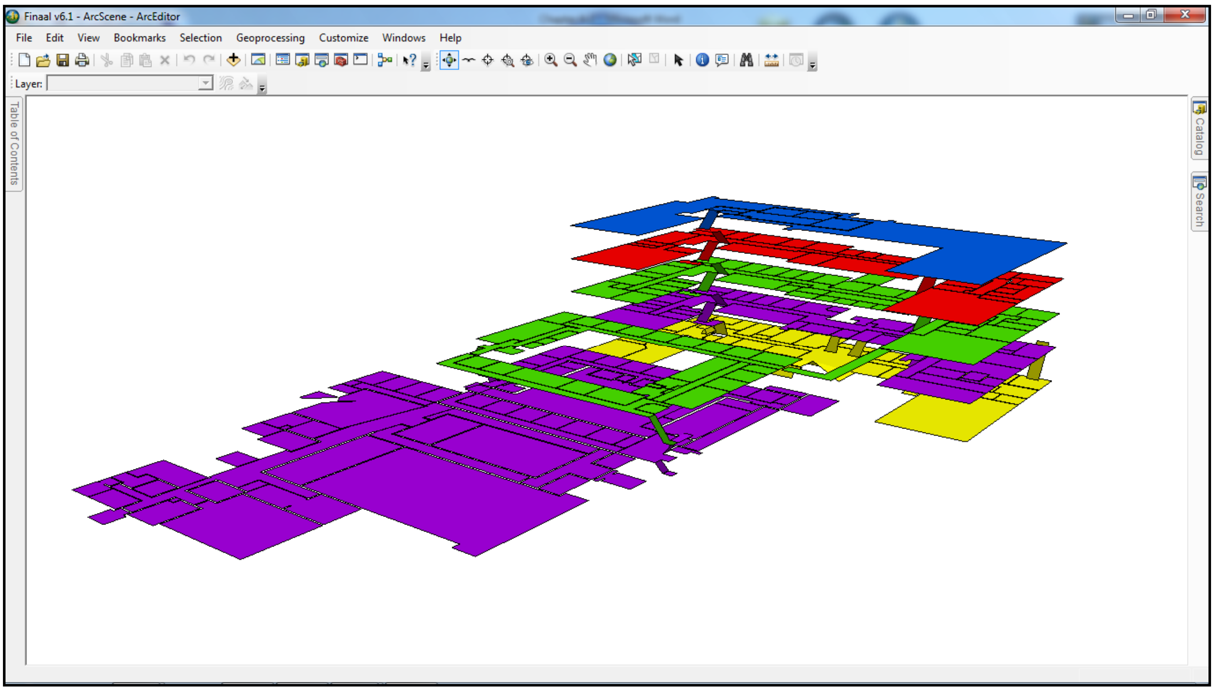

Figure 4.25: PUK_Rooms feature class in a 3D environment

After the vertexes of the polygon were assigned the appropriate z-values, and after the polygon sketch had been finished; the attribute table of each feature was populated with the correct information. One such example is the Unique_ID field. The final part of the unique identifier is the room number, which was also acquired from the annotations in the CAD dataset. The PUK_Rooms feature class in conclusion shows the rooms of each floor of each building, divided into floor types. In a 3D environment all the rooms on the same floor have the same z-values (Figure 4.25). The feature class also provides an attribute table which stores non-spatial information for each room.

Switches feature class

The switches serve as the source as well as the routers of the network, therefore it was decided to create the switches first. The switches feature class was divided into three subtypes, namely local switches, zonal switches and the source switch. Information about the location of the switches inside the three study area buildings (local switches) were derived from the building manager. Information about the location as well as the attribute fields of the source and zonal switches were unavailable and had to be estimated.

The three buildings contain one utility room each, which in turn contains the switches. These utility rooms operate as the main network distribution center for each building. The Lettie du Plessis building (E4), the De Klerks Huis building (E4) and the J.S. van der Merwe (E6) buildings have 4, 4, and 9 switches, respectively.

In the utility rooms, the switches are located above one another, and had to be displayed as such. This meant providing each switch with a z-value which is within the z-value bounds of the utility room. For example, if a utility room has a z-value of 0, and the next floor has a z-value of 5; then the switches' z-values have to be greater than 0 and smaller than 5. It was decided to increase the z-values by increments of 0.5 between switches. The z-value increments between floors (5) provide a round number which can be easily divided by 0.5 increments. A z-value range of 5 between floors provides a potential of 10 possible switches on top of one another (Figure 4.26).

The switches were created using the "Point at end of line" tool. By utilizing this tool, it was possible to award z-values to the points. While each sketch was being edited, the "Point at the end of line" tool was used to make two vertexes with the same x- and y-values. The "Sketch properties"-window was opened and the appropriate z-value was given to the second vertex, after which the sketch was finished, as seen in Figure 4.26. The last step was to populate the attribute table fields for the newly created switch feature.

