# Design, implementation and application of a generic framework for integrated regional land-use modeling

Dissertation

Zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.) im Fachbereich
Elektrotechnik/Informatik der Universität Kassel

vorgelegt von

Matthias Mimler

Kassel, 2007

Matthias Mimler

Center for Environmental Systems Research
Universität Kassel
Kurt-Wolters-Str. 3
34109 Kassel

## Danksagungen

An dieser Stelle möchte ich mich bei meinem Doktorvater Professor Joseph Alcamo für die Betreuung dieser Arbeit bedanken sowie bei Prof. Ralf Seppelt für die Übernahme des Zweitgutachtens. Ein herzliches Dankeschön gilt auch Prof. Andreas Ernst für seine Mitwirkung in der Promotionskommission.

Ganz besonders gilt mein Dank Jörg Priess für die hervorragende Betreuung, für eine Vielzahl an konstruktiven Gesprächen, viele wertvolle Ratschläge und zahlreiche Aufmunterungen.

Des weiteren bedanke ich mich bei meinen aktuellen und ehemaligen Kollegen für die großartige Arbeitsatmosphäre und die konstruktive Zusammenarbeit. Insbesondere möchte ich Achim Manche erwähnen für seine Unterstützung beim Aufbau des "Rechnerclusters" für die Modellkalibrierung.

Nicht zuletzt gilt mein Dank meiner Familie für ihre immerwährende Unterstützung!

## Erklärung

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und ohne unerlaubte Hilfe angefertigt und andere als die in der Dissertation angegebenen Hilfsmittel nicht benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen sind, habe ich als solche kenntlich gemacht. Kein Teil dieser Arbeit ist in einem anderen Promotions- oder Habilitationsverfahren verwendet worden.

Matthias Mimler

## Erklärung über Veröffentlichungen

Im Rahmen dieser Arbeit sind die folgenden Veröffentlichungen entstanden:

Mimler, M., Priess, J.A., 2007. SITE – A new generic framework for integrated regional land-use modelling. In preparation for Environmental Modelling and Software.

Priess, J.A., Mimler, M., Weber, R., Faust, H., 2007. Socio-environmental impacts of land use and land cover change at a tropical forest frontier. Accepted by MODSIM07, International Congress on Modelling and Simulation.

Mimler, M., Priess, J.A., 2007. An automated rule set calibration procedure for spatially explicit land-use change modelling. Submitted to Environmental Modelling and Software.

Priess, J.A., Mimler, M., Klein, A.-M., Schwarze, S., Tscharntke, T., Steffan-Dewenter, I., 2007. Linking deforestation scenarios to pollination services and economic returns in coffee agroforestry systems. Ecological Applications, 17, 407-417.

## Summary

Land use is a crucial link between human activities and the natural environment and one of the main driving forces of global environmental change. Large parts of the terrestrial land surface are used for agriculture, forestry, settlements and infrastructure. Given the importance of land use, it is essential to understand the multitude of influential factors and resulting land use patterns. An essential methodology to study and quantify such interactions is provided by the adoption of land-use models. By the application of land-use models, it is possible to analyze the complex structure of linkages and feedbacks and to also determine the relevance of driving forces.

Modeling land use and land use changes has a long-term tradition. In particular on the regional scale, a variety of models for different regions and research questions has been created. Modeling capabilities grow with steady advances in computer technology, which on the one hand are driven by increasing computing power on the other hand by new methods in software development, e.g. object- and component-oriented architectures. In this thesis, SITE (Simulation of Terrestrial Environments), a novel framework for integrated regional sland-use modeling, will be introduced and discussed. Particular features of SITE are the notably extended capability to integrate models and the strict separation of application and implementation. These features enable efficient development, test and usage of integrated land-use models. On its system side, SITE provides generic data structures (grid, grid cells, attributes etc.) and takes over the responsibility for their administration. By means of a scripting language (Python) that has been extended by language features specific for land-use modeling, these data structures can be utilized and manipulated by modeling applications. The scripting language interpreter is embedded in SITE. The integration of sub models can be achieved via the scripting language or by usage of a generic interface provided by SITE. Furthermore, functionalities important for land-use modeling like model calibration, model tests and analysis support of simulation results have been integrated into the generic framework. During the implementation of SITE, specific emphasis was laid on expandability, maintainability and usability.

Along with the modeling framework a land use model for the analysis of the stability of tropical rainforest margins was developed in the context of the collaborative research project STORMA (SFB 552). In a research area in Central Sulawesi, Indonesia, socio-environmental impacts of land-use changes were examined. SITE was used to simulate land-use dynamics in the historical period of 1981 to 2002. Analogous to that, a scenario that did not consider migration in the population dynamics, was analyzed. For the calculation of crop yields and trace gas emissions, the DAYCENT agro-ecosystem model was integrated. In this case study, it could be shown that land-use changes in the Indonesian research area could mainly be characterized by the expansion of agricultural areas at the expense of natural forest. For this reason, the situation had to be interpreted as unsustainable even though increased agricultural use implied economic improvements and higher farmers' incomes.

Due to the importance of model calibration, it was explicitly addressed in the SITE architecture through the introduction of a specific component. The calibration functionality can be used by all SITE applications and enables largely automated model calibration. Calibration in SITE is understood as a process that finds an optimal or at least adequate solution for a set of arbitrarily selectable model parameters with respect to an objective function. In SITE, an objective function typically is a map comparison algorithm capable of comparing a simulation result to a reference map. Several map optimization and map comparison methodologies are available and can be combined. The STORMA land-use model was calibrated using a genetic algorithm for optimization and the figure of merit map comparison measure as objective function. The time period for the calibration ranged from 1981 to 2002. For this period, respective reference land-use maps were compiled. It could be shown, that an efficient automated model calibration with SITE is possible. Nevertheless, the selection of the calibration parameters required detailed knowledge about the underlying land-use model and cannot be automated.

In another case study decreases in crop yields and resulting losses in income from coffee cultivation were analyzed and quantified under the assumption of four different deforestation scenarios. For this task, an empirical model, describing the dependence of bee pollination and resulting coffee fruit set from the distance to the closest natural forest, was integrated. Land-use simulations showed, that depending on the magnitude and location of ongoing forest conversion, pollination services are expected to decline continuously. This results in a reduction of coffee yields of up to 18% and a loss of net revenues per hectare of up to 14%. However, the study also showed that ecological and economic values can be preserved if patches of natural vegetation are conserved in the agricultural landscape.

## Zusammenfassung

Landnutzung stellt ein entscheidendes Verbindungsglied zwischen menschlichen Aktivitäten und der natürlichen Umgebung dar und kann zudem als eine der wichtigsten treibenden Kräfte des globalen Wandels betrachtet werden. Große Teile der Erdoberfläche werden land- und forstwirtschaftlich oder als Fläche für Siedlungen oder Infrastruktur genutzt. Aufgrund der Bedeutung von Landnutzung ist es unerlässlich, die Wechselwirkungen zwischen der Vielzahl von Einflussfaktoren und der daraus resultierenden Landnutzungsmuster zu verstehen. Die Modellierung von Landnutzungsänderungen und darauf aufbauende Simulationen sind hierfür geeignete Methoden. Landnutzungsmodelle ermöglichen es, die komplexe Struktur von Wechselwirkungen und Rückkopplungen zu analysieren und die Relevanz einzelner treibender Kräfte daraus abzuleiten.

Die Modellierung von Landnutzung und Landnutzungsänderungen kann auf eine mehrjährige Tradition zurückblicken. Insbesondere für die regionale Skala ist in den vergangenen Jahren eine Vielzahl von Modellen für unterschiedliche Regionen und Fragestellungen erstellt worden. Mit dem Fortschritt in der Computertechnologie wachsen auch die Möglichkeiten in der Landnutzungsmodellierung, einerseits durch steigende Rechenleistung, andererseits durch neue Methoden in der Softwareentwicklung, wie etwa objekt- oder komponentenorientierte Architekturen. In dieser Arbeit wird SITE (Simulation of Terrestrial Environments), eine neuartige Plattform für integrierte regionale Landnutzungsmodellierung vorgestellt und diskutiert. SITE zeichnet sich durch erweiterte Möglichkeiten zur Modellintegration sowie durch eine strikte Trennung von Anwendung und Implementierung aus, was schnelle und effiziente Modellentwicklung, -anwendung sowie -tests ermöglicht. Dabei stellt SITE auf der Systemseite generische Datenstrukturen (Grid, Zellen, Attribute etc.) zur Verfügung und verwaltet diese. Mittels einer Skriptsprache (Python), die um spezielle Modellierungsfunktionalitäten erweitert wurde, und eines in SITE eingebetteten Interpreters können diese Datenstrukturen für Modellierungsanwendungen verwendet und manipuliert werden. Die Integration von Teilmodellen ist über diese Skriptsprache oder eine allgemeine Schnittstelle, die von SITE bereitgestellt wird, möglich. Des weiteren sind für die Landnutzungsmodellierung wichtige Funktionalitäten wie Modellkalibrierung, Modelltests und die Analyse von Simulationsergebnissen in die generische Plattform integriert. Bei der Implementierung des Systems wurde besonderer Wert auf Erweiterbarkeit, Wartbarkeit und Benutzerfreundlichkeit gelegt.

Parallel zur Modellierungsplattform wurde im Kontext des Sonderforschungsbereiches 552 STORMA ein Landnutzungsmodell zur Untersuchung der Stabilität tropischer Regenwaldrandbereiche entwickelt. In einer Modellregion in Zentralsulawesi, Indonesien, wurden die Einflüsse von Landnutzungsänderungen auf die Umwelt sowie das soziale und wirtschaftliche Gefüge untersucht. Mithilfe von SITE wurden Landnutzungsänderungen in der historischen Periode von 1981 bis 2002 simuliert. Parallel wurde ein Szenario, das die Migration in der Bevölkerungsdynamik nicht berücksichtigt, analysiert. Zur Berechnung landwirtschaftlicher Erträge sowie Spurengasemissionen wurde das Ökosystemmodell DAYCENT integriert. In dieser Fallstudie konnte gezeigt werden, dass Landnutzungsänderungen im Untersuchungsgebiet hauptsächlich durch die Expansion landwirtschaftlicher Fläche auf Kosten des Regenwaldes charakterisiert sind. Ökonomische Verbesserungen können somit nicht als nachhaltig angesehen werden.

Aufgrund der Bedeutung von Modellkalibrierung wurde diese in SITE explizit durch die Implementierung einer entsprechende Komponente integriert. Die Funktionalität ist für alle SITE Anwendungen nutzbar und ermöglicht eine weitgehend automatisierte Modellkalibrierung. Sie wird als ein Prozess verstanden, bei dem für einen frei wählbaren Satz von Modellparametern eine optimale oder zumindest hinreichend gute Lösung hinsichtlich einer Zielfunktion gesucht wird. Die Zielfunktion ist typischerweise ein Kartenvergleichsverfahren, in dem ein Simulationsergebnis mit einer Referenzkarte verglichen werden kann. Es stehen mehrere Optimierungsverfahren sowie Zielfunktionen zur Verfügung. Das STORMA-Landnutzungsmodell wurde mittels eines genetischen Algorithmus in Verbindung mit dem *Figure of Merit*-Kartenvergleichsverfahren als Zielfunktion kalibriert. Der Kalibrierungszeitraum reichte von 1981 bis 2002, wobei entsprechende historische Landnutzungskarten zur Verfügung standen. Es konnte gezeigt werden, dass eine effiziente automatisierte Kalibrierung mit SITE möglich ist. Allerdings erforderte die Parameterauswahl detailliertes Wissen über das zugrunde liegende Modell und ist nicht automatisierbar.

In einer weiteren Fallstudie wurden Ertragsrückgänge und daraus resultierende Einkommensverluste aus dem Anbau von Kaffee unter vier verschiedenen Entwaldungsszenarien analysiert und quantifiziert. Dazu wurde ein empirisches Modell integriert, das den Fruchtansatz von Kaffee, der unter anderem von Bienenbestäubung abhängig ist, abhängig von der Distanz zur nächsten Regenwaldfläche beschreibt. Mit Landnutzungssimulationen konnte gezeigt werden, dass, je nach Größe und Ort fortschreitender

Umwandlung von Waldflächen, eine kontinuierliche Abnahme der Bestäubungsleistung zu erwarten ist. Daraus resultieren Ertragseinbußen für Kaffee um bis zu 18% und Einkommensverluste pro Hektar von bis zu 14%. Die Studie zeigte allerdings auch, dass diese durch den Erhalt von Stellen natürlicher Vegetation innerhalb der landwirtschaftlich genutzten Fläche deutlich gemildert werden können bei gleichzeitigem Erhalt ökologischer Funktionen.

# Contents

# 1 Introduction

## 1.1 Background

Land-use as “the total of arrangement, activities and inputs that people undertake in a certain land cover type”, in contrast to land-cover being the “observed physical and biological cover of the earth’s land, as vegetation or man-made features” (FAO, 1999), is a crucial link between human activities and the natural environment and one of the main driving forces of global environmental change (Lambin et al., 2000b). Large parts of the terrestrial land surface are used for agriculture, forestry, settlements and infrastructure. Concerns about land-use and land-cover change first emerged on the agenda of global environmental change research several decades ago when the research community became aware that land-surface processes influence climate (Lambin et al., 2006). While the focus in the beginning lay on the surface-atmosphere energy exchanges determined by modified surface albedo (Ottermann, 1974; Charney and Stone, 1975; Sagan et al., 1979), the view later on shifted to terrestrial ecosystems acting as sources and sinks of carbon (Woodwell et al., 1983; Houghton et al., 1985). A broader range of impacts of land-use change on ecosystems was identified since then. Besides being a major influencing factor on climate (Brovkin et al., 1999), land-use meanwhile is regarded the most important factor influencing both biodiversity and biogeochemical cycles on the global scale (Sala et al., 2000). To close the circle, land-use itself is strongly influenced by environmental conditions like climate and soil quality, affecting e.g. suitability for certain crop types and thus affecting agricultural use or biomass production (Mendelsohn and Dinar, 1999; Wolf et al., 2003).

Given the importance of land-use, it is essential to understand the interactions between the multitude of influential factors and resulting land use patterns. An essential methodology to study and quantify such interactions is provided by the adoption of land-use models. With land-use models it is possible to analyse the complex structure of linkages and feedbacks and to determine the relevance of driving forces (Heistermann et al., 2006). Land-use models are used to project how much land is used where and for what purpose considering different boundary conditions. After several years of research, land-use modeling has become an important technique for the projection of alternative pathways into the future, for conducting experiments that test our understanding of key processes, and for describing these processes quantitatively (Lambin et al., 2000b). Since land-use change models represent part of the complexity of land-use systems, they offer the possibility to test the sensitivity of land-use patterns to changes in selected variables. Through scenario building, they additionally allow testing of the stability of linked social and ecological systems (Veldkamp and Lambin, 2001).

## 1.2 Objectives

The objective of this thesis is to design, develop and implement a new and generic framework for regional and spatially explicit integrated land-use modeling called SITE (Simulation of Terrestrial Environments). Development took place in the context of the STORMA collaborative research project[1]. The main focus lies in the creation of a generic technical basis for land-use modeling applications. The design goal is to enable it to execute any spatially explicit land-use change rule set. In addition, the framework needs to integrate additional generic components required by modeling applications, e.g. functionality for model testing, model calibration and scenario management and analysis. As an integrated modeling framework, a standardized interface to sub models from third parties is required. Special focus is laid on the capability of the framework to calibrate and test land-use models and on a new and more flexible approach to scenario analysis enabling operator interaction.

The SITE modeling framework will be deployed an its applicability will be assessed in a comprehensive case study exmamining the dynamics of tropical rainforest margins on the island of Sulawesi, Indonesia. For this case study, a specific application rule set is developed and implemented as SITE application.

## 1.3 Thesis outline

The technical perspective, i.e. architectural and implementation details of the SITE framework, are described in chapter 3, followed by an introduction into the STORMA land-use model as the first SITE application in chapter 4. In addition, an analysis of socio-environmental impacts of land-use and land-cover change at the rainforest frontier is made using the STORMA model. In the following, the thesis will introduce and discuss innovative tools for land-use modeling applications. Chapter 5 describes the SITE calibration component as an example for a generalized methodology for the automated calibration of land-use model rule sets using optimization algorithms and map comparison procedures. Chapter 6 presents a study where a SITE application is used to estimate economic impacts due to land-use changes at a tropical rainforest frontier.

[1] Stability of rainforest margins in Indonesia, SFB 552, funded by the German Science Foundation (STORMA, 2003)

# 2 Current practice in land-use modeling

The objective of this chapter is to provide an overview over the current state-of-the-art in land-cover and land-use change modeling. After a short introduction into the theory of land-cover and land-use change, a brief overview over modeling concepts will be provided. A review of literature on the topic of land-use change will give an introduction to existing modeling applications and available modeling frameworks, focusing on the regional scale. Existing drawbacks will identified and requirements for the design, development and application of the SITE land-use modeling framework will be defined.

## 2.1 Land-use and land-use change

Land use plays a vital role in the earth system as it links human decision-making to the terrestrial environment and acts both as driver and target of global environmental change. The term land-use needs to be discriminated from land-cover. According to Watson et al. (2000), land-cover describes the biophysical properties of the earth surface and near-surface components such as vegetation, soils, topography, ground water and man-made features. Land-cover change can occur as either conversion or modification (Meyer and Turner, 1994). Conversion is understood an actual change in land-cover (e.g. forest being converted into grassland) and usually corresponds with a change in land-use. Modification is understood as a change in management (e.g. a different forest management). In contrast to that, land-use is defined based on the manner in which humans utilize land-cover. Examples for land-use could be settlement, agriculture or stock farming (Watson et al., 2000). One type of land-cover can be subject to several types of land-use. While forest seen as a land-cover class is defined by dominance of trees, regarded as land-use class it can be an area for wood production as well as one for human recreation. A conceptional framework that helps understanding and simulating changes in land-cover and land-use is provided by Meyer and Turner (1994) and – with a focus on tropical land-use – by Geist and Lambin (2001, 2002) who distinguish between proximate causes of land-cover and land-use changes and processes underlying those. An essential group among proximate causes are human activities that result in a change in land-use and conversion or modification of land-cover (e.g. slashing and burning of tropical forests, fertilizer application in agriculture, livestock farming or irrigation). Processes underlying proximate causes are termed driving forces (McNeill et al., 1994). Driving forces can be understood as a complex of social, political, economic, technological and cultural variables which define man-environment relationships (Geist and Lambin, 2001). These complex causes can be situated on different levels of scale. According to McNeill et al. (1994), they can be discriminated into political and economic determining factors, demographic evolution and availability or alteration of existing environmental resources. Typically, land-use dynamics are simultaneously caused by a combination of large scale drivers such as global markets or climate and region-specific forces like population dynamics and regional or local policies (Geist and Lambin, 2002).

Land-use change is of particular importance to environmental management since it significantly influences biodiversity, water and radiation budgets, trace gas emissions, the carbon cycle and livelihoods (Turner, 1994; Lambin et al., 2000a). Consequently, land-use planning one to influence dynamics of land-use change in order to achieve land-use configurations that balance environmental and stakeholder needs (Verburg et al., 2002). Especially for tropical regions, land-cover and land-use change have received much attention in recent years (Turner et al., 2001; Achard et al., 2002; Lambin et al., 2003; Tscharntke et al., 2007). This is mostly owed to the fact that tropical deforestation is one of the primary causes of global environmental change (Geist and Lambin, 2002). The study of deforestation in the tropics has become a major research topic (Mas et al., 2004; Agarwal et al., 2005; Etter et al., 2006; Wassenaar et al., 2007).

According to Lambin et al. (2000b), land-use change modeling has to address at least one of the following three questions. The first is environmental or socio-economic variables actually cause land-use changes. The second question is at which location land-use changes do occur. The third question is in which overall amount land-use change – independent of the location – does occur.

## 2.2 Land-use model classification

In the past years a multitude of land-use models have been developed addressing different applications, regions and scales. The following paragraphs give an overview over different established modeling methodologies that are currently applied in land-use change modeling.

### 2.2.1 Cellular automata (CA) models

A common method for spatially explicit representation of land-use dynamics in combination with the formulation of rules is provided by the use of cellular automata (CA). According to Wolfram (1984), CA are defined as spatially and temporary discrete systems. The modeling area is represented by a regular raster of identical cells. Each cell is characterized by its state, where the state is characterized by a certain combination of attribute values (e.g. current land use, elevation). Changes of grid cell states happen in discrete time steps, and an initial system state needs to be provided.

A cellular automaton is defined by the following parameters:

- Cell space $R$: Regular grid/raster of cells of identical size
- Finite neighborhood $N$: Based on a specific neighborhood definition (see below)
- A set of cell states $Q$
- A local transition function $\delta : Q^N \longrightarrow Q$

Changes in the state of a specific cell from time $t$ to $t + 1$ occur according to the local transition function (or rule set in the context of land-use modeling). They depend on the states of a cell and its neighbors at time $t$. State changes occur quasi-simultaneous by setting new attribute values at the end of a simulation step thus avoiding artificial effects

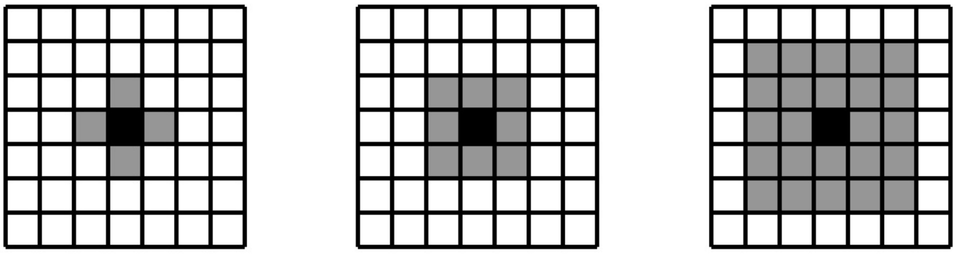

Figure 2.1: Examples for neighborhood definitions on CA. *Von Neumann* (left) and *Moore* neighborhood (center), extended *Moore* neighborhood with radius $r = 2$ (right).

due to the succession of cells being handled. Transition rules can be as simple as ordinary if-then statements. The global system behavior results from locally defined decisions in autonomous grid cells instead of being controlled at the top scale. Consequently, system state evolution usually cannot be anticipated other than by simulation experiments (Weimar, 1997). Thus, CA are classified as self-organizing systems, such as agent-based models (Manson, 2001).

Evolution of CA-based systems is strongly determined by the states of neighboring cells, thus, the definition of neighborhood is crucial. On a two-dimensional rectangular raster typically two types of neighborhood are distinguished. *Von Neumann* neighborhood includes the four adjacent neighbor cells that share one side with the central cell. *Moore* neighborhood also includes diagonal neighbor cells. The neighborhood definitions can be extended by specifying a radius. Figure 2.1 shows examples for different neighborhood definitions.

The dynamic behavior of CA is defined by the transition function, which is dependent from raster geometry, the neighborhood definition and the possible cell states. The transition function is applied to each single cell and describes the changes in a cell's state during one discrete step in time. Weimar (1997) and Wolfram (1984) list three different types of rule systems.

Via *direct specification* for each configuration of neighbor cells with respect to the current cell state a new cell state is assigned. More typical is the application of *totalistic rules*. The most popular example for totalistic rules is John Horton Conway's *Game of Life*. In a Game of Life simulation, each cell can either be populated (state = 1) or unpopulated (state = 0). Assuming *Moore*-neighborhood, transition rules are formulated as follows:

$$\delta(t+1) = \begin{cases} 0 & \text{if} \quad n \leq 1 \quad \text{or} \quad n \geq 4 \\ \delta(t) & \text{if} \quad n = 2 \\ 1 & \text{if} \quad n = 3 \end{cases},$$

where $t$ is the simulation time step and $n \in [1, 2, ..., 8]$ is the number of neighbor cells with state $\delta(t) = 1$ (populated). The third type are rules applied in combination with specific *transition probabilities*, where all probabilities for the achievement of all possible transitions have to sum up to 1.

In pure CA models, system dynamics are exclusively determined by the local transition rules. In modeling practice, however, often exogenous factors have to be considered in addition to the states of neighbor cells. This specific CA is referred to as constrained CA (White et al., 1997; Schaldach, 2004). Especially trends over the simulation period are implemented as exogenous factors. A typical example are global demands (e.g. crop yields, crop area) that have to be provided by the model (Schaldach and Alcamo, 2006) in each simulation step.

CA methodology is used in particular for models analyzing the development and diffusion of settlements and urban areas and resulting implications on the surrounding landscape (Clarke et al., 1996; Batty, 1997; Wu, 1998; Candau, 2000; Jenerette and Wu, 2001; Liu and Phinn, 2001; Messina and Walsh, 2001). However, CA models have also been used in the study of land-cover change influenced by agriculture (Rubiano, 2000).

### 2.2.2 Agent-based models

Comparable to cell states in CA-based models, the dynamics of agent-based models are based exclusively on the behavior of the individual agents. Agents are real or abstract entities (e.g. representing people, animals, organizations) capable of communicating with both other agents and their environment. Agents are autonomous, share an environment through agent communication and interaction and make decisions that tie behaviour to the environment (Parker et al., 2003). The main difference to cells in CA-based models is that agents are mobile instead of being bound to a fixed position. In addition, different rules can be defined for different agents, while for cells one uniform set of rules is provided (Schaldach, 2004). In contrast to CA-based models, which are usually focused on landscapes and transitions, agent-based models typically focus on human actions (Parker et al., 2003). Most agent-based models simulate decision processes and interactions of land-users in a study area (Verburg et al., 2002). For spatial representation, agent-based models can make use of underlying CA data structures (Berger, 2001).

### 2.2.3 Empirical/statistical models

Empirical/statistical models use statistical methods to relate driving forces to changes in land-cover and land-use that have been observed in the past. This relationship is assumed to also be valid in the future. For the explanation of existing land-use patterns, regression analysis is a prominent methodology (Veldkamp and Fresco, 1996; Schneider and Pontius, 2001; Serneels et al., 2001). Both biophysical and socio-economic variables can be considered under the assumption of sufficient statistical independence. Transfer of an empirical/statistical model to other study areas than the one it has been developed for is only possible if original and new areas are adequately comparable. In addition, only short-term studies can be conducted since analysis is based exclusively on the current system coherence. It is not possible to model system changes and trends (Lambin et al., 2000b).

### 2.2.4 Optimization models

Optimization models have their origin in economics and can comprise both micro- and macro-economic approaches (Angelsen and Kaimowitz, 1999). Both of them can be either spatially explicit or non-spatial. Micro-economic models tempt to explain the utilization of resources from an individual's view and consider economic variables such as prices, preferences, access to infrastructure and available technologies. The actual optimization is based on an objective function regarding possible constraints like political and planning allegations or available natural resources (Hilferink and Rietveld, 1999).

Macro-economic models work on country or world-region scale and are usually implemented as equilibrium models. Macro-economic models usually consider several sectors (e.g. agriculture, forestry, industry) and the connections between them (Kaimowitz and Angelsen, 1998).

### 2.2.5 Dynamic simulation models

Dynamic simulation models describe biophysical and socio-economic processes that produce patterns of land-cover changes in space and emphasize the interactions among all components forming a system. They are based on an a priori understanding of the system driving forces and describe complex ecosystems by a small number of differential equations (Lambin et al., 2000b). In a study analyzing land-use change in the sudanese sahel, Stéphenne and Lambin (2001) used a dynamic simulation model to calculate the yearly demand for different land-use class allocations using population dynamics, livestock numbers, precipitation and import of cereals as driving variables. Dynamic simulation models are application- and scale-specific, since they are typically parameterized based on local observations and data. Such relations cannot be used to straightforwardly model aggregate behavior (Lambin et al., 2000b).

### 2.2.6 Integrated modeling approaches

For integrated modeling, no unique definition is available. In the context of this thesis, integrated models are understood as models that include information from at least two disparate disciplines, represent this information in the form of discrete programming modules or sub models and link scientific findings with policy analysis, either implicitly or explicitly (Alcamo, 2002). A prominent example is the global-scale IMAGE model (Alcamo et al., 1998), which is a product of different global change research studies, integrating models from oceanic/atmospheric research, energy, industry and the terrestrial environment. The IMAGE land-use model is coupled with a dynamical simulation model to derive the demands for different agricultural products. A recent example of an integrated modeling application is MedAction (Van Delden et al., 2007), a policy support system focusing on the management Mediterranean river basins.

If integrated modeling is understood as combining existing models from different disciplines, this raises the question of how the actual coupling of models can be achieved. According to Lau et al. (1999), coupling of models happens on different levels:

- *Conceptual level:* abstraction of the overall framework, description of the entire system using suitable partial models
- *Methodological level:* conversion of model variables
- *Technical level:* solving implementation details, definition of interfaces, "nuts and bolts of linking the models"
- *Operational level:* coordination of research activities
- *Transitional level:* merging activities into mainstream

While the latter two levels have to be addressed especially for planning, coordinating or managing modeling projects, the conceptual, methodological and technical levels are of specific interest for this thesis. No matter how loosely or tightly models are coupled, the coordination of the single models regarding their spatial and temporal dimensions and the establishment of a flow control is crucial. It is frequently necessary to scale models to enable data exchange with other models (Lau et al., 1999). Implementation of feedback mechanisms between single models complicates the requirements to the flow control (Alcamo et al., 1998). However, feedback mechanisms of biophysical processes on land-use decisions are mostly not implemented in spatially explicit land-use models (Veldkamp and Lambin, 2001).

## 2.3 Short review of land-use models

In research on land-use dynamics, models are used in a variety of fields. The goal is to better understand the dynamics of the underlying system, to develop hypotheses that can be tested empirically and to derive predictions or scenarios for assessment activities (Brown et al., 2004). In the LUCC (Land-use and -cover change) project science plan of the International Geosphere-Biosphere Program (IGBP) and the International Human Dimensions Program (IHDP), land-use models have been defined as an important component of each of the three foci outlines (Turner et al., 1995). In the consecutive Global Land Project (GLP), integrated analysis and modeling are regarded key methodologies to assess land sustainability (GLP, 2005).

A large number of different land-use models have been described in literature on landscape ecology, geography, urban planning, economics, regional science, computer science, statistics, geographic information science and various other fields (Brown et al., 2004). Due to big differences among those models, both due to differing disciplinary perspectives as well as differing methodologies, data availability and modeling goals, a categorization of models gets rather complicated and literature reviews on land-use models remain a challenging task. Reviews have been made by several authors already. One of the earliest reviews has been created by Baker (1989) in the context of landscape ecology classifying models according to their modeling goals. None of the models discussed by Baker (1989), however, included explicit representation of human decision-making. During the 1990s, a significant amount of work was done on modeling tropical deforestation. A review on models focusing on observed land-cover change using empirical-statistical methods and spatial simulation is provided by Lambin (1997). Another group of LUCC models developed to describe micro-, regional or macro-economic aspects of the deforestation process and using similar methodological categories was summarized by Kaimowitz and Angelsen (1998). Irwin and Geoghegan (2001) compared non-economic models including several former approaches (Baker, 1989; Lambin, 1997). In their review, the authors distinguished between spatially explicit and non-spatial models. Agarwal et al. (2002) presented a review describing 19 models which they classified by dimensions of space, time and human decision-making; the authors emphasized the importance of how human decision-making is represented in these models. Another widely discussed method to represent human decision-making is available through the use of agent-based modeling, which represents a qualitatively different approach to ecological modeling. A review of agent-based models

Table 2.1: Overview over selected regional land-use models and their specific characteristics, compiled from Verburg et al. (2004) and Heistermann et al. (2006).

| Model name | Model type | References | Spatial and temporal resolution | Description |
|---|---|---|---|---|
| Urban Growth Model UGM / SLEUTH | CA model | Clarke and Gaydos (1998); Candau (2000) | San Francisco and Washington/Baltimore region, Mid Atlantic region. Possibility for urban regions in general. Temporal: 1900-21000 | Self-modifying CA model. Four processes included: spontaneous growth (neighborhood, suitability), diffuse growth (slope determined), edge growth (neighborhood), road influenced growth. Calibration is used to determine relative influence. |
| Constrained CA models: RamCo, LOV, MODULUS, SIMLUCIA | CA models and integrated/hybrid model | White and Engelen (2000) | St. Lucia (Caribean Island); Netherlands; City of St. John's (Newfoundland, Canada); Ujung Pandang (Sulawesi, Indonesia). Temporal: ±30 years | Constrained CA; demands for land use are calculated in sectoral models at an aggregated spatial level (LOV: gravity model) and allocated using CA. The CA are quantified using expert knowledge. |
| CLUE, CLUE-S, CLUE-Neotropics | Empirical-statistical model | Veldkamp and Fresco (1996); Verburg et al. (1999, 2002); Wassenaar et al. (2007) | Continental level: Central-America, China; National level: Ecuador, Central-American countries; Subnational level: Phillipines, Indonesia, Costa Rica. Temporal: ±20 years | Dynamic simulation model using empirically derived relations between land-use change and driving forces from cross-sectional analysis at multiple scales. |
| IIASA-LUC | Integrated/hybrid model and optimization model | Fischer and Sun (2001); Hubacek and Sun (2001) | China: agro-ecological analysis for detailed pixels; integrated analysis for 8 regions. Temporal: projections for 2025 | General equilibrium model based on multi-sector input-output tables and detailed agro-ecological characterization. |
| Cormas: a multi-agent simulation software for renewable resource management | Agent-based model | Bousquet et al. (1998) | Different case-studies at village level. Temporal: variable | Multi-agent simulations based on behavior of individuals towards other individuals and natural resources. |
| PLM: Patuxent landscape model | Empirical-statistical model | Bockstael (1996); Voinov et al. (1999); Irwin and Geoghegan (2001) | Patuxent watershed (several US counties); based on integrated ecological models for the Everglades. Temporal: decades | Integrated economic/ecological model. Land-use change allocation is based on economic modeling (hedonic modeling combining distance and location operators). The neighborhood of the location is taken into account by quantifying its effect on land value. |

in land change science is given by Parker et al. (2003).

