Plantation forestry: an analysis of the domain

C. Sue Price[1], Jules-Raymond Tapamo[2], Felicity Blakeway[3], and Fethi Ahmed[4]

[1] School of Information Systems & Technology, University of KwaZulu-Natal, 4041 Durban, South Africa
[2] School of Computer Science, University of KwaZulu-Natal, 4041 Durban, South Africa
[3] Forestry & Forest Products Research Centre, CSIR/UKZN, P. O. Box 17001, 4013 Congella, South Africa
[4] School of Environmental Science, University of KwaZulu-Natal, 4041 Durban, South Africa

Abstract. Plantation forests the world over have been established in order to supply industrial mills with wood. It is estimated that by 2050, over half of the world's requirements for industrial timber could be supplied from plantations, thus reducing the pressure on natural forests. Ensuring that the plantation, with its many stands, different planted species, terrains, etc., delivers the correct volume of timber over time is a complex problem. Although many forest harvest models and systems have been described in the literature, not many have been described from a computer science or information systems perspective. In this paper, the plantation forestry domain (based on South African experience) is described using formal models (Z notation), augmented by semi-formal (conceptual) models. Since plantations are generally planted to provide wood to mills, the forest-to-mill supply chain is described. This paper contributes toward an understanding of the plantation forestry domain.

Key words: Plantation forestry, forest-to-mill supply chain, semi-formal analysis, Zachman framework, formal analysis, Z notation

1 Introduction

It is estimated that by 2050, more than half the world's requirements for industrial timber could be supplied from plantations [14]. While natural forests have been used for centuries for their timber, in the last few decades, with increasing pressure to conserve forests and forest species habitats, timber is increasingly being sourced from managed or plantation forests [7]. Intensively managed plantations are able to produce increasing volumes of timber due to tree breeding and management, thus enabling the natural forests to be retained for other purposes such as maintaining biodiversity and recreation [7].

Foresters need to ensure that enough wood is available for mills to use in the long term, and that the appropriate volume of wood is delivered to mills in the short term [15]. This is because the industries supplied by the forests are often very capital-intensive. They therefore need assurance of a constant supply of wood to guarantee a return on the capital invested [8]. In plantations, this wood comes from stands which contain trees which typically have the same species and same age. The stand is the smallest homogeneous area of trees. The same activities will be applied to all the trees in the stand at the same time, thereby creating a crop which is silviculturally uniform [8].

Managing a large number of forest stands over vast tracts of land, and deciding when to harvest each stand, is a complex task. The simplest case would be if a single species of tree were grown, and each stand had a similar soil type and climate. If the trees were harvestable after n years, the harvesting decision would then be to cut $1/n$th of the plantation area each year to keep the mill(s) supplied [15]. Unfortunately, the situation is more complex than this. Often, the afforestable land covers a range of altitudes, soil types, and is subject to different weather patterns [8]. Trees are chosen to suit the sites in which they are grown [8] (for example, over 20 different species are used commercially in South Africa, excluding hybrids and clones [10]) and this variety means that the trees grow at different rates and become mature (harvestable) at different ages [21].

Deciding which stand to fell, when, over a different time frames and different forest areas is a complex, multifaceted problem [16]. Computer systems have been developed to support this decision making [16], but because of the large amount of data involved, these systems tend to concentrate on a particular time frame (long-term (strategic), medium-term (tactical), short-term (operational)), and tend to deliver plans which do not incorporate another time frame's plan's constraints [5, 17]. This area of decision support for forest harvest scheduling and plantation forestry management therefore needs further work.

Emphasis on understanding the domain (or, the problem area that is to be addressed) has received attention because costly and detrimental results have ensued from neglecting it during software development [11]. In order to develop a software system, it is necessary to determine what users need (i.e. draw up a user requirements statement). This also means that the domain in which the system will be active needs to be understood [4, 11] and recorded in the user requirements and specifications [6, 22].

Domain engineering aims at describing an environment as it is at present [4], possibly with no reference to requirements of future systems [3]. The descriptions could be formal (using mathematical notation) [3, 19], semi-formal (e.g. Entity-Relationship diagrams, Structured Analysis and OOA [19]) or informal (using narrative text) [3], and aim to capture all the aspects of the domain [3] in an accurate set of descriptions which domain experts can agree upon [4]. Although the semi-formal methods are widely-used in industry, they cannot be checked formally (mathematically) for inconsistencies. Formal models are represented by a formal specification language (like Z or VDM [1]), and can be used to record the states or activities in a domain with more preciseness [1]. Performing the formal analysis also helps highlight inconsistencies or oversights in the semi-formal analyses [22].