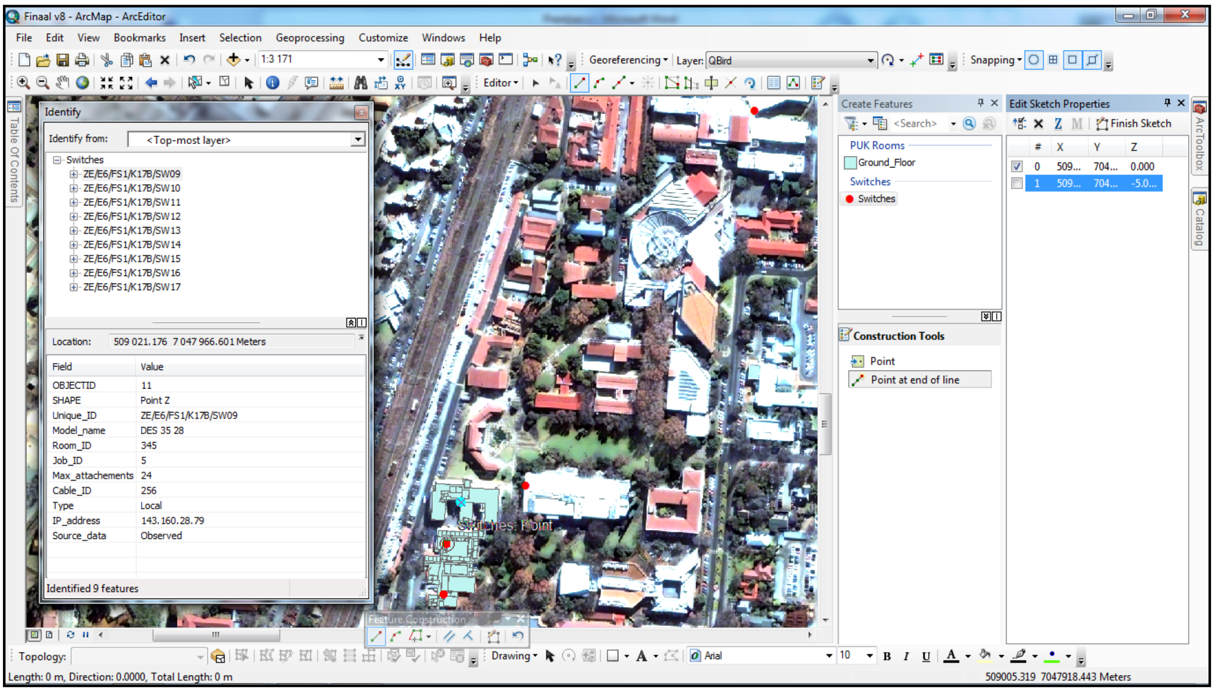

Figure 4.26: Switches feature class

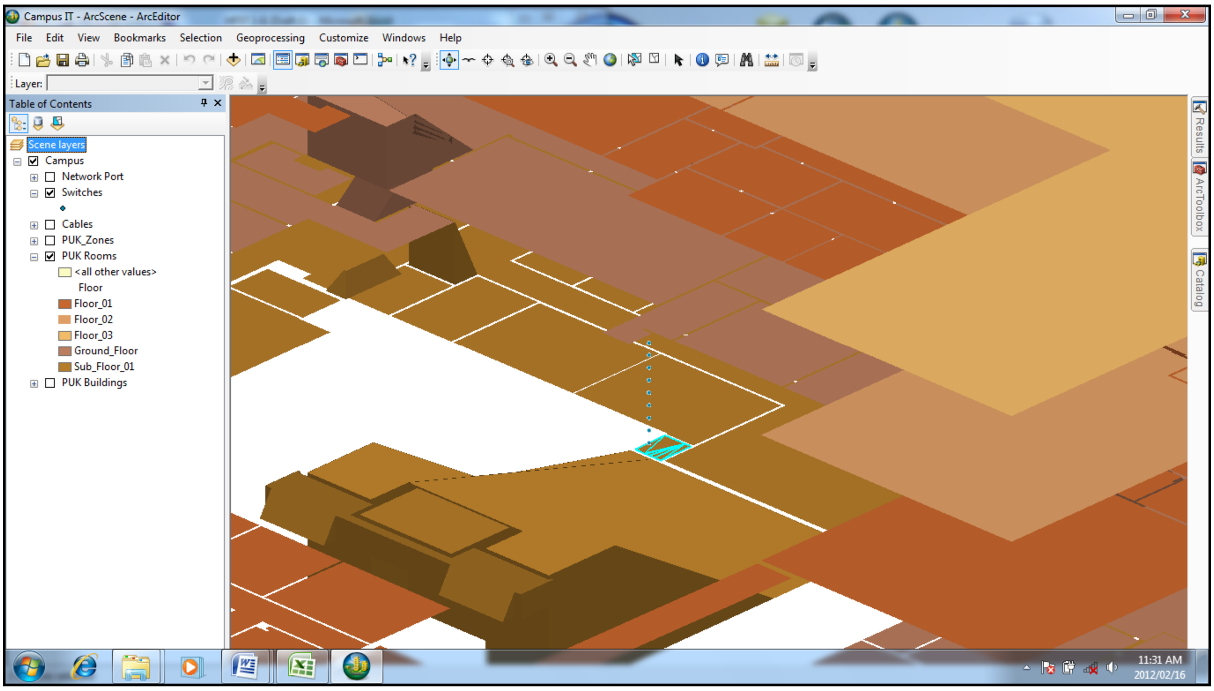

Figure 4.27: Utility room and switches of building E6 in 3D

The final product is a feature class containing switches which have z-values and are contained within their respective utility rooms. Figure 4.27 highlights the utility room of building E6, as well as its nine switches, in a 3D environment.

Network_Port feature class

The Network_Port feature class is a point feature class and was created in a similar fashion as the Switches feature class. The main difference between the two is that the switches' z-values had to be between floors, while the z-values of the network ports was equal to that of the floor it was located on. The z-values of the network ports therefore also increase with increments of 5 between floors.

Similarly to the Switches feature class, the "Point at the end of line" editing tool was employed to create the point features and provide the appropriate z-values to each. Due to the fact that most of the network ports are located along walls the snapping environment was set to snap to edges. Some exceptions do however occur, for example in a computer laboratory network ports might be located at a table in the middle of a room.

In the ArcMap (2D), the network ports resemble a collection of disorganized dots this is due to the fact that the different z-values cannot be distinguished from each other, as seen in Figure 4.28.

When the network ports are viewed in a 3D environment (Arc Scene) the network ports become organized according to each floor (Figure 4.29). After a new network port feature is created, its attribute table fields are populated. These fields include the Unique_ID, the Source_Data field as well as the foreign key fields (Room_ID and Job_ID).

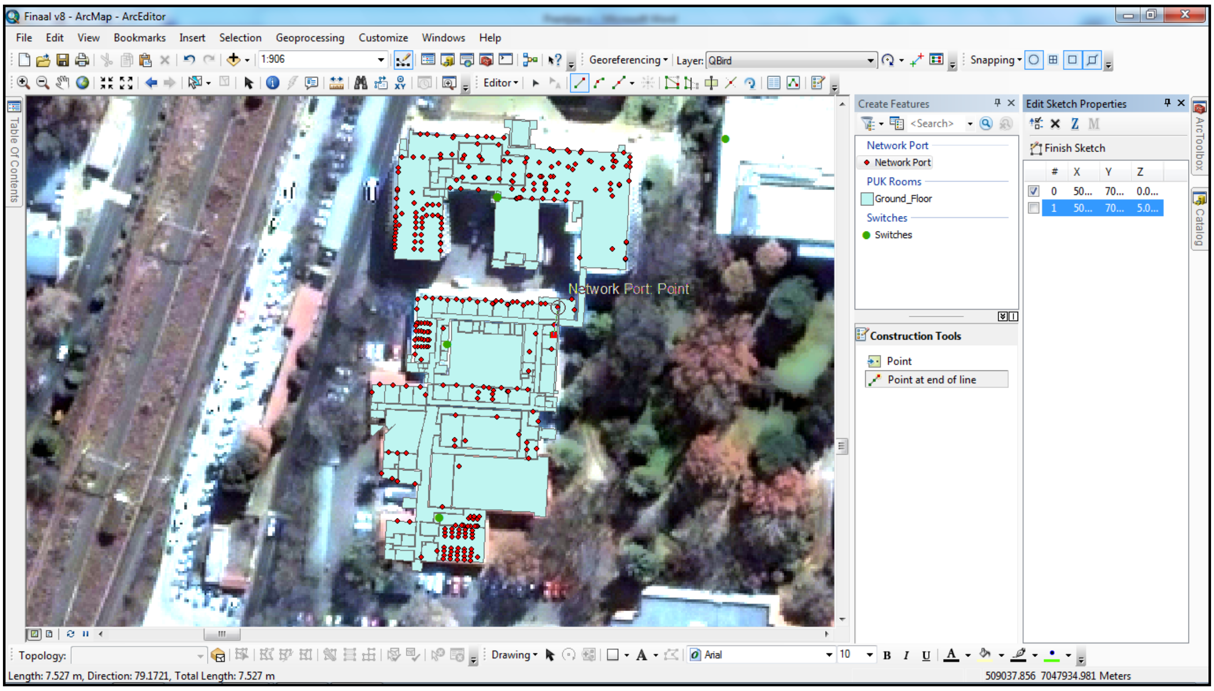

Figure 4.28: Network_Port feature class in Arc Map

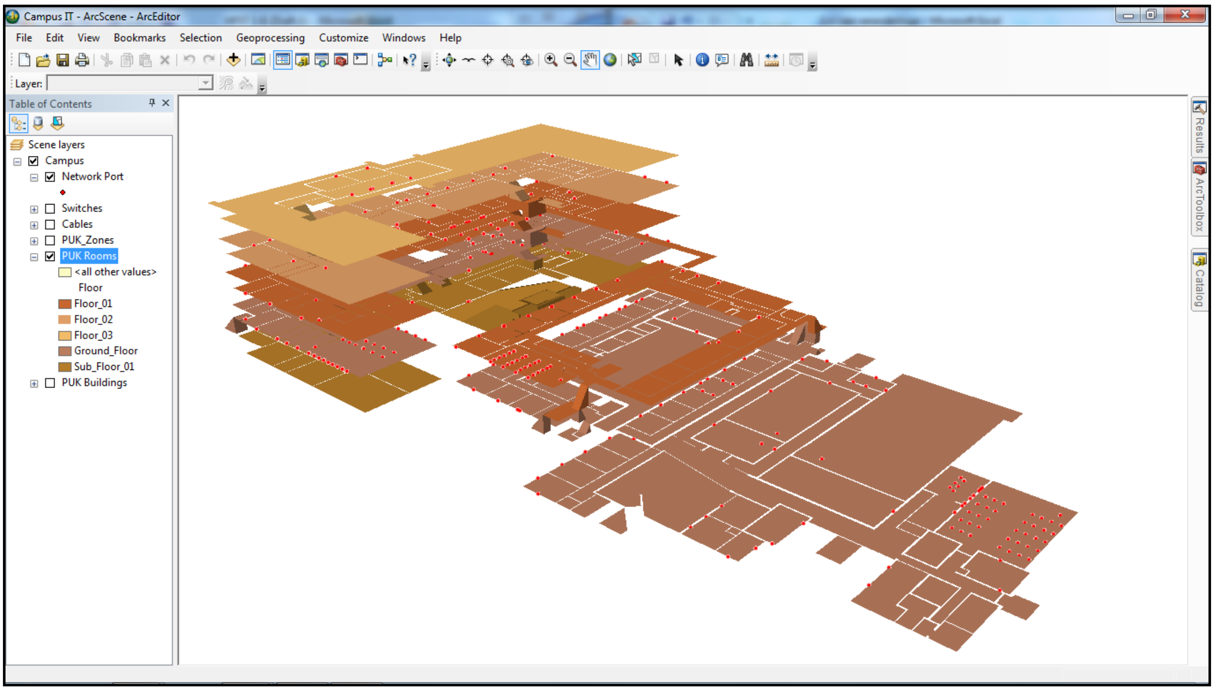

Figure 4.29: Network_Port feature class in ArcScene

Cables feature class

The cables feature class represents the connection between network ports and switches in the IT-network. This is the only line feature class among the utilities. Although the Cables feature class has five subtypes, this model only utilizes three.

The Category 5e cable is used to connect network ports inside a building to the correct switches. Each network port in a building is connected to the network by its own cable. It is thus important for a singular cable element to be able to be both horizontal and vertical. When a new cable feature is created it is very important to assign the correct z-values in the "Sketch properties"-window to the appropriate vertex. Where a line changes from a horizontal line to a vertical line, two vertexes are awarded the same x- and y-values, but different z-values. This ensures that a cable can be created horizontally on a specific floor as well as vertically between floors (Figure 4.30).