Recent reviews have been provided by Verburg et al. (2004) for regional scale models and by Heistermann et al. (2006) for continental to global scale models. Verburg et al. (2004) classify models according to six different criteria: (1) Level of analysis, (2) cross-scale dynamics, (3) driving factors, (4) spatial interaction and neighborhood effects, (5) temporal dynamics and (6) level of integration. In contrast to that, Heistermann et al. (2006) differentiate between geographic and economic models, which they distinguish based on the type of land-use allocation.

Table 2.1 shows a cross-section over different existing land-use models. The selection of models also represents the main model types. Since this thesis focuses on regional land-use modeling, only models operating on regional scale are listed. Most models use spatially explicit representation of data, typically by defining a spatial raster reflecting the modeling goals and data availability. Modeling methodologies include CA-based, empirical/statistical, agent-based and integrated approaches. In particular SLEUTH and CLUE can be regarded as sort of a model family, since they have been additionally adopted for other than their originally intended applications. SLEUTH, a CA-based land-use model primarily applied to urban change studies in the San Francisco and Washington/Baltimore area, can be transferred to be used for urban regions in general (Clarke and Gaydos, 1998; Candau, 2000). From the original CLUE model, applied to analyze and simulate land-use change in Costa Rica (Veldkamp and Fresco, 1996), a number of other different applications have evolved (Verburg et al., 1999, 2002; Wassenaar et al., 2007).

## 2.4 Modeling environments

Implementing a model technically implies to formulate it in a set of computer instructions. Two complementary possibilities to achieve this are to either reuse and adapt an existing model that resembles the actual modeling task, or by programming a new model. It is obvious that both ways have several advantages and drawbacks. Reusing an existing model definitely saves cost and time. However, it is likely that a model created for a different though potentially similar task needs to be adapted to a certain degree. This might not always be possible to achieve in a satisfying manner as too many compromises might need to be accepted. Programming a new model by using a general-purpose programming language (e.g. C++, Fortran) will definitely result in a tailor-made application, but this approach is usually cost- and time-intensive depending on the complexity of the desired application. In addition one will usually be able to identify certain structure in the model, that can be equally used in other applications. For cellular automata (CA) models, e.g., the underlying cell lattice is such a generic structure, which only differs among models in its size and the spatial resolution it represents. The actual differences of CA models lie in their transition rules and constraints.

Another complication is caused by the fact that landscape ecologists, planners and modelers, i.e. the persons actually developing models, are not necessarily programmers. To provide efficient model development and model adoption conditions, model developers should be able to transform their conceptual models into computer simulations without having to implement them in a general-purpose computer language or at least without having to deal with specific details of software development, e.g. memory management. This conflict has been recognized by the modeling community in the past years and a

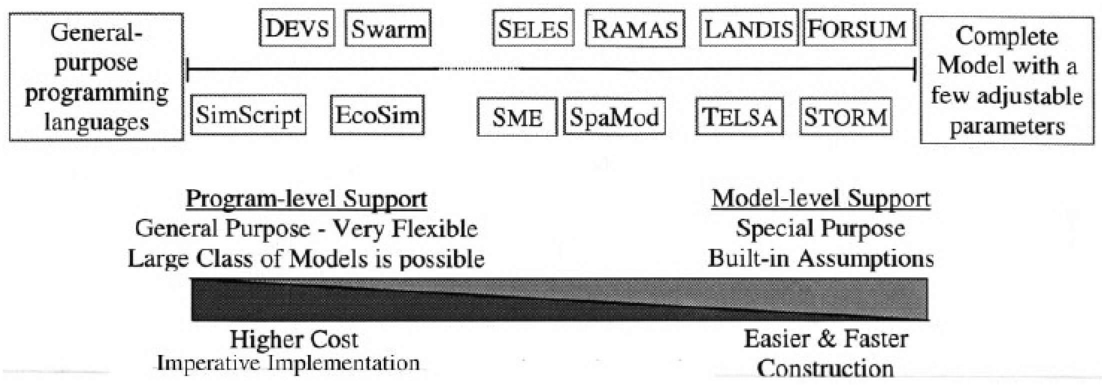


Figure 2.2: Spectrum of approaches to implementing spatial landscape models (Fall and Fall, 2001).

number of possible solutions, in the following termed modeling environments, have been proposed to "untangle the beauty of a model from the beast of its implementation on a computer"(Fall and Fall, 2001).

Fig. 2.2 shows several examples of existing modeling environments along a spectrum of specificity. On the left lie general purpose programming languages, opposed to complete specific models on the right edge whose possibilities for adaption are restricted to a number of adjustable parameters. Along that gradient, a number of solutions is listed. Examples on the left side are program-level support tools. Simscript and DEVS (Clark, 1992; De Vasconcelos and Zeigler, 1993) are extensions to programming languages providing modeling functionality. EcoSim (Lorek and Sonnenschein, 1998) is a software library that reduces complexity by providing modeling functionality, thus enabling model developers to concentrate on writing high-level code, i.e. rule set and model parameterization. A similar concept is pursued for multi-agent simulation by Swarm (Minar et al., 1996), which additionally provides a set of graphical tools for display and analysis of modeling results. The use of the latter solutions still requires programming knowledge. They can reduce development time and cost, but result in specialized modeling applications.

In contrast to program-level support tools, model-level support tools can assist model construction without requiring the creation of source code (Fall and Fall, 2001). FORSUM (Frelich and Lorimer, 1991) and STORM (Krauchi, 1995) are applications that implement a template for a set of similar modeling applications. Portability to other applications is enabled by providing an extensive set of parameters for which values or definitions can be altered. Flexibility is increased with approaches like LANDIS (Mladenoff et al., 1996) and TELSA (Klenner et al., 1997) that provide extended features like user-definable spatial layers or variables.

Largest flexibility combined with minimum programming requirements is given by tools that provide model-level support for classes of models rather than individual model types. Examples can be found in the middle of the spectrum depicted by Fig. 2.2. Different approaches are used to achieve this target. SpaMod (Gao, 1996), for example, enables high-level formulation of mathematical models through differential equations which are automatically translated into C code and subsequently linked to the system run-time environment, thus creating a simulation tool. SME (Maxwell and Costanza, 1997) uses a similar technique as it provides a high-level language for the specification of modular spatio-temporal models that is later translated into C++ code and linked to the SME run-time environment. STELLA (Costanza et al., 1998), suitable for non-spatial simulations, allows to freely design a model via building blocks on a graphical user interface. SELES (Fall and Fall, 2001) supports spatially explicit, CA-based land-use modeling applications by providing the necessary generic infrastructure, in particular the simulation grid. For the formulation of application rule sets, Fall and Fall (2001) developed a domain-specific language. This language enables a rule set developer to concentrate on the actual modeling task and fulfills a number of specific criteria like simplicity, flexibility capability modularity, transparency, efficiency, reusability, adaptability and communicability.

The increased use of models in ecological sciences, the advent of modeling environments and frameworks and the trend toward integrated systems of models from different disciplines imply increasing complexity regarding the implementation as software. A number of concepts for handling software complexity is provided from computer science. In particular, modularization, object-oriented design and encapsulation of models by means of components with clearly defined interfaces are of specific interest (Argent, 2004). Modularization and object-oriented design contribute to an improved quality of the implementation of single models. The creation of model components can help to facilitate model coupling and assembling of integrated systems. The advantages of a component-based design are numerous. For example, making changes in the implementation of one particular model will not affect other parts of an integrated system due to encapsulation. Models responsible for specific processes in the integrated system can simply be replaced by other models, provided they have the same interface.

In the past years, studies on synergies between environmental modeling and computer science have been conducted and object- and component-oriented tools and designs have entered the field of land-use modeling (Maxwell and Costanza, 1997; Fall and Fall, 2001). Villa and Costanza (2000) adopted a component-based software architecture for their Simulation Network Interface (SNI), emphasizing the advantages of encapsulated complexity, exchangeability and reusability of single models. In a similar manner, the DEVS framework was advanced to enable the definition of components (Filippi and Bisgambiglia, 2004). In a study focusing on forest landscape modeling, He et al. (2002) experienced that the use of component-based model integration supports scientists with few expertise in computer science in building complex, but still manageable applications. GEONAMICA (Engelen et al., 1999) combines a generic modeling platform with the possibility of integrated modeling using a component-based approach (Oxley et al., 2004; Van Delden et al., 2007). With Eclpss (Ecological Component Library for Parallel Spatial Simulation; Wenderholm, 2005) this component-based approach is enhanced by enabling parallel processing. In the future, integrated models could be constructed from models implemented as components in a so called “plug and play” approach, where a model component such as a water balance algorithm could simply be replaced by an alternative component (Argent, 2004). Modern tools that support the development of component-based architectures like Microsoft .Net or Mono offer information about components using meta data and introspection. ICMS (Interactive Component Modeling System; Rahman et al., 2004) takes advantage of these features to improve usability of modeling frameworks by generating self-documenting components and custom meta data. In the combination with other advantages of modern programming platforms (e.g. multi-language development, web enabled models), Rahman et al. (2004) see strong impulses for model developers and uses.

A further benefit from advances in computer science is the increasing use of graphical tools that support the design, development and the integration of models. An early example is the adoption of the STELLA framework for ecological modeling (Costanza et al., 1998). In the recent years, additional tools became available. Filippi and Bisgambiglia (2004) provided a graphical editor to establish model integration and flow for the JDEVS framework, Rahman et al. (2004) chose a comparable technique (called model linking canvas) for their ICMS framework. SimuMap (Pullar, 2004), a tool that formally represents a model as a set of coupled sub-models, uses a visual modeling interface to characterize and run system models.

# 3 The SITE modeling framework

In this chapter, SITE (**SI**mulation of **T**errestrial **E**nvironments), a generic framework for integrated land-use modeling on the regional scale, will be introduced. In the following, the requirements to be accomplished by a generic framework for regional land-use modeling will be identified and discussed with particular focus on the application of the framework in interdisciplinary projects. Subsequent to that, a detailed description of the architectural design and the implementation of SITE will be presented. The chapter closes with a discussion of the system characteristics and an evaluation of the accomplishment of the drafted requirements.

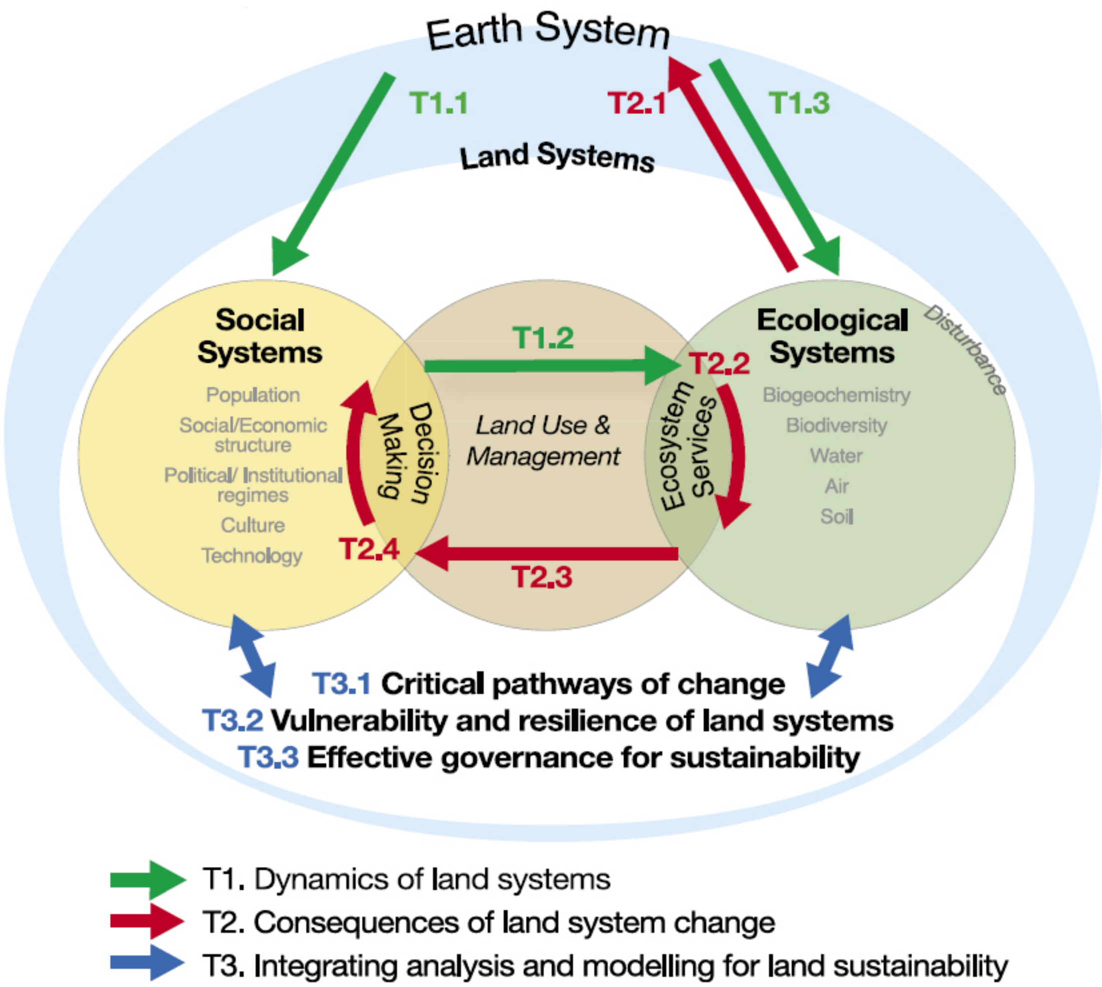


Figure 3.1: Analytical structure of the interdisciplinary Global Land Project (GLP, 2005).

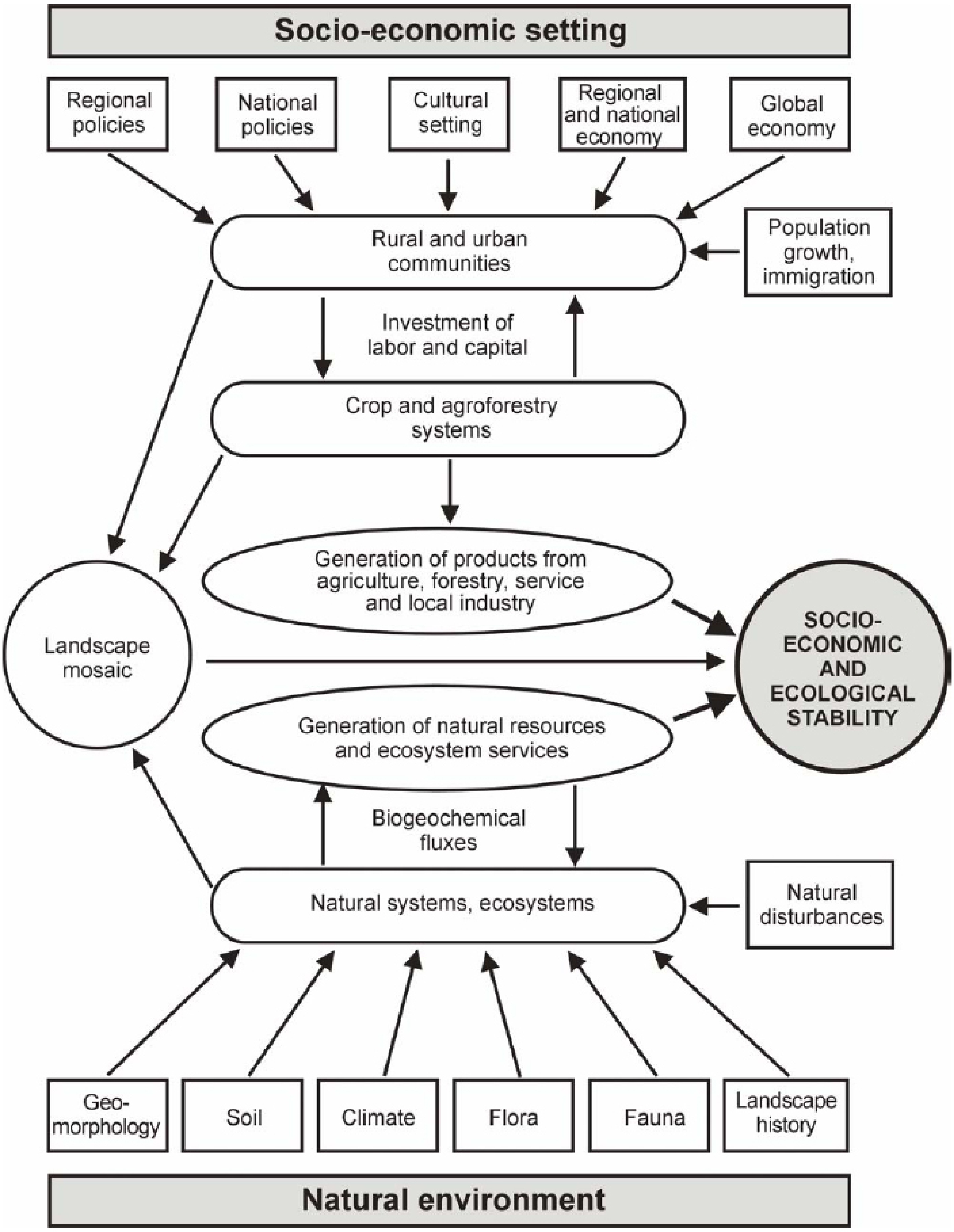


Figure 3.2: Interactions between socio-economic (upper half) and natural (lower half) systems. The landscape mosaic is strongly determined by both domains and is a major influencing factor for socio-economic and ecological stability (STORMA, 2003).

## 3.1 Requirements

Land-use dynamics are driven by a variety of factors, biophysical as well as socio-economic. Therefore, comprehensive research on land-use dynamics needs to be organized in interdisciplinary projects. Figure 3.1 shows the structure of the Global Land Project (GLP, 2005), which is a long-term research framework for land systems and a structural template for regional projects. Figure 3.1 delineates the large variety of interactions between the earth system, terrestrial subsystems and land use. A more detailed view of interactions between the socio-economic and the biophysical subsystem and their influence on land-use is provided by Figure 3.2. It is the challenge of integrated land-use modeling to include the insights of different sectoral views and disciplines and their respective interactions for the simulation and assessment of land-use changes.

According to that, the main target in the development of SITE was to create a land-use modeling framework capable of integrating scientific results of the STORMA[1] research project. STORMA is an interdisciplinary long-term project funded by the German Research Foundation. The target is to analyze the stability of rainforest margins in a research area in Central Sulawesi, Indonesia (STORMA, 2003). The proposed modeling outline for STORMA is guided by a similar scheme as the Global Land Project. The nature of SITE, being an integrative tool in the context of STORMA and potentially other similar research projects, directly implies a number of requirements with respect to the system design. In turn, certain technical demands result from the scientific requirements. In the following, these requirements and technical issues will be identified and defined.

#### Integrated modeling

The requirement of integrated modeling capability was implied directly by the definition of SITE, being a framework for modeling land-use changes in an interdisciplinary research context. Along with generic applicability, it can be considered the most important demand.

Integrated modeling is already established in the land use modeling community (Alcamo et al., 1998; Voinov et al., 1999; White and Engelen, 1997, 2000; Oxley et al., 2004; Van Delden et al., 2007). When reviewing literature on regional land-use modeling frameworks, however, it is noticeable that there are approaches that either support the implementation of models (Gao, 1996; Fall and Fall, 2001) or ones that enable integration or combination of existing models (He et al., 2002; Filippi and Bisgambiglia, 2004; Argent, 2004).

Following project requirements, the framework needed to enable both model development on the application side and interfacing capabilities to existing models. The only comprehensive solution for both providing a generic modeling platform and integrated modeling is GEONAMICA (Engelen et al., 1999; Oxley et al., 2004; Van Delden et al., 2007), which, however, is a commercial product. In addition, model integration in the context of the STORMA project additionally required high flexibility concerning feed back mechanisms from sub models to the actual land-use model as a basis for further decision making (e.g. about environmental and economic indicators, see chapters 4 and 6). Consequently, interfaces that allow model integration in SITE had to include functional-

[1]SFB 552 "Stability of rainforest margins in Indonesia" (STORMA, 2003)

ity to feed back results to the calling instance. To handle possible performance problems regarding run-time that can arise from model coupling, the SITE interface was designed to enable parallel processing if allowed by the modeling methodology.

#### Generic platform

The definition of SITE as a framework for land-use modeling implies that it needs to provide a generic platform for operating land-use modeling applications. As a generic platform, the use of SITE is not restricted exclusively to STORMA. The framework is also suitable for other similar research projects with an interdisciplinary outline.

A generic modeling framework is mainly characterized by the separation of the actual modeling application (rule set specification) from the implementation of structures that are shared by all potential models that can be operated within the framework. For spatially explicit models, these structures basically are the simulation grid, the single grid cells and structures for the handling of cell attributes. Model-specific values, like the grid dimensions as well as concrete attributes and attribute values characterizing grid cells are in turn specified by the modeling application. Connection to the data structures supplied by the framework is established via a specific interface. For the implementation of the model itself, a variety of solutions are imaginable. Costanza et al. (1998) used a graphical front end for the definition of model semantics. Another solution is the introduction of an application specific programming or description language (Fall and Fall, 2001). With GEONAMICA (Engelen et al., 1999), also a commercial product is available that supports both model definition and model integration. In the SITE framework, a scripting language, of which the functionality was specifically extended to match land-use modeling demands, was chosen for the task of implementing modeling applications. The main advantage of a framework is that model developers can concentrate entirely on their modeling task, ignoring implementation details. Thus, a faster and more efficient formulation of land-use models is possible. This is of specific interest for interdisciplinary working environments, in which model demands are likely to be altered frequently throughout the communication process among different parties. It also facilitates the execution of further case studies and the development of model prototypes since model code can be rapidly altered without having to consider any side effects on implementation details.

#### Integration of calibration methodology

Modeling practice reveals the necessity for model calibration. This field is widely discussed among the scientific community (Boumans et al., 2001; Oliva, 2003; Straatman et al., 2004). Despite its importance, there is no framework available yet, that implements calibration as an integral part. As a novelty, the goal of allowing rapid and effective development of land-use models and model integration was extended to also include model calibration. The design of the system even allows for a simultaneous calibration of different component models, which are both influencing land allocation (e.g. the land-use model and the biophysical sub model). Generic model calibration in SITE is based on parameter optimization with respect to an objective function. Consequently, beside optimization algorithms or heuristics, additional methods serving as objective functions needed to be integrated. The SITE calibration procedures use map comparison algorithms as objective function. The quality of simulation and calibration results is benchmarked based on reference maps.

### Integration of scenario handling

As simulations of land-use dynamics are generally conducted under specific scenario assumptions, SITE was required to provide functionality to handle scenario information. Like calibration functionality, the handling of scenarios is not explicitly addressed by available modeling frameworks. SITE was designed to explicitly represent the handling of quantified representations of scenarios in its implementation. In addition, the scenario implementation supports interactive use of scenarios, during which simulations can be stopped to examine if simulation targets have been achieved. Based on this intermediate analysis, scientists can alter the underlying scenario (e.g. by adjusting management parameters), thus simulating the interaction of policy makers. This method was proposed by Alcamo et al. (2006) to overcome one of the major limitations of the current scenario methodology and is currently implemented in no other modeling framework.

### Usability and cummunicability

As an integrative tool, SITE was required to be able to transport insights back to all other parties involved in the respective research project. To increase the acceptance among participating scientists, especially of those with backgrounds which are not easily related to computer science, it was crucial that the modeling system was designed for simple handling and transparent delineation of the modeling concept. Moreover, it needed to be capable of communicating its modeling and simulation results. In recent approaches, model developers became increasingly aware of this fact and introduced different solutions ranging from display of the simulation grid (Fall and Fall, 2001) to graphical tools for editing and viewing the model structure (Costanza et al., 1998; Filippi and Bisgambiglia, 2004). For SITE, a detailed graphical user interface (GUI) was implemented that facilitates model operation and understanding by e.g. providing 3-dimensional multiple views of the simulation grid.

Besides model operation, also model development, which usually involves programming, can be simplified, thus increasing usability by enlarging the potential user community. Graphical model builders (Costanza et al., 1998; Filippi and Bisgambiglia, 2004) are attractive but potentially unflexible approaches to this task, since they typically force specific modeling methodology. For SITE, we took advantage of the large potential that lies in modern scripting languages, which are powerful but nonetheless relatively simple and extendable programming tools.

As there is practically no restriction to the number of model parameters and settings for SITE applications, it is likely that single simulation runs cannot be reproduced, as the selected parameter values for an earlier simulation are not saved persistently. This situation conflicts with the requirement of communicability of modeling results. To overcome this limitation of existing frameworks, this feature was integrated in the SITE usability concept.

Table 3.1: Comparison of SITE to selected spatial modeling frameworks with respect to the requirements identified for a generic framework for integrated land-use modeling.

| | **SELES** (Fall and Fall, 2001) | **DEVS/JDEVS** (Filippi and Bisgambiglia, 2004) | **SimuMap** (Pullar, 2004) | **Eclpss** (Wenderholm, 2005) | **GEONAMICA** (Engelen et al., 1999) | **SITE** (this study) |
|---|---|---|---|---|---|---|
| **Purpose** | Land-use modeling | Environmental modeling | Spatial modeling of environmental processes | Grid-based ecosystem modeling | Dynamic land-use modeling and spatial decision support | Integrated land-use modeling |
| **Integrated modeling** | No | Yes | No | Yes, capable of parallel processing | Yes | Yes, capable of parallel processing |
| **Generic land-use modeling platform** | Yes (using a domain-specific language) | No | Yes (proprietary language MapScript, manipulates rasters based on map algebra) | Yes | Yes (CA-based) | Yes (grid-based, using an extended scripting language) |
| **Integrated calibration** | No | No | No | No | No | Yes |
| **Scenario handling** | Implicit[a] | Implicit (experimental frames) | No | No | Yes | Explicit representation, interactive handling |
| **Usability/ Communicability** | GUI available | Detailed GUI, graphical model builder | GUI including graphical model builder | Detailed GUI | Detailed GUI | Detailed GUI |
| **Software availability** | Upon request | Free for non-commercial use | Upon request | Free | Commercial | Free, upon request |

[a]In SELES, a simulation scenario is defined as a complete set of initial state information and the definition of landscape event.

**Expandability**

Although expandability is a general requirement of state-of-the-art software development, it was of specific interest for the development of SITE, as it strongly supports long-term usability of the framework, which is of particular importance in projects with an envisaged duration of 12 to 15 years. One particular benefit from creating expandable software lies in facilitating the integration of new features, which keeps the system open for new developments and allows long-term use of the system. This way, a system is also prepared to effectively handle short-term demands. In the context of the SITE framework, two aspects were of special importance. The first was the established understanding of expandability by means of keeping the basic system open for the integration of further functionality. This goal could mainly be achieved by the adoption of modern software development concepts, such as component design, implementation by object oriented programming and the heavy use of design patterns. The second aspect addressed the interface between generic system implementation and application (i.e. simulation rule sets).

**Summary**

The listed requirements were derived from the demands implied by the application of the framework in interdisciplinary projects. They revealed, that such a land-use modeling framework had to be a generic platform for operating land-use models. The system had to be capable of integrating sub models and of feeding back sub model results for further decision-making. In addition, we identified specific requirements concerning usability and communicability of modeling results. Further requirements were consequence of drawbacks we recognized in existing frameworks, such as the integration of calibration functionality and the explicit representation of scenarios. An analysis of published modeling environments revealed that existing generic approaches do satisfy single requirements, but not their entirety (see Table 3.1). With SITE, we present a holistic approach for a generic framework for spatially explicit land-use modeling. The system was designed to overcome some of the limitations of previous approaches. Major innovations in the field of land-use modeling are the the high degree of integration of components of the land system, the integration of calibration functionality allowing the simultaneous calibration of interacting component models.

## 3.2 System design features

### 3.2.1 Aspects of software quality

As mentioned above, with the SITE land-use modeling framework it is intended to provide a generic platform for performing regional land-use change modeling and simulation tasks. Especially the aspect of generic enforces a number of specific design features on which the implementation of the framework is based.

The task of creating a generic platform for running different land-use change models implies the development of complex software. Handling software complexity is a focus of research of its own (Gamma et al., 1995). One basic method to handle complexity is encapsulation, which means specific tasks are implemented in separate modules. To allow

Table 3.2: Software quality attributes defined by the international standard ISO/IEC 9126

| Attribute | Sub-attributes |
|---|---|
| *Functionality*: A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs. | *Suitability, Accuracy, Interoperability, Compliance, Security* |
| *Reliability*: A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time. | *Maturity, Recoverability, Fault Tolerance* |
| *Usability*: A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users. | *Learnability, Understandability, Operability* |
| *Efficiency*: A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions. | *Time Behavior, Resource Behavior* |
| *Maintainability*: A set of attributes that bear on the effort needed to make specified modifications. | *Stability, Analysability, Changeability, Testability* |
| *Portability*: A set of attributes that bear on the ability of software to be transferred from one environment to another. | *Installability, Replaceability, Adaptability* |

interaction of such modules, a way for communication by defining adequate interfaces has to be established. Thus, by altering functionality in one module, only the module itself is affected. Following this strategy in the process of software development leads to component-oriented design for the entire system and object-oriented design for the implementation of single components.

Modularizing software also is a way to create more robust software and thus contribute to the achievement of a certain level of quality. As a matter of fact, the SITE framework needs to meet a high level of software quality since it is intended to be used over a longer time period and for a larger number of applications. Software quality is defined by the non-functional requirements of the system and is not obvious from the catalog of functional requirements. Important non-functional requirements are, amongst others, changeability, interoperability, efficiency, reliability (error tolerance, robustness), testability and reusability (Buschmann et al., 1998). A comprehensive collection to gauge software quality that should not only be respected during software architecture development but also in all other phases of the development process, is given by the international standard ISO 9126. Table 3.2 gives a short description of the quality criteria defined by ISO 9126.

Apart from a few minor sub attributes listed in ISO 9126, this standard was crucial in the development of the SITE framework. For instance, SITE development is not critical concerning security issues. Maturity, to mention another aspect, must of course be achieved over time and adoption in a number of projects. Other features like usability or changeability can already be found in the list of system requirements. The SITE system is designed to show a high usability both in aspects of user-friendliness by providing a graphical user interface and being able to house a wide range of different modeling tasks due to the establishment of a generic structure.

A central focus lies on the maintainability of the SITE system. The capability to enable a large number of modeling applications implies that the software might be utilized over a longer period in time and thus has to be administered accordingly. It can also be expected that the persons being in charge of the maintenance change over time. Multiple applications will most probably raise the need to change and expand the system. Due to these reasons the design will be characterized by strong modularization of the software, resulting in a component-based implementation and the use of object-oriented analysis.

### 3.2.2 Separation of implementation and application

One main target in the development of the SITE regional land-use modeling framework is to provide a generic platform for implementing regional land-use models of different character and running the respective simulations. This goal is achieved by strictly separating the units dealing with project specific aspects and the units providing functionalitiy for all applications. In the following paragraphs, the unit housing all generic functionality will be referred to as the system domain, while the project-specific unit will be named the application domain (Fig. 3.3). Analogous to that, it will further be distinguished between system developer and application developer. Defining such a separation implies accepting a number of compromises, since modeling projects might be very specific in some details and thus it may not be possible to provide a generic platform for all eventualities that may arise. In consequence, the way of separation is strongly determined by what the parties involved in the design of the SITE system define as relevant to all modeling studies.

Utilizing a system design which strictly distinguishes between system and application domain has a number of advantages that compensate for extra efforts required by implementing the desired generic. Since the system provides elements needed by all applications, the application developer will not have to deal with them, save time and efforts and can focus on his or her specific application. Thus, generic functionality will not be implemented repeatedly and redundancy will be avoided. In addition, multiple applications using the same system will lead to a robust system implementation exhibiting a minimal number of errors and consequently improving its reliability.

A second advantage is the possibility to administrate the implementations of system and application separately. In case there is no separation, making changes to either a system- or application-specific feature means altering the whole implementation. System and application domain disjunction will avoid this and thus contribute to error prevention and improved quality of both system and application.

For a definition of the system domain of the SITE framework, it is necessary to recall the classification of land use models and modeling techniques. The SITE system is designed to support regional land-use models. Besides scale differences concerning the rules and processes underlying land-use models on the continental or global scale, regional land-use models usually also differ in aspects like data volume (which is usually smaller) or the representation of the spatial data. The SITE modeling framework is capable of supporting all regional land-use models following the criteria listed below. These features

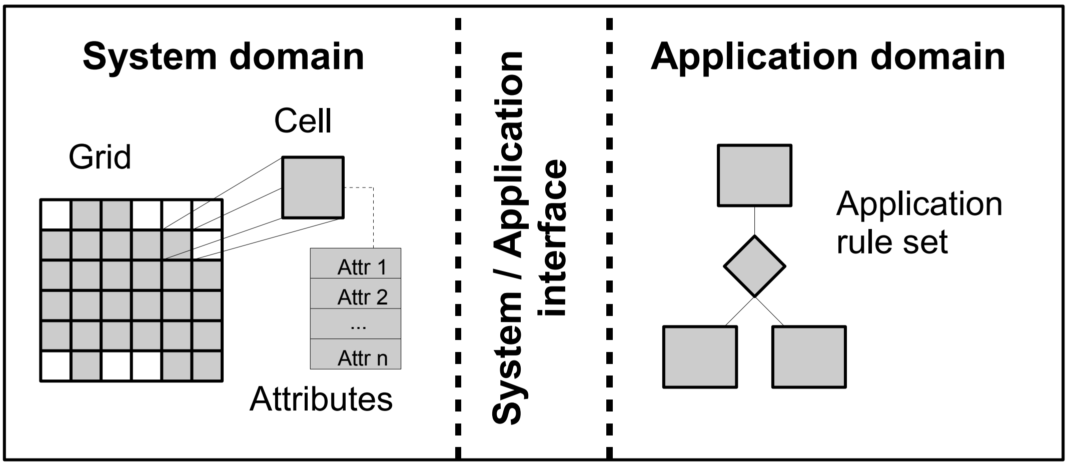


Figure 3.3: Separation of the SITE framework into system domain and application domain. The system domain provides generic data structures that are used and initialized by applications.

and respective data structures are an integral part of the SITE system domain and all functionality for their maintenance is implemented there.