Modelling the domain is beneficial because it captures understanding and knowledge about a particular domain [4]. Customers of future systems can check the description to ensure that their 'reality' is adequately captured by the models [13]. It will also enable specifiers, and later developers, to create software which more accurately reflects the environment because the descriptions are stated explicitly [22].

While many authors have described mathematical models which can be used to aid forest harvesting decision making and forest management decisions in general, very few have described the forestry domain from a computer science or information systems point of view. Baskent *et al.* [2] uses object-oriented techniques to give a conceptual framework for the design of forestry management problems. This work covers natural forestry management, but could be applied to plantation forestry. Two papers give semi-formal descriptions of the plantation forestry industry: Nobre and Rodriguez [18] describe the data modelling aspect of large forest harvest scheduling problems, and Ribeiro *et al.* [20] use the Zachman framework to design the enterprise architecture for an integrated forest planning system. The complex forestry domain is therefore not well described, and when describing it, authors have used semi-formal methods. This makes it a possible area for future work.

This paper contributes to knowledge of the plantation forestry domain in that it uses the formal notation Z, supported by semi-formal models, to describe the plantation forestry domain. This domain includes the transportation of timber to the mill and the activities at the mill, but most attention is given to the plantation forestry aspect. Because it is a simpler case, the forestry supply-chain for pulp and paper manufacture is described. The analysis represents ongoing work currently being undertaken in South Africa as a first step to specifying a forest harvest scheduling system.

2 Methods and Techniques

The understanding of the forest-to-mill domain was gained during a user requirements gathering exercise aimed at developing a forest harvest scheduling system. For this system, various role players in an integrated plantation forestry company in South Africa were interviewed; these included the planning forester, regional foresters, estate foresters, the systems analyst, the database administrator, the IT specialist and the logistics manager. During the requirements gathering phase, every attempt was made to understand plantation forestry as it is undertaken world-wide, rather than concentrating on the particular company's implementation. This was done by comparing the outcomes of the interviews with books (e.g. [8], [15]) and other plantation forestry literature.

An abstract Entity-Relationship diagram for the entire forest-to-mill domain was drawn. From this, the main actors of each sub-domain were identified. For each sub-domain, a list of actions which take place in that domain, and the constraints experienced by the domain, was made. Thereafter, semi-formal and formal models were developed (see Fig. 1).

The Zachman framework [23] was used to structure the semi-formal models. This proposes that different models be created to describe the system from

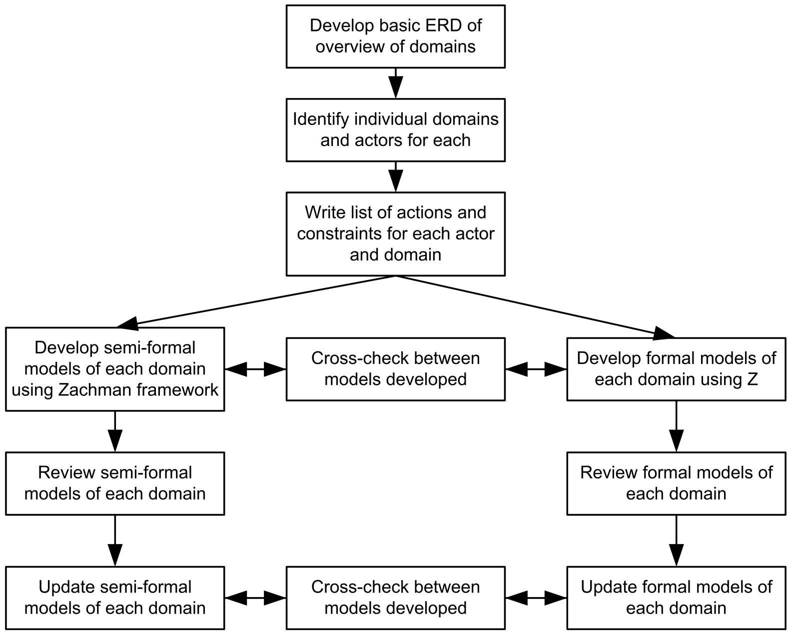

Fig. 1. Method followed in analysing the domain

different points of view; in addition, different models are used for the various stages of the system's development. The Business owner's view of the system was used in this analysis (see Table 1). The models thus developed can be cross-checked to improve consistency and completeness.

Table 1. Business owner's row in the Zachman framework (from [9, p.3])

	Data (what?)	**Activities (how?)**	**Locations (where?)**	**People (who?)**	**Time (when?)**	**Motivation (why?)**
Enterprise model (Business owner's view)	Language, divergent data model	Business process model	Logistics network model	Organization chart	State/ transition diagram	Business strategies, tactics, policies, rules

Semi-formal methods used in the analysis of the domain were Entity-Relationship diagrams, Business Process diagrams (also known as Swimlane diagrams) and State Charts. Because of space, Business Process diagrams are not included.