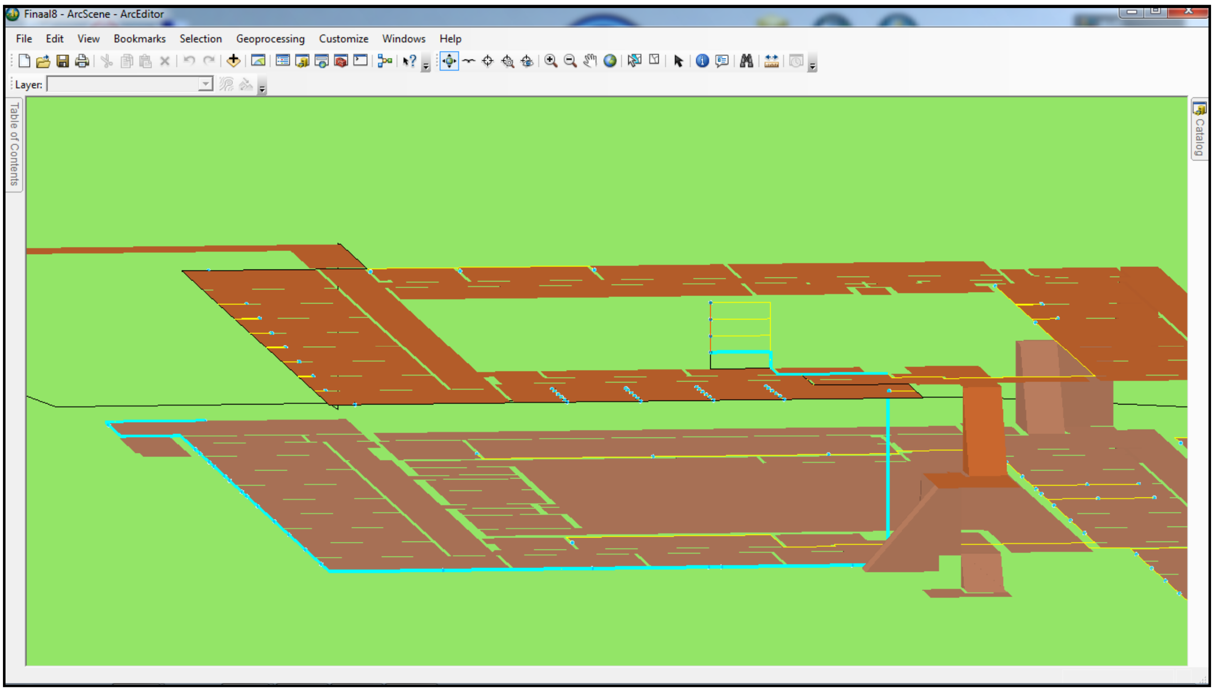

Figure 4.30: A cable can be horizontal as well as vertical

The multi-mode optical fibre cables are used on the Potchefstroom campus to connect the switches to one another. These cables are very short in this model and takes part in the network as vertical connections between switches. These cables are essential to the network and ensure that the switches do not create stand-alone errors when later creating a network dataset.

The single-mode fiber optical cable is utilized to connect the source and the zonal switch with one another. The single-mode optical fiber cable is also employed to link the zonal switch to one of the local switches in each building. This cable type can run for long distances without losing data and is therefore used to connect geographically distant switches on campus. The location of this cable subtype was derived from CAD data depicting the outdoor campus infrastructure. After the CAD data was georeferenced, the single-mode optical fiber cables were digitized.

The final product is a feature class which makes up the bulk of the network. The Cables feature class is the connection between macro campus switches (source and zonal switch) and micro switches. The cables also connect each individual network port to the network as seen in Figure 4.31.

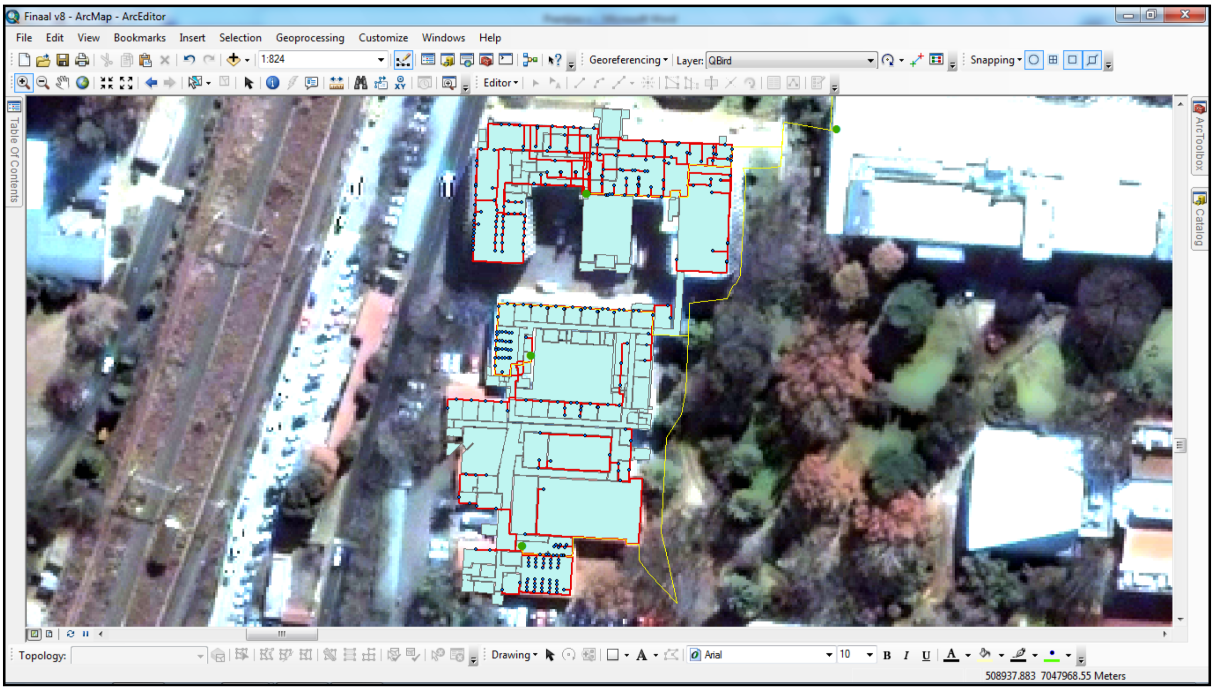

Figure 4.31: Cables feature class in ArcMap

When the Cables feature class is viewed in a 3D environment, its scope becomes more apparent. Figure 4.32 displays how the single-mode optical fiber cables link the indoor network with the rest of the campus, as well as how the network ports on different floors are connected to the network in a 3D environment.

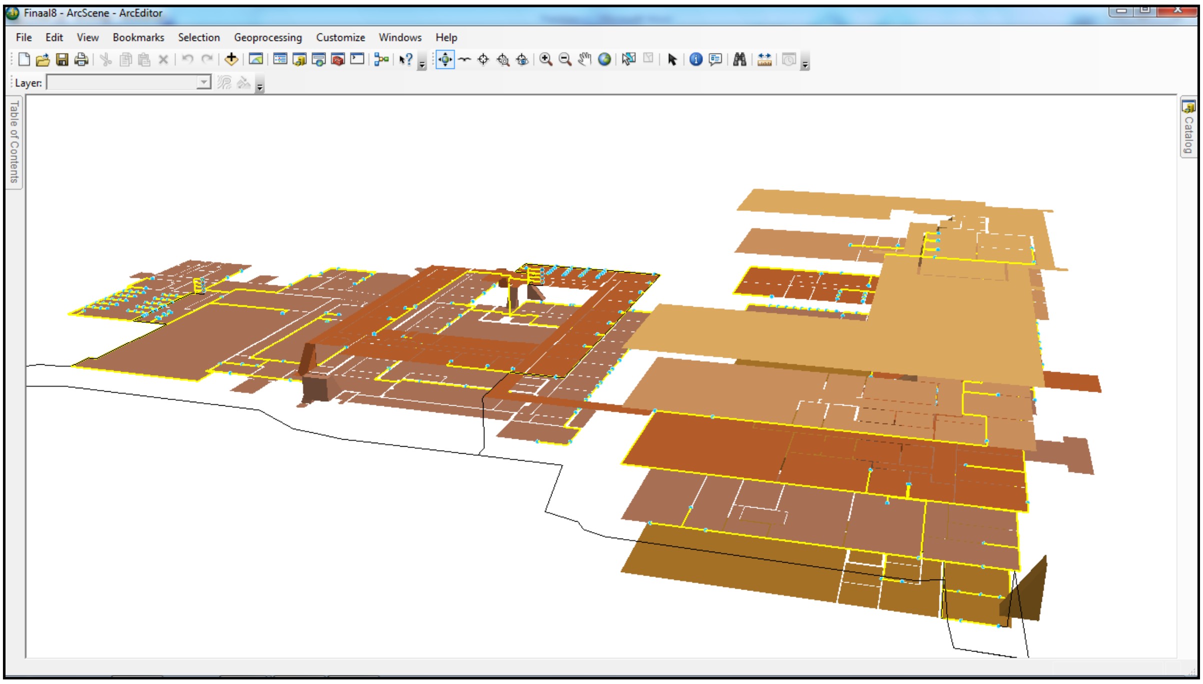

Figure 4.32: Cables feature class in a 3D environment

4.1.5 Creating non-spatial tables and relationship classes

The pilot model contains three non-spatial information tables. The Owners table lists the individuals responsible for each room, while the Maintenance_register and List_of_Contractors contain records of maintenance tasks done on the utilities, and the person who performed these tasks, respectively. The tables do not have spatial reference, and therefore cannot be stored in the feature dataset, instead they are saved as stand-alone tables in the geodatabase. After the tables have been created in Arc Catalog, they have to be opened in Arc Map and populated during an editing session.