- *Spatial explicitness based on cellular automata:* The way in which spatial data is represented in a model is a basic feature which determines the data structures of the modeling system. One could think of a solution which puts the representation of spatial information in the responsibility of the application domain, but that would result in increasing complexity when implementing the respective application code and hence collide which design features like usability or efficiency. Cellular automata based models are numerous and very prominent in the community, so this confinement proves to be a good compromise.

- *Georeference based on coordinate systems:* This means that each grid cell represents a piece of land of the same size or a size that can be derived by a functional dependency (based on cell position, e.g. dependent on geographic latitude). In its current state, however, the SITE model only supports rectangular coordinate systems (e.g. UTM), but it can be easily extended to support other coordinate systems as well.

There are no restrictions with respect to the spatial resolution of the model, i.e. the value can be freely defined by the application. All other criteria used in the classification of land-use modeling systems can be addressed in the application domain.

Based on this definition of responsibilities the system domain can be configured. As a central data structure, it houses a class representing a two-dimensional grid of application-defined size and resolution plus respective iteration functionality and methods to access each grid position. Every single grid cell needs to be represented by an own instance of a data structure which specifically addresses the problem of handling cell attributes. Analogous to grid size and resolution, the number of attributes, their names and data types are only known at run-time as soon as an application is selected. Hence attribute handling must be implemented in a highly dynamic manner. The definition of attributes must be conducted via the system-application interface.

The grid-cell-attribute complex provides the main data structures needed for modeling and simulation functionality. Besides that, one could think of additional data structures and respective functionalities that are useful for modeling tasks and can be provided by the system domain for use in applications such as the aggregation of cells to clusters based on application-defined rules. Despite of the fact, that such functionalities can be implemented as part of an application, there are cases where an implementation as generic service by the system side proves as useful. The SITE system domain houses such functionality, which will be described and discussed below.

The data structures for the grid, cells and attributes represent the static aspect of the SITE system domain. For simulation dynamics, the system domain also needs to provide respective functionality. Since the SITE system integrates cellular automata (CA) as the fundament for land-use modeling and simulation, it must also provide the basic operations typically performed by CA (Weimar, 1997; White et al., 1997). Three basic CA operations were implemented:

- *Initialization:* This operation is performed upon connecting an application to the system domain. The basic procedures during initialization are (i) to create a grid with size and resolution defined by the application, (ii) create the required number of attributes and (iii) set their initial values. Depending on the application, additional procedures can be carried out during the initialization step.

- *Start simulation step:* This system domain operation signals the application to start executing the code containing the logic for performing a simulation step.

- *Allocation of new attribute values:* In CA, the actual assignment of new attribute values is done after finishing a simulation step, so this task is done automatically by the framework as soon as the respective application code has been executed. Nevertheless, an application might require that attribute value changes become effective immediately, thus the system/application interface comprises a respective method.

Until now, only the term simulation step was used, but no actual time was assigned to that. The definition of simulation time and the appropriate temporal resolution depend on the model in use and have to be defined in the application domain.

### 3.2.3 Technologies applied in SITE

Due to the strict separation of the SITE framework into system domain and application domain, different technologies can be used for the implementation of each module. However, a minimum compliance between both domains is required. Following the strategy of providing generic functionality for a number of yet undefined modeling applications means that the system domain must provide a possibility to load application code and execute it. A number of technical options are available to achieve this aim. In the SITE framework the code of the system domain is written in the C++ language, embedding an interpreter for the Python scripting language. The Python language is used to code modeling applications (Fig. 3.4).

Due to the requirement that the SITE system has to run on standard PC hardware and shall be made available to different research groups, it was developed on the Win32 platform. The entire handling of the software is compliant with the Windows philosophy.

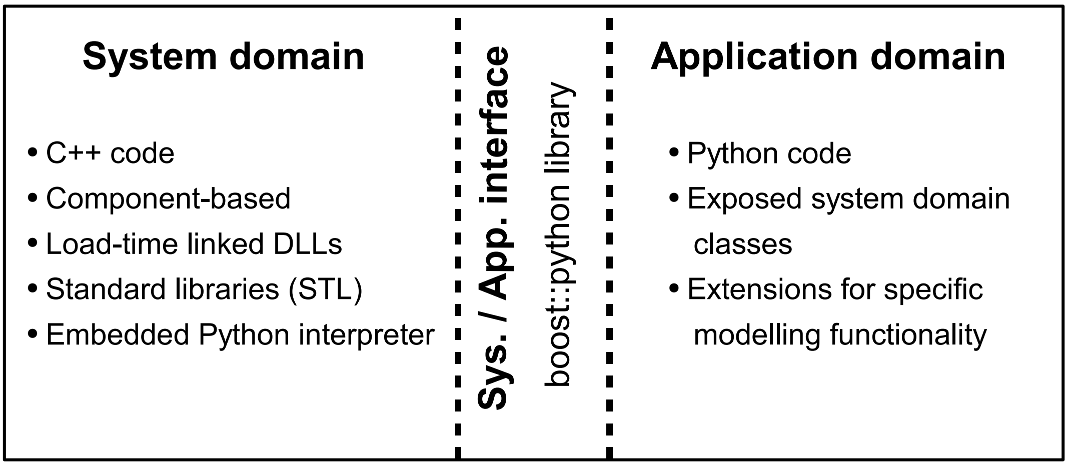


Figure 3.4: Technology used for the implementation of the SITE framework's system and application domains. The system domain is C++ software capable to interpret application code programmed in Python. The interfacing between the two domains is implemented using the Boost Python library.

### System domain

For the implementation of the system domain the C++ programming language was selected. As an object-oriented language it allows a structured implementation ensuring further expandability combined with high efficiency concerning run time and system resources. Although portability (e.g. to a Linux platform) was not a major design goal, the effort necessary for porting the system to another platform is reduced to a minimum by using libraries that are provided for both platforms respectively.

The system domain consists of a number of components with each component being responsible for specific tasks. However, due to portability reasons, no technology like COM was adopted to wrap the component binaries. Instead, each component is built as a Windows dynamic link library (DLL) with a clearly defined interface. This facilitates porting since a similar concept known as shared libraries is applied on Linux platforms. On startup of the system domain, the components are load-time linked and invoked by either a GUI or a command line application.

To enable the execution of Python application code, the system domain embeds an appropriate interpreter and provides all respective functionality like Python language extensions to expose system domain data structures (especially the grid-cell-attribute framework) to the application domain.

The system domain also includes a number of libraries providing basic data structures and functionality. One example is a wrapper library which provides access to the xerces DOM (document object model) implementation which in turn provides an interface to access XML files. The SITE system domain uses XML for configuration files and to save and restore system states.

### Application domain

SITE modeling applications need to be coded in the Python scripting language. It is an excellent tool for the formulation of rules, processes and logic to describe the rules underlying a land-use model. Since it is a full object-oriented programming language, there are no limitations with respect to the complexity of an application rule set. In addition, Python has a very large user community and is also used as a scripting language for commercial software packages (e.g. ESRI's ArcView GIS software).

The Python language can be extended by using its C API. In the SITE framework such extensions have been created to make the data structures for the grid, cells and attributes defined in the system domain available on the application domain. In addition, the implementation of such extensions is especially useful when time critical operations have to be performed. Functionality used on the Python side is in fact carried out by a C or C++ module. The SITE framework provides such extensions for time-critical operations like the calculation of distance maps.

#### System/application domain interface

As described above, generic data structures and functionality are exposed to the application domain via the system/application interface. With its C API, Python already offers a solution to implement this interface. However, being a low-level interface, it is relatively complicated and error-prone to utilize this API. Due to this reason and the fact that Python is widely used, there are a number of libraries which support the exposure of functionality coded in C++ to the Python language. In the SITE framework, the Boost Python library was used. Boost Python is a subset of the Boost library, which itself is a collection of free libraries that extend the functionality of the C++ programming language. Boost Python offers a concise syntax for exposing whole C++ classes and the necessary subset of methods. Especially this property makes it favorable for use in the SITE framework as compared against tools like SWIG (Simplified Wrapper and Interface Generator) where exposing of classes can only be achieved indirectly by a workaround. A detailed description of the SITE system/application interface is provided in appendix A.

## 3.3 System architecture

The objective of software design is to develop an adequate software architecture that meets the predefined requirements. In its Unified Modeling Language Specification, the Object Management Group (OMG) has defined the term *Architecture* as an organized structure and associated behavior of a system which can be decomposed recursively into different parts. These parts interact and include classes, components and subsystems (OMG, 1999). This definition of an architecture considers aspects like the fragmentation of the entire system into multiple components, communication between single components and relations of components among each other. An appropriate definition of software architecture which matches the context of SITE framework development is provided by Endejan (2003), where software architecture is the basic structure of a software system that describes an assemblage of defined components interacting via interfaces. The architecture specifies the components' scope and their relationships among each other. A component is defined as an enclosed binary software module that implements application-oriented and semantically mated functionality that is provided to clients via interfaces (Balzert, 2000).

### 3.3.1 Existing standards and architectures for integrated modeling

As discussed in chapter 1, integrated land-use modeling is a tool of in increasing importance throughout the scientific community. A number of models already exists and consequently, there have already been efforts to bring the different modeling approaches together regarding both modeling concepts and underlying technology.

Beside the International Organization for Standardization (ISO) there are other organizations which have issued recommendations or standards for the technical realization of integrated modeling systems that are also relevant for land-use modeling. Among those, the Institute of Electrical and Electronic Engineers (IEEE), the World Wide Web Consortium (W3C) and the Open Geospatial Consortium (OGC, the former OpenGIS Consortium) are the most relevant ones.

#### High Level Architecture (HLA)

The High Level Architecture is an architecture to combine interacting sub models to aggregated models pursuing the target to significantly increase the interoperability of simulation models (Kuhl et al., 1999). It was originally developed for military applications but is increasingly adopted in the civil domain (Schulze et al., 1999; Lindenschmidt et al., 2005). In the year 2000 it became an IEEE standard. Since HLA is a generic architecture it only provides functionality to increase the interoperability of simulation models. This functionality is encapsulated in the HLA run-time infrastructure (RTI). There are several commercial implementations of the RTI available.

#### NIST/ECMA reference model

Having been developed as an architecture to integrate different applications in the context of computer aided software engineering (CASE), the NIST/ECMA reference model provides an extendable framework to establish communication among single applications and a consistent graphical user interface for data representation. The problem of the consistent integration of data and software that led to the development of this architecture is comparable to those arising when setting up an integrated modeling framework. Chen and Norman (1992) provide further information on the NIST/ECMA reference model. The NIST/ECMA reference model was one basis for the development of the reference architecture issued by the Open Geospatial Consortium.

#### Open distributed processing reference model

Since single components of an integrated modeling system do not necessarily have to run on the same machine, one has to consider the possible distributed character of the system. A distributed system makes special demand to the underlying software architecture, therefore ISO created a framework to facilitate and encourage the creation of standards for such distributed systems and published it as the standard ISO/IEC 10746-(1 to 4). Four basic elements are postulated for standardization: System description using object-oriented analysis, system description via five separate but related viewpoints (enterprise viewpoint, computational viewpoint, information viewpoint, engineering viewpoint and technology viewpoint), the definition of a system-infrastructure to ensure transparency regarding the distribution of applications and finally a framework to assert that the system is compliant to the respective ISO standard. An application of the open distributed processing reference model is the OpenGIS service architecture introduced below. Further information is provided by Farooqui et al. (1995) and Schürmann (1995).

### OpenGIS service architecture

The OpenGIS service architecture (Percivall, 2002), issued by the Open Geospatial Consortium, is a technical reference model. It has been taken over by the International Organization for Standardization as standard ISO 19119 in April 2001. It assumes that underlying target systems are distributed and implemented using object-oriented analysis. It provides a taxonomy for geographic services and regulates how platform-independent specification for services have to be created and how to derive respective platform-dependent specifications. The goals pursued by the OpenGIS service architecture standard are to

- provide an abstract framework, that allows the development of specific services,
- enable interoperable services by standardization of interfaces,
- support the development of service catalogs through the definition of meta data of services,
- enable the separation of specific data and services,
- allow the use of services from one provider to work on data of another provider,
- define an abstract framework, that can be implemented in different manners.

The OpenGIS service architecture refers to the Open Distributed Processing Reference Model by adopting four of the five viewpoints defined there. Viewpoints considered are the computational, information, engineering and technological viewpoint. The enterprise viewpoint is described in other parts of the ISO 19100 series of standards (e.g. in the ISO 19101 reference model). For a detailed description and discussion of this and the above introduced architectures, see Endejan (2003).

### SISA architecture

Based on a review of existing architectures for integrated modeling, Endejan (2003) developed an architecture for a system for integrated simulation-based assessment (SISA). He defines a system for integrated simulation-based assessment as a software system that combines both data and simulation models from different disciplines dealing with the "system earth" in a consistent frame and that computes and provides new data describing state and possible long-term changes of the "system earth". This is basically done to support policy-makers. Referring to the quality of assessment results the consistent frame is considered being of special importance since it contributes to the transparency and comprehensibility of results.

Figure 3.5 gives an overview over the SISA architecture. Not considering a component implementing the client side of whatever kind (e.g. command line or GUI), the architecture features twelve different components.

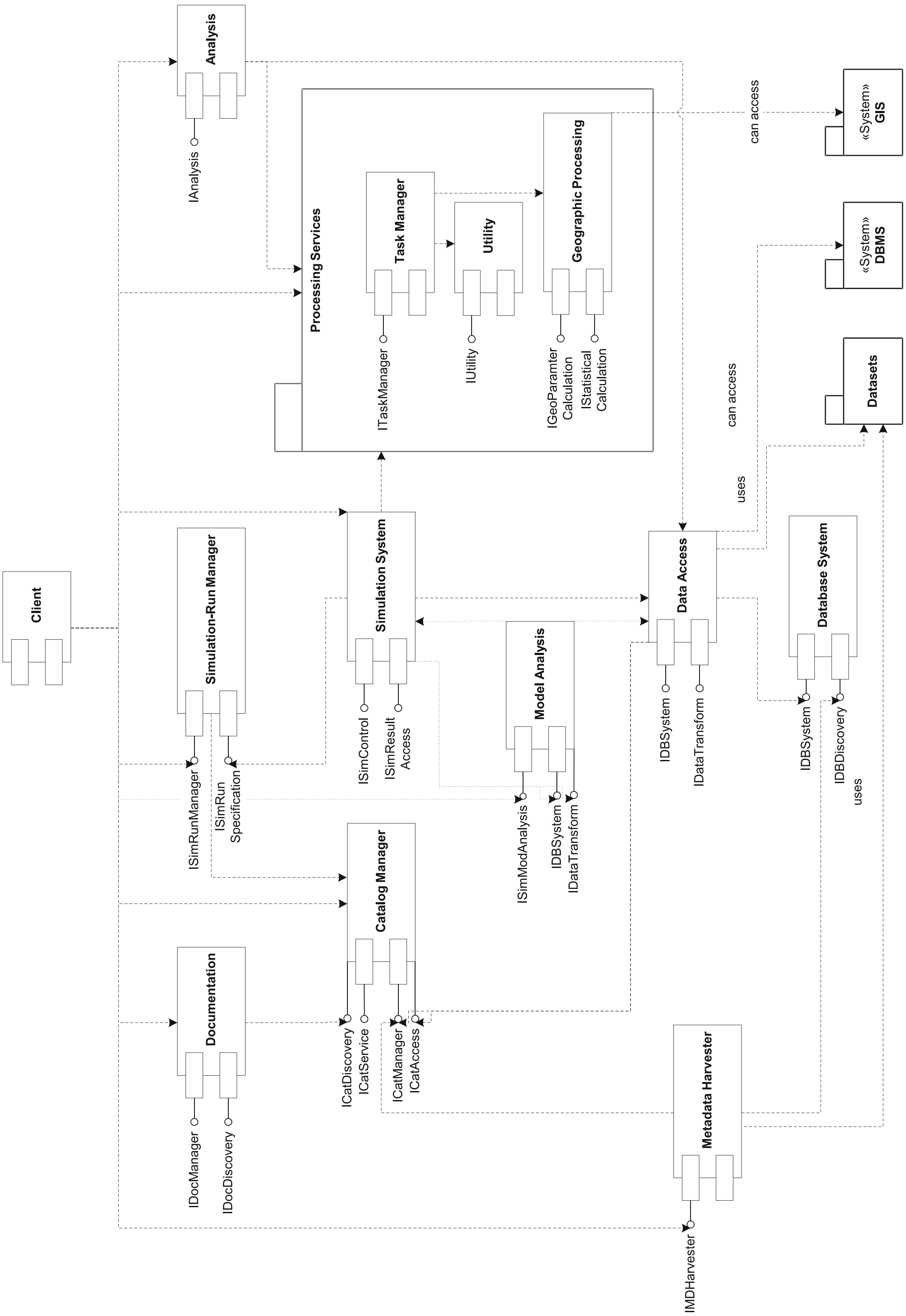


Figure 3.5: Architecture of a system for integrated simulation-based assessment (SISA) as proposed by Endejan (2003). Arrows referring to complete components instead of single interfaces denote dependency to all interfaces of the respective component.

**Simulation System** The simulation system component is the central component. It is responsible for computation, storage and propagation of simulation results. For the propagation of results it features a specific interface which ensures interoperability and reusability of the simulation system.

**Simulation-Run Manager** The Simulation-Run Manager ensures comprehensibility and reproduceability of simulation results. Its responsibility lie in both managing the specification of simulation runs and propagating them to the actual simulation system.

**Data Access** This component contributes to data integration and the allocation of simulation results. Services to transform data to make them consistent with the data format required by the simulation system are also assigned to it. Thus, the data access component supports requirements such as interoperability and ex changeability.

**Database System** The actual data used during simulation runs are housed in the database system component which encapsulates database functionality. Beside having an interface providing common database access operations, it features a separate interface for performing queries. Nevertheless, data are not accessed directly by the simulation system, this is done via the data access component which serves as an integration layer and performes necessary data transformations.

**Catalog Manager** One function of a system for integrated simulation-based assessment is the provision of meta data describing data sets used for simulation. Since indexing of resource meta data can be seen as an independently functionality inside a SISA, this functionality is implemented as a separate component responsible for the management and delivery of meta data concerning SISA resources.

**Metadata Harvester** There is a conflict between centrally managing meta data and storing them locally. The first option is favorable for the integration inside a SISA while the latter one can be advantageous in other aspects since it is sensible to store meta data at the same location as the data they describe. The implementation of a meta data harvester can solve this conflict. The harvester is a program that automatically searches the a defined file system for the desired meta data and thus can simulate a central storage for the catalog manager component.

**Documentation** While management of meta data lies in the responsibility of the catalog manager, the documentation component administrates all information about executed simulation runs and underlying scenarios. In addition it gives model users information about the handling of the system.

**Utility** This is a component reserved for any kind of data processing that can be realized independent from the simulation system.

**Task Manager** The single components in the SISA provide reusable operations which naturally can be used by all other components inside the framework. To facilitate the use of these services, functionality for the claim and control of services should be provided. Thus, the task manager component is responsible for the program controlled invocation of other SISA services.

**Geographic Processing** This component provides services for geographic data processing and can encapsulate an existing GIS system ir implement GIS functionality itself.

**Model Analysis** The responsibility of this component is to manage procedures to analyze model behavior like sensitivity or uncertainty analysis. Usually this is done by altering specific model parameters and evaluate their influence on the simulation result. Respective functionality is implemented here.

**Analysis** Set up upon data processing services and services provided by the data access component the analysis component is responsible for supporting the model user in the analysis of simulation results.

Realization of the described SISA architecture can be achieved using simple technical tools and free software as has been proven by Endejan (2003) who set up a SISA to run the GLASS (Global Assessment of Security) integrated model (Alcamo et al., 2001). Especially the meta data framework and the simulation run manager component contribute significantly to increased transparency and reproducibility of simulation runs. Applying the simulation system interfaces to sub models leads to improved reusability and interoperability among sub models. In addition, the information provided by the documentation component has proven useful for a transparent assessment.

### 3.3.2 SITE system domain architecture

The SISA architecture provides a well suited template for the development of the SITE architecture. Many features of SISA match well with the requirements listed for the SITE framework. However, the SITE architecture has to be a more compact solution due to requirements concerning ease of use or ease of distribution. In particular, this means that SITE components are more closely coupled than their SISA counterparts. Nevertheless, loose coupling of components like in a SISA is not required due to the basic design feature of the SITE system, which is its separation into a system and application domain. For applications, there is only a minimum of constraints and thus it is imaginable to implement a SISA-based architecture in the SITE application domain. In contrast to SITE, the simulation system component inside the SISA architecture already represents an application.

Figure 3.6 shows the architecture of the SITE system domain. Four components are defined to house all necessary functionality and to meet the defined requirements. In addition, two client components are specified. For model development, performing simulation runs and presentation tasks, a graphical user interface is provided that features a variety of possibilities to work with an application interactively. For elaborate tasks like performing multiple simulations or a calibration run, a command line client is provided which enables the use of SITE inside a batch processing framework. A detailed description of the client components is given in section 3.4.3.

The components *SmltnEnvironment* (simulation environment) and *SmltnDynamics* (simulation dynamics) are the central building blocks of the SITE system domain and together will be referred to as the system core engine in the following. Component *SmltnEn-*

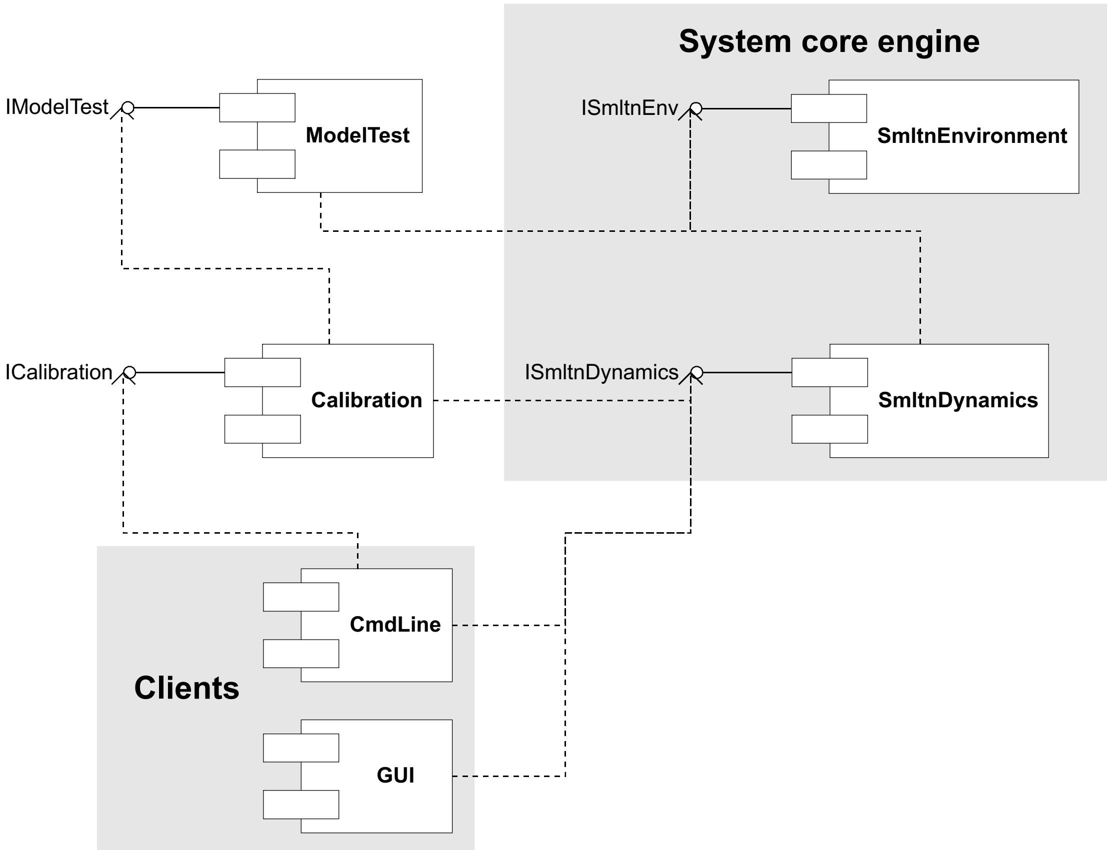


Figure 3.6: Architecture of the SITE system domain. Each displayed component represents a binary entity. While the client components are executable files, the functional components are realized as DLLs and are linked to the GUI ore CmdLine component respectively.

*vironment* is responsible for managing all static aspects of the simulation. It provides all data structures necessary for representing spatially explicit modeling data. In particular, these are classes that make grid, grid cell and cell attribute functionality available for use in the application domain. Other classes support the application-defined organization of grid cells into layers determined by attribute values. For the grid class adequate iterators for different purposes (e.g. entire grid iteration, attribute layer-specific iteration, moving window iteration) are supplied. The component exposes its functionality via the *ISmltnEnv* interface.

As can be assumed by its name, the *SmtlnDynamics* component, to be utilized via the *ISmltnDynamics* interface, implements all functionality dealing with dynamic aspects of a simulation. On one side, this is a framework for the basic cycle of cellular automata operations (initialization, simulation step, attribute allocation). On the other side, since change rules are project-specific and thus are integral part of an application, the *SmltnDynamics* component is the instance where the system/application interface (which is technically the connection of the SITE system C++ part to the Python scripts) is implemented. In addition to the interfacing technology the component features extensions to the Python language that have been found useful and can be expanded respectively. The *SmltnDynamics* component operates on the data structures provided by *SmltnEnvironment* using the *ISmltnEnv* interface.

In addition to the basic simulation functionality, the system domain provides other generic services supporting quality aspects of modeling applications. Functionality to assess the quality of simulation results through a number of map comparison algorithms that work on categorical data is implemented in the *ModelTest* component. It is designed for simple expandability and its functionality is exposed via the *IModelTest* interface. The featured map comparison algorithms work on grid cell attributes representing the classification and thus operate on the data structures implemented in the *SmltnEnvironment* component via its *ISmltnEnv* interface.

The *Calibration* component provides methodology for automated calibration of application rule sets. Basically, it is a collection of algorithms that are capable of finding optimal or adequate solutions for an application-defined parameter set with respect to an objective function (currently it only includes the implementation of genetic algorithms). The implementation is analogous to the implementation of the *ModelTest* component and allows simple integration of new algorithms. The evaluation of candidate solutions created by an optimization algorithm is based on map comparison algorithms provided by the *ModelTest* component. Therefore there is a dependency of component Calibration from component *ModelTest* which is accessed via its *IModelTest* interface. Due to the fact that performing model calibration using an optimization algorithm means running multiple simulations (one for each candidate parameter set), the *Calibration* component must be enabled to repeatedly start simulation runs with altered parameter sets until an application-defined termination criterion is met. To ensure this, a dependency of the *Calibration* component to the *SmltnDynamics* component via its *ISmltnDynamics* interface is defined. For details on the *Calibration* component see section 3.4.2.

As has already been noted, all SITE system domain components are either compiled into dynamic link libraries (DLLs) or executable files. No higher level technology like COM or .Net for the component definition has been used, since this is not imperative due to all components being implemented in the C++ language and their rather close coupling. Each component specifies exactly one interface. Exchangeability of components on the binary level is given provided that interfaces are identical.

## 3.4 Implementation of the SITE system domain components

In this section all components that are part of the SITE system domain architecture introduced in section 3.3.2 will be described in detail especially regarding their static structure, dynamic behavior and they way they are interfaced among each other. In class diagrams, not all methods and class members will be displayed. Instead, there will be a focus on methods contributing to component interfacing and to exposure of functionality to the application domain.

### 3.4.1 System core engine

#### Simulation environment

The *SmltnEnvironment* component contains and provides all data structures that are required to run cellular automata-based spatially explicit land use change simulations. It can be seen as the main component of the SITE framework since it implements the largest share of generic modeling functionality. As a consequence, it is also the most complex

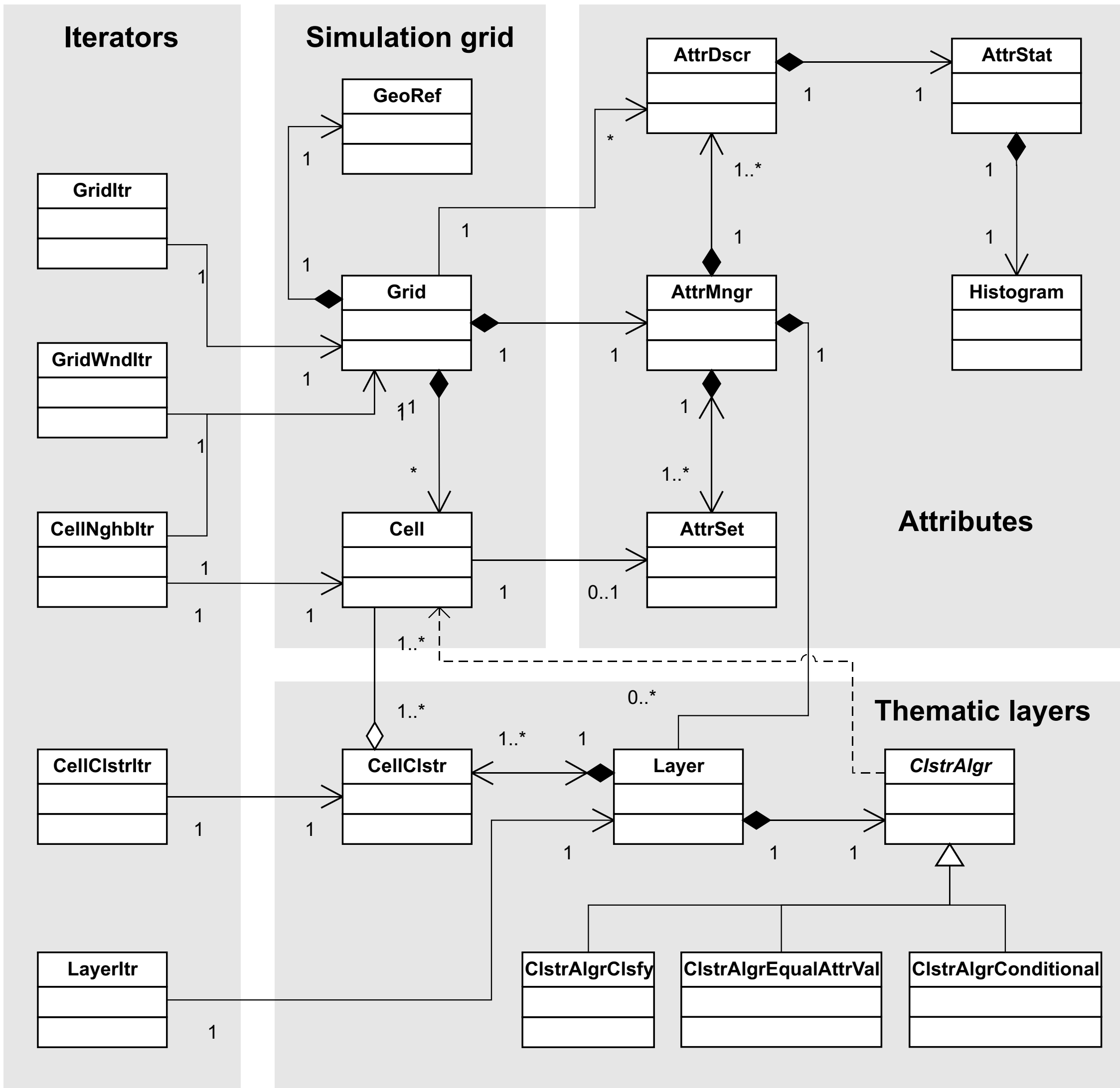


Figure 3.7: Class diagram showing the overall static structure and class relations of the *SmltnEnvironment* component. For clarity reasons class methods and data members are not displayed.

component of the SITE system, both regarding the number of implemented classes and the relations among classes. Most other components depend directly or indirectly from *SmltnEnvironment*.

The overall class layout is displayed in figure 3.7. Inside the *SmltnEnvironment* component four functional parts can be identified: The actual simulation grid, a framework for managing and handling of attribute data and data structures for the representation of information layers determined by cell attribute values. The fourth part contains different iterators that enable the operation on the main functional classes and provide adequate access to them.