The semi-formal models were verified by the main interviewee (the planning forester). Thereafter, specifications were developed using Z. The narrative text describing these were checked by a forestry expert (not familiar with the Z notation) and the specifications checked by two Z experts, one of whom is familiar with forestry.

Z is a formal specification notation which uses mathematical features such as sets and predicate logic [1]. Z presents the model in small 'chunks' called schemas, which describe the initial state of the model and the behaviour which would change the initial state to some other state [22]. The Z notation is useful to show pre- and post-conditions of the states present in the domain. The constraints placed on an entity or action can often be shown in the schemas' pre- and post-conditions. The Z descriptions were typechecked using Z Type Checker (ZTC) [12].

3 Analysis of the Domain

3.1 Forest-to-mill Domain Overview

The forest-to-mill domain can be summarised by Fig. 2. The plantation forest produces logs, which are transported to the mill, which makes pulp from them. In this analysis, the pulping or papermaking processes are not included. Fig. 3 shows the main entities involved in the forest-to-mill domain (viz. the plantation forests, the transport and the pulp mill) together with their actors. These are the foresters (growers) (including planning, silvicultural and harvesting foresters), transporters (anyone involved with moving timber from one place to another, or the planning thereof), and millers/processors (anyone involved with receiving timber and processing it). Tables 2, 3 and 4 show the actions and constraints relevant to each of the main areas.

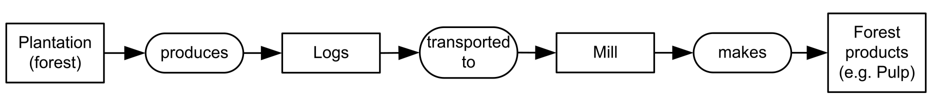

Fig. 2. Entity-Relationship diagram of the forest-to-mill domain

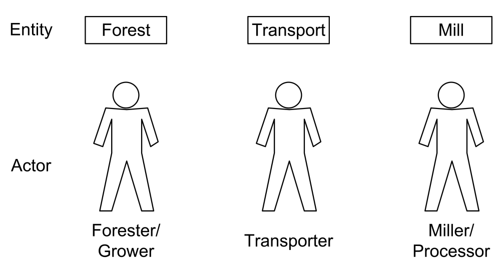

Fig. 3. Main entities and actors of the domain

Table 2 shows the actions and constraints which apply to the forest and forester (grower). Many types of foresters would be involved in the actions (e.g. silvicultural, harvesting and planning foresters). The constraints are subdivided into three main sections: planting, harvesting and planning.

Table 2. Actions and constraints for the forester (grower)

Actions	• Plant • Maintain stands/estates • Perform other silvicultural activities • Ensure that roads are upgraded prior to harvesting stands • Harvest trees in stand & extract to roadside • (Make logs from tree-lengths) • Develop long-term, medium-term and short-term plans • Find markets for excess trees (if more need harvesting than can be processed)
Constraints	• Afforestable/afforested area is widely spread • Many different species of trees could be planted • Trees' growth rates differ, depending on where planted and species
	Planting: • Should plant as soon after harvesting as possible • May have to wait for rainy season to commence planting • Not all species can be planted everywhere (e.g. some are frost sensitive) • Should only plant a species suitable for the stand • Should plant tree species which are acceptable to mills
	Harvesting: • Forester is paid for tonnage of timber, so don't want to cut trees too young or old • Aim to harvest many stands in one area and then move to another to save costs of moving harvesting staff & equipment, and to upgrade fewer roads prior to harvesting
	Planning: • There are many stands, each with trees of differing levels of maturity • Want enough timber now, and each year into the future • Make sure all plans (short-, medium- and long-term) agree with each other

Fig. 4 shows a state chart of the domain. Logs/tree-lengths are produced from the plantation on harvesting (felling) and are stacked at the stand's roadside; they are transported to the mill (via a depot) where they are processed to make pulp. The formal models of the transport and mill domains are not included, for space reasons.