A total of 11 relationship classes were created for the data model. Relationship classes can only be created between feature classes located in the same feature dataset, and are stored in the feature dataset. A relationship class can however be created between a feature class stored in a feature dataset and a stand-alone table in the geodatabase, this relationship class is stored as a stand-alone relationship class in the geodatabase.

The data model contains 6 relationship classes between feature classes, 4 relationship classes between feature classes and tables, and 1 relationship class between two tables. A relationship class is created by either right-clicking on the geodatabase or the feature dataset. The “New Relationship Class” wizard runs the designer through the processes of creating a new relationship class such as: assigning a name to the relationship class; defining the tables which are participating and assigning the cardinality. The wizard also helps to define the origin table, destination table, primary key and foreign key (Figure 4.33).

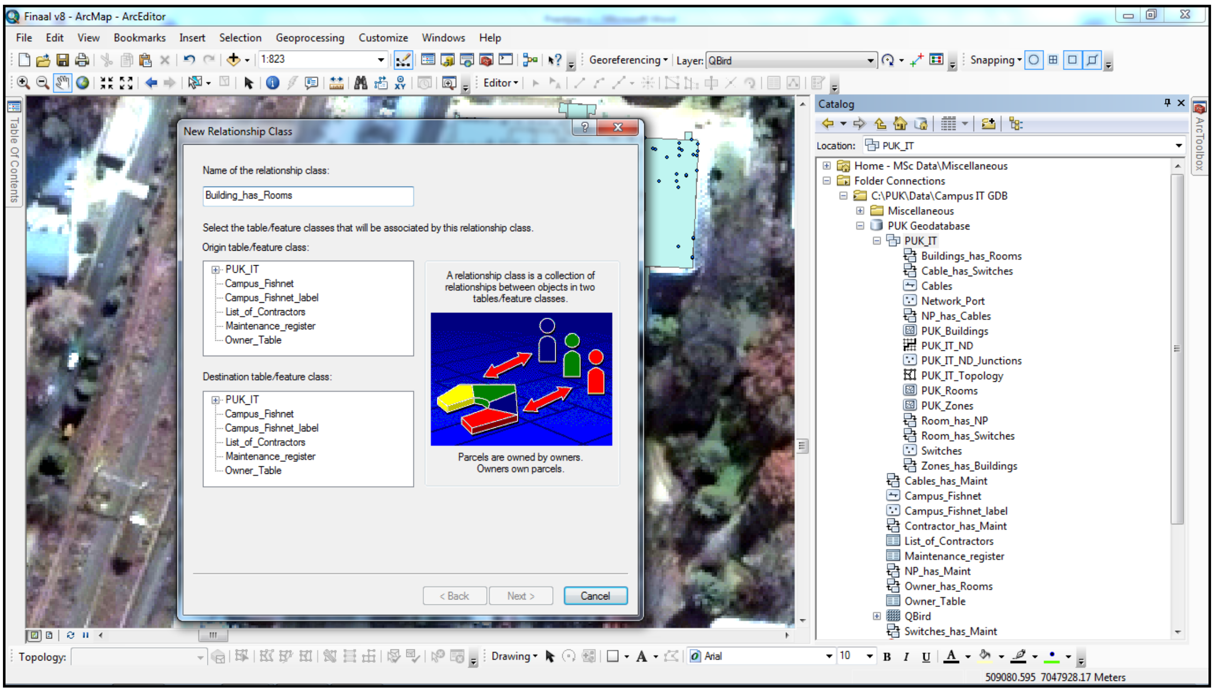

Figure 4.33: Creating a relationship class

The data model contains a single many-to-many relationship class between Cables and Switches. A many-to-many relationship class is created differently from other relationship classes. In this instance the “New Relationship Class” wizard is also employed to create the relationship class. The wizard automatically creates an intermediate table between the two related features.

The intermediate table can be viewed by dragging the relationship class from ArcCatalog to the table of contents in ArcMap, where it can be opened. Initially the intermediate table is blank. It is necessary for the geodatabase developer to manually populate this table.

The intermediate table can only be edited during an editing session. The first step is to open the attribute tables of the related features, as well as the intermediate table. In this case the Switches attribute table and the Cables attribute table were opened alongside the Cables_has_Switches relationship class. The second step is to open the attribute window by clicking on the “Attributes” button on the Editor-toolbar.

The final step of linking two features through a many-to-many relationship starts with selecting the specific elements in their respective tables. As the elements are selected they appear in the attributes window. Open the folder of either one until the layer of the linked feature class is visible. Right-click on the related layer and choose the “Add selected” option (Figure 4.34).

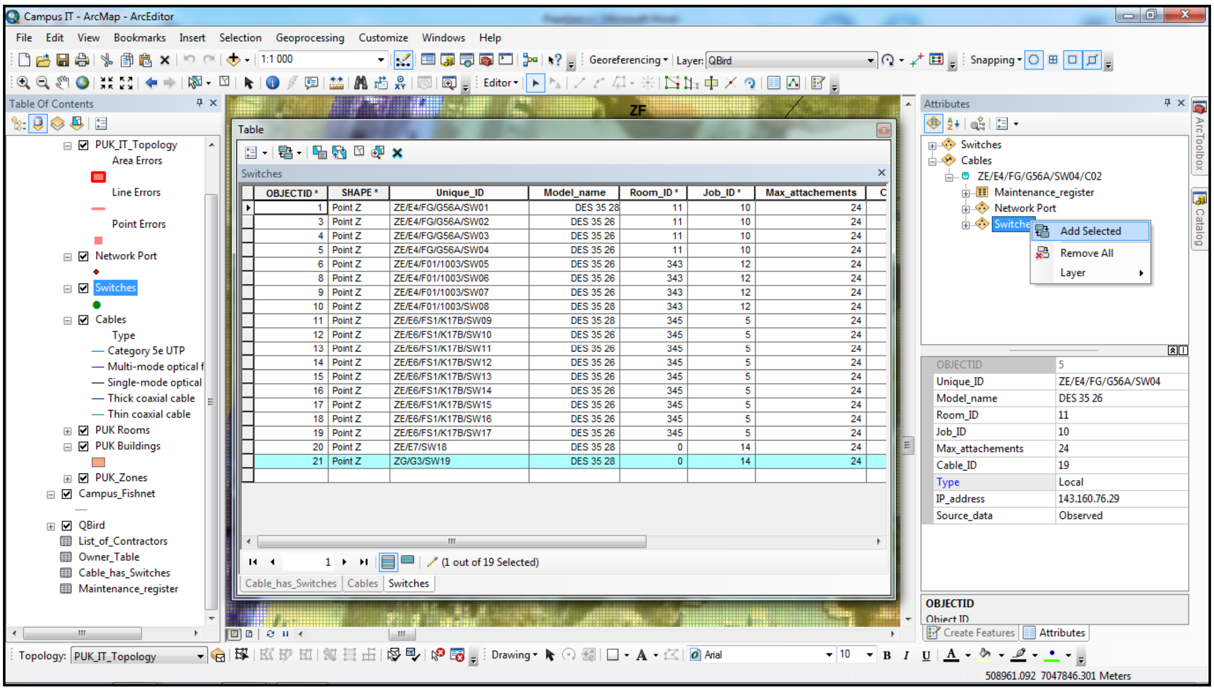

Figure 4.34: Populating an intermediate table

After the selection has been added, the intermediate table is populated for the records that were selected. The intermediate table depicts the primary key for both of the related records. This process has to be repeated manually for each record, until the intermediate table is fully populated. The Cables_has_Switches relationship class intermediate table, containing 350 entries, can be seen in Figure 4.35.