Since the main characteristic of applications designed to be operated by the SITE framework is the spatial explicitness represented by a cellular automata approach, the central functional part is the cluster of classes referred to as the simulation grid, with the *Grid* class being the fundamental data structure. Figure 3.8 displays the static structure

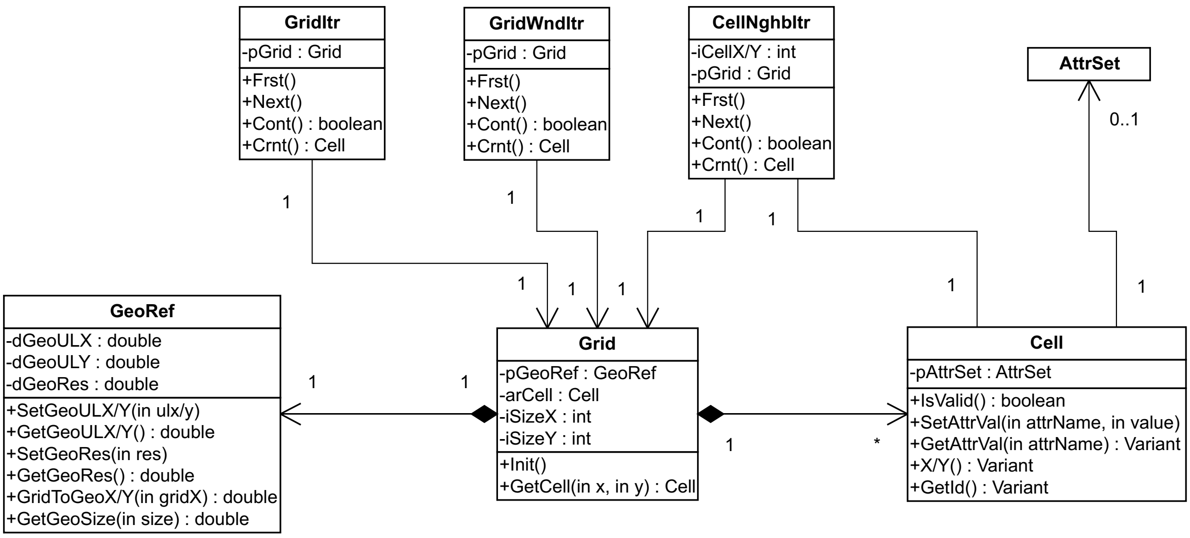


Figure 3.8: Detailed class layout of the SITE simulation grid. Class methods and members involved in the grid representation are displayed.

of this part in more detail together with class methods and class members involved. The *Grid* class acts as entry point to all *SmltnEnvironment* objects, implements a two-dimensional rectangular array of cells and defines a number of methods to enable access to grid information and single cells. The size of the simulation grid, its georeference and spatial resolution are dynamic and defined by an actual application. Georeferencing is handled in class *GeoRef* which is owned by *Grid*. Class *GeoRef* provides a small number of methods to specify georeference and to perform simple scaling calculations. In the current version, georeferencing must be based on UTM (Universal Transverse Mercator; see Snyder, 1987) coordinates. The single grid positions are instances of the class *Cell* aggregated in the *Grid* class. *Cell* instances basically contain a reference to an instance of class *AttrSet* (attribute set) describing the overall cell state and a number of methods to provide access to attribute values.

Regions examined in land use change simulations typically do not have the shape of a rectangle. The two-dimensional cell array implemented in the *Grid* class therefore represents the smallest possible bounding box for the application region. Consequently, not all grid cells are part of the project regions and have to be marked as invalid. In SITE framework, only valid cells have a reference (realized by a pointer to class *AttrSet*) to an attribute set. This is the criterion checked in the *Cell.IsValid()* method. There is a 1 $\rightarrow$ 0..1 relation between the classes *Cell* and *AttrSet* and no attribute values are managed for invalid cells. As pictured in figure 3.8, the *Cell* class defines methods to directly access a cell's x and y coordinate and its cell ID (*X()*, *Y()*, *GetId()*). These three criteria are internally handled as normal attributes and could also be accessed using the method *GetAttrVal(attrName)*, but since they represent integral information for every cellular automata-based simulation, these methods are hard coded and respective attributes in applications must be named accordingly (*OBJECTID*, *x*, *y*)). Attribute values are returned using a *Variant* data type.

Although class *Grid* provides direct access to single cells by specifying cell coordinates, this is not the usual procedure when analyzing the grid and applying land use change

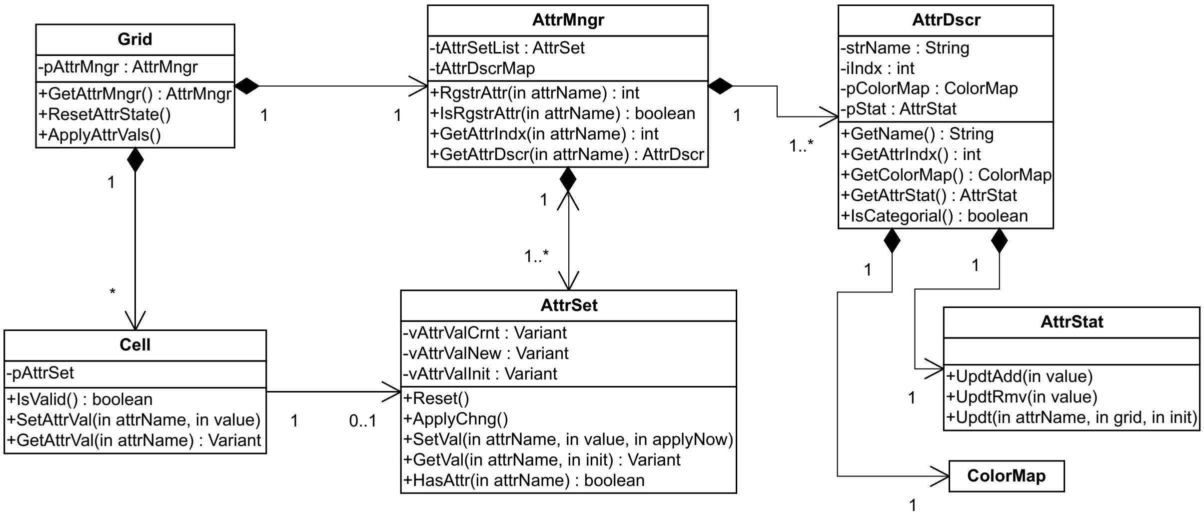


Figure 3.9: Detailed class layout of the SITE attribute framework. Classes *Grid* and *Cell* are also shown since they serve as entry points to the attribute framework; for those classes only methods and members dealing with handling of attributes are displayed.

rules. Typically, only valid cells are of interest in such an analysis and each cell has to be accessed. Therefore a safe way to traverse all valid cells of the grid has to be established. This is done by implementing iterator classes (applying the iterator design pattern; see Gamma et al., 1995) that can be configured to traverse the grid in a way required by the application. Three different iterators are offered. The *GridItr* class encapsulates functionality to iterate the entire grid. By default, invalid cells are ignored. This iterator can be configured to only return cells that serve as start cells for iterations handled by the second iterator, *GridWndItr*. This iterator traverses the cells of a rectangular grid subset while the subset size is defined by the current application. Combining these first two iterators, a moving window iteration is possible. This service is used in the *Validation* component for map comparison algorithms that utilize such moving windows. The third iterator, *CellNghbItr*, traverses all neighbor cells of a given center cell (specified by its grid coordinates). It can be configured to apply Von-Neumann (4-pixel) or Moore (8-pixel) neighborhood. Extended Moore neighborhood is currently not implemented but the system can be easily extended.

All iterator classes implement the same interface which consists of the methods *Frst()* (set iterator to first element of underlying object collection), *Next()* (step iterator to next element), *Cont()* (check if current element is valid or if there is a current element at all) and *Crnt()* (return the current element of whatever type). Using this interface, iterators can easily be employed in C++ *for* and *while* loops. Depending on the purpose of the iterator and the complexity of the underlying object set, there are additional methods for configuration. All iterator classes are available globally throughout the SITE system and iterator objects are instantiated when needed. They require a reference to the underlying object set and establish a directed association of the multiplicity $1 \rightarrow 1$.

The attribute framework of the SITE *SmltnEnvironment* component is designed to enable generic handling of cell attributes while causing a minimum number of restrictions for applications. The only restriction so far is that an application must specify three attributes representing describing the cell ID and a cell's x and y-coordinate on the sim-

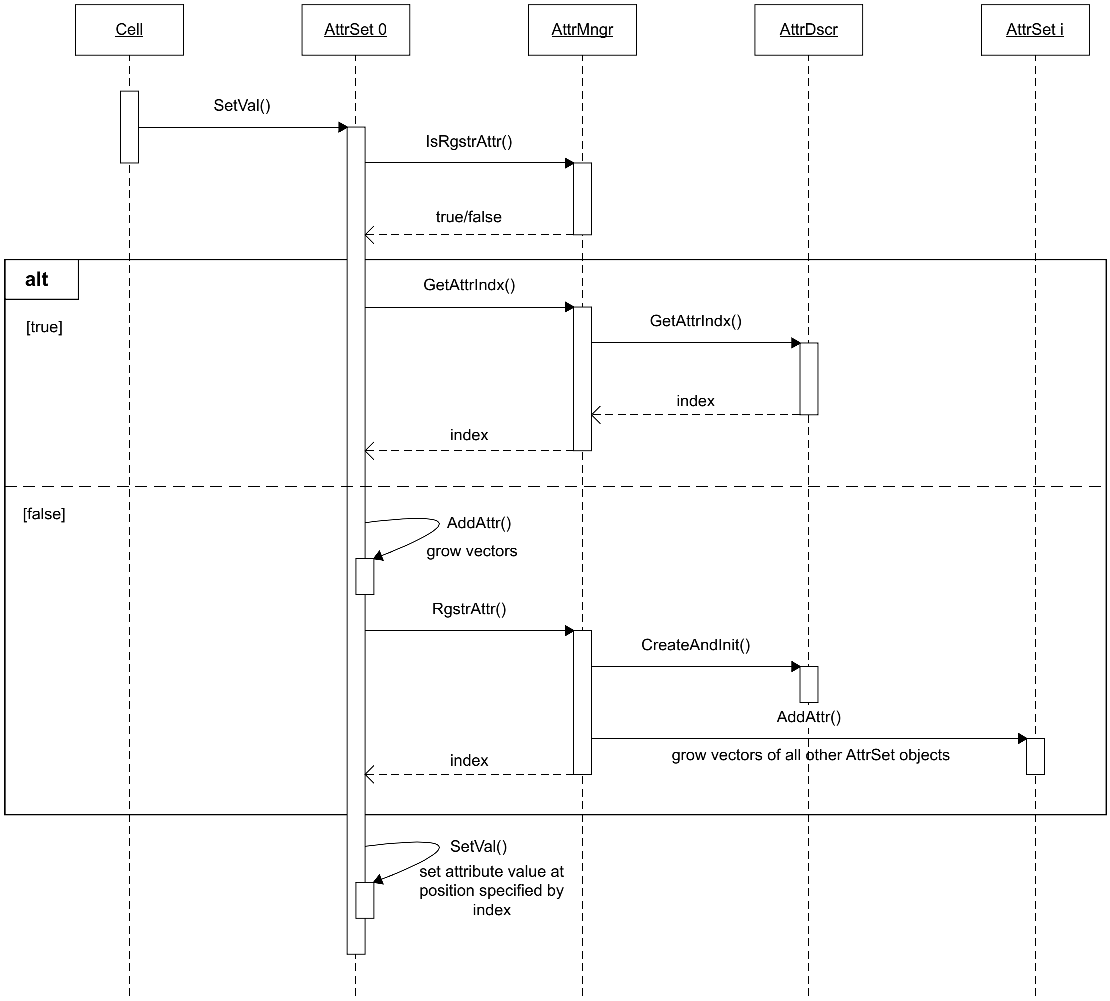


Figure 3.10: Dynamics implemented in the SITE attribute framework. When accessing an attribute (in htis case to set an attribute value), it is first checked, whether the attribute is registered or not. For registered attributes, the *AttrMngr* object derives the index of the attribute values by accessing the attribute meta data object of type *AttrDscr*. For yet unregistered attributes, a new meta data object is created and an index is assigned. In addition, the data vectors managed by *AttrSet* objects must be enlarged.

ulation grid. Figure 3.9 depicts the class layout of the attribute framework. It shows a straightforward implementation approach by associating an attribute set object (instance of class *AttrSet*) to each valid grid cell object to retain the attribute values for the respective cell, thus establishing a $1 \rightarrow 0..1$ relationship between the classes *Cell* and *AttrSet*. To allow easy handling of attributes on the application side, it is reasonable to address specific attributes by their names, i.e. by using an ID of string type. However, using a map for associating attribute names to attribute values inside the multitude of *AttrSet* objects is not advisable as attribute names would have to be stored for each instance of *AttrSet* which results in high redundancy and an increased consumption of memory. This is even more problematic when meta information for each attribute has to be managed. A more efficient way to store attribute values, both regarding memory usage and access time, is to use a vector as container for attribute values. Attribute values thus have to be addressed by an index. In the SITE attribute framework implementation, management of attribute data lies in the responsibility of the three classes *AttrMngr* (attribute manager), *AttrSet* (attribute set, attribute values for one cell) and *AttrDscr* (attribute descriptor, attribute meta data).

The grid object delegates management of attributes and attribute data to an instance of class *AttrMngr*. The attribute manager is the central object of the attribute framework. It aggregates all *AttrSet* and *AttrDscr* instances, handles adding of new attributes (also during simulation runs) and mediates between *AttrSet* and *AttrDscr* objects. Each attribute has to be registered by the *AttrMngr* instance. On registration of an attribute, specified by the attribute name, an *AttrDscr* object is created to maintain meta information for the attribute. To store *AttrDscr* objects the attribute manager uses a map that associates the attribute name with the actual descriptor object. On creation of a new attribute, an index is assigned to it by which respective cell attribute values can be accessed from the data vectors encapsulated in *AttrSet* objects. This index is stored as part of the attribute meta data. Meta data maintained by the descriptor objects additionally includes the reference to an attribute-specific statistics object providing basic descriptive statistics and to a color map object for categorially-scaled attributes (see figure 3.10 for attribute framework dynamics).

To access an attribute value for a cell, the respective member function (*GetAttrVal()*) is called using the attribute name. This request is forwarded to the *AttrSet* object. The attribute set object retrieves the index of the specified attribute from the attribute manager and accesses and the desired attribute value. The *AttrSet* class encapsulates three different data vectors. The first vector (*vAttrValCrnt*) contains the attribute values at the current point in time of the simulation. In cellular automata, new attribute values are not necessarily applied directly; this is done after finishing the respective simulation step. Attribute values to be allocated after finishing a simulation step are stored in vector *vAttrValNew*. To replace the current values with the marked new values, the SITE framework calls the *ApplyChg()* (apply change) method for each *AttrSet* instance at the end of a simulation step. Delaying attribute value allocation to the end of a simulation step is not mandatory. To specify whether to directly allocate a value or not, the *AttrSet::SetVal()* method provides the *applyNow* flag. The third vector (*vAttrValInit*) is used to store initial attribute values (value of creation time). This functionality is desirable for use in application rule sets, for map comparisons and for the possibility to reset a simulation to its initial state. If an attribute is not registered but requested by the application, the attribute manager creates a new attribute, respective meta information and access index, and enlarges the data vectors for all attribute sets.

Simulation grid and attribute framework represent the basic generic functionality that has to be provided for cellular automata-based simulation systems. Technically, it is possible to implement all other functionality in the application domain. However, there are other features that can be seen as fundamental to a great extend so that it is reasonable to provide respective structures by the system domain. Throughout the development of the STORMA application based on the SITE framework (see chapter 4) it became obvious that it is very useful to aggregate grid cells to cell clusters that are themselves organized in thematic layers. For the clustering cells, it is possible to specify similarity measures (cluster algorithms). Based on such a clustering it is possible to e.g. aggregate grid cells that belong to the same spatial unit (determined by an identical attribute value representing

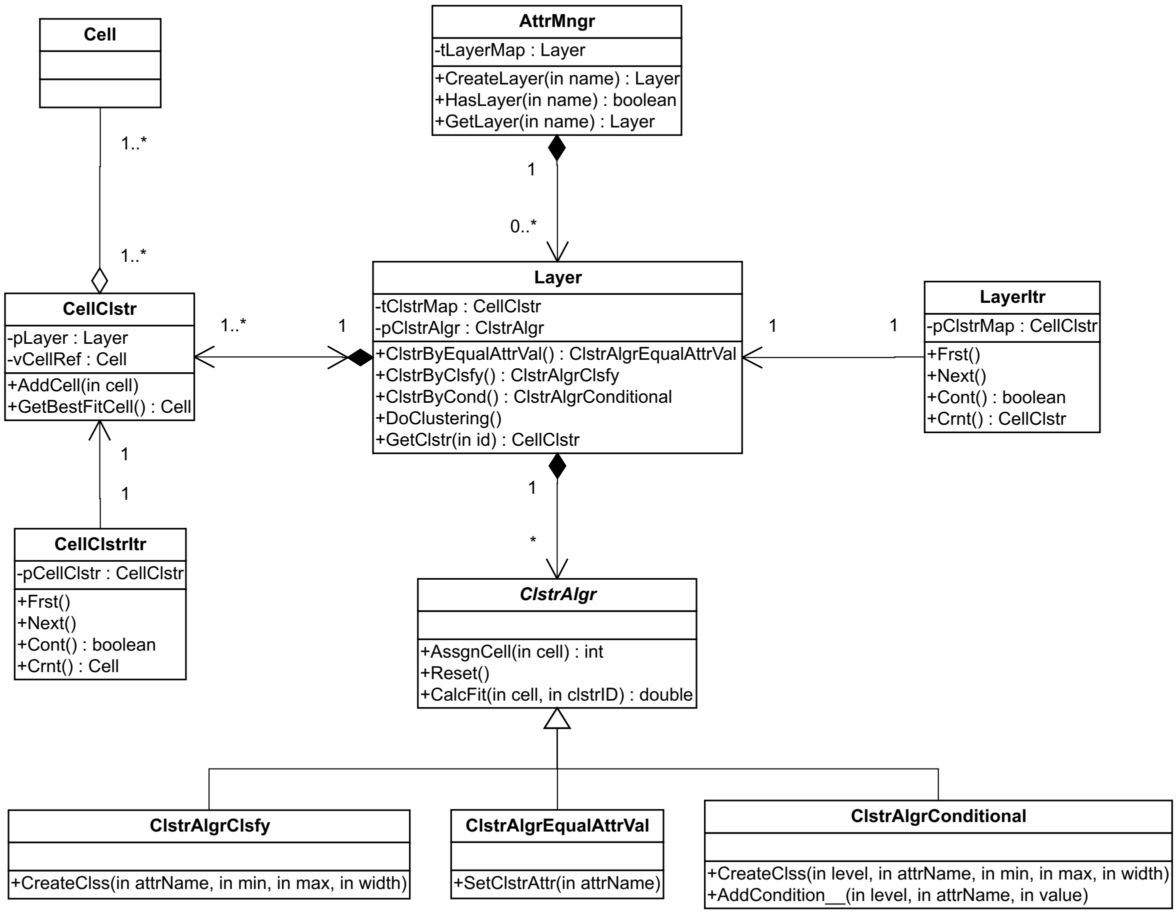


Figure 3.11: Detailed class layout of the SITE thematic layers implementation together with the three available clustering algorithms.

the ID of the respective spatial unit) enabling an application to specifically analyze single spatial units. Another important application is the aggregation of similar cells resulting in a significant reduction of the amount of data for time consuming processing by only performing computations for representative cells.

Figure 3.11 depicts the implementation of thematic layers. Since a thematic layer and the cell clusters it consists of is determined by a selection of attributes and respective attribute values, objects representing thematic layers are composed in the *AttrMngr* instance. The attribute manager object establishes a $1 \rightarrow 0..n$ relationship resulting in an arbitrary number of layers that can be instantiated by an application. In addition, it implements an interface to create and access thematic layers; this interface is exposed to the application side (see appendix A for a complete description of the system/application domain interface). On creation of a thematic layer, a unique string ID (layer name) has to be specified by the application.

An instance of class *Layer* is the representative for an actual thematic layer. It consists of a collection of cell clusters (instances of class *CellClstr*) and a specification of how to assign single grid cells to cell clusters. This specification is provided by a clustering algorithm object (instance of class *ClstrAlgr*). The Layer interface is exposed to the application domain and consists of methods to select the clustering algorithm, to start the actual clustering and to access single cell clusters after their creation. The process of the actual assignment of grid cells to cell clusters (the actual clustering) is decoupled from the specification of the underlying clustering algorithm. This is necessary to enable the recalculation of a thematic layer for the case that attribute values for grid cells have changed after a simulation step.

The *SmltnEnvironment* component implements three different clustering algorithms each represented by a respective class. The three algorithm classes are derived from the base class ClstrAlgr which declares an interface for assigning cells to clusters, resetting the clustering process and calculating how good a given cell matches a cell cluster. Method *AssgnCell()* takes a cell as argument and return the ID of the cluster to which the cell has been added. Method *CalcFit()* takes a cell and a cluster ID and returns a value between 0 (not in cluster) and 1 (ideal fit) describing how good the passed cell represents the cluster. The public interfaces of the algorithm classes themselves consist of methods to parameterize the clustering process and are exposed to the application domain.

The simplest algorithm, *ClstrAlgrEqualAttrVal*, is configured by an attribute name; for this attribute it reads the value for all cells and assigns cells with equal values for the specified attribute to respective clusters. Using this algorithm, it is possible to e.g. create a thematic layer representing administrative units like districts or villages (provided that cells have attributes like district ID or village ID). Due to the nature of clustering, each cell assigned to a cluster is a perfect representative.

The second clustering algorithm, *ClstrAlgrClsfy*, executes cell assignment through classification. Using the *CreateClss()* method, an application can specify value intervals for an arbitrary attribute, thus defining classes. Other attributes value intervals can be added by repeatedly calling this method. Two cells are assigned to the same cluster if they fall in the same value interval for all specified attributes. How well a cell fits to a cluster depends on how its respective attribute values lie relative to their intervals. A value close to an interval border leads to a decreased fitness value.

The third clustering algorithm (*ClstrAlgrConditional*) is an extension of *ClstrAlgrClsfy*. Based on an initial classification an application can establish further evaluation for specific outcomes of the initial classification by defining conditions to be evaluated for those results. To formulate such conditions it provides a set of methods called *AddCondition()* (equality/inequality, larger than, less than) in addition to the *CreateClss()* method, each implementing a comparison to an application-defined value determining the decision tree path. For the ongoing tree paths below the conditional nodes, further classification of values can be defined. Decision trees can be of arbitrary depth. In contrast to class *ClstrAlgrClsfy*, methods require the current level in the decision tree as argument. This algorithm is especially useful if clustering has to be specific to categories (e.g. land use classes).

Each thematic layer manages at least one cell cluster (composition of multiplicity $1 \rightarrow 1..n$). Each cluster can be identified by a unique ID assigned by the clustering algorithm during its creation. The *CellClstr* class defines methods to add new cells and to access the cell which is the best representative of the respective cluster. Each *CellClstr* instance hold references to all grid cells assigned to it. Each single grid cell, in turn, can be assigned to only one cluster for each thematic layers but to more than one cluster in different layers. Cell clusters can be accessed directly by their ID or via an instance of *LayerItr*, an iterator that traverses all *CellClstr* object of a thematic layer. Access to cells aggregated by *CellClstr* instances is provided by a specific iterator (*CellClstrItr*).

### Simulation dynamics

The handling of all dynamic aspects of a simulation lies exclusively in the responsibility of the *SmltnDynamics* (simulation dynamics) component. It operates on the data structures provided by component *SmltnEnvironment*, hence the dependency of *SmltnDynamics* from *SmltnEnvironment* defined in the overall layout of components. Dynamic aspects include performing actual simulation steps and providing scenario data as well as the storage and export of simulation data for single time steps. In the implementation of the *SmltnDynamics* component three major sections can be identified: a central instance representing the entire application rule set functionality inside the system domain, a framework for handling the different kinds of dynamic information and functionality for interfacing the application rule set implementation with the system domain, i.e. the integration of the Python interpreter.

Figure 3.12 depicts the class layout of the SITE *SmltnDynamics* component. An application inside the SITE framework is represented by a single instance of class *RuleSet*. This object provides access to the underlying data structures implemented in the *SmltnEnvironment* component, to all other objects managing dynamic information and to objects representing the integration of the Python scripting language. As the representative of an application's simulation logic, it also provides the necessary methods to control simulation runs. The *Load()* method is used to import a user-specified rule set script into the framework. The initialization of the rule set script is done explicitly using the *Init()* method. An imported rule set script can be executed repeatedly through the use of a reset mechanism invoked by method *Reset()*. An arbitrary number of simulation steps can be performed through the *DoSmltnStep()* method which accepts the number of simulation steps as an argument.

The integration of the Python scripting language together with the establishment of information exchange between the Python (application domain) and C++ (system domain) side is delegated to a set objects of classes encapsulating both the low level Python C API and the *boost::python* library. The central class in this context is *PythonEmbd* (Python embedding) which is instantiated once. The rule set object holds and manages a reference to this instance. The python embedding class contains the Python interpreter which is initialized and terminated by the class methods *StartUp()* and *ShutDown()*. A SITE rule set script must contain two functions called *Initialize()* and *SimulationStep()* featuring initialization logic and simulation rules for one time step respectively. To run a simulation, the SITE system domain calls these functions on the application side through the *PythonEmbd* methods *CallFnctInitialize()* and *CallFnctSmltnStep()*.

In addition to the invocation of these basic functions, the Python integration framework establishes the exchange of information between the SITE system and application domain. In the subject of information exchange, one has to distinguish between two different types: The manipulation of the SITE system domain data structures (e.g. grid cell attributes) from the Python side and the manipulation of variables in the application rule set script by the SITE system domain (e.g. parameter sets for rule set calibration). To allow accessing of data structures defined in C++, the respective C++ classes need to be exposed to Python, i.e. the Python language has to be extended in a way that data structures

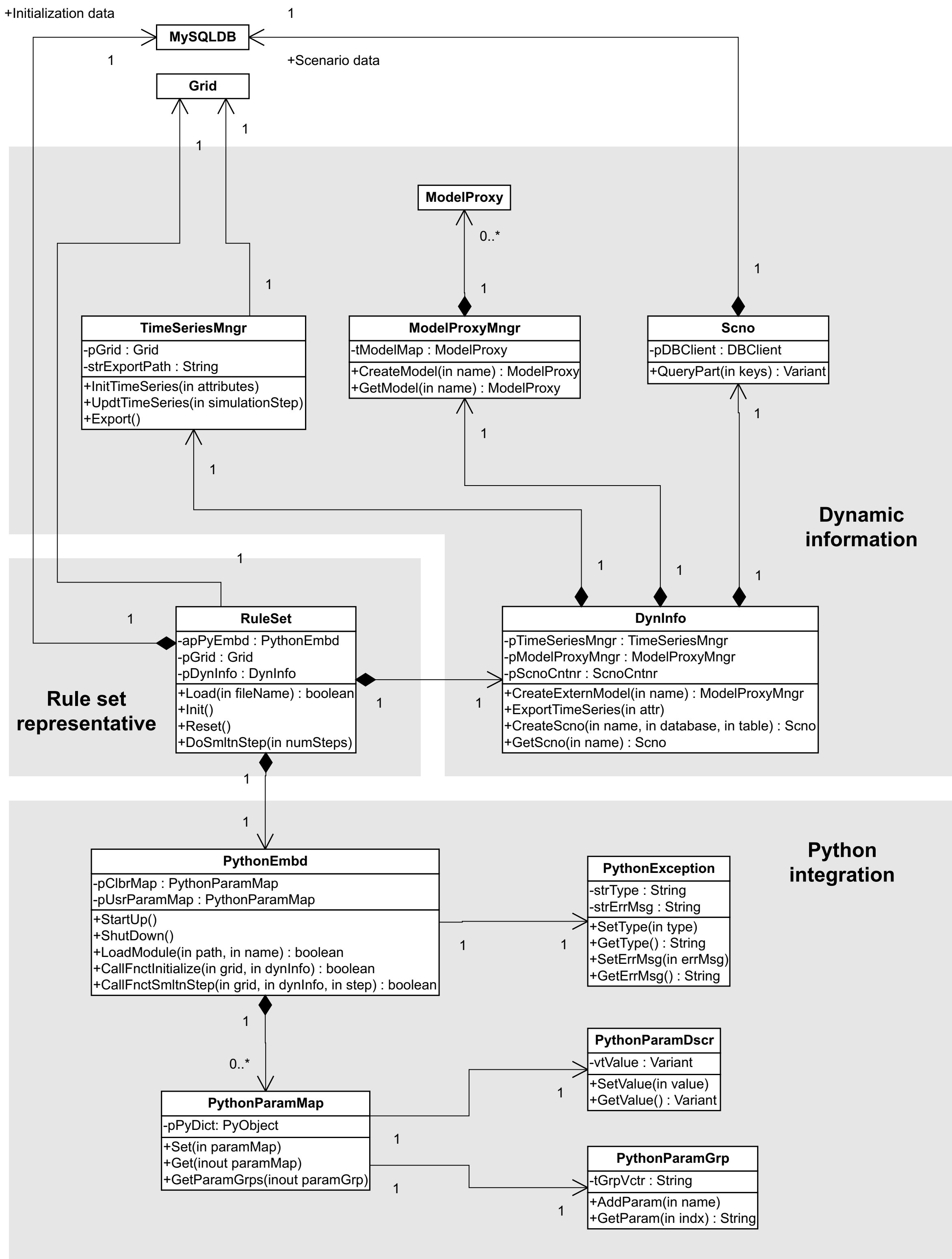


Figure 3.12: Class layout of the SITE *SmltnDynamics* component. Dynamic information and Python scripting language integration are managed by a single object representing the entire application rule set.

like the SITE system domain classes *Grid*, *Cell* etc. plus a set their methods become part of it. To allow manipulation of these classes, an instance of the SITE simulation grid and the dynamic information object are passed to the required Python functions *Initialize()* and *SimulationStep()* (see their C++ counterparts *CallFnctInitialize()* and *CallFnctSmltnStep()* in Fig. 3.12).

While language extensions represent a way to access and manipulate system domain information from the application domain, the access and manipulation of Python variables from the system domain is realized differently. Variables to be accessed by the system domain need to be defined exclusively in separate Python modules. Information exchange is done through the *PythonParamMap* class which is an encapsulation of such a Python module (a *PyDict* object in the Python C API). The helper classes *PythonParamDscr* and *PythonParamGroup* provide meta information on these variables. In the SITE framework variables of the rule set scripts need to be accessible for two different aspects. First, during rule set calibration, the calibration algorithm (see section 3.4.2) will determine candidate solutions for the respective parameter set. Since the calibration methodology is implemented in the SITE system domain, these values need to be transferred to the application side. Second, to support the design goal of user-friendliness, a number of rule set parameters can be made editable via the GUI, which also is part of the SITE system domain (see Fig. 3.6).

Dynamic information in the SITE framework refers to all aspects of information that are dependent on the simulation time step. It includes several characteristics of information like output data (simulation time series), model driving forces and input data (scenarios) or the processing of necessary information through the use of sub-models. The management of dynamic information is delegated to a special object of class *DynInfo* which is passed to the application rule set script together with the *Grid* object and thus can be accessed from inside the application code. The *DynInfo* object itself holds references to management objects dealing with the specific types of dynamic information. Time series are created for selected attributes. The selection can be made using the GUI or by defining attributes in the application rule set. Time series tables (saving the change of attribute values for every single cell) are exported in form of a csv file for each selected attribute. Each column of a time series table represents the attribute state after completing a simulation time step.

Scenario data are managed by an instance of class *Scno* (scenario manager). In SITE, scenarios are technically represented by compilations of different input data sources in form of both database tables and specifically tagged rule set parameters. *Scno* objects are configured via an XML file which the user must select together with the rule set script prior to performing a simulation run. Based on this configuration, the *Scno* object provides rule set parameterization and establishes the connection to database tables that represent time series. Simulation runs are always determined by the combination of a rule set with one specific scenario. The SITE scenario functionality is designed to allow user interaction. This means that a simulation run can be stopped at on predefined step, The results produced so far can be analyzed to extrapolate whether certain targets will be achieved. Depending how trends to target achievement are, scenario parameters (e.g. a rule set parameter representing a management parameter) can be edited, thus simulating policy interaction. After that, the simulation can be continued. When editing scenario parameters, SITE automatically tracks these changes to ensure reproducibility and filing of simulation runs.

Third-party models are represented by model proxy objects that are made accessible inside application code via the passed *DynInfo* object. Depending on the application, an arbitrary number of sub models which are contained by a model management object (*ModelProxyMngr*) can be integrated. For details on the integration of sub models and the respective interface provided by the SITE framework see section 3.4.5.

#### Summary

The implementation of the SITE core engine already provides a solution to most of the listed requirements (see section 3.1). It houses all functionality for generic CA-based land-use modeling. Through the use of an established scripting language extended by a concise interface to manipulate the modeling data structures, it makes a maximum of functionality available for different modeling applications. No directives for specific modeling methodologies are made. Thus, applicability to a multitude of modeling projects is established. Expandability and maintainability are ensured by the consequent use of object-oriented programming paradigms. Innovative functionality compared to existing approaches are automated documentation of simulation runs and rule set parameterization as well as the possibility to interactively handle scenario analysis. Integrated modeling is supported via model proxy objects through which an interface to third-party models is made available (see paragraph 3.4.5 for a detailed description).

### 3.4.2 Model calibration and model testing components

Calibrating complex application rule sets is crucial to achieve valuable simulation results (see chapter 5 for the detailed description and discussion of the calibration of rule set parameters for a selected application). The SITE framework provides generic functionality to find an adequate solution for application-specific parameter sets. Figure 3.13 shows the class layout of the *Calibration* component in connection with component *ModelTest*. The calibration component basically consists of a management object that possesses a reference to an algorithm capable of optimizing the defined parameter set. For all optimization algorithms to be used in the context of the SITE framework, a base class (*OptmzAlgr*) declaring a generic interface is provided. In the current version, an optimization procedure utilizing genetic algorithms (*OptmzAlgrGntc*) is implemented. Other algorithms can be added and used polymorphically. Optimization algorithms have a reference to the rule set object representing the rule set to calibrate. From this object, also the parameter set to optimize is retrieved. Via the rule set object, the optimization algorithm triggers the repeated reset and restart of simulation runs (hence the dependency of component *Calibration* from *SmltnDynamics*). After each single simulation run based on a specific solution for the parameter set to calibrate, the optimization algorithm evaluates the respective simulation result via its assigned objective function. In the SITE framework, the quality of a simulation result is assessed by comparison of a result map with a reference map. Depending on the application, different map comparison algorithms might be favorable. Map comparison algorithms to be used as objective functions for rule set calibration are not implemented in the *Calibration* component, but in the *ModelTest* component. Optimization algorithms utilize functionality provided by *ModelTest*.