Table 3. Actions and constraints for the transporter

Actions	• Load timber (logs/tree-lengths) at stand's roadside • Transport timber to depot (short haul) • Unload timber at depot • (Make logs from tree-lengths) • Load timber at depot • Transport timber to mill (long haul) • Unload timber at mill
Constraints	• Want equipment available when ready to load/unload • Paid for no. of tonnes hauled over the haulage distance • May decide to drive around the clock to maximise vehicle R.O.I. • Number of hours a driver may work per day or per week is limited

Table 4. Actions and constraints for the miller (processor)

Actions	• Accept timber from own forests and/or other suppliers • (Make logs from tree-lengths, if not already done) • (Debark logs) • Remove timber from logyard & feed into pulping process • Find source of additional timber (if not enough produced by own plantation)
Constraints	• Need constant timber supply so process can work 24/7 • Need buffer timber stock in logyard in case transport fails or stands can't be accessed • Some species of timber are not acceptable • Want certain timber species, or timber species mix • Want timber's collection point to be near mill, because transport costs make up a large proportion of delivered timber costs

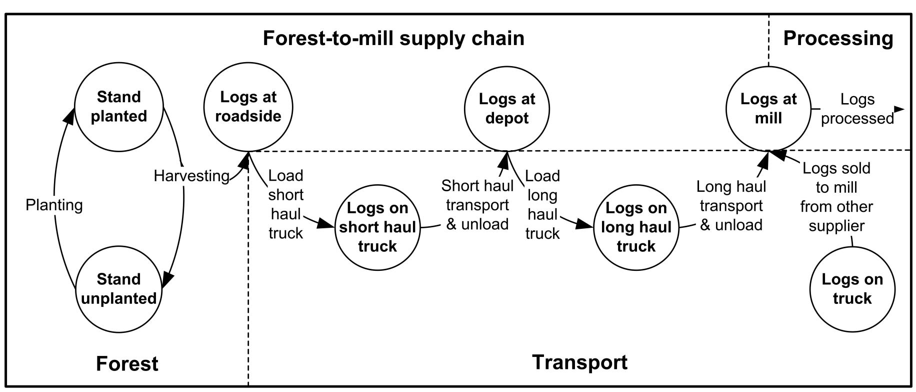

Fig. 4. State chart of the forest-to-mill domain's scope

3.2 Plantation Forest Domain

The plantation forest is divided into smaller management units, the smallest of which is the stand. A group of stands forms an Estate. Stands are usually bounded by roads [10]. Although roads are sometimes only developed just prior to harvesting, basic quality roads are often built before planting [8]. There is a depot located in or near each estate, to which harvested timber is transported. From there, it is transported to the mill.

The Z specification begins with the definitions needed. Each stand, mill and species is uniquely identified. The stand's planting state is either unplanted or planted. The type AGE describes the age of the trees planted in a stand; $MASS$ describes the mass of timber harvested from the stand. AGE and $MASS$ are defined here as a finite set of natural numbers (although they are actually real numbers; they are so defined because of a restriction in Z).

$$[STANDID, MILLID, SPECIESID]$$

$$PLANTINGSTATE ::= unplanted \mid planted$$

$$AGE, MASS : \mathbb{F}\,\mathbb{N}$$

Two schemas follow: *StandSuitableSpecies* contains a function (*suitableSpecies*) which gives a list of suitable species which could be grown on each stand (determined by site-species matching). Each stand must have at least one suitable species. Since this specification describes an integrated forestry company, tree species would not be planted which were not acceptable to a mill. *MillAcceptableSpecies* contains the list of species (*acceptableSpecies*) acceptable to each mill. Each mill must have at least one acceptable species.

StandSuitableSpecies

$$suitableSpecies : STANDID \rightarrow SPECIESID$$

$$\forall\, sID : STANDID \bullet \\ \quad sID \in \mathrm{dom}\, suitableSpecies \land \\ \quad \#\{(suitableSpecies\ sID)\} \geq 1$$

MillAcceptableSpecies

$$acceptableSpecies : MILLID \rightarrow SPECIESID$$

$$\forall\, mID : MILLID \bullet \\ \quad mID \in \mathrm{dom}\, acceptableSpecies \land \\ \quad \#\{(acceptableSpecies\ mID)\} \geq 1$$

Schema *StandOfTrees* governs the relationships the stand's land and the trees that are planted on it. The schemas *StandSuitableSpecies* and *MillAcceptableSpecies* are included in this schema, but cannot be changed by it. Three functions are included: *plantingStatus* records whether the stand is unplanted or planted; *plantedSpecies* records the species planted; and *treeAge* monitors the tree's age. If the stand is planted, the planted trees' species will be one of the stand's suitable species as well as one of the mills' acceptable species, and the trees' age will always be zero or above. If the stand is unplanted, the tree age and planted species will be undefined.