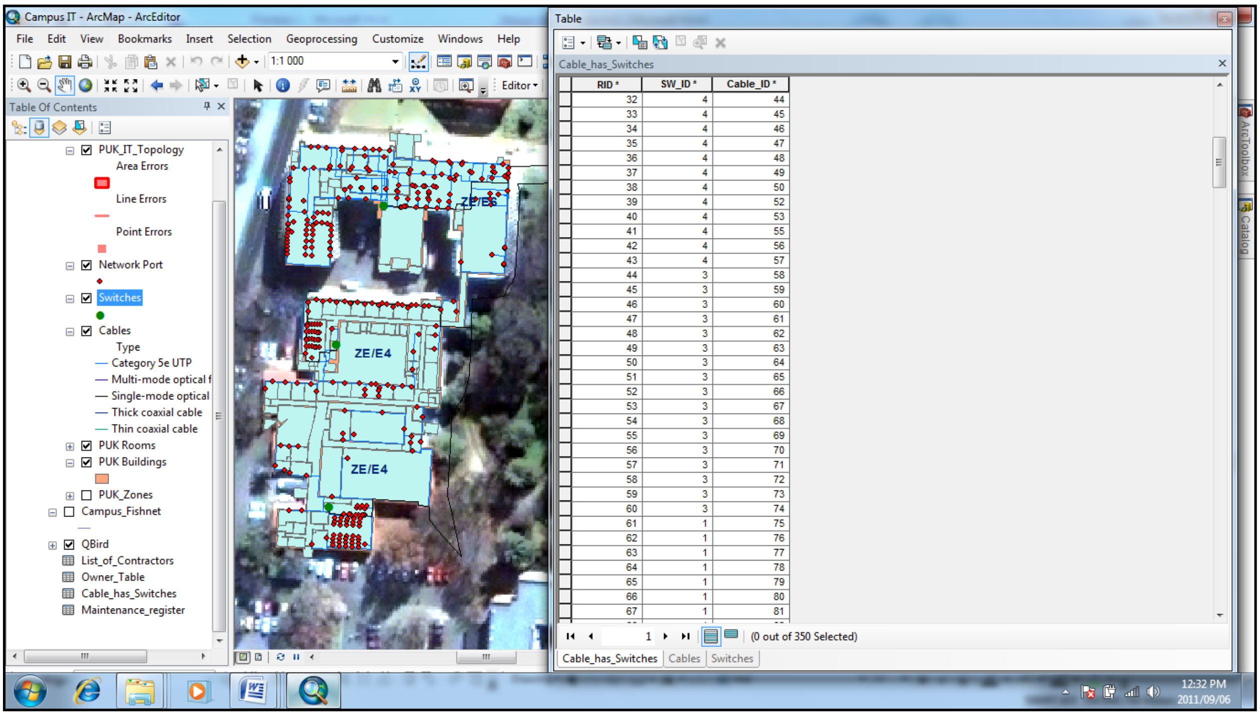

Figure 4.35: Cable_has_Switches relationship class intermediate table

4.1.6 Creating a topology and a network dataset

Creating topology rules ensures the data model's spatial integrity. As described in Chapter 3, the data model has a total of 7 topology rules. The rules are stored in a topology. Feature classes participating in the topology have to be stored in the same feature dataset. Each feature dataset can only contain one topology.

A topology is created by right-clicking on the specific feature dataset in ArcCatalog and choosing the "*New*" > "*Topology*" option. The topology creation wizard appears which helps the designer to assign a name to the topology as well as define and enter a list of topology rules that will be enforced on the feature classes. After the topology is created it has to be verified in order to become active. The topology is stored in the feature dataset.

A topology has to be placed in the table of contents in ArcMap in order to see the topological errors which are present in the model. Figure 4.36 displays 5 errors on this data model.

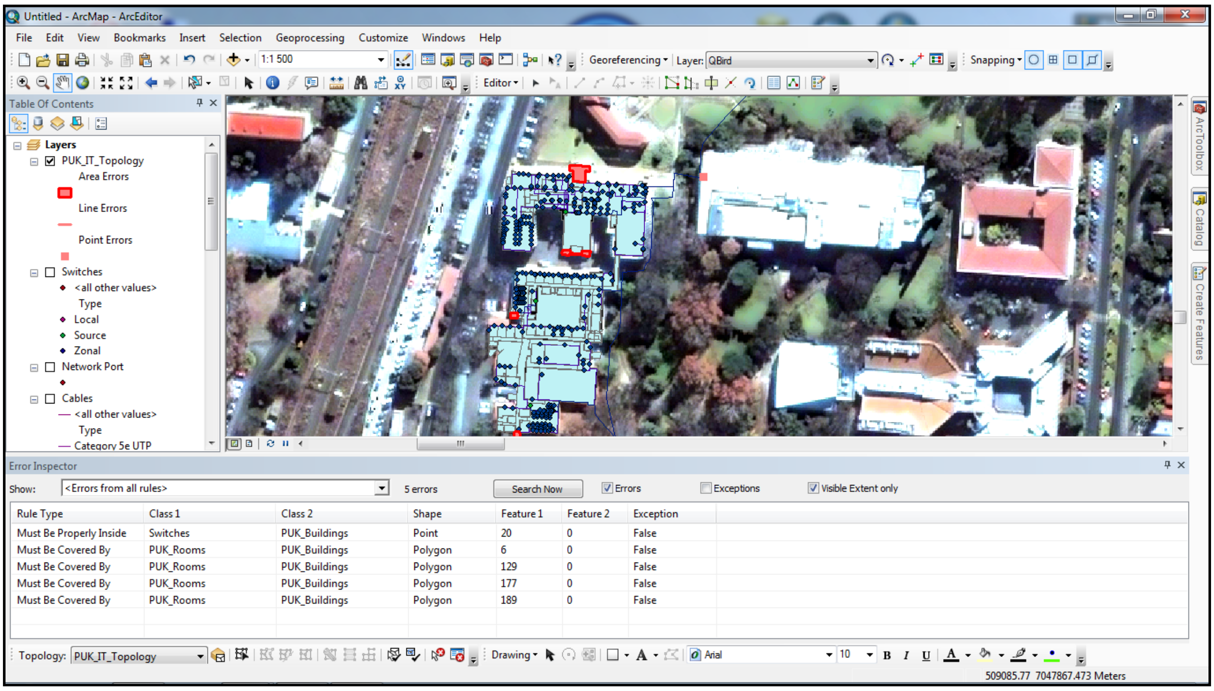

Figure 4.36: Topological errors on the model

Four of the errors are derived from the “PUK_Rooms must be covered by PUK_Buildings” topology rules. This rule ensures that a room is created inside of a building. The errors which occur are all external areas where one or more of the boundaries have the same x- and y-values as the PUK_Buildings. Thus the error occurs because a boundary of the external area is not covered by PUK_Buildings but aligns with the buildings.

The second error type occurs from the “Switches must be properly inside polygon PUK_Buildings” topological rule. In this case there were no data available to create PUK_Buildings or PUK_Rooms polygons for either the Zonal or Source switches. The error occurred due to the fact that the zonal switch was not covered by a building. It was decided to create a topological exception for the switch error, instead of creating false building and room polygons.

Due to the fact that the “PUK_Rooms must be covered by PUK_Buildings” errors were in fact real-life exceptions, it was decided to create topological exceptions for each. An exception is created during an editing session. On the Topology toolbar, the “Error Inspector” is opened. This window allows the user to search for errors and exceptions. Each error is listed as a row in the error inspector, as seen in Figure 4.37. On the exception field of an error, right-click on the value and choose the mark as exception option. After an error is marked as an exception, it is no longer seen as a topological error. An error can be unmarked as an exception at any time during an editing session.

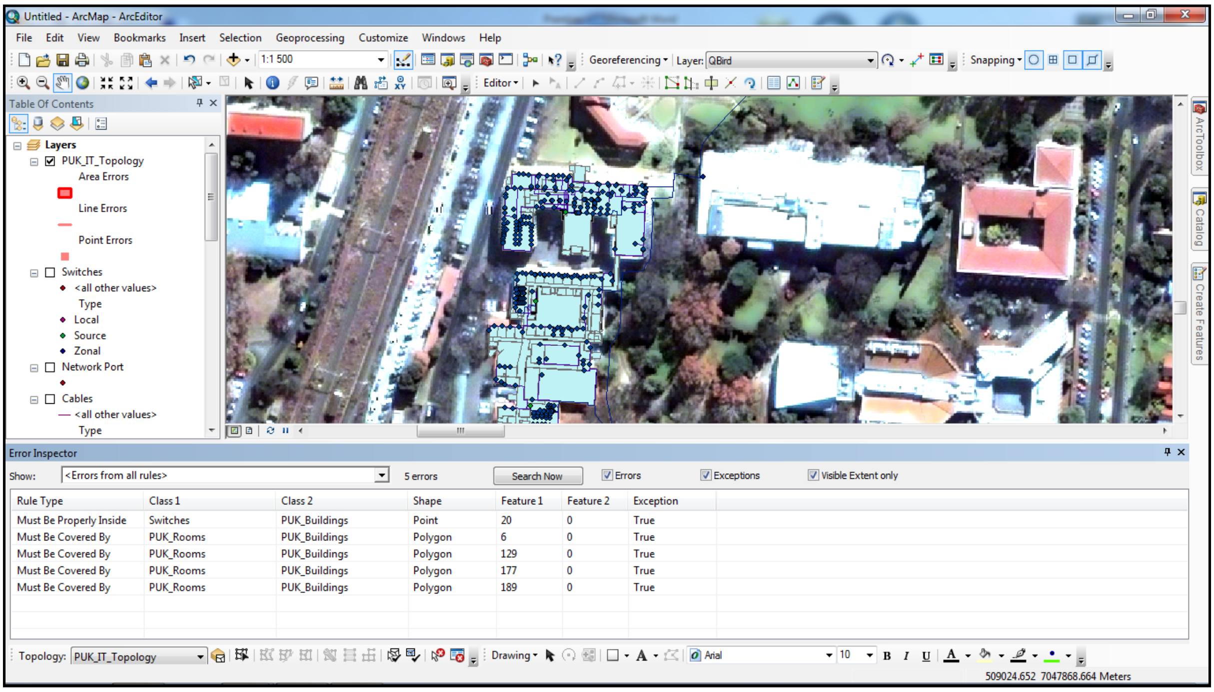

Figure 4.37: Marking topology errors as exceptions

In order to represent the connectivity between features in the IT-network on campus sufficiently, a network dataset is utilized. A network dataset is created and stored in a feature dataset, and can only consist of line and point feature classes from within that feature dataset.