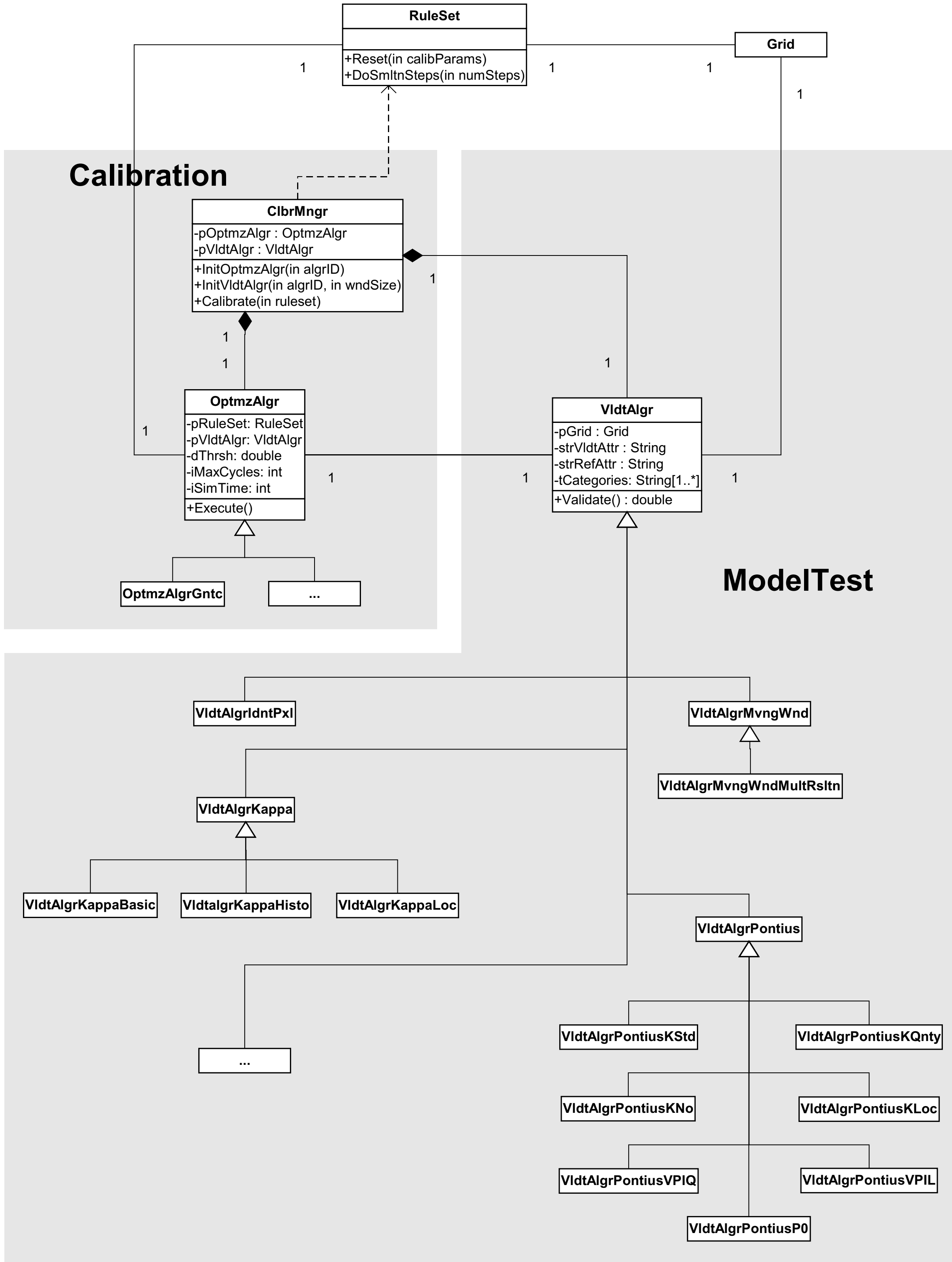


Figure 3.13: Class layout of the SITE *SmltnDynamics* Class layout and connection between the SITE calibration and model testing components. In this constellation, the model testing component is used to provide an objective function for the calibration methodology.

Table 3.3: List and description of map comparison algorithms implemented in the SITE *ModelTest* component.

| Class name | Algorithm description |
|---|---|
| *Simple pixel comparison* | |
| VldtAlgrIdntPxl | Compares all corresponding grid cells and calculates the ratio of the number of all matching cells and the number of all cells. |
| *Pixel-based Kappa* | *Reference: Pontius (2000)* |
| VldtAlgrKappaBasic | Calculates Standard Kappa value based on contingency matrix |
| VldtAlgrKappaHisto | Calculates Kappa value based on contingency matrix specifically regarding quantification error |
| VldtAlgrKappaLoc | Calculates Kappa value based on contingency matrix specifically regarding location error |
| *Moving window-based Kappa* | *Reference: Pontius (2002)* |
| VldtAlgrPontiusKStd | Standard Kappa value |
| VldtAlgrPontiusKNo | Kappa value without considering quantification and location error |
| VldtAlgrPontiusKQnty | Kappa value specifically regarding quantification error |
| VldtAlgrPontiusKLoc | Kappa value specifically regarding location error |
| VldtAlgrPontiusVPIQ | Value of perfect information of quantity |
| VldtAlgrPontiusVPIL | Value of perfect information of location |
| VldtAlgrPontiusVP0 | Observed proportion correct |
| *Moving window-based ratio* | *Reference: Kuhnert et al. (2005)* |
| VldtAlgrMvngWnd | Ratio of correctly classified cells for specified moving window size |
| VldtAlgrMvngWndMultRsltn | As above, but integrated over multiple resolutions up to moving window size |
| *Figure of Merit* | *Reference: Klug et al. (1992); Pontius et al. (2007)* |
| VldtAlgrFigMerit | Pixel-based figure of merit, assessing actual change |

In the SITE framework, assessing the quality of simulation results and thus the quality of the underlying models is done by comparing the simulation result maps with a reference map (e.g. historical land use maps throughout rule set calibration). For this purpose, the *ModelTest* component provides a number of map comparison algorithms (Fig. 3.13). All algorithms share the same base class *VldtAlgr* which basically holds the main information required by all concrete map comparison algorithms. In addition, it defines an interface valid for all algorithms and allows their equal utilization by clients (e.g. an optimization algorithm from the *Calibration* component) through polymorphism. The *VldtAlgr* base class includes a reference to the *Grid* object (component *SmltnEnvironment*) which is the data structure on which all map comparison algorithms operate. The class hierarchy of map comparison algorithm can be arbitrarily extended. The *ModelTest* component features different families of algorithm like direct pixel-to-pixel comparisons, pixel-based comparisons using the Kappa with regard on quantification or location error (Pontius, 2000), moving window-based Kappa algorithms also considering quantification and location error (Pontius, 2002) and moving window-based algorithms based on the quantities category occurences inside an actual window (Kuhnert et al., 2005). Table 3.3 lists all map comparison algorithms that are currently available in the SITE framework. These map comparison algorithms typically deliver result values between 0 (no match) and 1 (perfect match, identical maps). Results delivered by Kappa-based algorithms need to be interpreted differently: A value of 1 means perfect classification while values out of the interval $]0;1[$ indicate, that the proportion of cells classified correctly is greater than the expected proportion classified correctly due to chance. A value of 0 or smaller means that there are no more correctly classified cells than there would be due chance. The categories to be regarded for map comparison can be selected freely. All categories that are not selected are considered belonging to a rest category.

So far, only algorithms comparing categorial maps are implemented. However, the *ModelTest* component data structures are technically not restricted to categorial maps. Integration of other types of validation algorithms like the ROC method (Pontius and Schneider, 2001; Pontius and Pacheco, 2004) which assesses the quality of suitability maps or algorithms applying an amount of fuzziness to both location and category of cells (Power et al., 2001) is possible.

The *ModelTest* component is designed to be adopted at all places where the quality of simulation results needs to be assessed. The SITE framework provides the functionality to assess the quality of simulation results directly via its GUI. In addition, the provided map comparison algorithms are used by the *Calibration* component as objective functions (as depicted in Fig. 3.13).

**Summary**

With the calibration and model test components, the respective methodology is consistently integrated into the SITE framework. All models operated within the framework can utilize the functionality. The implementation satisfies the respective requirements listed in section 3.1. An application of the calibration and model test components can be found in chapter 5. Calibration methodology is restricted to algorithms that find an optimal or adequate solution for a set of rule set parameters with respect to an objective function. In the particular case of the SITE framework, map comparison algorithms have to serve as objective functions.

### 3.4.3 Clients

All SITE components introduced so far physically are libraries that have to be linked dynamically. Consequently there is no fixed directive how to put the functionality they offer into operation. Typically, software is operated either using a graphical user interface or via the command line. Both ways have their specific advantages and drawbacks depending on the respective software is being used for. The SITE component architecture is suitable for both kinds of clients. While due to the requirement of user friendliness a detailed graphical user interface has been created, there is also a command-line client available which can be used for time-consuming calculations (typically rule set calibration runs) and for integration of the SITE framework into batch processes.

### Graphical user interface

The GUI developed for SITE is designed to enable simple handling of the rather complex framework. Hereby, the requirement of simple applicability is addressed. It supports both the utilization by non-expert users and by rule-set developers. The SITE components are linked dynamically to the GUI. The dependencies between client and the components and the inter-component dependencies are displayed in Fig. 3.6. For inputs and for controlling simulation runs, the GUI directly calls the respective methods from the component interfaces. The visualization of simulation results is based on an observer pattern (Gamma et al., 1995) through which respective GUI elements are notified as soon as an update is necessary. The GUI is based on the Microsoft Foundation Classes (MFC). Additional functionality like more elaborated GUI elements (e.g. dockable control bars) and a flexible layout which is restored at each startup are provided by a commercial extension to the MFC (Business Components Gallery). Consequently, using SITE in combination with its GUI is only possible on Windows systems.

A snapshot of the SITE GUI is given by Fig. 3.14. As depicted, there are two different kinds of visualization of simulation data. The first one is the view on the simulation grid, where one attribute, which can be selected by the operator, is displayed for the entire area using a specific color code. For continuous data (e.g. elevation), attribute values are coded by a transition of colors from green (lowest value) to red (highest value). Categorial data (e.g. land use classes) use a color coding that has to be specified in the rule set script (see appendix A). Two instances of this view can be displayed simultaneously to facilitate the examination or comparison of two different attributes. The views are capable of rendering data in three dimensions and can be freely navigated and zoomed. Therefore, it is possible to use the z-axis to display a second attribute value inside each view. Typically, one would select the elevation attribute (if available) to get a true representation of the project area, but the 3D functionality might also prove useful for other data, e.g. to check whether distance maps are calculated correctly. For the rendering of the simulation grid, it is represented by a graphical model which is displayed using the OpenGL graphics hardware interface. The second aspect of data visualization is the display of basic attribute statistics. Statistics maintained for each attribute include simple descriptive statistics (mean, minimum, maximum, variance, standard deviation) and a distribution of attribute value frequencies.

The current status of a simulation run is displayed by the control bar labeled as "Simulation control" in Fig. 3.14. It consists of two tabbed pages. The first one gives information about the current simulation time step and about existing thematic layers and allows the selection of attributes for display. The second page allows editing the value of selected rule set parameters. Which rule set parameters appear is specified inside the application rule set script. It is this functionality which makes SITE accessible to a wider range of users. Two types of users are addressed: expert users capable of writing their own application code and non-expert users working on finished but parameterizable rule sets.

The basic control of simulation runs is done via the GUI tool bar where an operator can load and reset a simulation and start single or multiple simulation steps. Further control and configuration of input and output is provided by specific menu items. Via the menu bar, specific tools for managing interactive scenarios and assessing the quality of simulation runs by means of map comparison procedures, are provided.

By default, user system messages and warnings are directed into a message console.

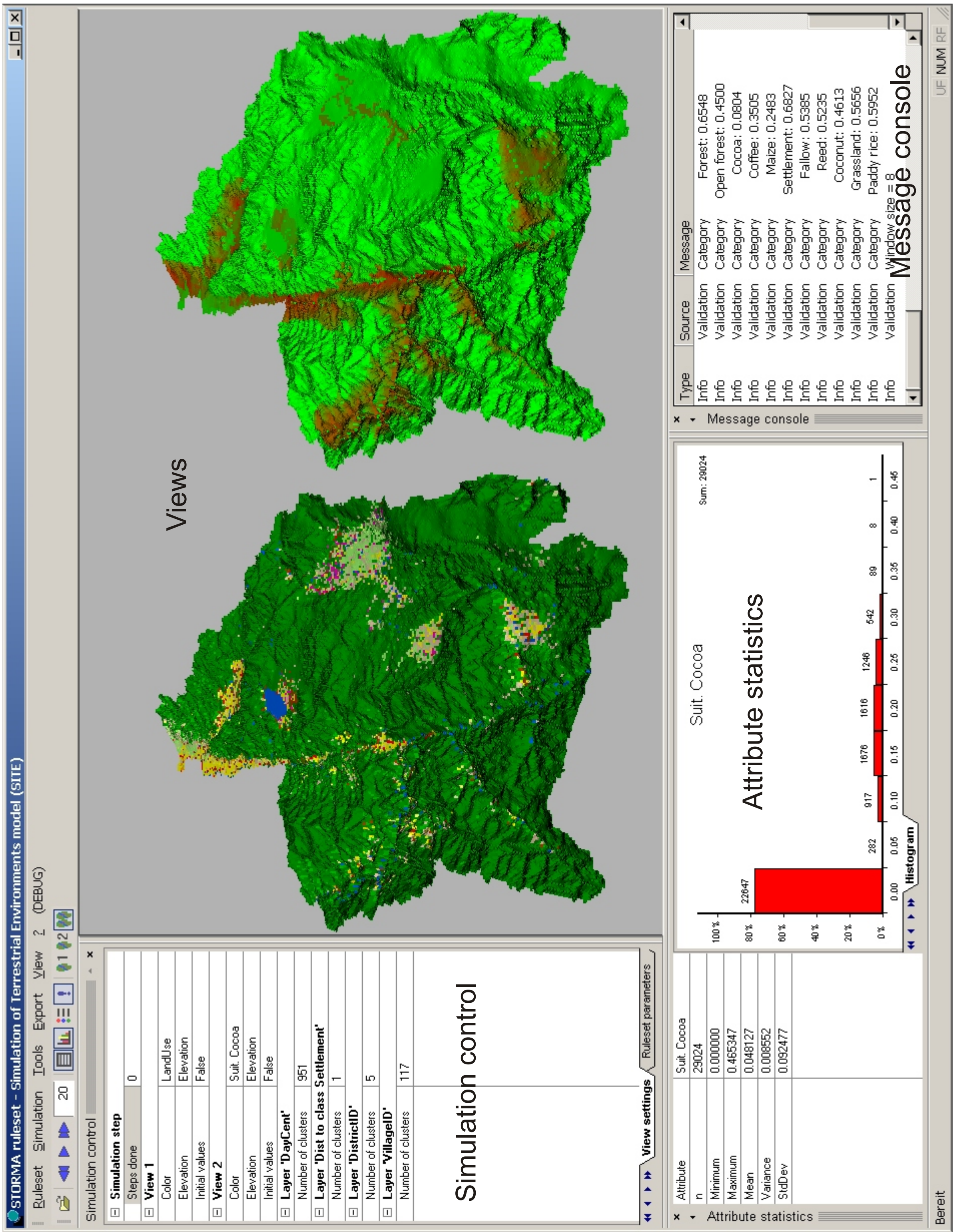


Figure 3.14: Snapshot of the SITE graphical user interface in typical configuration. The left view shows current land use, the right view a suitability map. For both views, elevation is displayed on the z-axis.

In addition, a file labeled “SITELog.txt” is written containing the same output as the message console. Analogous to the other GUI elements, an observer pattern is used for the output of system messages and warnings. Output devices are specified in the code of the respective client.

#### Command-line client

While the GUI is specifically suited for interactively performing simulation runs and assisting in the development of application rule sets, the use of a command-line based client addresses the application of the SITE framework with particular consideration of productivity and automation. The command-line client is intended to be used inside batch processing frameworks and for time-consuming jobs, especially for the calibration of application rule sets. Therefore, a special dependency of the command-line client from the *Calibration* component is established (see Fig. 3.6). Compared to the GUI, the SITE command-line client is a very lightweight application which can easily be ported to different platforms.

When using the command-line client, system messages and warnings are directed to the standard output and standard error channels respectively. The output can be redirected to other devices like e.g. files.

### 3.4.4 Import/export, database connectivity

A SITE application is initialized based on data stored in a database table. Database clients for Microsoft Access and MySQL are available. The database connection details (database name or file path and table name) need to be specified in the main module of the application script. Initialization data is read upon initialization of the simulation grid.

In its current version, the SITE framework writes output data to files as comma separated lists (using semicolons as separators) which can be easily imported into other software for evaluation. There are two types of output, output of values of a selected attribute at the current step in simulation time and output of time series for selected attributes. Attribute selection can be done via the respective menu items in the graphical user interface or alternatively by Python methods provided by the system/application interface in the application script. The latter alternative has to be used if the SITE is operated by its command-line client.

Configuration of components and functional parts is done based on XML files. Configuration files need to be edited whenever basic selection have to be made (e.g. selection of the database client via file *DBConfig.xml*, specification of working directories for DayCent integration via file *DayCentDrvrConfig.xml*, etc.).

System messages are directed to specific message devices depending on the SITE client used. If SITE is operated using the GUI all messages are displayed in their own message console GUI element. In addition, messages are written to a log file. The command-line client directs messages to the standard output and standard error channels. Specific messages are issued by the calibration component during model calibration runs. This output contains all data associated with a calibration run.

### 3.4.5 Integration of third-party models

One of the main requirements defined for the SITE framework is the integration of sub models. Sub models provide services for the superior land use model by performing specific calculations and making their results available for the land use model which in turn uses these results in its own process of decision making. This way a feedback loop between the land use model and its sub models is established.

The technical realization of integrating external sub models basically depends on complexity and character of the models to integrate. In case of very low complexity, e.g. if the sub model is none more than a regression function, the best way of integration is to simply code the model in the application script. However, for more complex models that already have been implemented as software this is not applicable. A mechanism is required by which it is possible to configure and invoke the sub model and to read in the respective modeling results. Since SITE is designed to be a generic framework for regional land use modeling applications, it is also desirable to establish a universal solution for the integration of existing sub models. Although this goal is rather impossible to achieve for all different kinds of models, by making a number of compromises an adequate solution can be achieved. Things get particularly complicated if it is desired to integrate a model that defines its own spatial grid, which, among other difficulties, results in the need of synchronization with the SITE grid.

SITE development occurred largely parallel to the development of the first major SITE

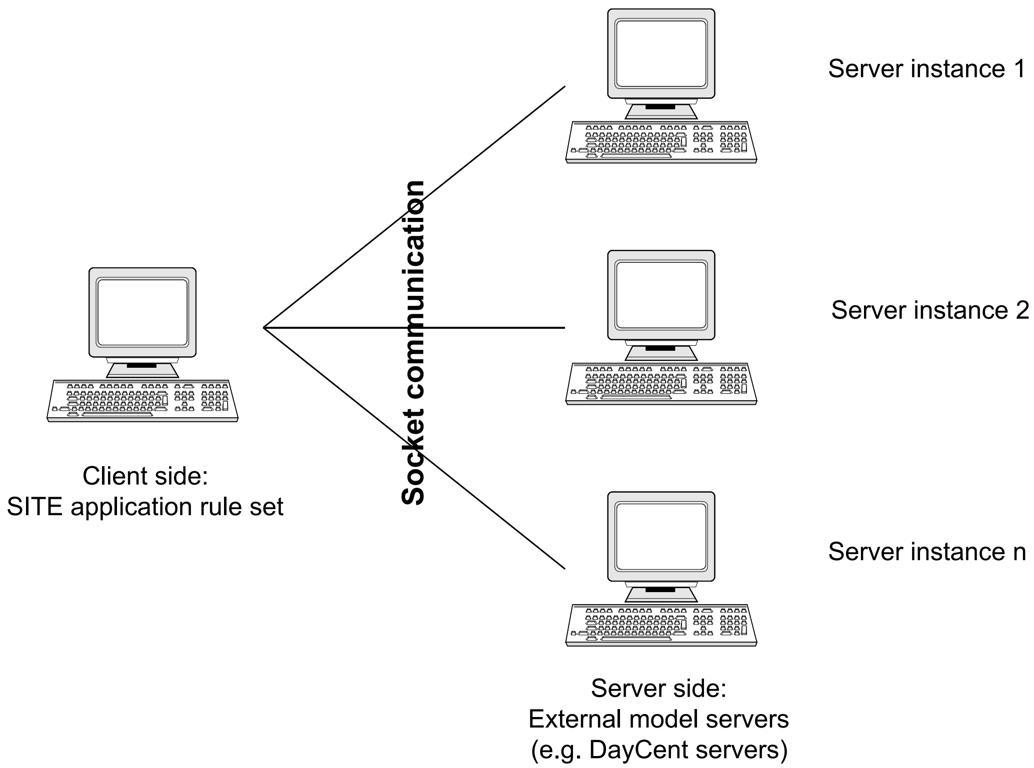


Figure 3.15: A client/server approach is used inside the SITE framework to integrate external models. External models are wrapped and operated via server applications. The actual land use model has the client role and configures and send modeling jobs. The system is capable of parallel processing.

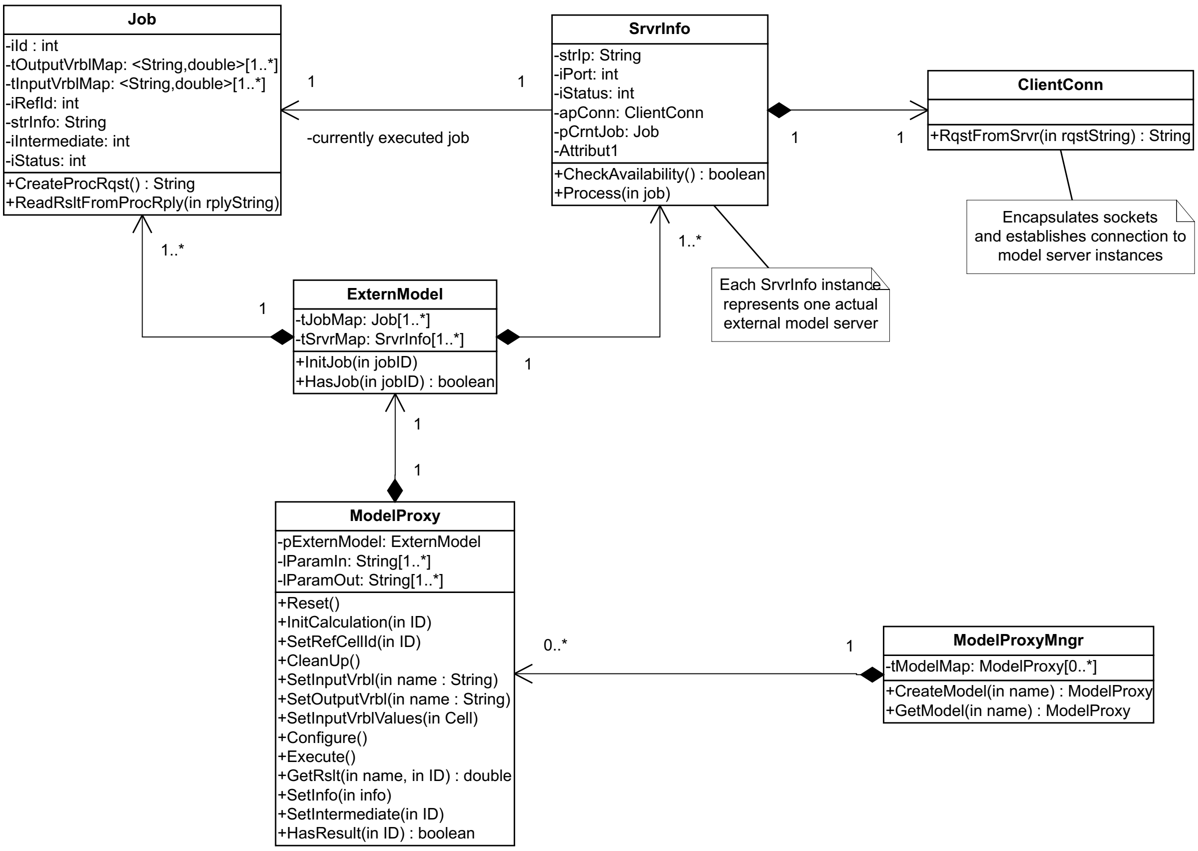


Figure 3.16: Class diagram of the client side of the SITE sub model integration. The *ModelProxy* methods to configure and operate sub model can also be found in the classes ExternModel and *Job* and are exposed the application domain. Each instance of *SrvrInfo* is connected to one server application.

application, a rule set to model the stability of tropical rain forest margins in the context of the STORMA project (see chapter 4). This application required the integration of the DAYCENT agro/ecosystem model (Parton et al., 1998) to calculate the productivity and soil parameters for single grid cells representing crop areas. The DAYCENT model represents a class of models that have no explicit spatial reference and are only valid for one single location, or, referring to the spatial explicit SITE framework, for one of its grid cells. In addition, it can be assumed that there is no influence by neighboring grid cells when performing DAYCENT calculations for a particular cell. Accepting these criteria, no explicit spatial reference and no influence from neighboring cells, as compromises, it is possible to define an interface to integrate a variety of external models. Technically, the fact that sub model calculations for single grid cells can be seen as isolated processes, they have the property of being concurrent and thus can be executed in a parallelized framework with a significant gain in run time.

The technical framework to integrate third-party models established in SITE is depicted by Fig. 3.15. It is based on a client/server architecture with the SITE application (land use model) acting as client that requests modeling jobs from the actual third-party models on the server side. Communication between client and the model servers is established via sockets. Socket functionality is encapsulated in a library providing simple functionality to send and receive job requests and replies. Based on the integration of the DAYCENT

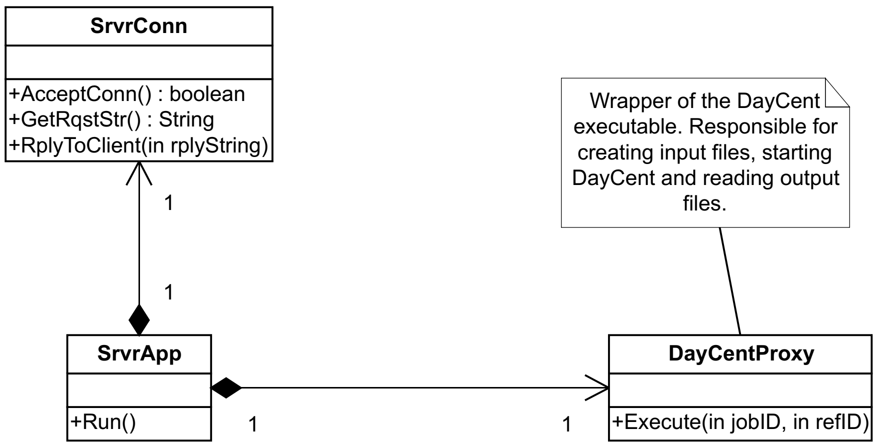


Figure 3.17: Integration of the DayCent model as an example for integration of third-party models. A server instance receives calculation requests. Configured jobs are started by a wrapper instance includeing the actual third-party model.

model, an interface to configure and execute processing jobs and receive the respective modeling results for use in the application has been created. Additional effort was put into keeping this interface generic for use with other models.

### Client side

The interface is implemented in the *ModelProxy* class. An instance of this class represents the use of a specific third-party model (e.g. DAYCENT). The methods forming the interface can be seen in class *ModelProxy* in Fig. 3.16 which shows the static structure of the sub model framework client side. All displayed classes are part of the *SmltnDynamics* component, the classes *ExternModel*, *Job* (processing job), *SrvrInfo* (server info) and *ClientConn* (client connection) being implemented in libraries other that the SmltnDynamics DLL, but statically linked to it. The sub model interface from class *ModelProxy* is exposed to the Python language to enable its use inside application rule set scripts. *ExternModel* is responsible for creating and configuring the single processing jobs (e.g. for single grid cells), for configuring the single server instances that do the actual processing and for distributing processing jobs to server instances. Each object representing a processing job holds the complete configuration data. Based on this, it provides two functions, *CreateProcRqst()*, which generates a request string transmitted to a server instance, and *ReadRsltFromProcRply()* (read result from processing reply), that is able to interpret the reply string received from a server instance after the job processing is finished. Actual server instances are represented and accessed by objects of type *SrvrInfo*. These objects provide information about the connection to the server (using IP address and port) and whether a server is available (function *CheckAvailability()*). To start a process, they define a function *Process()* which takes a reference to the job to be processed as argument. The job object, in turn, is capable of providing a request string to be sent to the server instance. Each *SrvrInfo* object starts a separate thread in which the request string is transmitted to the server. Consequently, it is not necessary to wait until the server instance replies after finishing the job processing. Instead, the *ExternModel* object can directly send the next job request to an idle serve, if available. Availability and status of model servers is indicated by a set of status flags (unknown, not available, ready, processing). In addition, each job object indicates its current status using a set of flags (initialized, configured, processing, finished). Based on these flags, the *ExternModel* object carries out the distribution of processing jobs. The communication between client and servers is done using functionality provided by a *ClientConn* object that encapsulates socket functionality behind its function *RqstFromSrvr()* (request from server).

**Server side**

Which servers are potentially available has to be defined prior to a SITE simulation in an XML file, where the IP address and port number of each instance have to be specified. Currently, it is not possible to add new servers during a simulation run. However, out of this specified set, not all servers must necessarily be running. The system is robust concerning failure and temporary unavailability of servers. In this XML file, a directory for intermediate data used by a third-party model can be defined. The DayCent model requires this information to store intermediate files which it needs to resume its status based on former simulation steps.

The actual job processing takes place on the server side (Fig. 3.17). Job requests are received by an object of class *SrvrConn* (server connection), which is the counterpart of *ClientConn* and is implemented in the same messaging library. It defines the functions *AcceptConn()* (accept a connection from a client), GetRqstStr() (receive an incoming request string) and *RplyToClient()* (send a reply string containing processing results back to client). A *SrvrApp* (server application) instance is capable of interpreting request strings and creating reply strings. This *SrvrApp* object then configures the third-party model using a model proxy object. The proxy object is a wrapper of the actual third-party model. Due to the variety of models that can be used, this wrapper has to be a specific development. Figure 3.17 shows the wrapping of DAYCENT model. DAYCENT needs a set of specific input files and preprocessing steps typically performed manually or inside batch jobs. It delivers results by means of result files and files describing its intermediate status. The *DayCenProxy* wrapper is designed to carry out all these steps programmatically (e.g. starting the actual DAYCENT executable). An advantage of this approach using a wrapper is that it is not necessary to manipulate the code of the integrated model which avoids the introduction of additional complexity.

### 3.4.6 Extendibility and portability issues

The entire SITE system is designed to be extendable. The component-based architecture defines clear functional units and encapsulates the respective complexity. Implementation of components themselves has been done with strict use of the object oriented programming paradigm. Design patterns (especially factory, observer, iterator) have been used heavily. The system/application interface can be extended by simply adding new classes or class methods to expose to the Python language.

Another focus during the development of the SITE framework lay on the minimization of efforts necessary if components need to be ported to other platforms, in particular to Linux systems. With the exception of the graphical user interface, which uses the Microsoft MFC library, only libraries have been used that are available on both platforms.

However, the technical solutions of how dynamic linking of components differs between Windows and Linux, which requires a certain amount of work when porting the system.

## 3.5 Discussion

Land use and its dynamics are determined by a wide variety of factors. Since research on land-use dynamics is mostly interdisciplinary research, a modeling framework used in this context consequently has to be able to reproduce and utilize interactions between the different factors determining land-use change. The increasing availability of sectoral models (e.g. for population dynamics, crop growth, ecosystem services) favors a modular assembly resulting in integrated modeling systems. Since SITE was developed to be used as such an integrative tool, its value for the modeling community has to be benchmarked largely based on its capability of model integration and the way this capability is combined with other innovative features.

SITE provides two different ways of model integration. On the one hand, integration can be achieved via a specifically designed interface. This interface facilitates the coupling of complex models that are available in the form of components. Component-based coupling of models has become a popular approach in landecological modeling, as it supports modularity, and interchangeability of integrated models (He et al., 2002; Argent, 2004). However, SITE advances this functionality by establishing a mechanism to feed back results from the coupled model to the calling instance. Beside the capability to establish feedback loops, the SITE model coupling interface supports parallel processing, provided that the modeling methodology allows concurrent processes (e.g. the calculation of yields for crop cells which does not interfere with any processes in the cell neighborhood). This results in a significant reduction of processing time for simulation runs. As a second method, model coupling can be achieved by creating extensions to the SITE scripting language. For the integration of relatively simple models (e.g. regression models, functional dependencies) this method is even superior to the component approach, since respective language extensions can be implemented quickly. The implementation of feedback loops is also possible for the latter case. The applicability of the integration and feedback functionality has been shown in case studies, where SITE was linked to an agro-ecosystem model and a model integrating ecosystem services (see chapters 4 and 6).

The advanced possibilities for model integration are combined with a generic land-use modeling platform. As for model integration, a number of solutions for generic platforms are available ( e.g. GEOMOD2, Pontius et al., 2001; SELES, Fall and Fall, 2001). However, these solutions gain simplicity at the cost of flexibility (e.g. the SELES domain-specific language requires the definition of so-called landscape events and thus does not allow the use of other modeling methodology). Generic applicability is ensured by SITE through its central design characteristic, which is the strict separation of implementation and application into system and application domains with the use of a modern high-level scripting language (Python) for the implementation of land-use modeling applications. The Python language was extended to match requirements specific to land-use modeling (e.g. by adding classes for the simulation grid, cells and attributes). Thus, a full-fledged programming language is available for model implementation; no restrictions remain regarding modeling methodology as opposed to existing solutions. In addition, Python is already being used as scripting language in a number of established software products with significance to the land-use modeling community (e.g. GIS-Software), which enables further possibilities with respect to synergies with these products.