$$
\begin{array}{l}
\textbf{StandOfTrees} \\
\hline
\Xi \mathit{StandSuitableSpecies} \\
\Xi \mathit{MillAcceptableSpecies} \\
\mathit{plantingStatus} : \mathit{STANDID} \rightarrow\!\!\!\!\!+ \ \mathit{PLANTINGSTATE} \\
\mathit{plantedSpecies} : \mathit{STANDID} \rightarrow\!\!\!\!\!+ \ \mathit{SPECIESID} \\
\mathit{treeAge} : \mathit{STANDID} \rightarrow\!\!\!\!\!+ \ \mathit{AGE} \\
\hline
\forall \mathit{sID} : \mathit{STANDID} \bullet \exists \mathit{mID} : \mathit{MILLID} \bullet \\
\quad \mathit{sID} \in \operatorname{dom} \mathit{suitableSpecies} \land \\
\quad \mathit{mID} \in \operatorname{dom} \mathit{acceptableSpecies} \land \\
\quad \mathit{sID} \in \operatorname{dom} \mathit{plantingStatus} \land \\
\quad \mathit{sID} \in \operatorname{dom} \mathit{plantedSpecies} \land \\
\quad \mathit{sID} \in \operatorname{dom} \mathit{treeAge} \land \\
\quad ((\mathit{plantingStatus}\ \mathit{sID}) = \mathit{planted} \Rightarrow \\
\qquad ((\mathit{plantedSpecies}\ \mathit{sID}) \in \{(\mathit{suitableSpecies}\ \mathit{sID})\}) \land \\
\qquad ((\mathit{plantedSpecies}\ \mathit{sID}) \in \{(\mathit{acceptableSpecies}\ \mathit{mID})\})) \land \\
\quad ((\mathit{plantingStatus}\ \mathit{sID}) = \mathit{planted} \Rightarrow (\mathit{treeAge}\ \mathit{sID}) \geq 0) \land \\
\quad ((\mathit{plantingStatus}\ \mathit{sID}) = \mathit{unplanted} \Rightarrow \{(\mathit{treeAge}\ \mathit{sID})\} = \varnothing) \land \\
\quad ((\mathit{plantingStatus}\ \mathit{sID}) = \mathit{unplanted} \Rightarrow \{(\mathit{plantedSpecies}\ \mathit{sID})\} = \varnothing) \\
\hline
\end{array}
$$

Prior to harvesting, the mill to which the stand's timber will be sent is determined and stored in function *millForStandsTimber* in schema *MillForStandsTimber*. Each stand has only one mill to which its timber will be sent.

$$
\begin{array}{l}
\textbf{MillForStandsTimber} \\
\hline
\mathit{millForStandsTimber} : \mathit{STANDID} \rightarrow\!\!\!\!\!+ \ \mathit{MILLID} \\
\hline
\forall \mathit{sID} : \mathit{STANDID} \bullet \exists_1 \mathit{mID} : \mathit{MILLID} \bullet \\
\quad \mathit{sID} \in \operatorname{dom} \mathit{millForStandsTimber} \land \\
\quad \mathit{mID} \in \operatorname{ran} \mathit{millForStandsTimber} \\
\hline
\end{array}
$$

Schema *MassOfFelledTrees* gives the mass of the stand's trees, when felled. The function *massOfFelledTrees*, which takes as inputs the stand's ID and that stand's harvesting age, gives the utilizable mass of the trees (i.e. the mass of the part of the trees which will eventually become logs). This mass is greater than or equal to zero.

$$
\begin{array}{l}
\textbf{MassOfFelledTrees} \\
\hline
massOfFelledTrees : (STANDID \times AGE) \pfun MASS \\
ageToFell? : AGE \\
\hline
\forall sID : STANDID \bullet \\
\quad (sID, ageToFell?) \in \operatorname{dom} massOfFelledTrees \land \\
\quad massOfFelledTrees\ (sID, ageToFell?) \geq 0 \\
\hline
\end{array}
$$

Once harvested, the stand's logs or tree-lengths are piled at roadside ready to be transported to the depot and then to the mill. Schema *TimberAtRoadside* contains information about the timber which is piled at roadside. It includes the unchangeable schema *MillForStandsTimber*, and a function, *timberAtRoadside*. This has as inputs the stand's ID and the mill's ID, and outputs the mass of timber. The mill's ID is that of the mill to which the stand's timber has been allocated. The mass of the timber at roadside is greater than or equal to zero.

$$
\begin{array}{l}
\textbf{TimberAtRoadside} \\
\hline
\Xi MillForStandsTimber \\
timberAtRoadside : (STANDID \times MILLID) \pfun MASS \\
\hline
\forall sID : STANDID \bullet \exists_1 mID : MILLID \bullet \\
\quad sID \in \operatorname{dom} millForStandsTimber \land \\
\quad mID \in \operatorname{ran} millForStandsTimber \land \\
\quad (sID, mID) \in \operatorname{dom} timberAtRoadside \land \\
\quad millForStandsTimber\ sID = mID \land \\
\quad timberAtRoadside\ (sID, millForStandsTimber\ sID) \geq 0 \\
\hline
\end{array}
$$