A network dataset is created by right-clicking on the appropriate feature dataset and selecting the "New" > "Network dataset" option. This action allows the "New Network Dataset"-wizard to appear. The wizard assists the geodatabase developer to define a name for the network dataset as well as determine the parameters such as selecting the participating feature classes (Figure 4.38), as well as modeling turns in the network and how to model elevation.

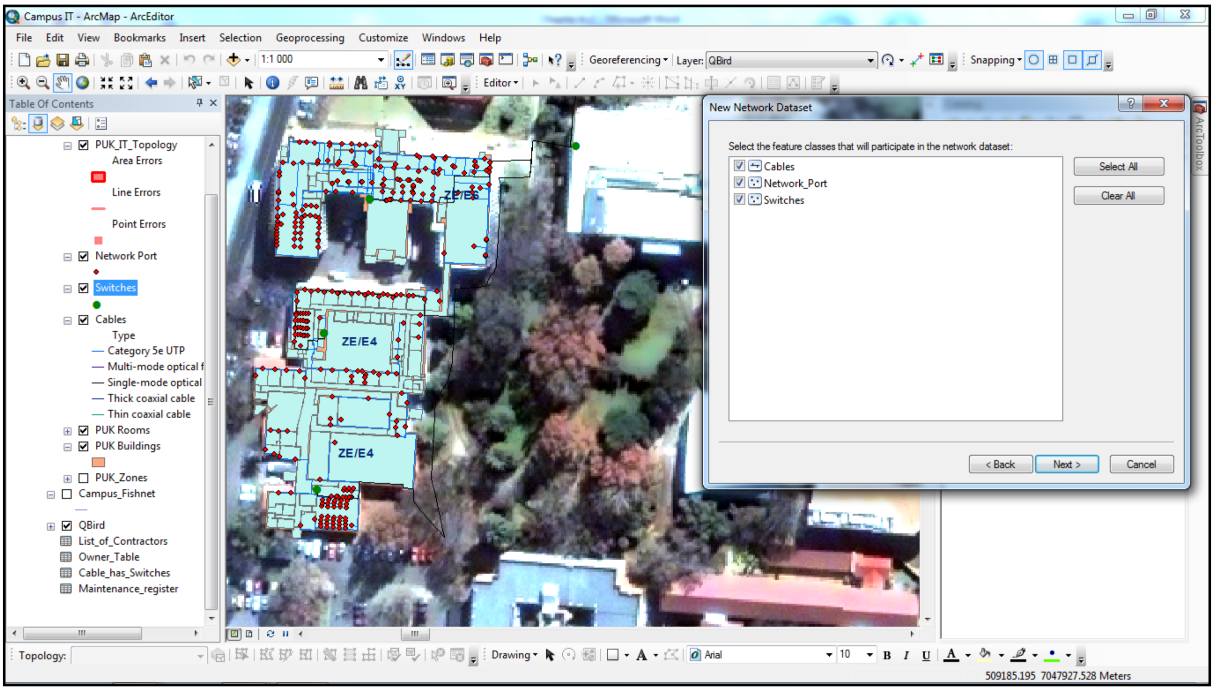

Figure 4.38: Creating a network dataset

Due to the fact that this model is not a transportation model, some of the parameters were not applicable. These parameters included modeling turns, defining a cost attribute for the network and establishing driving directions. It was decided to assign these parameters the default value, provided by the wizard.

One of the parameters of the network dataset that were applicable was the selection of the participating feature classes. The network dataset for this model consisted out of only the three IT-network elements (Network_Port, Switches and Cables) as seen in Figure 4.38. It was decided to use the z-values that were provided to the utility elements during development to model elevation in the network dataset. After all the parameters have been defined or assigned the default value, the “Finish” command on the wizard creates a new network dataset.

The network dataset has to be “Build” after it has been created. This action verifies the spatial connectivity between the feature elements. If any errors occur it reports back to the developer with a list of errors. Errors may include stand-alone network ports or switches, or cables which are not connected to an end-point. The list of errors provides the Object_ID of the relevant feature. The network features can be edited spatially and subjected to the “Build” process again until no more errors are present.

4.1.7 Creating a Fishnet and the 3D Shortest Route Model

For the purpose of this study there is no certain way to determine the location of outdoor utilities, especially underground infrastructure. A Fishnet provides an approximated estimation of the location of outdoor utility infrastructure.

A fishnet is created by utilizing the Create Fishnet tool stored in the Data Management toolbox in ArcToolbox. The Create Fishnet tool helps the developer to define a location for the Fishnet to be created. It was decided to store the fishnet as a stand-alone feature class in the PUK Geodatabase, named Campus_Fishnet. The tool also requires the developer to define the extent of the fishnet. The fishnet was given the same extent as PUK_Zones which covers the whole campus.

It was decided beforehand to create a fishnet with a cell size of 1m x 1m. Therefore the cell width as well as the cell height was equal to 1. The final parameters that needed to be determined in the fishnet tool were the number of rows and columns (Figure 4.39).

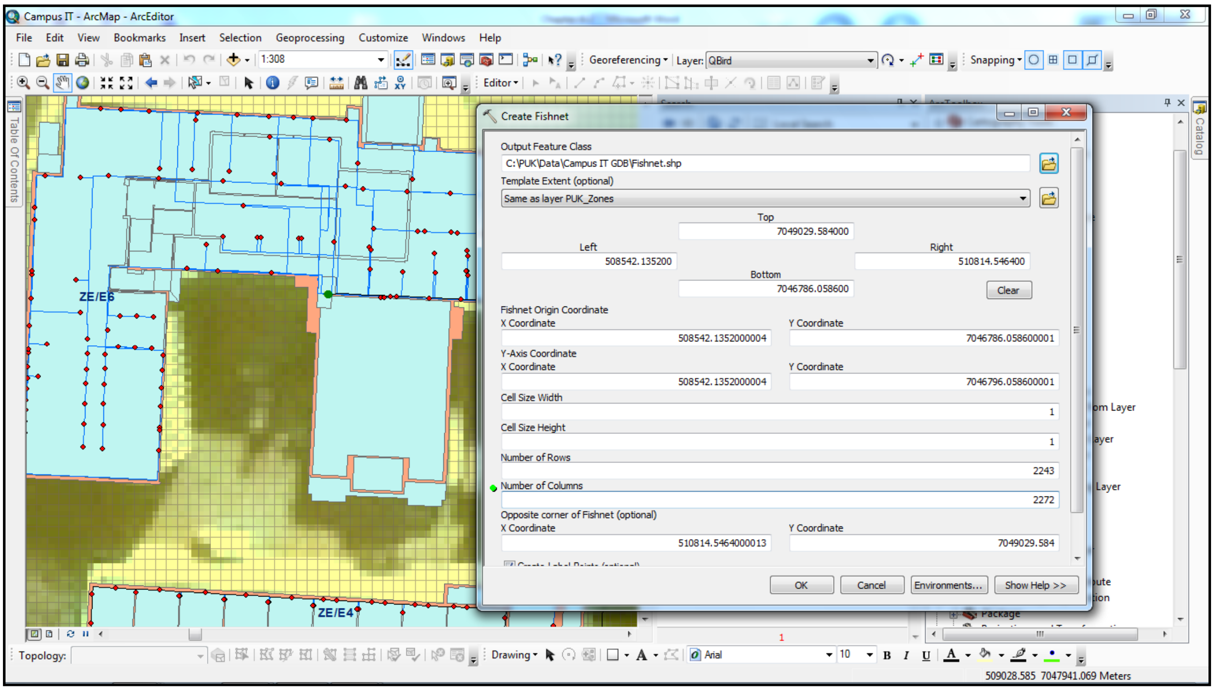

Figure 4.39: Create fishnet tool parameters

The number of rows parameter is determined by subtracting the Bottom extent from the Top extent and dividing the answer by the cell height. The final answer is rounded to the nearest meter. In this case the total number of rows is 2243.

In the same way the number of columns is determined by subtracting the Left extent from the Right extent and dividing the answer by the cell width. The final answer is also rounded to the nearest meter. The fishnet created for the data model had a total of 2272 columns.