Model calibration, although indispensable (Boumans et al., 2001; Oliva, 2003; Straatman et al., 2004), is not integrated in most of the available modeling frameworks. In the SITE framework, calibration functionality is implemented in an integrated system component. In the current version, only genetic algorithms are available, but the component can be extended to house additional methodologies. Calibration algorithms used by SITE aim to find an optimal or adequate solution of an arbitrarily defined parameter set (defined in the application script) based on an objective function. The objective function, in turn, can be freely selected from another system component (*ModelTest*), which provides a selection of map comparison algorithms. This design enables model operators to freely combine optimization algorithms and algorithms for objective functions. Apart from the process of parameter selection, which requires expert knowledge of the underlying rule set, SITE is capable of automated rule set calibration. The component that implements the different map comparison algorithms, can also be used independently from the calibration as an integrated tool for model tests based on map comparison methodologies. The SITE calibration methodology seamlessly interacts with the generic modeling functionality and integrated models, thus it can be used for all applications that are operated within SITE. Moreover, the calibration methodology is not restricted to the land-use model, but also allows to simultaneously calibrate different integrated models (see chapter 5).

The explicit representation of scenarios in SITE is a further innovation in the field of land-use modeling frameworks. Performing a simulation in SITE always implies to use the underlying model rule set in combination with a quantified scenario. Model rule set and quantified scenario are separate instances. This concept allows simulation runs under different scenarios without having to edit model code, thus improving system handling and facilitating maintenance. With the possibility to interactively handle and alter scenarios based on an analysis of interim simulation results, it was possible to overcome a major limitation of scenario analysis (Alcamo et al., 2006).

Although the SITE concept of integrating a scripting language significantly facilitates model implementation, programming knowledge is still requested. To enable scientists without programming knowledge to also work with the SITE framework, a detailed graphical user interface (GUI) has been created. In this GUI, arbitrary rule set parameters (i.e. variables in the Python application code) can be edited directly. Thus, one can distinguish between two different application levels for SITE: (i) model development, performed by users that are capable of writing application domain code, and (ii) application of complete parameterizable models via the GUI. The latter application level is open for non-expert users. With this compromise, the requirement of simple accessibility to researchers of different scientific background is satisfied. Furthermore, model handling and operation is strongly facilitated by the design of the SITE GUI, which provides two 3-dimensional views on the simulation grid. All attributes of a case study grid can be displayed. With these features, the GUI is also capable of supporting model development since it can give rapid feedback through its configurable views. A high communicability of simulation results is provided. Although the SITE GUI does not directly support definition of rules like other frameworks, that e.g. provide a graphical interface for the definition of rules (Costanza et al., 1998) or model component assembly (Filippi and Bisgambiglia, 2004), it is open for further development in that direction. Another innovative contribution to usability is the automated logging of simulation and model settings for every simulation run, which guarantees reproducible results combined with minimum administration efforts.

Much effort was laid on the architectural design of the SITE framework. The system architecture was developed with respect to the requirements posed for integrative tools. The SITE component design is based on a study by Endejan (2003), who developed a system architecture for integrated simulation-based assessment of global change, emphasizing the advantages of a component-based approach. In fact, recent developments in integrated modeling show that there is a trend toward model encapsulation into components (He et al., 2002; Argent, 2004). Compared to the architecture proposed by Endejan (2003), the SITE architecture represents an advancement with respect to land-sue modeling in the context of interdisciplinary projects. Due to the target of usability, the design is more compact, as several components have been merged (e.g. documentation and simulation-specific components), while on the other side additional components were introduced, based on the set of scientific and technical requirements (e.g. calibration, model test, simulation environment and dynamics components).

In contrast to other publications available on land-use modeling frameworks, this study stresses the importance of a well designed architecture, accurate implementation and the overall software development process for the final system. These technical aspects ensure that the framework can be successfully applied for modeling applications while at the same time being open for further developments in both information and land-use change research. A long term availability of the SITE framework for land-use modeling applications can be expected.

The development of SITE included a couple of innovations in the field of land-use modeling. In particular, however, it was the combination of these features that made SITE an innovative and valuable tool for land-use modeling. As it enables flexible integration of models, including the implementation of feedback loops, together with a generic platform for the formulation of land-use models it is a flexible integrative tool in interdisciplinary land-use modeling projects and an advancements to existing solutions. In addition, it is the only comprehensive approach so far available (see Table 3.1). The underlying architecture ensures expandability of the system and the integration of new functionality, thus enabling long-term usage.

# 4 A study on socio-environmental impacts of land-use change

Land use and land cover change, especially in tropical regions, have received much attention in recent years (Turner et al., 2001; Achard et al., 2002; Lambin et al., 2003; Tscharntke et al., 2007). In the case study from Indonesia presented here, large areas of forest were replaced by cocoa agroforestry plots during the last two to three decades. These processes are intimately linked with agricultural intensification and ongoing rural immigration. Simultaneously, the food crop paddy rice, but also coffee agroforestry were loosing importance in the regional agricultural production systems. In this study we used the SITE framework to examine the dynamics of major land-use and land cover types and quantify selected environmental and economic impacts. The results presented in this study enable both scientists and policy makers to analyze some of the consequences of regionally important policies such as (i) access to land, (ii) rural immigration and (iii) protection of the Lore Lindu National Park (LLNP) area. In addition, food security issues, which are more relevant on the national scale, can be assessed.

## 4.1 Introduction

In tropical Asia land use and land cover change is mostly driven by a combination of 4 - 5 underlying and 2 - 4 proximate factors (Geist and Lambin, 2002). Among them, the processes of deforestation and expansion of agricultural land are very obvious forms of land cover and land use change, often observable by remote sensing (Achard et al., 2002). Simultaneously, less obvious though very important processes of change occur. In forests timber and other products may be extracted, or the understory is removed to install agroforestry systems. In any case, such changes are mostly undetectable by satellites, because they do not necessarily go hand in hand with land cover change (Erasmi and Priess, 2007). In combination with other underlying drivers such as economic, technological and cultural factors, migration can act as a significant driver of land use change (Lambin and Geist, 2003). In the following paragraphs we focus on Indonesia, in particular the island of Sulawesi, to discuss different forms of migration and their relevance for land use change processes. Two major types of rural migration can be observed in Indonesia: state-directed migration and spontaneous migration, both of which take place between and within the different islands of the archipelago and in divers dimensions since the past 100 years.

Indonesia shows enormous disparities in the distribution of population. While the central islands of Java and Bali are the most densely populated regions with 944 and 565 people per km$^2$ respectively, the density in the so-called Outer Island regions ranges from 141 persons per km$^2$ in Nusa Tenggara to 86 in Sulawesi and only 6 in Papua (BPS, 2001). Between 1970 and 2000 the population density on Java and Bali increased by 65% and 47% respectively but by 95% on Sulawesi and by 200% in Papua (BPS, 2001). One main reason for the rapid population growth in the Indonesian periphery is the national inter-island transmigration program. Initiated by the Dutch colonial rulers at the beginning of the 20th century, the program was resumed in the 1950s (Fearnside, 1997a). Between 1969 and 1994, around 6.8 million people have been resettled from the Inner to the Outer Islands with the majority of transmigrations after 1980 (Adhiati and Bobsien, 2001). No specific skills or knowledge relevant for agricultural land use can be attributed to this very heterogeneous group of migrants.

In contrast to the transmigration program, most spontaneous migrants move between different regions of the outer islands. The destinations of spontaneous migration in Indonesia are mainly forest frontier zones, relatively low-populated areas with abundant arable land. The number of immigrants to these regions is largely linked to the improvement of infrastructure, which was part of the national upliftment programs for remote regions of the Outer Islands since the late 1970s (Boomgard, 1993).

In this paper we discuss the question how land use change processes at a tropical forest frontier region are driven or influenced by rural immigration, and regional/local "access to land" policies. In addition to empirical evidence, we present simulation results of historical land use change and a "No Immigration" scenario to assess the potential impact of immigrants on land use and land cover change and some of the associated positive and negative environmental and economic impacts such as nitrogen and carbon emissions and per capita gross margins.

## 4.2 Characterization of the study area

The study site is located in the province of Central Sulawesi, Indonesia and covers 7200 km$^2$ (Figure 4.1). At the time of the field research, the region was divided into five kecamatans (= sub districts) and 118 desas (communities), surrounding the Lore Lindu National Park (LLNP), which was established in the mid 1990ies, partly on land traditionally used by villagers. The provincial capital of Palu (approx. 300,000 inhabitants) is the most important market place and situated north of the northern most kecamatan Sigi Biromaru, which belongs to the tropical lowland zone, while the other four kecamatans belong to the sub-mountainous and mountainous zones (highest peak 2800 m). With a coverage of more than 80%, forest ist the dominating land-use class of the region (Erasmi and Priess, 2007). However, it is increasingly used for agricultural production. Both inside and outside the LLNP, forests are used for the extraction of timber and other forest products such as rattan. Furthermore, forests are converted to agricultural land and agroforestry plots for subsistence and market production. Figure 4.2 shows the land-use/land-cover situation in the year 1981, which is the beginning of our study period.

The provincial capital of Palu (approximately 300,000 inhabitants) is located north of the northernmost kecamatan Sigi Biromaru, which belongs to the tropical lowland zone. It is the most important marketplace for coconuts (copra), cocoa and coffee. The largest fractions of the other four kecamatans are located in the sub-mountainous and mountainous zones.

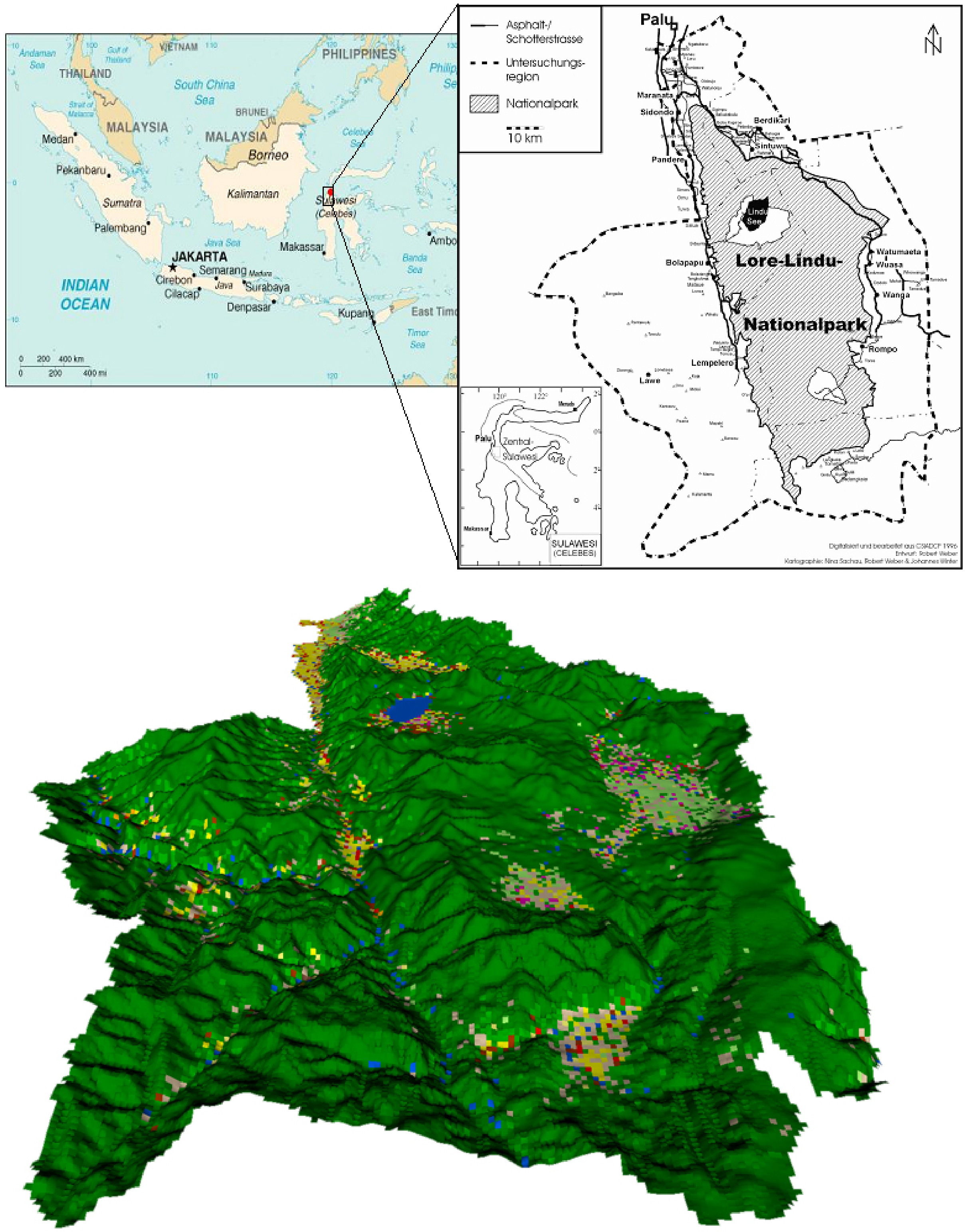


Figure 4.1: Location and overview over the STORMA study area.

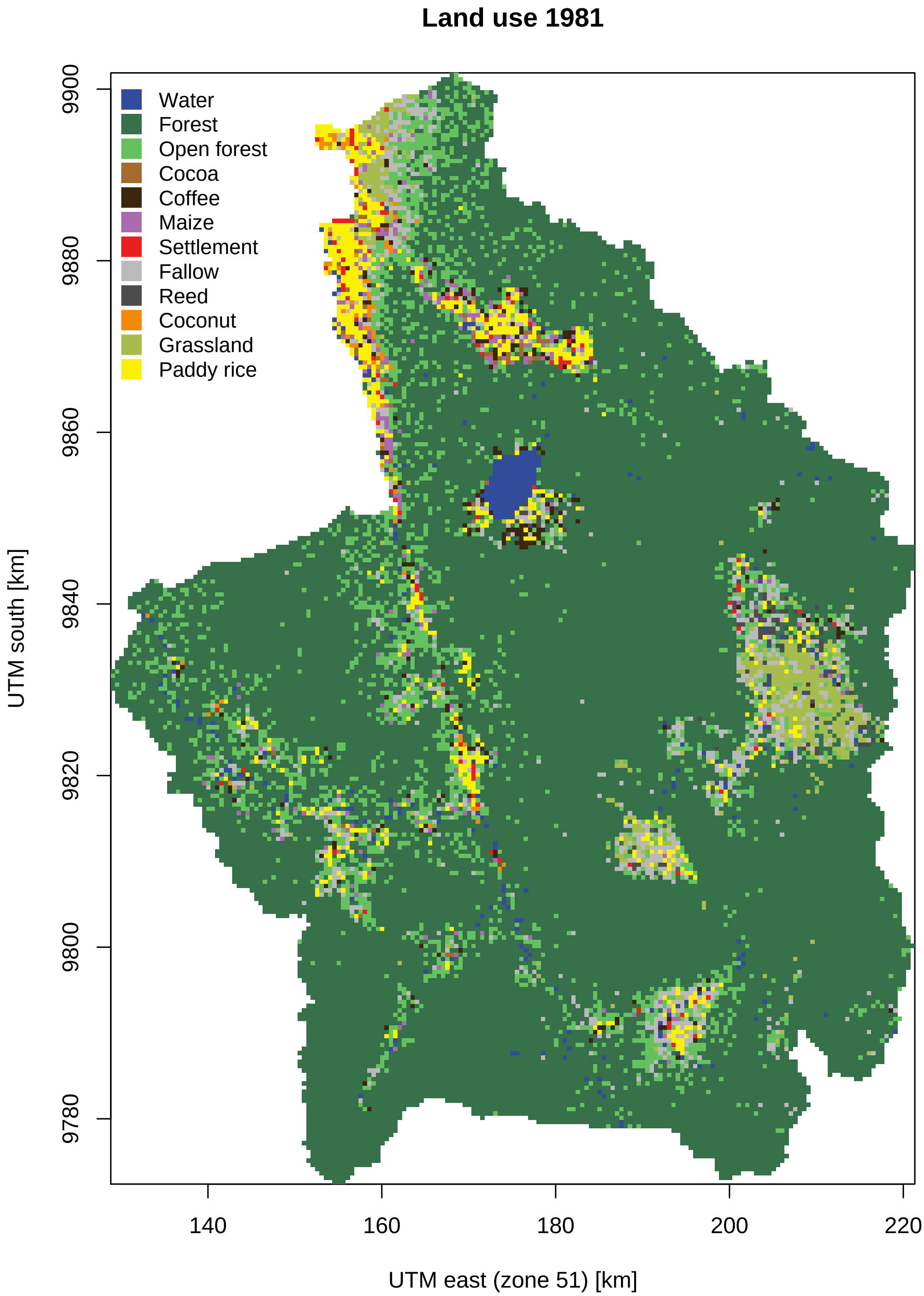


Figure 4.2: Land use map of 1981. This is the initial land-use/land-cover distribution in our simulation.

Table 4.1: Assumptions made for crop production in the *NoMig* scenario.

| Crop | Land-use strategy |
|---|---|
| Cocoa | In case of increasing historical production, increase was reduced by 50%; in case of no production in starting year, increase production to the rate of the kecamatan with the lowest production in the end year 2002. |
| Coffee, Coconut, Maize | In case of decreasing historical production, production was decreased with 50% of the historical rate; in case of increasing historical production, increase rate was doubled. |
| Paddy rice | In case of decreasing historical production, production was decreased with 50% of the historical rate; in case of increasing historical production, the same rate was used. |

## 4.3 Materials and methods

### 4.3.1 Scenario analysis

This study covers the period from 1981 to 2002. Various data sources were used to identify the most important land-cover and land-use types, and quantify the land-use and land-cover changes, which have occurred during the two decades. Based on land-use maps, which we established for 1981 and 2002 following the method developed by Erasmi and Priess (2007), we selected the most important land-use and land cover types according to their spatial coverage in 1981 and 2002.

Immigration into the study region in Central Sulawesi is occurring since the 1970ies, causing a population increase of 60% during the 1981 - 2002 study period. Based on demographic trends of rural population, a *No Migration* (*NoMig*) scenario, in which the population only increased by 9%, was developed. The *NoMig* scenario is based on the assumption that immigration into the five kecamatans of the research area ceases in 1980, caused by restrictive policies of the provincial government in Palu and local leaders in the villages. To simulate the demographic changes of the *NoMig* scenario, we used the population growth rates for rural population of Indonesia derived from the United Nation Population Division (UN Population Division 2006). The growth rates for rural population were applied to each kecamatan starting in 1980, to ensure that starting conditions were the same as during the historical development 1981 - 2002. Additionally, we applied results from our empirical studies, specifically the information that the knowledge how to grow cocoa was mainly introduced to the region by immigrants from South-Sulawesi. In consequence, the transition from strongly paddy-based subsistence farming including some coffee growing both for subsistence and the market, to coffee and more and more cocoa based market-oriented farming is much slower in the *NoMig* scenario than observed in the historical period. All other crops besides cocoa have been cultivated in the region for longer periods of time and follow the same trends as in the historical development. Table 4.1 provides a summary of the assumptions made for agricultural commodities in the *NoMig* scenario.

### 4.3.2 Land-use change modeling

Land use change was simulated using the SITE framework. The respective application (in the following referred to as the land-use model) has been developed as an integrative scientific tool in the context of the STORMA project. STORMA (Stability of Rainforest Margins, SFB 552) is a long-term interdisciplinary Indonesian-German research project funded by the German Science Foundation. The main scope of the land-use model is to quantify scenarios of land-use change, which can either be driven by scientific questions (e.g. impacts of climate change, deforestation, or rural immigration), stakeholder-driven (e.g. by assessing perceived risks and concerns of farmers or local leaders in household and village surveys) or include both stakeholder and science questions. The feasibility to assist in decision-making is considered to be less an issue of operating the model, but to (i) communicate the features and limitations of the model and (ii) communicate the simulations results in terms of socio-economic and environmental impacts of different potential pathways of the recent past or into the future. Policy makers like the provincial government, the Lore Lindu National Park authorities and local farmers have repeatedly demanded tangible results from STORMA scientists. As this was foreseen as part of the collaboration, so far, many research groups have presented their studies – mainly results from field work – in the form of leaflets and posters, but not yet including recent results from the SITE model. Thus at present it is still unclear, whether it will be possible to translate e.g. modeling results, which are partly presented in this paper in a way, which is adequate i.e. credible and understandable for different stakeholders, and whether the desired feedback loop in terms of comments, suggestions and criticism from stakeholders can be established and lead to improved model parameters and scenario assumptions.

The land use model is spatially explicit and based on a CA approach, in which the biophysical environment is represented by a regular 500m $\times$ 500m grid. Four spatial scales are represented in the model: the grid cell level (29,024 units), the village level (116 units), the kecamatan level (5 units) and the entire research area (7,256 km$^2$). Land use decisions are simulated once a year, while the growth of crops, estates and forests is simulated in daily time steps, using an adapted version of the DAYCENT model (Parton et al., 1998), including a new trace gas module developed by Stehfest (2005). The model explicitly simulates the following twelve land-use classes: Natural forest, open/secondary forest, reed, coconut, cocoa, coffee, maize, paddy rice, grassland, fallow, settlement and water. An overview of the entire land-use model is provided by Figure 4.3. The land-use model is divided into two major compartments, SUIT (suitability analysis) and ALLOC (allocation). Biophysical models (e.g. DAYCENT) are integrated. Simulated land-use decisions are based on a multi-criteria suitability analysis of (i) the natural environment and (ii) the socio-economic conditions. Based on a sensitivity analysis, selected model parameters without sufficient empirical basis are calibrated using a genetic algorithm to minimize the difference between a reference land-use map for the year 2002 and the simulated map (see chapter 5). Technically, the land-use model is implemented in a set of Python scripts. Each Python module represents a specific compartment of the rule set (e.g. suitability analysis, settlement allocation, crop allocation, etc.).

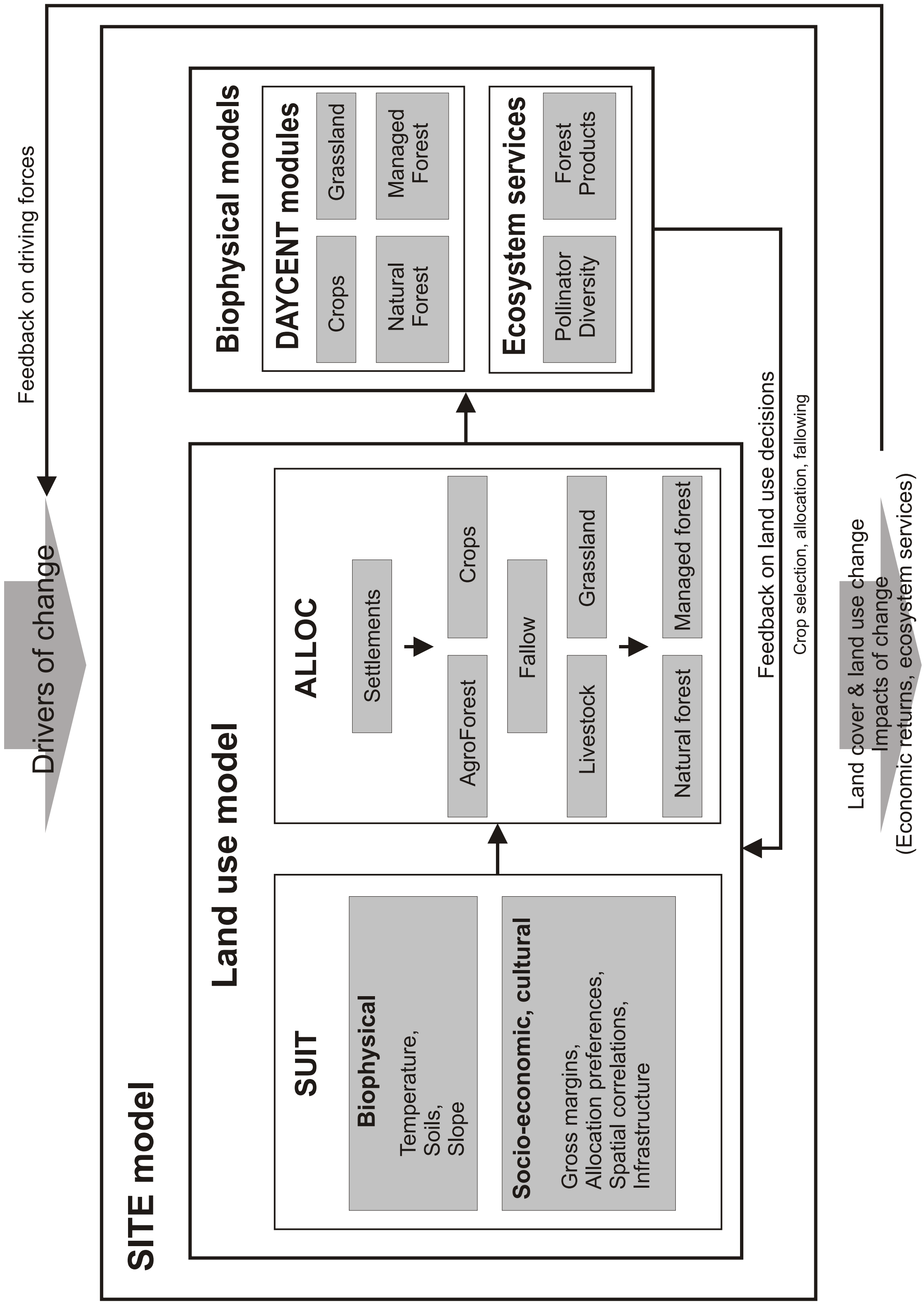


Figure 4.3: Schematic view of the STORMA land-use model.

**Suitability assessment (SUIT)**

The task of the SUIT module is the calculation of suitability maps for each land-use type (see Figures 4.4, 4.5, 4.6 and 4.7). These suitability maps are the input for the subsequent compartment, ALLOC, and provide the main basis for all land-use decisions ind the STORMA land-use model. Suitability values range between 0 (not suitable) and 1 (perfectly suitable) and are assembled from different partial suitabilities. The method behind the calculation of suitabilities is derived from multi criteria analysis (MCA) (Eastman et al., 1995). Using a method based on MCA is advantageous since it is transparent enough to incorporate expert knowledge and flexible enough to incorporate new additional data layers (Schaldach and Alcamo, 2006). Based on the MCA method, an overall suitability value $s_{kl}$ for grid cell $k$ and land-use class $l$ is calculated as follows:

$$s_{kl} = \underbrace{\sum_{i=1}^{m} \omega_i s_{ikl}}_{\text{suitability}} \cdot \underbrace{\prod_{j=1}^{n} c_{jkl}}_{\text{constraints}}, \quad \text{with } \sum_{i=1}^{m} \omega_i = 1 \text{ and } s_{ikl}, c_{jkl} \in [0,1]. \tag{4.1}$$

As can be seen in equation 4.1, the calculation is split into two parts. The first is the mean value of $m$ partial suitabilities $s_{ikl}$, each weighted by $\omega_i$. The second part reduces this preliminary suitability value by applying $n$ constraints $c_{jkl}$, resulting in the overall suitability $s_{kl}$.

The STORMA land-use model explicitly discriminates suitabilities regarding biophysical factors and suitabilities for socio economic parameters (see Figure 4.3). This is also reflected by the calculation of overall suitability values, which is described by equation 4.2.

$$s_{(kl)} = (\omega_B s_B + \omega_E s_E) \cdot c_B c_E \tag{4.2}$$

$$= \left(\omega_B \sum_{i=1}^{m} \beta_i s_{Bi} + \omega_E \sum_{i=1}^{n} \epsilon_i s_{Ei}\right) \cdot \prod_{j=1}^{o} c_{Bj} \prod_{j=1}^{p} c_{Ej} \tag{4.3}$$

$$\text{with } \omega_B + \omega_E = 1; \ \sum_{i=1}^{m} \beta_i = 1; \ \sum_{i=1}^{n} \epsilon_i = 1; \ s_{Bi}, s_{Ei}, c_{Bj}, c_{Ej} \in [0,1]$$

Due to reasons of readability, the subscripts $k$ and $l$ have been omitted in the latter two equations. $s_B$ and $s_E$ are biophysical and socio-economic suitabilities and $c_B$ and $c_E$ are biophysical and socio-economic constraints, respectively. The advantage of this procedure is, that the entirety of biophysical suitabilities can be weighed against the entirety of socio-economic suitabilities. Table 4.2 lists the partial suitabilities applied to the suitability calculation in the STORMA land-use model. In its current version, only constraints from the socio-economic domain are used (e.g. national park protection, resource limitation factor for new crop areas).

Each single partial suitability has to be expressed in the interval $[0, 1]$. Depending on the suitability factor, different methodologies for the assignment of the respective suitability value are used. In most cases, this is done by defining a two threshold values, the first marking the limit for the assignment of 0 (not suitable) and the second marking the limit for assignment of 1 (perfectly suitable). For values in between these borders the calculation of the respective suitability is usually based on a functional relationship (e.g. a linear function). Examples in the STORMA land-use model can be found in the

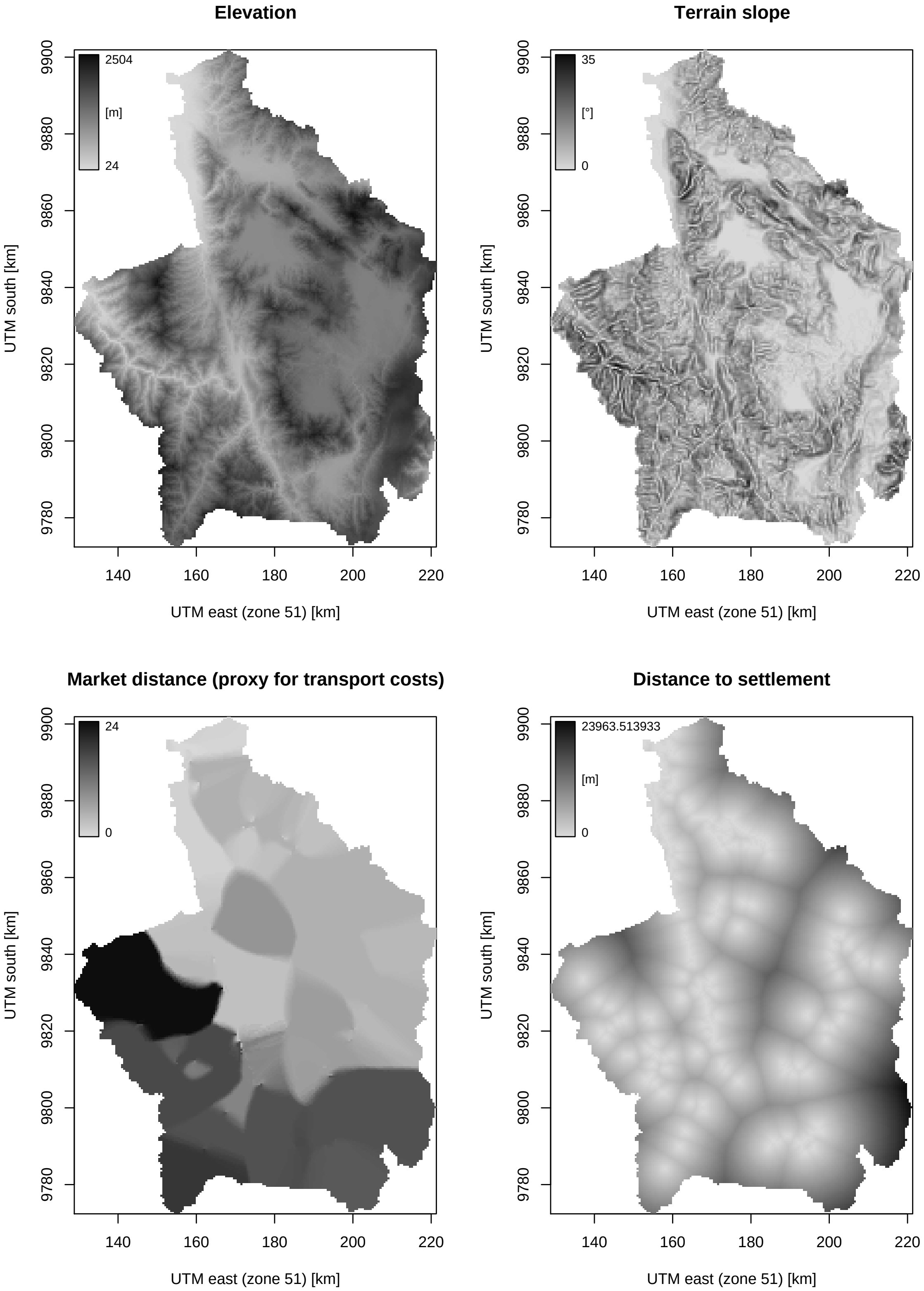


Figure 4.4: Examples for different suitability factors used to calculate overall crop suitabilities in this study. From these factors partial suitabilities in the range of $[0, 1]$ were derived that contributed to the calculation of overall suitabilities.