The action schema *PlantStand* describes the planting activities. This schema includes unchangeable schemas, *StandSuitableSpecies* and *MillAcceptableSpecies*, and changeable schema *StandOfTrees*. Inputs to this schema are *whichStand?* (the ID of the stand to be planted) and *speciesToPlant?* (the species to be planted). The species to be planted must be in the list of suitable species, and a mill must exist for which the species to be planted is in the list of acceptable species. The stand's status prior to planting must be unplanted. After planting, the planting status becomes planted, the trees' age changes to zero and the planted species is assigned the value of *speciesToPlant?*.

$$
\begin{array}{|l}
\textbf{PlantStand} \\
\hline
\Xi StandSuitableSpecies \\
\Xi MillAcceptableSpecies \\
\Delta StandOfTrees \\
whichStand? : STANDID \\
speciesToPlant? : SPECIESID \\
\hline
whichStand? \in \operatorname{dom} suitableSpecies \\
whichStand? \in \operatorname{dom} plantingStatus \\
whichStand? \in \operatorname{dom} plantedSpecies \\
whichStand? \in \operatorname{dom} treeAge \\
speciesToPlant? \in \operatorname{ran} suitableSpecies \\
speciesToPlant? \in \operatorname{ran} plantedSpecies \\
speciesToPlant? \in \{suitableSpecies\ whichStand?\} \\
\exists\, mID : MILLID \bullet \\
\quad mID \in \operatorname{dom} acceptableSpecies \land \\
\quad speciesToPlant? \in \{acceptableSpecies\ mID\} \\
(plantingStatus\ whichStand?) = unplanted \\
plantingStatus' = plantingStatus \oplus \{whichStand? \mapsto planted\} \\
treeAge' = treeAge \oplus \{whichStand? \mapsto 0\} \\
plantedSpecies' = plantedSpecies \oplus \{whichStand? \mapsto speciesToPlant?\} \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\textbf{HarvestStand} \\
\hline
\Delta StandOfTrees \\
\Delta TimberAtRoadside \\
\Xi MassOfFelledTrees \\
whichStand? : STANDID \\
fellingAge? : AGE \\
\hline
whichStand? \in \operatorname{dom} suitableSpecies \\
whichStand? \in \operatorname{dom} plantingStatus \\
whichStand? \in \operatorname{dom} plantedSpecies \\
whichStand? \in \operatorname{dom} treeAge \\
fellingAge? \in \operatorname{ran} treeAge \\
(whichStand?, fellingAge?) \in \operatorname{dom} massOfFelledTrees \\
plantingStatus\ whichStand? = planted \\
fellingAge? = (treeAge\ whichStand?) \\
timberAtRoadside' = timberAtRoadside \oplus \\
\quad \{(whichStand?, millForStandsTimber\ whichStand?) \mapsto \\
\qquad massOfFelledTrees\ (whichStand?, fellingAge?)\} \\
plantingStatus' = plantingStatus \oplus \{whichStand? \mapsto unplanted\} \\
\{(plantedSpecies'\ whichStand?)\} = \varnothing \\
\{(treeAge'\ whichStand?)\} = \varnothing \\
\hline
\end{array}
$$

The action schema for harvesting (felling) the stand (*HarvestStand*) includes changeable schemas *StandOfTrees* and *TimberAtRoadside*, and unchangeable schema *MassOfFelledTrees*. It has two inputs, *whichStand?* (the stand to fell) and *fellingAge?* (the age at which to fell the stand). Before harvesting, the planting status must be planted and the stand's age must be the same as *fellingAge?*. After felling, the logs/tree-lengths are piled at roadside: their mass takes the value of the function *massOfFelledTrees* (evaluated at *whichStand?* and *fellingAge?*). The stand's planting status becomes unplanted, and the planted species and tree age become undefined.

ForestSchema combines the two forestry action schemas *PlantStand* and *HarvestStand*.

```
ForestSchema
PlantStand
HarvestStand
```

3.3 Transport and Mill Domain Overview

Due to space limitations, the transport and mill domains cannot be described at length. Two Entity-Relationship diagrams capture the essence of these domains respectively (Fig. 5 and Fig. 6). The second figure shows how business rules can be shown using Entity-Relationship diagrams (i.e. "the mill consists of both a logyard and one or more processes".)