By creating a fishnet with a cell size of 1m x 1m, it becomes possible for the end-user to determine the possible location of underground infrastructure on parts of the campus between buildings. For example in Figure 4.40, a user can estimate that if a cable is located 3 cells from a building it is approximately 3 meters from the building. Although a fishnet cannot be used to predict the location of infrastructure with 100% accuracy, it can narrow down the number of potential possible locations.

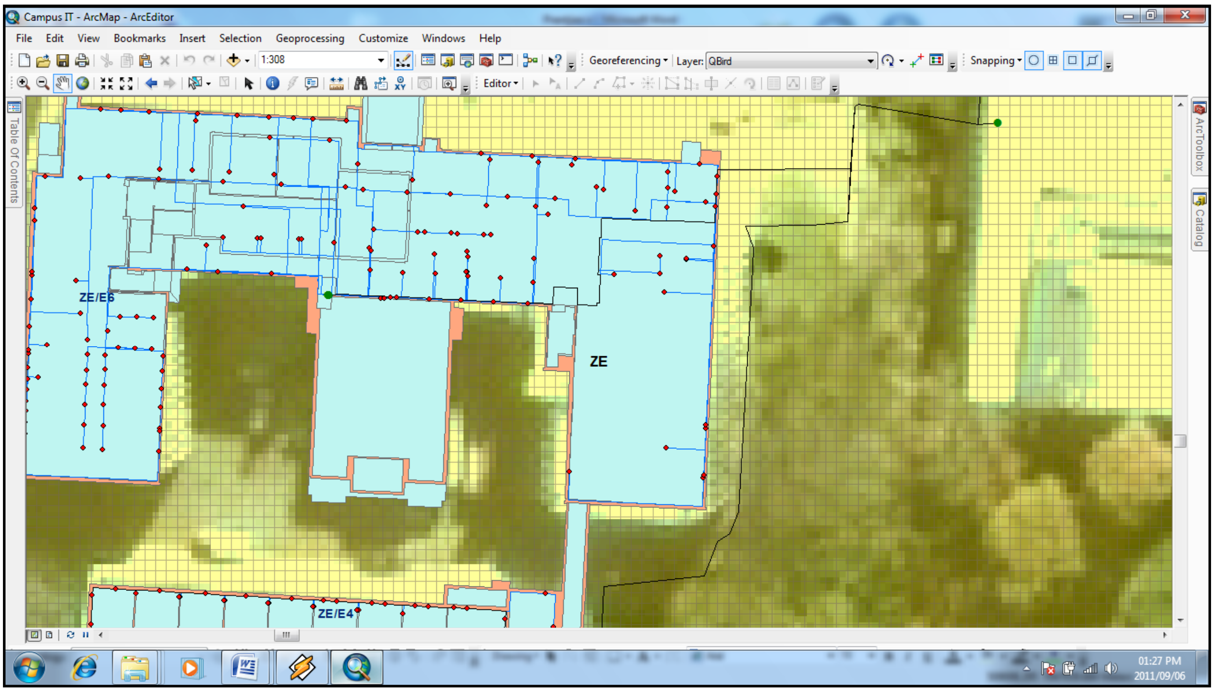

Figure 4.40: Campus_Fishnet

The final element that was added to PUK Geodatabase was the inclusion of a Toolbox containing a 3D Shortest Route model. The model was implemented in ArcScene to determine the best route between two or more points in the IT-network.

The model was created in *ModelBuilder* and is similar than the model described by ESRI (2011). ModelBuilder provides the geodatabase developer with a blank canvas, where different tools can be used in order to create a geoprocessing model.

The 3D Shortest Route Model consists of four types of tools. The four tools are: "Make route layer", "Add locations", "Solve" and "Apply Symbology from layer". The following section which describes the model will be divided according to these four geoprocessing tools.

The first tool that is added to the canvas is the "Make route layer" tool. The tool takes the PUK_IT_ND network dataset as input and creates a route layer from the network as output, named Route.

The next tool in the model is the "Add locations" tool. This tool allows the user to manually select the points that will serve as the start- and end-points of the route when the model is run. The first step after the tool was added to the canvas was to right-click on the tool and select the "Make variable" > "From parameter" > "Input locations" option. This action presented an "Input location" parameter as input for the "Add locations" tool. The input locations parameter is set as the Switches feature class and is the first stop in the route analysis. A stop is one of the points which serve as a mandatory stoppage-point along a route. Opening the properties of the Switches parameter is essential when implementing this tool. In the Switches' properties window the data type has to be set to "Feature set". This means that when the model is run, it will provide the user with the opportunity to manually choose any switch as a stop along the route. The "Add locations" tool also takes the result of the "Make route layer" tool as input. The tool therefore adds Switches stops manually to the route created by the previous tool, and delivers an altered route to the next tool as output, named Route (2).

Similarly another “Add locations” tool is added, with the only difference that the “Input locations” parameter is set as Network_Port. This allows the network ports to take part in the network analysis in the same way that Switches does. The second “Add locations” tool delivers another altered route named Route (3) as output.

The penultimate tool that was added to the canvas was the Solve tool. This tool analyses the network and determines the shortest route between the stops that were provided. It takes the last route (Route (3)) as input and presents two outputs. The first output is a final route, named Route (4), which represents the answer to the network analysis problem presented by the stops.

The final tool in the 3D Shortest Route model is the “Apply Symbology from Layer” tool. This tool allocates a set of predefined symbols to the final route. The set of symbols is saved as a layer file in the miscellaneous folder named RouteSymbology. The “Apply Symbology from Layer” tool takes Route (4) as input and applies the RouteSymbology layer as symbology. The final output, Route (5) is an altered route with symbology which answers the route analysis problem. The final step is to mark the PUK_IT_ND, Switches, Network_Port parameters as well as Route (5) as the model parameters. A flow-chart of the model processes can be seen in Appendix A.

4.2 Designing workflows for maintaining each layer

Step 9 in the geodatabase design method requires that the database developer create workflows which specify how each layer should be updated. In compliance with step 9, it was decided to create a standard operating procedure (SOP) which list rules that have to be followed when the geodatabase is updated. Step 9 also provides metadata for each layer.

4.2.1 Standard operating procedure

The following section lists rules for each layer that has to be enforced when the geodatabase is altered in order to ensure spatial-, data- and relational integrity.

PUK_Zones

- When a new zonal polygon is created, its attribute table has to be populated correctly. This is done during an editing session, by opening the feature class attribute table from the table of contents.
- PUK_Buildings is related to PUK_Zones. When a zonal polygon is deleted, it has to be verified that no PUK_Buildings records relate to the erased zone in the Zone_ID field (foreign key), otherwise resulting in an invalid relation.

PUK_Buildings

- When a new building polygon is created, its attribute table has to be populated correctly. This is done during an editing session, by opening the feature class attribute table from the table of contents.
- It has to be assured that a new building polygon has the correct value for its Zone_ID field (foreign key).
- PUK_Rooms is related to PUK_Buildings. If a building polygon is deleted it has to be verified that no PUK_Rooms records relate to the erased building through the Building_ID field (foreign key), otherwise resulting in an invalid relation.

PUK_Rooms

- When a new room polygon is created, its attribute table has to be populated correctly, including the subtype field Floors. This is done during an editing session, by opening the feature class attribute table from the table of contents.
- A new room element has to be assigned the correct z-values in terms of the floor it is located on, before the sketch is finished.
- It has to be assured that a new room polygon has the correct value for its Building_ID field (foreign key).
- It has to be assured that a new room polygon has the correct value for its Owner_ID field (foreign key).
- Switches is related to PUK_Rooms. When a room polygon is deleted it has to be verified that no Switches records relate to the erased room through the Room_ID field (foreign key), otherwise resulting in an invalid relation.

- It also has to be verified that no Network_Port records relate to a room polygon via the Room_ID field (foreign key), after it has been deleted, otherwise resulting in an invalid relation.

Switches

- When a new switch point is created, its attribute table has to be populated correctly. This is done during an editing session, by opening the feature class attribute table from the table of contents.
- A new switch feature has to be assigned the correct z-values in terms of the floor it is located on, before the sketch is finished.
- Switch elements with z-values that are not equal to 0 is created by utilizing the “Point at end of line” tool.
- It has to be assured that a new switch point has the correct value in its Room_ID field (foreign key).
- It also has to be assured that a new switch point has the correct value in its Job_ID field (foreign key).
- When a new switch element is created it has to be connected to a cable. Therefore it has to be assured that the intermediate table of the Cables_has_Switches relationship class is correctly updated, and that the appropriate links are depicted.
- When a switch element is deleted it cannot be connected to a cable. Therefore it has to be assured that the intermediate table of the Cables_has_Switches relationship class is correctly updated, and that the appropriate links are removed.