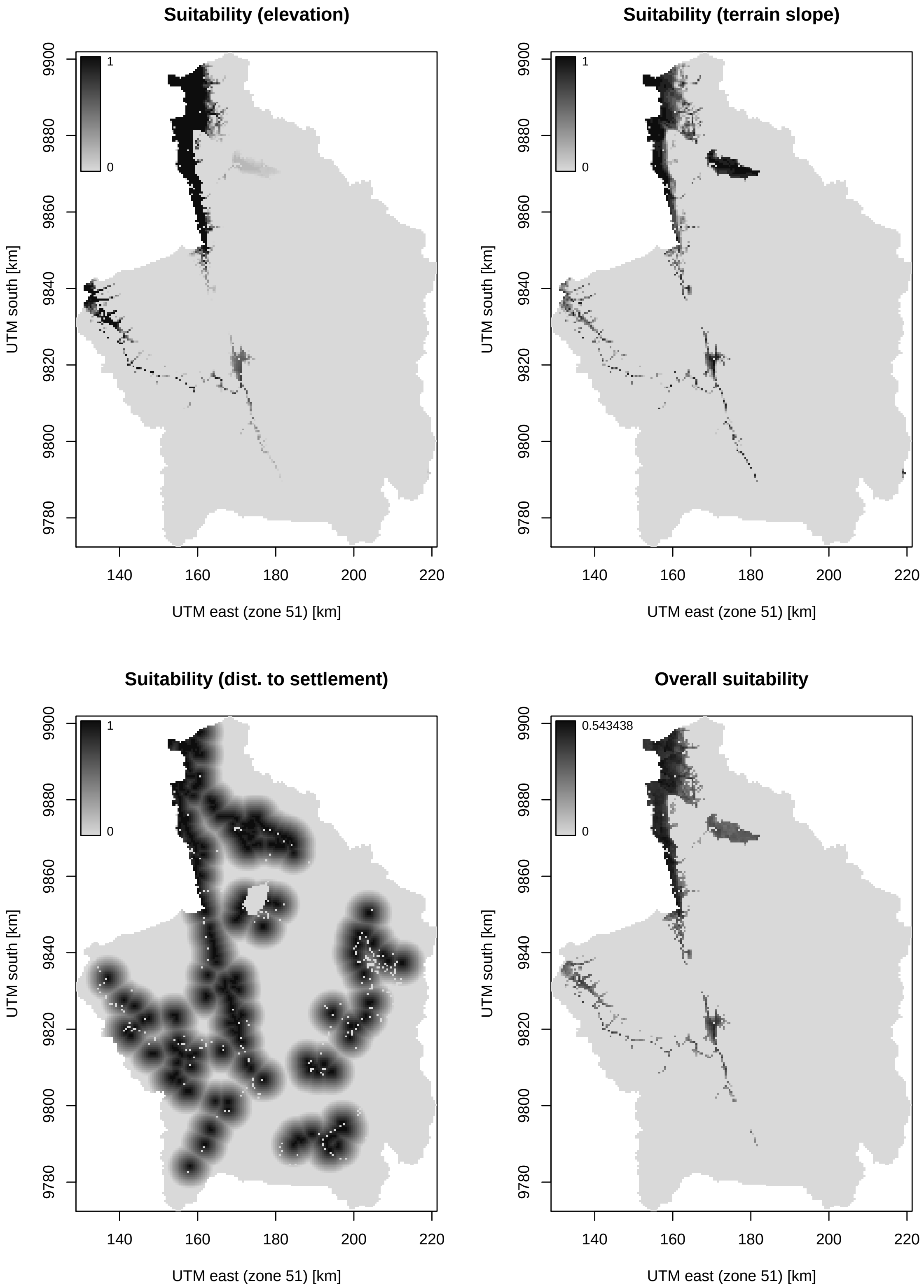


Figure 4.5: Selected partial suitabilities used for the calculation of the overall suitability map for coconut (bottom right).

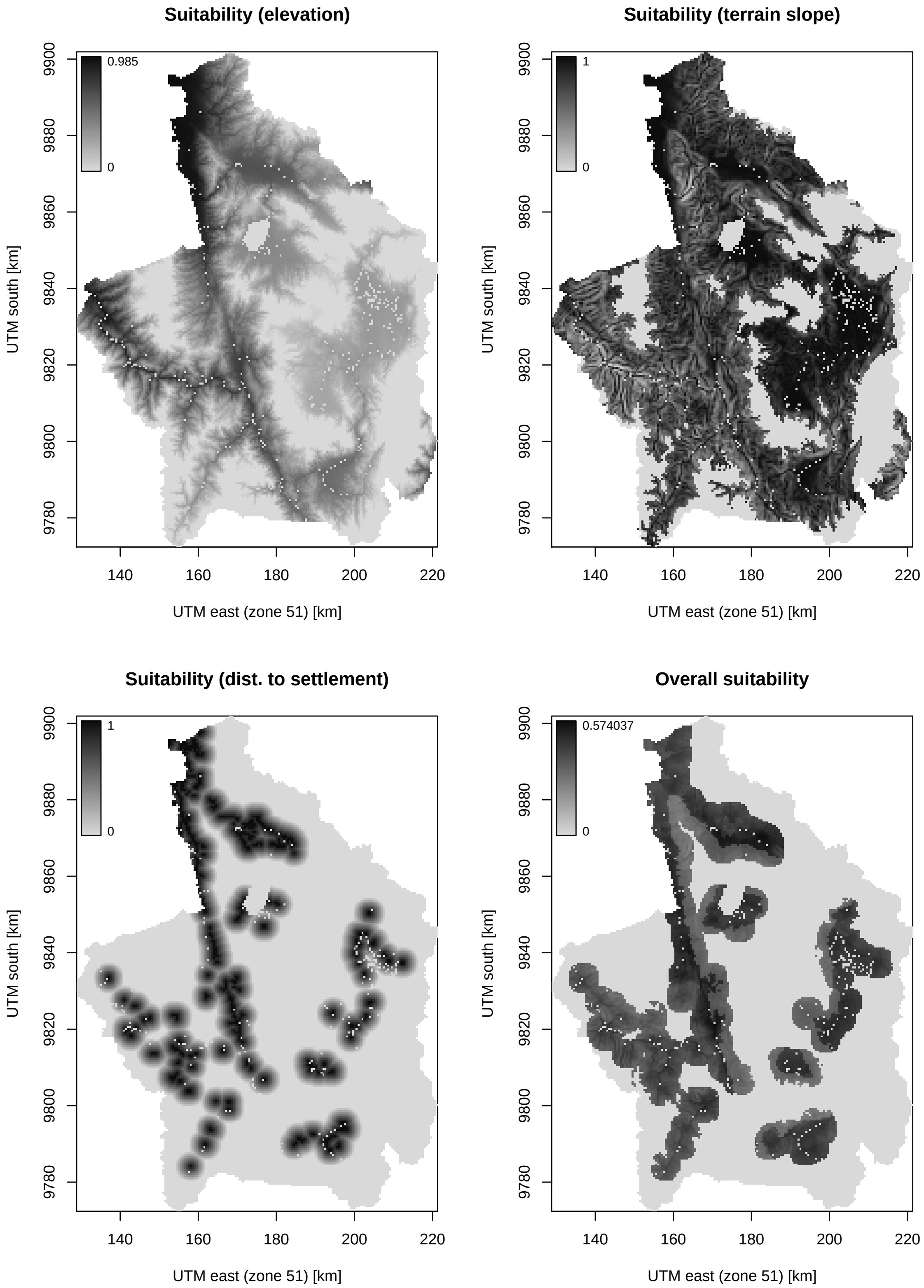


Figure 4.6: Selected partial suitabilities used for the calculation of the overall suitability map for maize (bottom right).

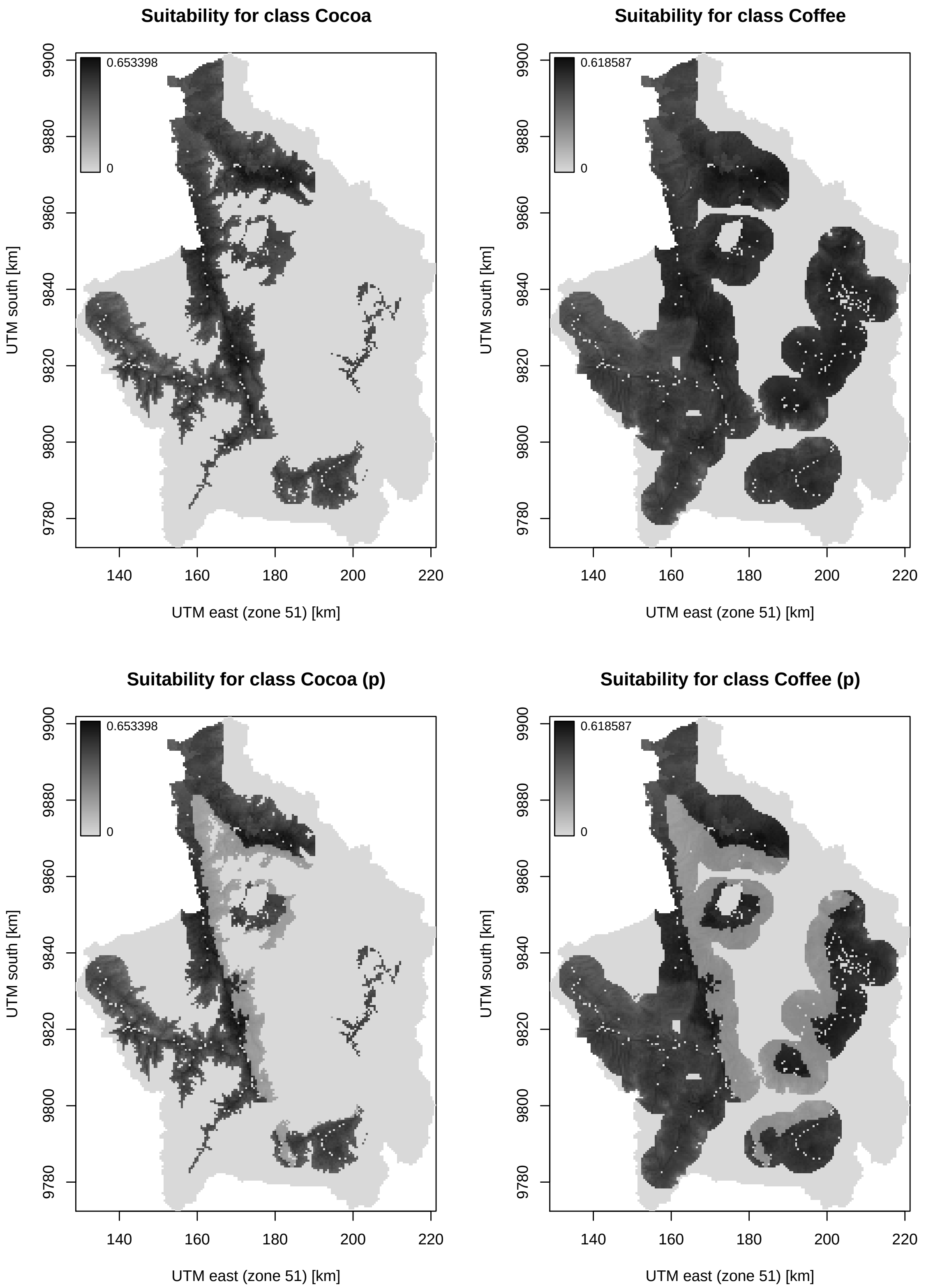


Figure 4.7: Suitability maps for the land-use classes Cocoa (left) and Coffee (right). In the upper two maps, no constraint simulating the protection status of the national park has been applied. The lower two maps have been created with a protection constraint of 0.5. Note the differences in overall suitability in the national park area.

Table 4.2: Suitability factors and constraints implemented in the STORMA land-use model. *B* denotes biophysical factors, *E* denotes socio-economic factors. Land-use classes that are not listed in this table either remain static throughout the simulation or are allocated based on a succession definition (fallow, open/secondary forest).

| Land-use class | Suitability factors | Constraints |
|---|---|---|
| Settlement | *B:* terrain slope – *E:* distance to road; distance to water; number of inhabited neighbor cells; fraction of available flat area | *E:* forest protection; area already tilled |
| Paddy rice | *B:* Terrain slope; elevation; soil fertility – *E:* distance to settlements; distance to water; adjacent paddy plots; profit margins; transport costs | *E:* area already cropped; population density in superordinate town council |
| Cocoa | *B:* terrain slope; elevation; soil fertility; precipitation – *E:* distance to settlements; adjacent crop plots; profit margins; transport costs | *E:* area already cropped; population density in superordinate town council |
| Coffee | *B:* terrain slope; elevation; soil fertility; precipitation – *E:* distance to settlements; adjacent crop plots; profit margins; transport costs | *E:* area already cropped; population density in superordinate town council |
| Coconut | *B:* terrain slope; elevation; soil fertility – *E:* distance to settlements; adjacent crop plots; profit margins; transport costs | *E:* area already cropped; population density in superordinate town council |
| Maize | *B:* terrain slope; elevation; soil fertility; precipitation – *E:* distance to settlements; adjacent crop plots; profit margins; transport costs | *E:* area already cropped; population density in superordinate town council |
| Forest use | *E:* biomass availability; recent use; distance to settlements | *E:* population density in superordinate town council |

suitability calculations for cocoa, coffee and maize. The suitability factor precipitation e.g. is assigned a suitability of 0 for precipitation values lower than 700mm. A value of 1800mm or more results in a suitability of 1 regarding precipitation. In between the interval of 700mm and 1800mm, suitabilities are calculated using linear interpolation between 0 and 1.

Constraints are calculated in a similar manner as partial suitabilities. Based on the constraint value, a reduction factor in the range of $[0, 1]$ is calculated using a functional relationship. In the present version of the land-use model, sigmoid and linear functions are used. An important constraint in the land-use model is the forest protection factor, which is applied in the overall suitability assessment of all land-use classes.

### Allocation module (ALLOC)

The set of suitability maps calculated by SUIT serves as basis for the actual land-use change decisions implemented in the ALLOC module. The ALLOC module implements a hierarchy of modules responsible for the allocation of settlements, crops, fallow land and forests. Reflecting the importance of housing and the absence of information about a land market, the allocation of new settlement areas has the highest priority and thus is situated on top of the allocation hierarchy. On the next hierarchy level, allocation of the different crop types is performed using a competitive and demand-driven approach. Rules for both the allocation of settlements and crops are implemented in form of a decision tree. Decision criteria (symbolized by the tree nodes) are evaluated consecutively, negative outcomes resulting in either no land-use change for the active grid cell, or conversion to fallow (in case a crop cell drops below a certain productivity). Allocation criteria are based on the crop-specific SUIT results, productivity (comparison of a cell's actual productivity with the mean productivity of the respective administrative unit), and on a conversion-matrix, which debars conversions that do not occur in the real world (e.g. conversion of agricultural or built up land to primary forest). If a cell can potentially change to a new land-use class or crop type, the algorithm will identify the most suitable crop. In case the cell state is already a crop class, the model checks whether the suitability for the new crop class exceeds the one of the current crop by a predefined value. Subsequently, the cell is marked for a potential change to the new crop class. However, new crops are only allocated if they are ranked higher than the actual crop in a number of consecutive years, simulating farmers' reasoning. The lag phase is crop-specific and implemented based on regional agricultural practice, expert knowledge and other sources (e.g. longer for agroforestry crops like cocoa, that require a more cost- and labor-intensive establishment compared to field crops like maize). On the next lower level of the allocation hierarchy we simulate forest areas, which can be divided into natural (both protected and unprotected) and managed forest. Fallows turn into secondary forest if they are not reconverted to agricultural land after ten years. The influence of livestock and pastures on land-use is not modeled spatially since in the current case study region, animals such as cattle, buffaloes and horses are mainly grazing on the roadside and harvested paddy fields and thus do not require additional pasture area.

**Integrated biophysical models**

The allocation step is followed by the simulation of crop growth and the growth of forests and fallows using the DAYCENT model. Environmental impacts derived from the biophysical models (e.g. changes in soil fertility, pollinator diversity or trace gas emissions) and socio-economic changes (gross margins, crop yields, etc.) are fed back to the decision algorithm to be taken into account in the suitability analysis and land-use decision simulated for the next year. In addition, we simulate changes in land-use intensities based on trend in fertilizer application over the years. Yields for the currently allocated crop classes are calculated in yearly steps. Yields that can potentially be expected for any crop class on grid cells suitable for growing any of the simulated crops are calculated every ten years. It would be preferable to simulate potential crop yields for each single year, but this is not possible due to the high run time demands. Following the calculation of crop yields, the yield for coffee are reduced according to a pollinator diversity model (Klein et al., 2003b), which is described in detail in chapter 6.

### 4.3.3 Data sources

For this study we used spatial data and statistical information from different sources to characterize the time period 1981 - 2002. Land use maps were established for the years 1981 and 2002, based on regional statistics, topographic maps, surveys and Landsat images of 1981 and 2002 classified by Twele et al. (2006). A detailed description of the

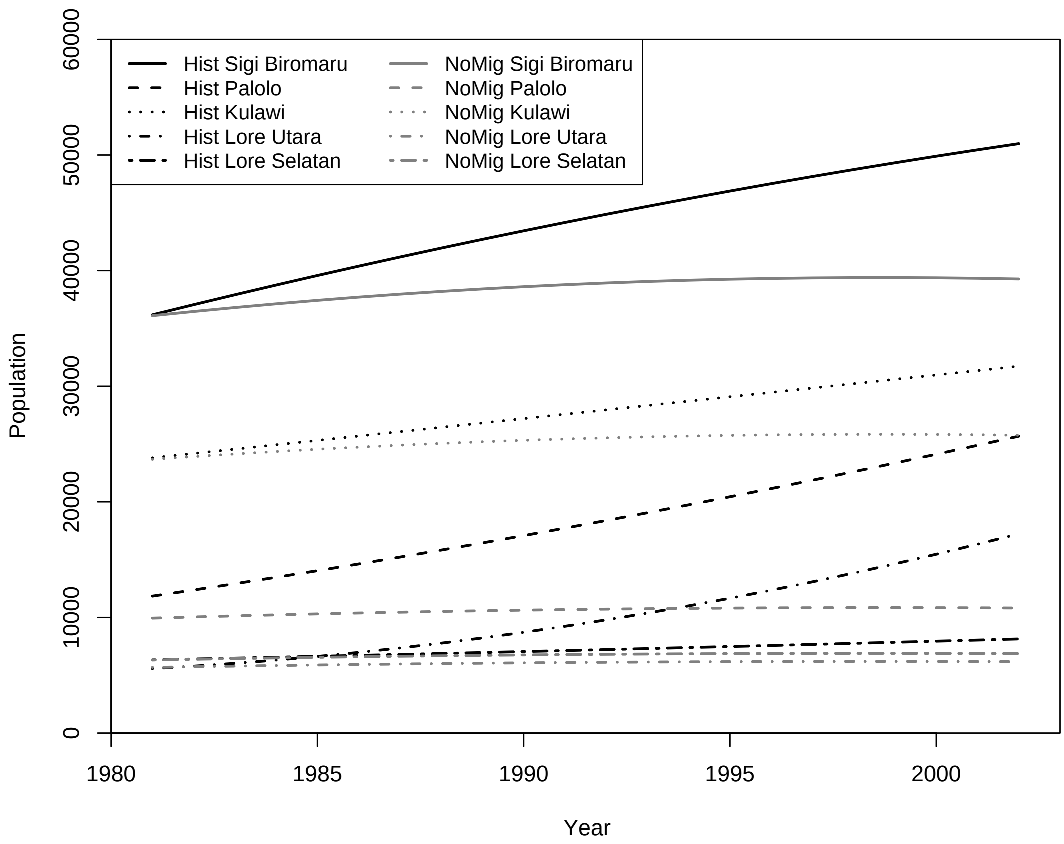


Figure 4.8: Population growth in five kecamatans of Central Sulawesi from 1980 - 2002. The black lines represent historical data while the grey lines represent the population data used for the *NoMig* scenario.

integration of spatial data sources is provided by Erasmi and Priess (2007). Times series of demographic changes, crop yields and other factors were established based on statistical yearbooks published by BPS-Palu between 1982 and 2004. Time series of producer prizes for regionally and internationally traded commodities were calculated as constant Rupiahs (base year 2000) using various sources.

## 4.4 Results

### 4.4.1 Population growth and immigration

In Sulawesi, highest net rates of immigration are found in the province of Central Sulawesi, followed by Southeast Sulawesi, while South and North Sulawesi are characterized by net outmigration (BPS, 2001). At the latest since the 1970ies the influence of inter-region migration on the population development in Sulawesi rose respectably. The population of Central Sulawesi increased by only 14% between 1930 and 1970 but by more than 41% during the period 1970-1980 (own calculations based on Kruyt (1938); BPS (1971, 1981)). In the research area, which comprises five kecamatans of Central Sulawesi, the population increased by 60% during the period 1981 - 2002. Large intra-regional differences in

Table 4.3: Land-use changes 1981 - 2002.

| Land-use type | Historical changes [ha] | *NoMig* scenario [ha] |
|---|---|---|
| Forest | -28,150 | -12,075 |
| Cocoa | 16,950 | 5,500 |
| Coffee | 4,350 | 2,550 |
| Coconut | 175 | -275 |
| Fallow | 5,875 | 10,100 |
| Grassland | -3,825 | -2,075 |
| Maize | -2,275 | -1,700 |
| Paddy rice | 4,100 | -2,475 |
| Settlement | 2,800 | 450 |

population growth were observed, ranging from 29% in Lore Selatan to a tripling of the population in Lore Utara (Figure 4.8).

Based on the annual growth rates for the rural population of Indonesia provided by the Population Division of the Department of Economic and Social Affairs of the United Nations Secretariat (2006), and the regional population statistics (BPS, 2004), we calculated the hypothetical size of the population without rural immigration, based on the population data of the year 1980. We estimated that during the 80ies and 90ies the rural population would have increased by 7,000 people (9%) without immigrants, while 37,000 people immigrated into the region, (assuming identical fertility and mortality rates of autochthonous people and migrants). Large differences in immigration rates between the five kecamatans could be observed. Sigi-Biromaru, Palolo and Lore Utara were subject to high immigration rates, while the rates calculated for Kulawi were lower and almost neglectable for Lore Selatan. In the following chapters we will link the demographic trends of the historical development and the *NoMig* scenario to land-use dynamics, biophysical and economic indicators.

### 4.4.2 Land-use change

The spatial extend of agricultural and agroforestry land, including fallows increased by 47% during the historical period, while the cultivated area inside LLNP tripled from 3,000 ha in 1981 to almost 9,000 ha in 2002. In the *NoMig* scenario agriculture expanded by 22% in the entire region, while the cultivated area inside LLNP increased by 70%. Thus, in both cases the expansion of agriculture and agroforestry within the protected areas was much faster than the general trend. The 2002 land-use situation for the historical period and the *NoMig* scenario are displayed by Figure 4.9.

As Table 4.3 shows, the contribution to land-use change varied considerably between land-use types. The by far largest expansion was observed in cocoa agro-forestry (12 fold), followed by coffee, fallows and settlement areas. Notably, the historical expansion of agricultural land used for the dominant staple food paddy rice was very moderate (20%) and even negative in case of the *NoMig* scenario (-12%). The spatial extend of settlements, (including infrastructure, public buildings and home gardens) doubled, but increased only by 20% in the *NoMig* scenario. The dominant land cover 'forest' including secondary and

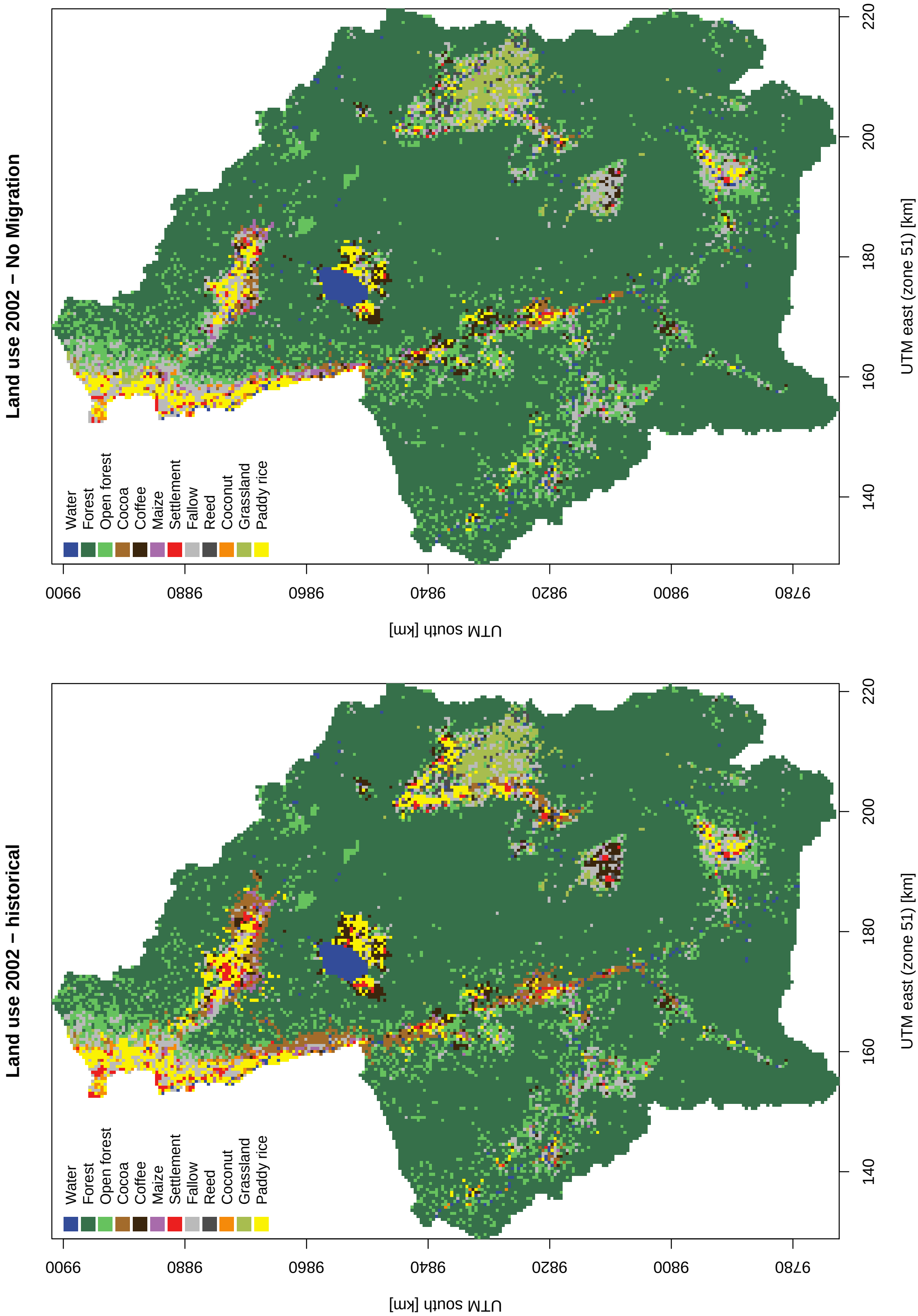


Figure 4.9: Land use maps after 21 simulated years (2002) for the historical period and the *NoMig* scneario.

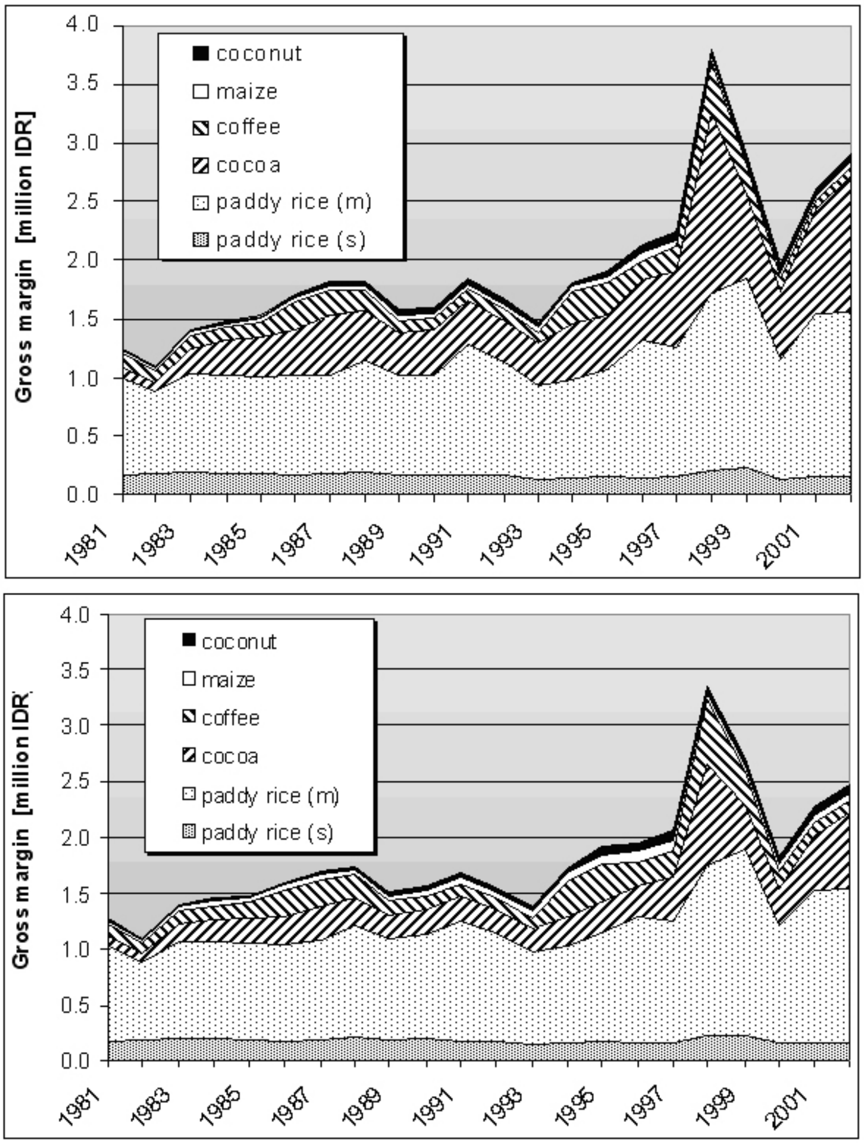


Figure 4.10: Per capita gross margins from agricultural production for the period 1981 - 2002. Upper: historical period; Lower: *NoMig* scenario; (m) indicates market production, (s) indicates subsistence production.

managed forests was reduced by 4%, while in the *NoMig* scenario forests lost 2% of their coverage, which is equivalent to 20,500 ha and 10,500 ha respectively. Large differences were observed between the five kecamatans, in which deforestation summed up to more than 13% of the low land forests of Sigi Biromaru in the North, while in the South-East less than 1% were converted in Lore Selatan.

### 4.4.3 Economic impacts

Our simulations are based on producer prices paid in Indonesia, as constant Indonesian Rupiah, base year 2000. While the prices of cocoa and especially of coffee were highly volatile, prizes for maize and coconuts (copra) increased steadily, with a minor depression at the end of the 1990ies. Prices for paddy rice paid in the provincial capital, the city of

Palu, remained stable almost over the entire period. Contrasting to falling prices of all other commodities, the rice price increased for three years during the late 1990ies, before it fell back to the original level in the year 2000.

In Sulawesi traditional agriculture was based on paddy rice, some maize, and vegetables, of which the latter are not subject of this study. After Indonesia became independent in 1945, first coffee and later cocoa were introduced as cash crops. Agriculture was and still is also the main source of income for most of the rural households (Schwarze and Zeller, 2005). Since the early 1980ies, the gross margins per capita from agricultural production slowly increased from 1.2 Mio. Rupiah to 2.3 Mio. Rupiah until 1997, when a sharp increase followed by a more pronounced sharp decrease disturbed the trend, while the last three years until 2002 gross margins recovered and finally surpassed the mid 90ies level and reached 2.9 Mio. Rupiah in 2002 (Figure 4.10). The gross margins of the *NoMig* scenario were generally lower, but followed the same general trend reaching 2.5 Mio. Rupiah in 2002. On average in the entire research area the gross margins in the historical past were 10% higher than in the *NoMig* scenario (15% in the last years). Contrastingly, due to a high fraction of cocoa, the income generated in the protected forests of LLNP was twice as high in the historical past than in the *NoMig* scenario.

During the study period the importance of maize and coconuts remained low, while the contribution of paddy rice and coffee decreased and the one of cocoa increased. In the *NoMig* scenario, the contribution of coffee remained higher, while the contribution of cocoa increased slower than in the historical past. The fraction of paddy produced for subsistence varied between 11% and 20% in both cases.

### 4.4.4 Greenhouse gas emissions

GHG emissions were simulated with new trace gas module of DAYCENT (Stehfest and Bouwman, 2006) and are presented as annual means of the sum of $N_2O$ and NO emissions. Figure 4.11 shows emission levels simulated for unfertilized and fertilized land-use and land cover types. The emissions of cocoa and paddy rice have been calculated separately for periods the 1981 - 1990 (unfertilized) and 1992 - 2002 (fertilizer applications of 20 - 60 kg N $ha^{-1}a^{-1}$). Emissions from the dominant land cover type 'forest' were on average 1.2 kg N $ha^{-1}a^{-1}$. Elevation had a negative impact on emission rates. Low land forests (0 - 1000m) emitted on average 1.4 kg N $ha^{-1}a^{-1}$, while mountain forests ($>$ 2000m) emitted only 0.9 kg N $ha^{-1}a^{-1}$. Emissions of secondary / logged forests and from maize fields were 4 times higher than those of natural forests and were the highest of all land-use types. Emissions from coffee and cocoa agro-forestry and fertilized paddy rice summed up to around 2.8 kg N $ha^{-1}a^{-1}$, while unfertilized paddy lost only 2.3 kg N $ha^{-1}a^{-1}$to the atmosphere. No differences were observed for unfertilized and fertilized cocoa plots. The lowest N emissions were simulated for fallows (1.1 kg N $ha^{-1}a^{-1}$), only slightly surpassed by coconut plantations (1.3 kg N $ha^{-1}a^{-1}$). Except for the last two categories, the nitrogen emissions of all other land-use types were 2 - 4 times higher than the emissions of natural forests.