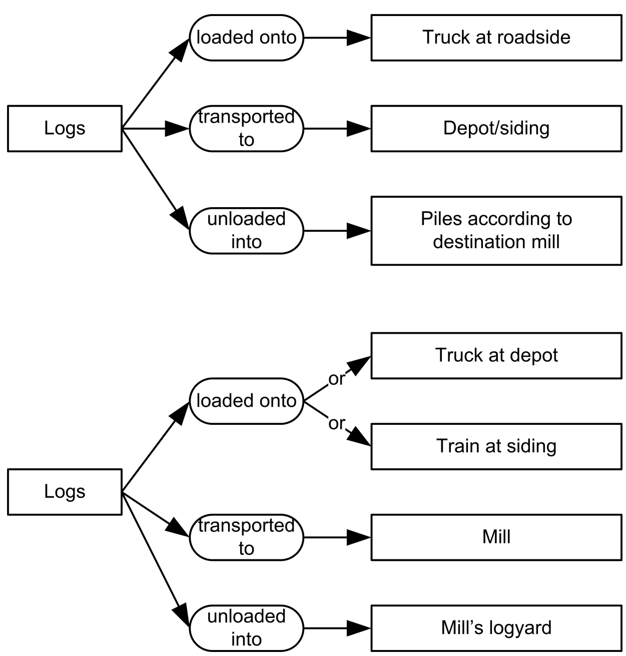

Fig. 5. Entity-Relationship diagram showing the transport of logs to the mill

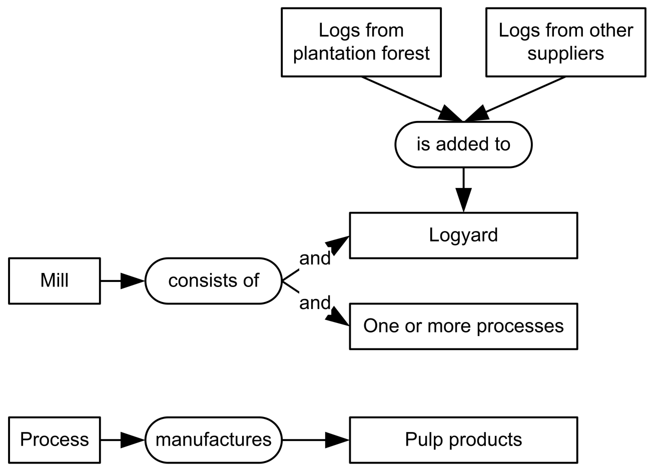

Fig. 6. Entity-Relationship diagram showing the entities of the mill

4 Discussion and Conclusion

The identification of domains, the actions which would typically occur in them, and the constraints acting on them was useful for both the semi-formal and formal modeling. For the semi-formal modelling, they acted as a reminder of issues to include in the models. For the formal (Z) models, the actions identified became actions schemas which changed the normal state to another state. The constraints were added in the second (predicate) part of the Z schemas.

The use of the Zachman framework for developing the semi-formal models of the domain was beneficial, because it allowed the models to be cross-checked. Deficiencies highlighted when developing one model could then be adjusted in others. Developing the formal models of the domains was beneficial because the formal approach required much more rigour and thought in their development. For example, if a formal model of the domain had not been developed, the fact that at the depot, the logs are put into piles according to their destination mill (see Fig. 5) would not have been uncovered. Using a formal notation such as Z also makes it easier to capture the business rules, as required by the last column of the Zachman framework. This is because the predicates which is declared in the second part of a schema is actually a constraint on the functions declared in the first (signature) part.

An analysis of the plantation forestry domain, with particular reference to pulp manufacture, has been presented from a computer science/information systems perspective. The main entities of the domain (forest, transport and mill) were described in terms of their actions and constraints on those actions. The actions and constraints for the plantation forestry domain were described using the formal notation Z. Not every action and constraint mentioned in the table was modelled with Z; this is the subject of ongoing research.

The approach taken was to model an abstraction of the domain, knowing that with each future refinement, more detail can be added. Using this approach, the model can be expanded to include other features, e.g. capturing the genus-species hierarchy, or expanding it for other end uses (such as sawmilling).

Although this is an analysis of the plantation forestry domain, many of the rules and relationships hold for natural forests. This work can be used, in conjunction with other works (e.g. [2, 18, 20]), when developing systems or processes for forestry companies: by checking the constraints, actions and relationships described here against their customer's, a more complete set of requirements could be obtained. An expanded version of this work is currently being used to specify a forest harvest scheduling system for plantation forestry which supplies wood to pulp mills.

Acknowledgements. The primary author would like to thank: the integrated forestry company for exposure to their business processes; the planning forester for reviewing earlier versions of the analysis; Hans Peters for reviewing forestry concepts; and the Forestry & Forest Products Research Centre, Durban, South Africa, under whose auspices this study took place.