Network_Port

- When a new network port point is created, its attribute table has to be populated correctly. This is done during an editing session, by opening the feature class attribute table from the table of contents.
- A new network port feature has to be assigned the correct z-values in terms of the floor it is located on, before the sketch is finished.
- Network_Port elements with z-values that are not equal to 0 are created by utilizing the “Point at end of line” tool.
- It has to be assured that a new network port point has the correct value in its Room_ID field (foreign key).
- It also has to be assured that a new network port point has the correct value in its Job_ID field (foreign key).

- When a Network_Port point is deleted it has to be verified that no Cables records relate to the erased room through the NP_ID field (foreign key).

Cables

- When a new cable line is created, its attribute table has to be populated correctly. This is done during an editing session, by opening the feature class attribute table from the table of contents.
- In order to prevent connection errors in the network, a cable's end-points have to be covered by either a switch or a network port.
- A cable may exist on several levels with several z-values, moving horizontally or vertically, as long as the cable remains a single extended cable feature.
- It has to be assured that a newly created cable element has the correct value in its NP_ID field (foreign key). If a cable is not connected to a network port the NP_ID field is allocated a 0.
- It also has to be assured that a new cable has the correct value in its Job_ID field (foreign key).
- When a new cable element is created it has to be connected to a switch on at least one of its end-points. Therefore it has to be assured that the intermediate table of the Cables_has_Switches relationship class is correctly updated, and that the appropriate links are depicted.
- When a cable element is deleted it is no longer connected to a switch. Therefore it has to be assured that the intermediate table of the Cables_has_Switches relationship class is correctly updated, and that the appropriate links are removed.

Owner_Table

- When a record is added to the owners table the attributes has to be populated correctly. This is done during an editing session, by opening the table from the table of contents.
- PUK_Rooms is related to Owner_Table. When a record is deleted from the owners table, it has to be verified that no rooms longer relate to the erased owner through the Owner_ID (foreign key), otherwise resulting in an invalid relation.

Maintenance_register

- When a record is added to the maintenance register the correct values has to be awarded to the fields. This is done during an editing session, by opening the table from the table of contents.

- It has to be assured that a new maintenance register record is awarded the correct value for its Contractor_ID field (foreign key).
- The Maintenance register is related to Switches, Cables and Network_Port. When a maintenance register record is deleted it has to be verified that no Switches features, Cable features or Network_Port features relate to the erased record through their respective Job_ID fields (foreign keys), otherwise resulting in an invalid relation.

List_of_Contractors

- When a record is added to the contractors list the correct value has to be awarded to its various fields. This is done during an editing session, by opening the table from the table of contents.
- When a List_of_Contractors record is deleted it has to be verified that no Maintenance_register records relate to the erased record via its Contractor_ID field (foreign key).

Topology

- When one or more feature classes are altered, the whole topology has to be validated again.
- An existing topology can be validated during an editing session.
- During an editing section, open the "Error inspector" on the Topology toolbar.
- Zoom to the PUK_Zones feature class' extent. This ensures that all the participating feature classes are all within the extent.
- On the Topology toolbar, select the "Validate Topology In Current Extent" option. This validates all the topology rules for that extent.
- On the error inspector search for errors. Any new infringements of topology rules will be depicted in the list.
- To correct an error, right-click on the error in the list and either mark it as an exception or select one of the available fixes.
- Validate the topology again until no errors occur.

Network alterations

- When features has been added or deleted from the utility feature classes which participate in the network dataset (PUK_IT_ND), the network dataset has to be rebuilt.
- After the feature class has been altered, open the network dataset properties. In the properties window, select the "Source" tab. The source tab lists all the participating

feature classes. Select the ones that have been altered and remove them from the network dataset (Figure 4.41).

- After the feature classes have been removed, simply add them again and click on OK.
- Right-click on the network dataset and select rebuild.
- It is not necessary to edit the 3D Shortest Route model, because if the network dataset is rebuilt, the model automatically uses the modified network dataset.

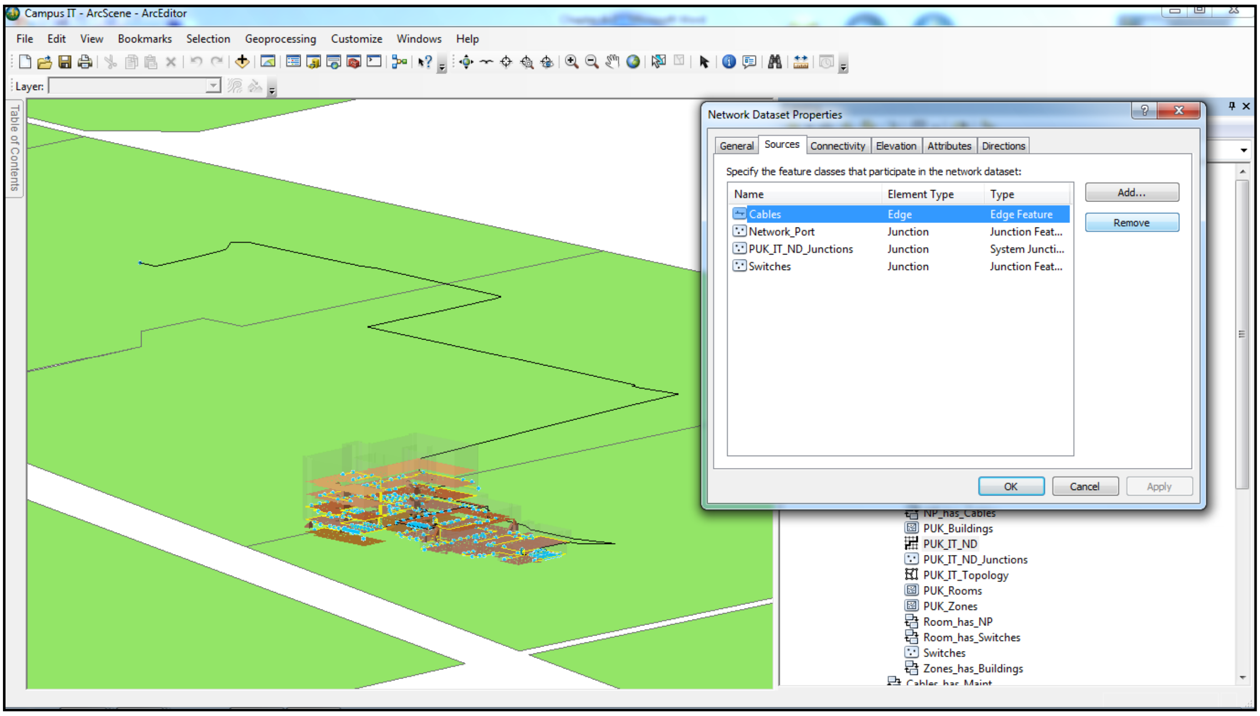

Figure 4.41: Rebuilding the network dataset

Opening the data model

The DVD accompanying this dissertation contains the data model. ArcGIS 10 (with the network analyst extension) is required to open, view or run analysis on the data model. The following steps will allow the user to open the data model:

- Open the DVD and copy the folders to a permanent location on the PC hard drive (approximately 109MB).
- Open ArcCatalog and create a *Folder connection* to the location where the folders have been saved.
- Navigate to the *Miscellaneous* folder at the saved location and open both the Campus IT.mxd and Campus IT.sxd files.

- Right-click on any layer in the table of contents and choose the *Data* > *Repair data source* options. Then navigate to the location where the folders are saved and choose the appropriate feature class/raster.
- The map should appear correctly for all the layers.
- Right-click on the *3D Shortest Route Model*, located in the Toolbox of the PUK Geodatabase, and choose the *Edit* option. This opens the ModelBuilder canvas.
- Open the *Make Route Layer* tool and make sure that the *Input Analysis Network* is the PUK_IT_ND network dataset (located in the PUK Geodatabase).
- Open the *Apply Symbology from Layer* tool and make set the *Symbology layer* to *RouteSymbology* (located in the *Miscellaneous* folder).
- Save and close the ModelBuilder canvas. The data model is now ready to be used.
- These instructions can also be found in the ReadMe PDF in the *Miscellaneous* folder.

4.2.2 Metadata

Step 9 in the geodatabase design method also provides the geodatabase with information about the data called metadata. The metadata of the pilot model can be viewed in Appendix B: Metadata for the PUK Geodatabase. The metadata was constructed according to some of the standards presented by Biodiversity GIS (2004). The metadata describes:

- A short description of the data,
- The intended purpose of the data
- The source where it was obtained from as well as the source scale
- Data attributes
- The spatial reference of the data
- The spatial representation of the data
- The date the data was created as well as he person responsible for developing the data

4.3 Documenting the design

The final step in the geodatabase design method is to create documentation which explains how the model design was shaped and implemented. For most part this dissertation serves as the requirements of step 10, which explains the concept, design, implementation, review and maintenance of the data model. However more detailed summarized diagrams depicting the model design and the domains can be seen in Appendixes C: Geodatabase schema diagram and Appendix D: PUK Geodatabase domains.

Chapter 5 will study the implementation methods of the data model. It will give a detailed explanation of how the data model can be applied by the end-user, the types of analysis that are possible as well as discuss some of the limitations.