References

1. Alagar V., Periyasamy, K.: Specification of Software Systems. Springer-Verlag New York Inc., (1998).
2. Baskent, E., Wightman, R., Jordan, G., Zhai, Y.: Object-Oriented Abstraction of Contemporary Forest Management Design. Ecol. Model. 143, 147164 (2001)
3. Bjørner, D.: A Triptych Software Development Paradigm: Domain, Requirements and Software Towards a Model Development of a Decision Support System for Sustainable Development. In: Olderog, E.-R., Steffen, B. (eds.), Correct System Design. LNCS, vol. 1710, pp. 29-60. Springer, Heidelberg (1999)
4. Bjørner, D.: Software Engineering 3: Domains, Requirements, and Software Design. Springer, Berlin (2006)
5. Cea, C., Jofré, A.: Linking Strategic and Tactical Forestry Planning Decisions. Ann. Oper. Res. 95, 131-158 (2000)
6. DeMarco, T.: Structured Analysis and System Specification. Prentice-Hall Inc., Englewood Cliffs (1979)
7. Dyck, B.: Benefits of Planted Forests: Social, Ecological and Economic. In: UNFF Intersessional Experts Meeting on the Role of Planted Forests in Sustainable Forest Management, pp. 1-10 (2003)
8. Evans, J., Turnbull, J.: Plantation Forestry in the Tropics: the Role, Silviculture, and Use of Planted Forests for Industrial, Social, Environmental, and Agroforestry Purposes. Oxford University Press, 3rd Ed. (2004)
9. Hay, D.: Requirements Analysis: from Business View to Architecture. Prentice-Hall PTR, Upper Saddle River (2003)
10. Herbert, M.: Site Classification and Forest Management: prepared for University of Natal School of Agricultural Sciences and Agribusiness. Fractal Forest AFRICA, P.O. Box 1497, 3650 Hillcrest, South Africa (2000)

11. Jackson, M.: Problem Frames: Analyzing and Structuring Software Development Problems. Addison-Wesley, Harlow (2001)
12. Jia, X.: ZTC: A Type Checker for Z Notation ver. 2.2. School of Computer Science, Telecommunication, and Information Systems, DePaul University, Chicago (2002)
13. Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, A.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Carnegie Mellon University, Software Engineering Institute, Pittsburgh (1990)
14. Kirilenko, A., Sedjo, R.: Climate Change Impacts on Forestry. Proc. Natl. Acad. Sci. 104, 19697-19702 (2007)
15. Leuschner, W.: Forest Regulation, Harvest Scheduling and Planning Techniques. John Wiley and Sons, New York (1990)
16. Martell, D., Gunn, E., Weintraub, A.: Forest Management Challenges for Operational Researchers. Eur. J. Oper. Res. 104, 1-17 (1998)
17. McNaughton, A., Ronnqvist, M., Ryan, D.: A Model which Integrates Strategic and Tactical Aspects of Forest Harvesting. In: Powell, M., Scholtes, S., (eds.), 19th IFIP TC7 Conference on System Modelling and Optimization, pp. 189-208. Kluwer (2000)
18. Nobre, S., Rodriguez, L.: A Relational Data Model for the Generation of Large-Scale Forest Scheduling Problems. In: Bevers, M., Barrett, T., (eds.): System Analysis in Forest Resources: Proceedings of the 2003 Symposium. General Technical Report PNW-GTR-656, pp. 299-306. USDA Forest Service, Pacific Northwest Research Station, Portland (2005)
19. Pohl, K.: The Three Dimensions of Requirements Engineering. In: Rolland, C., Bodart, F., Cauvet, C., (eds.), CAiSE 93, LNCS, vol. 685, pp. 275-292. Springer, Heidelberg (1993)
20. Ribeiro, R., Borges, J., Pereira, C., Sousa, P., Lé, J.: Designing an Integrated Forest Planning System for the Forest Industry: an Application in Portugal, In: Bevers, M., Barrett, T., (eds.): System Analysis in Forest Resources: Proceedings of the 2003 Symposium. General Technical Report PNW-GTR-656, pp. 89-96. USDA Forest Service, Pacific Northwest Research Station, Portland (2005)
21. Smith, C., Cunningham, L.: The Effect of Stand Density on the Growth of *Eucalyptus grandis* Hybrids at 5 to 7 Years of Age. Institute for Commercial Forestry Research Bulletin series 03/2002, Pietermaritzburg, South Africa (2002)
22. Sommerville, I.: Software Engineering. Addison-Wesley, Harlow, 8th ed. (2007)
23. Zachman, J.A.: A Framework for Information Systems Architecture. IBM Syst. J. 26, 276-292 (1987)