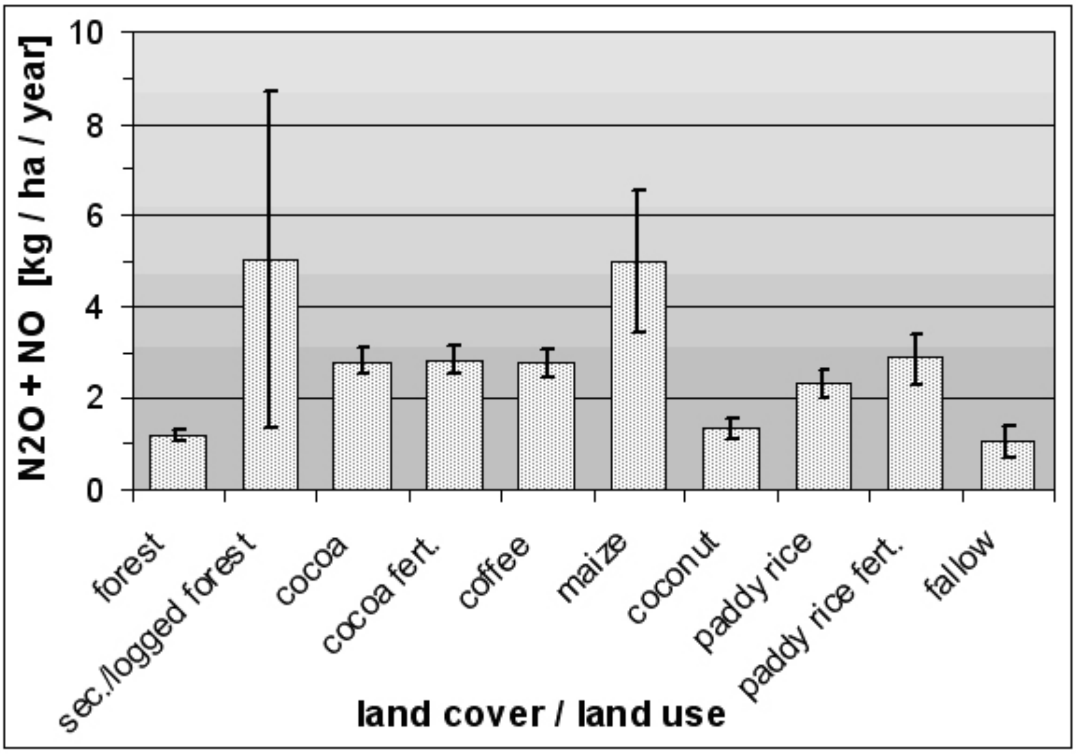


Figure 4.11: Sum of annual $N_2O$ and NO emissions of different land cover and land use types simulated with SITE/DAYCENT; columns: annual means of 5 kecamatans; error bars: standard deviation.

## 4.5 Discussion

### 4.5.1 Migration

Information networks within the ethnic groups between pioneer migrants and their families, which were still living at the places of origin, facilitated the succeeding chain migration to the forest frontier zones of Central (and Southeast) Sulawesi. Besides the disparities of land availability between areas of origin and areas of destination, the migrating farmers' knowledge about the cultivation of cacao and their financial potential to purchase land are major driving forces of the increasing migration since the late 1970ies to those regions, where the environmental conditions are favorable for cocoa cultivation (Akiyama and Nishio, 1996; Li, 1999; Ruf and Yoddang, 2001). Additionally, higher average commodity prices for cocoa (10,000 IDR/kg), as compared to lower and even more volatile coffee prices (6,500 IDR/kg) contributed to the successful establishment of cocoa plantations and the decreasing attractiveness of the already established cash crop coffee. As Weber et al. (2007) could show, the impact of migrants on deforestation is mostly indirect, as they tend to buy land from locals, rather than to cut the forest themselves. As a consequence, locals either replace the land sold, opening new plots for agriculture or agro-forestry, or move on to new forest frontiers. Thus, the social and economic status of migrants in Central Sulawesi differs considerably from those of migrants e.g. reported from Amazonia and Central America (Lopez, 2004; Renner et al., 2006). Local leaders or village headmen frequently seemed not only to be involved in land transactions, but to benefit from legal and illegal transactions, which might partly explain the "open access" policies observed in many villages, and the policy failure in protecting the LLNP.

Contrasting with the historical development, the alternative *NoMig* scenario is not only based on different immigration policies at the village and kecamatan level, but assumes the continuation of a more traditional food-crop and subsistence oriented land-use strategy and life style, which is still to be found in the remote corners of the study area, in villages which are only connected via foot-paths to the rest of the world. The environmental and economic consequences of the two different live styles are discussed in the following paragraphs.

### 4.5.2 Land-use change

During the study period, three different processes of land use change occurred simultaneously. Firstly, agricultural land use was intensified, mainly with respect to fertilizer use, a practice which started in the 1990ies. Simultaneously, a slowly advancing mechanization with two-wheeled hand tractors occurred around the same time, tractors mainly being used to prepare paddy fields (not simulated). Secondly, crop land and agro-forestry land were expanded, mainly on the expense of forests. The conversion of land cover and land use occurred not only on protected and unprotected forest land, but also from agricultural land to agro-forestry land. The latter type of conversion occurred in the context of the third process, the transition from a traditional, mostly subsistence oriented production of food crops towards an increasing production of cash crops. It is noteworthy that fertilizer subsidies intended for the stimulation of paddy rice production were found to be applied on cocoa plantations. A larger fraction of the new cocoa agro-forestry plots (as compared to the average agricultural expansion) was established within the LLNP. The fact that gross margins per hectare for cocoa are higher than for agricultural crops explains why the differences in per capita gross margins between the two scenarios are much higher in the national park (more agro-forestry) than outside (more agriculture). The still large fraction of forest cover and – as a consequence – the relatively low forest conversion rates, conceal the highly dynamic agricultural expansion, which have caused an almost 50% increase of agricultural land in the research area and an extraordinary increase of 300% increase inside the National Park.

During the period 1981 - 2002 the forest conversion rates in Sulawesi still appear moderate when compared to other tropical regions e.g. Central America where 23% of forest were converted in the same period (Carr et al., 2006). However, large scale studies and assessments may mask or overlook regional sets of socio-environmental drivers and conditions (presented in this study, or e.g. reported by Renner et al. (2006) from Guatemala), which might enable us to identify options to mitigate unsustainable land use, for example pathways to increase the still low technical efficiency of cocoa agro-forestry as proposed by Keil et al. (2007).

Comparing historical agricultural expansion to the one calculated for the *NoMig* scenario, reveals that immigration is directly or indirectly (via activities of the local population) related to deforestation, especially inside the LLNP. Obviously, local and regional authorities did not succeed in protecting the LLNP. This lack in protection was clearly reflected during calibration runs of the SITE model, during which the conceptual "park protection parameter" was repeatedly set to a value of 0 (or even small negative values) by the calibration algorithm, indicating "no protection" or even "preferred allocation" within the LLNP.

### 4.5.3 Economic impacts

At the beginning of the study period the agricultural production was mainly subsistence oriented and predominantly based on paddy rice. However, during the entire period, food security – which is a permanent concern of the central government – was never endangered, as the fraction of paddy rice needed to feed the local population varied from 11% - 20% of the regional production, assuming a per capita annual rice consumption of 131 kg (see BPS). In Figure 4.10, the fraction of gross margin used for subsistence has been calculated based on the assumptions that 90% of the rural population was engaged in paddy rice production in 1981, a fraction linearly decreasing to 60% in 2002 (to 70% in the *NoMig* scenario). If we take subsistence rice consumption into account, per capita gross margins have to be reduced on average by 9% (up to 19% at the beginning, 6% at the end of the period).

During the study period, the per capita gross margins of agricultural commodities increased more than twice, a fact which can be assumed to be the most important economic impact of land-use. Comparing the historical past to the *NoMig* scenario, the first clearly benefited from the introduction of cocoa plantations, which can mainly be attributed to the knowledge of immigrants from South-Sulawesi. Considering an average household of 5 persons, gross margins would sum up to 14.5 Mio. Rupiah in 2002, which is equivalent to 1,320 Euro, or 1,120 Euro or 15% less in case of the *NoMig* scenario (exchange rate IDR - Euro: 11,000). In other words, rural households on average profited 200 Euro per year from the ongoing immigration. Though we are aware that this calculation is a strong simplification, (for example not taking into account environmental damages, nor income distribution between locals and migrants, nor other factors), it still provides us with a means to evaluate some aspects of the economic impacts of rural immigration in Central Sulawesi. Not only immigrants, but also adaptive locals were benefiting from the cocoa boom. However, both Li (1999) and Weber et al. (2007) highlighted the complex economic and social transformations going hand in hand with land-use change, and which are partly causing tensions between ethnic groups. Additionally, pests and diseases are increasingly affecting the quality and quantity of cocoa yields and thus the profit of cocoa growers, and Neilson (2007) poses the question whether or not the economic benefits for the cocoa growers of Sulawesi can be sustained or will follow the decline which has occurred e.g. in Brazil and Malaysia.

### 4.5.4 Greenhouse gas emissions

The background emissions of $1.2\,\mathrm{kg\,N\,ha^{-1}a^{-1}}$for tropical rainforests reported in this paper are low compared to emissions from Amazonian forests ($\tilde{3}.4\,\mathrm{kg\,N\,ha^{-1}a^{-1}}$) (Neill et al., 2005), but are in good agreement with measurements from the same region by Purbopuspito et al. (2006), who reported emissions of $0.5$ - $1.6\,\mathrm{kg\,N\,ha^{-1}a^{-1}}$. Secondary/logged forests released four-fold increased amounts of N into the atmosphere. In Rondonia, grasslands also showed enhanced N-emissions after forest conversion, an effect which exponentially declined over a decade (Neill et al., 2005) Similar effects were reported from Sumatra, where natural forest emitted extremely low amounts of $0.1$ - $0.39\,\mathrm{kg\,N\,ha^{-1}a^{-1}}$, while logged-over forests released $0.69$ - $1.5\,\mathrm{kg\,N\,ha^{-1}a^{-1}}$(Ishizuka et al., 2002).

Weitz et al. (2001) reported emissions of 1.2 kg N $ha^{-1}a^{-1}$and 2.7 kg N $ha^{-1}a^{-1}$from unfertilized and fertilized maize fields in Costa Rica. The high emissions of 5.0 kg N $ha^{-1}a^{-1}$of maize fields reported in this study, can partly be attributed to the facts that they were located on recently converted forest land and can be expected to decrease with time, as reported by Neill et al. (2005) from Amazonia.

All land cover and land-use changes, except fallows, caused considerably higher emissions of $N_2O$ and NO. While the high emissions from logged forests and recently converted maize fields can be expected to decrease over time, the average emission levels of all other land-use types are still at least twice as high as the level of natural forests, and can be expected to increase in the future due to increasing application rates of Nitrogen fertilizers. While this type of information is important in a global change context, especially in a region for which data and estimates are still scarce, see e.g. Weber et al. (2007), we assume no relevance for local stakeholders' land-use decisions.

## 4.6 Conclusions

In this paper we presented a study on land-use dynamics and selected socio-environmental impacts in Sulawesi, Indonesia. We used the integrated land-use modeling framework SITE as a tool to simulate historical land-use change and an alternative *No Migration* scenario. The modeling framework was capable to process relevant driving forces and to simulate major components and impacts on the socio-environmental systems of Central Sulawesi.

Land-use dynamics in the study region are characterized by strong expansion of agricultural and agro-forestry land, and the consumption of natural resources. While the livelihood of the rural population clearly improved during the study period, the introduction of cocoa plantations – mainly triggered by the knowledge and the capital of immigrants – provided additional economic benefits. The economic achievements are considered unsustainable, because they are based on forest conversion and an increase in unproductive fallow land. Major challenges in Central Sulawesi seem to be (i) to slow down the consumption and conversion of forest resources, and (ii) find means to regulate access to land. In addition, it seems essential to reduce the growing impacts of pests and diseases in cocoa agro-forestry in order to avoid the classical cocoa pathway of "boom and bust" as depicted by Neilson (2007).

In the context of the STORMA research project, the dissemination of scientific findings is facilitated by pathways of communication with villagers and authorities and mutual trust established during several years of cooperation. Following Matthews et al. (2006), the established social structures are considered more important for communication than technical and scientific aspects of the SITE model. Potential limitations in communication with stakeholders are expected (i) to arise from conflicting interests of different stakeholders (and scientists), for example the regional government and local farmers on the topic of agricultural expansion, (ii) lack of awareness and understanding of the concerns and risks perceived by the respective other parties, and (iii) a limited set of scenarios and indicators provided by SITE, (of which only a subset has been presented in this study).

The current discussion among STORMA scientists how to communicate simulation results like the ones presented in this paper, includes aspects of (i) whether or not to confront stakeholders with (ii legal aspects of land-use change, (ii) whom to address from the large and very heterogeneous group of stakeholders, (iii) which format would be appropriate (narrative, tables, maps, images) and to which degree risk perceptions of stakeholders – which are currently evaluated – should be discussed before integrating them into scenarios of the future. We currently discuss to use the "good news" discussed in section 4.5.3, to transport the message both to farmers and politicians that "everything comes at a price" (e.g. the negative impacts discussed in section 4.5.2) and that we not just want to inform stakeholders, but also seek their help and advice to (i) address relevant issues in the scenarios and (ii) identify potential pathways into the future decreasing socio-environmental costs and increase the benefits e.g. via improved efficiency (Keil et al., 2007) and better control of agricultural expansion.

# 5 Automated model calibration

Modeling and simulation of land-use change is a field of increasing importance and a broad variety of techniques have been developed for this discipline. Especially for highly parameterized models, the subject of model calibration is crucial concerning the accuracy of simulation results. Different calibration approaches have been employed by the scientific community. In this study we present the genetic algorithm (GA)-based calibration of a complex land-use model to examine the dynamics of tropical rainforest margins in Sulawesi, Indonesia. We used SITE, a new land-use modeling framework that provides generic functionality for running, testing and calibrating land-use models. The land-use model explicitly simulates twelve land-use types and exposes 24 different parameters as calibration candidates. Due to model complexity and high run-time requirements, the number of parameters was reduced by performing a sensitivity analysis to select calibration parameters and employing a two-step calibration.

## 5.1 Introduction

State of the art land-use change models increasingly are simulating both environmental and socio-economic aspects of land-use (Verburg et al., 2004; Heistermann et al., 2006). Modeling approaches include regression techniques (CLUE) (Veldkamp and Fresco, 1996; Verburg et al., 2002) or multi-agent systems (MAS) (Parker et al., 2003), rule based cellular automata (CA) models (Rubiano, 2000; Jenerette and Wu, 2001), or hybrid approaches (Manson, 2001; Berger, 2001). While empirical models based on regression techniques may be easier to set up and do not require calibration, MAS and CA approaches are more process oriented and thus provide much more explanatory power. It has been shown that CA models are suitable for adequately modeling and simulating spatial dynamics of land change and urban growth (Itami, 1994; White and Engelen, 1997, 2000; Clarke and Gaydos, 1998; Candau, 2000). The quality of the results is dependent on adequate formulation of the transition rules, determining when, where and how the grid cells, which are representing the environment, change from one land-use type to another. Typically, the rule sets, in which such transition rules are formulated, are generated based on empirical information and a number of parameters whose values cannot be derived empirically, but are subject to calibration.

Manual model calibration is not an appropriate method for both its unreliability and its lack of reproducible outputs (Oliva, 2003). For this reason, the topic of developing methodologies for automatically calibrating land-use and other modeling systems got an increasing relevance over the past years (Boumans et al., 2001; Oliva, 2003; Straatman et al., 2004). Since for most applications the problem can be tracked down to finding an adequate set of values for the defined tuple of parameters to calibrate, a number of methods based on mathematical optimization algorithms or heuristics and techniques from the field of machine learning are adapted and applied by the LUCC community (Recknagel, 2001; Seppelt and Voinov, 2002). Prominent methods are Monte-Carlo derived algorithms (Xian et al., 2005) or neural networks (Li and Gar-On Yeh, 2001; Pijanowski et al., 2005). Those methods, however, usually are computationally expensive. Especially neural networks require intensive training, which may imply a large number of simulation runs and thus result in unacceptable CPU time. Desired computationally less demanding optimization heuristics need to be capable of searching over non-linear objective functions and have to be robust to deal with noise, discontinuities and search spaces of high dimensions. Depending on non-linearities of the underlying model, those objective functions may be fulfilled for local optima. Discontinuities in the search space may arise due to the nature of the underlying model parameters. These factors plus the need to economize on the number of simulations during calibration runs show the need for algorithms that can direct their sampling inside the search space (Miller, 1998).

Examples for such optimization heuristics are hill-climbing, simulated annealing, genetic algorithms and gradient estimation techniques (Miller, 1998; Ndiritu and Daniell, 2001; Otten, 1989). Hill climbing algorithms are simple and widely used procedures which start upon an initial random solution within the search space which is defined as the status quo. Based on a definition of neighborhood inside the search space, a new solution is chosen randomly from that neighborhood in each iteration. In case the new solution results in an improved value of the objective function, it is regarded the new status quo. Otherwise, the status quo remains unaltered. This process usually is continued for a fixed number of iterations with the last status quo being used as the ultimate solution (Miller, 1998).

Simulated annealing algorithms (Kirkpatrick et al., 1983)differ from hill climbing in that they might accept inferior solutions with a probability that is decreasing in the size of the loss of “temperatur” and “cooling speed” parameters that make such acceptance less likely over time (Miller, 1998). A large number of applications of simulated annealing can be found in the field of calibrating hydrological models (Cooper et al., 1997; Sumner et al., 1997; Thyer et al., 1999; Abdullah and Al-Badrahnih, 2000).

For complex optimization problems, genetic algorithms (GAs) have proven to be an effective search technique. They were first developed by Holland (1975) as a method of mimicking meiosis in cellular reproduction to solve complex problems (Goldstein, 2003) and were later adapted as an optimization procedure (Goldberg, 1989). However, GAs should rather be seen as a search process than strictly as an optimization process, since they do not necessarily yield globally optimal solutions (Holland, 1975; Whitley, 2001). Nevertheless, they are capable of finding an adequate solution of a parameter set in a relatively small number of iterations (Holland, 1975; Mitchell, 1998), which is a good argument to use them for computationally demanding applications. There are already a few examples in applying GAs in the field of land-use modeling, especially for models dealing with urban growth (Wong et al., 1998; Jenerette and Wu, 2001; Wong et al., 2001; Goldstein, 2003; Stewart et al., 2004).

In this paper we present the application of an automated generic calibration approach using genetic algorithms implemented in the SITE model The case study in which the SITE model is used is an analysis of the stability of tropical rainforest margins in Indonesia (see chapter 4). To examine land-use dynamics at the forest frontier, it is crucial to calibrate a number of parameters of the underlying rule set using datasets from a historical time period (1981 to 2002). In contrast to previous work, we will apply the calibration methodology to a highly parameterized model that explicitly addresses the twelve main land-use classes that are found in the study area. In addition, we present a generically applicable calibration approach instead of a tailor-made implementation for a specific land-use model.

## 5.2 Methods

### 5.2.1 Optimization procedure

Rule set calibration within the SITE framework is performed by finding an adequately good solution for a parameter set defined in the respective rule set script using an optimization algorithm. Since running simulations with the SITE framework is computationally expensive depending on the rule set (for the STORMA application, the run time of a 21 year simulation ranges between three hours on a dual core PC with 2 parallel DAYCENT server instances and a minimum of 1 hour, using up to 10 parallel DAYCENT instances), we were looking for an optimization methodology that is capable of searching the defined parameter space efficiently, i.e. to find an adequate solution for the parameter set in a minimum number of iterations.

Among the methodologies available we decided to use GAs to calibrate the land-use model, since they deliver adequate results after a relatively small number of iterations and thus decrease the computational expenses of SITE simulation runs. To implement GA functionality, we used the freely available GALib software library (Wall, 1996) which was linked to the calibration component of the SITE framework.

Figure 5.1 schematically displays the workflow of GAs. The algorithm starts based on a set of potential candidate solutions, which form the initial population. This population can be created by random or non-random methods. The initial population is then evaluated by applying the objective function to each candidate solution. The objective function returns a score value on which a fitness value is assigned to the respective candidate solution. Based on the score values the algorithm checks whether the GA termination criteria are met. A number of different methods are available for this task, including a predefined number of simulation runs, a shrinking variance in the solutions presented, a predefined threshold value for the candidate with the best fitness, or, ideally, obtaining the optimal solution. During each iteration, the GA creates a new population of solutions. New populations consist of reproduced old solutions, and new solutions created by so-called genetic operators. Typically, GAs adopt two such operators: crossover and mutation. Crossover recombines pieces of existing solutions in a way that tends to result in better solutions. The selection of candidate solutions for crossover is based on their respective fitness. The mutation operator randomly applies small alterations to a solution in order to prevent the system from getting prematurely trapped in local optima. As soon as the new population has been created, it is evaluated as described above. Iterations are performed subsequently until the termination criteria are met (Miller, 1998).

GALib provides a number of possibilities to parameterize a GA. We applied the following parameterizations in our study:

**Algorithm class:** Simple GA or Steady-state GA. While simple GAs create an entirely new population for each generation, Steady-State GAs use overlapping populations, in which the amount of the population to be replaced in each generation is also subject to parameterization. For our study, a Steady-state GA was used.

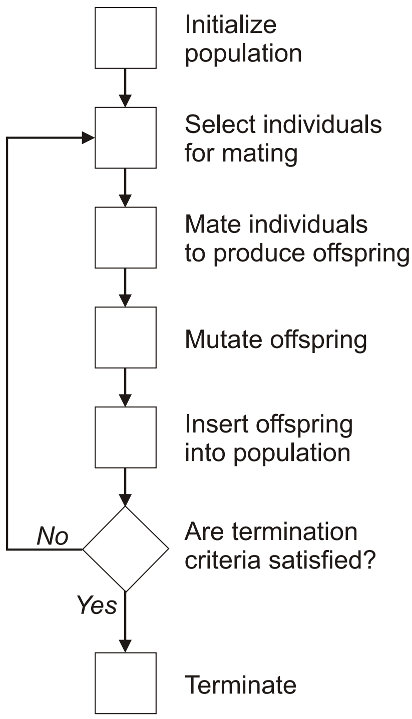


Figure 5.1: Flowchart of a generic GA.

**Population size:** Populations in our study were small, consisting of 25 individuals for the calibration of settlement parameters and 20 individuals for the overall model calibration.

**Crossover probability:** set to 90%

**Mutation probability:** Due to the small population used in our study, mutation probabilities were set to 10% for settlement calibrations and 20% for the overall model calibration.

**Maximum number of generations:** Second termination criterion in case threshold cannot be reached within a predefined number of iterations. In this study it was set to 30 generations for settlement calibrations and 15 for the overall model calibration.

Since the rule set of this case study implements a hierarchical structure, performing settlement allocation first, followed by crop allocation and simulation of forest use, it was possible to use a two-step approach for model calibration. In the first step, we calibrated parameter values that exclusively have effect on the settlement class while disabling the crop and forest allocation hierarchy level in the rule set script. After the best parameter values had been derived from this first calibration step, the calibrated parameters were established in the rule set and other parameters that do not have an effect on settlement allocation were calibrated in a subsequent step. Since exclusively simulating settlement allocation did not involve the use of the DAYCENT model which consumes most of the run time, we were able to run a larger number of calibrations for settlement allocation.

The initial GA populations were created by a random process. For each parameter we defined value ranges and increments, thus creating a pool of values of identical probability for the initialization procedure.

### 5.2.2 Objective function

The objective function evaluates the quality of every GA candidate solution during the optimization process. Depending on the type of model to be calibrated, different methodologies to calculate the fitness of a single solution can be applied as objective function. The SITE calibration component assesses simulation results using one of several map comparison algorithms provided by its model test component. The map comparison value is used as score for the respective candidate solution. In this case study, an independently compiled 2002 land-use map serves as reference for the simulation quality assessment. It represents the final state of the calibration period. For the SITE model this means, the better the outcome of a simulation corresponds with the 2002 reference land-use map, the higher is the fitness value assigned to the respective parameter set. Map comparison algorithms provided by the SITE framework summarize the degree of similarity in one single value, typically in the range between 0 (no fit) and 1 (identical). This makes them suitable for application within the optimization methodology which requires a single objective value to derive the fitness or quality of a candidate solution.

Research on map comparison methodologies is considered highly important and received growing interest in the modeling community over the past years (Monserud and Leemans, 1992; Winter, 2000; Pontius, 2000; Pontius and Schneider, 2001; Power et al., 2001; Kuhnert et al., 2005). However, many map comparison techniques have limitations compared to human visual comparison, especially with respect to the comparison of patterns. Addressing this, the prominent Kappa method has been examined for its capability to discriminate errors concerning the quantity and the location of corresponding cells (Pontius, 2000). Pontius (2002) combined the kappa method with moving window procedures to allow kappa calculations for multiple resolutions. However, kappa tends to underestimate changes. A recent comparison of the performance of different land-use models by Pontius et al. (2007) utilizes the figure of merit measure, which is the ratio of the intersection of the observed and predicted change to the union of the observed and predicted change (Klug et al., 1992; Perica and Foufoula-Georgiou, 1996). We found figure of merit to be best suited for our task. In addition, it allows to range our model performance in the model comparison study by Pontius et al. (2007). As goal for a successful calibration we defined a figure of merit for our calibrated model, that is of similar value as model of comparable observed net change listed in the Pontius et al. (2007) study (a value between 0.05 and 0.07).

### 5.2.3 Sensitivity analysis

Due to the high run time requirements which limit the population size and the number of possible generations, it is essential to analyze which rule set parameters are most promising for model calibration, i.e. which parameters have the strongest influence on model results. We defined two formal criteria for parameters to be used in the model calibration:

Table 5.1: Set of rule set parameters that address the allocation of settlements. Values assigned to those parameters after calibration are shown in the right column. Note that weights (S4 to S8) are normalized after value selection so that they sum up to 1.

| No. | Description | Calibrated value |
|---|---|---|
| S1 | Maximum fraction of cell area to be covered by settlements in case not all neighboring cells are inhabited, representing lower population densities at settlement margins | 0.6 |
| S2 | Maximum initial fraction of area of a cell covered by settlements | 0.09 |
| S3 | Parameter simulating resource limitation by reducing settlement suitability, a proxy for increasing land prices | 5.0 |
| S4 | Weight of suitability regarding distance to the closest road | 0.9 |
| S5 | Weight of suitability regarding distance to closest occurrence of water | 0.0 |
| S6 | Weight of suitability regarding number of populated neighbor cells | 0.3 |
| S7 | Weight of suitability regarding share of flat area inside the administrative boundaries of a village | 0.7 |
| S8 | Weight of suitability regarding terrain slope of current grid cell | 0.6 |

1. The parameter value is either unknown or cannot be derived with sufficient precision from empirical evidence.
2. The model outcome reacts sensitive to a change of the respective parameter values.

A list of all parameters that meet the first of the two prerequisites is given in Table 5.2, including a short description of the function of each parameter. Parameters P1 to P6 are distance thresholds which, if exceeded, lead to a suitability of 0 for the respective land-use class. Parameters P11 and P12 are weights assigned to partial suitabilities in the multicriteria analysis. Parameters P9, P13, P14 and P15 contribute to the calculation of a reduction factor, which is part of the multicriteria analysis used to determine the suitability of a cell for each land-use class. P7 and P8 do not take effect during suitability assessment, but are used as decision thresholds throughout the land-use allocation process. Parameters P16 and P17 are part of the DAYCENT crop model. They influence the efficiency of the conversion from solar energy to cocoa and coffee biomass respectively. In contrast to the other crops, yield levels for cocoa and coffee had a relatively weak empirical basis, which is the reason why we selected these parameters as candidates for calibration.

To answer the question which of the selected parameters fulfill criterion 2 (sensitivity of the model outcome to changes of a parameter value) we performed a sensitivity analysis. Each parameter was assigned different values covering the respective value range (see Table 2). For any parameter value a simulation run was executed, while we kept all other parameters constant at their respective default values (bold numbers in Table 2). The influence on the simulation result was quantified by calculating the figure of merit based on the initial 1981 map, the simulated map (period of 21 years) and the reference map of the year 2002. Since parameters P16 and P17 directly influence crop yields, we

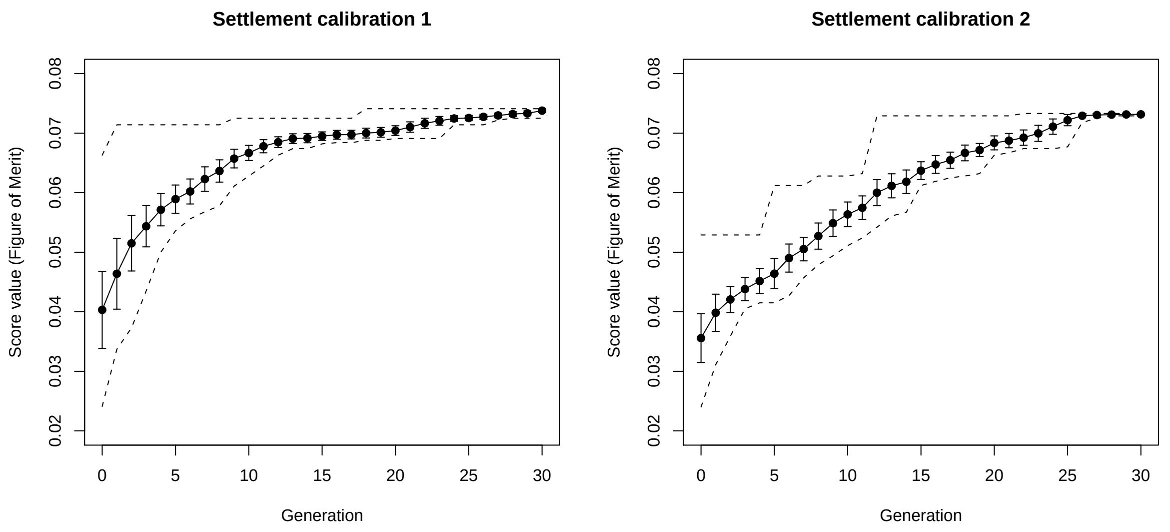


Figure 5.2: Results of the calibration of the settlement class. The GA generation number on the x-axis is displayed against score value statistics on the y-axis. The dotted center line represents the progression of population score mean values throughout the GA execution. Score standard deviations for each generation is depicted by the bars. The upper dashed line indicates the score of the best candidate solution, while the lower dashed line represents the lowest score value for each generation.

regarded it obvious that they have an impact on the overall simulation result and did need to be included in the sensitivity analysis. Thus, the sensitivity analysis resulted in fifteen simulation series with each series consisting of 5 to 8 simulations. In addition, we calculated corresponding figure of merit values for each land-use class separately by merging all other classes to a rest class to derive more detailed information about how single land-use types contributed to the overall objective measure.

## 5.3 Results

### 5.3.1 Calibration of the land-use class settlement

Three different calibration samples were the base for the following analysis (The parameters used to calibrate the settlement class are listed in Table 5.1). Figure 5.2 shows the progression of GA populations for two different settlement calibration runs. All progressions followed the same pattern. Starting from a randomly created set of candidate solutions forming the initial population (generation number 0), the score mean values were continuously increasing (a decrease of score mean values is impossible based to the definition of how new populations are established for each new GA generation). Usually, the lowest score value increased in each generation. Advancements of the score maximum typically occur discretely. In calibration sample 1, for example, the maximum score value of generation 0 is already high, and was slightly increasing in generations 10 and 19. Calibration sample 2 showed a different progression, starting with a lower maximum score value. This score was improved four times resulting in a similar final maximum score as in sample 1. The parameterization of the worst candidate of generation 0 and the best candidate of generation 30 improved by 310% in all 3 calibration runs, in terms

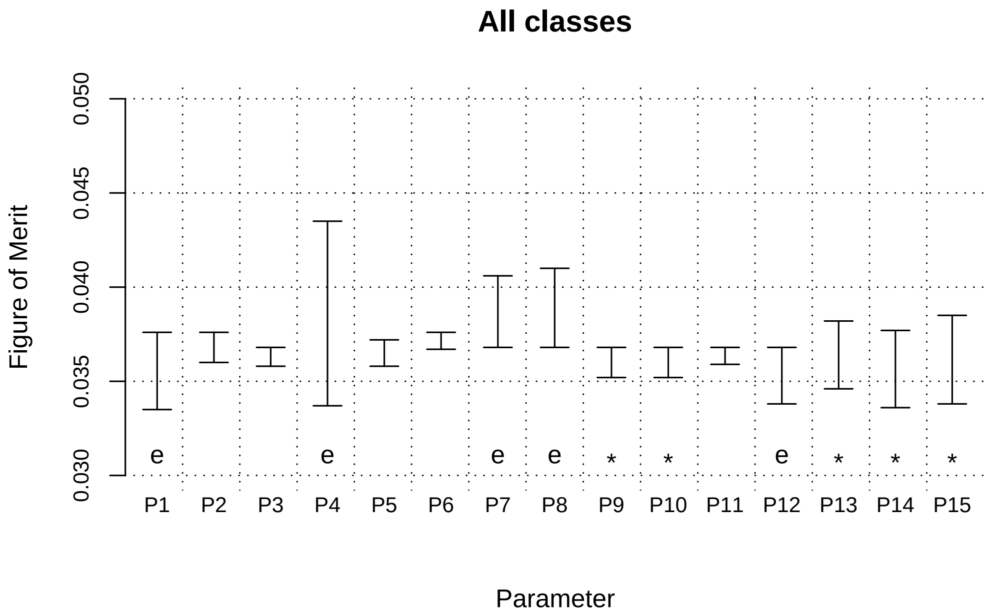


Figure 5.3: The calculation of the figure of merit was based on all classes. Parameters labeled with "c" were not considered for calibration since respective values could be derived empirically after sensitivity analysis. Values labeled with "*" were selected for calibration.

of figures of merit. Comparing the best candidates of the 1st and the last generation yielded parameter sets improved by 10% to 40%. As expected, the variability strongly decreased with the increasing number of generations, which means that most parameters of the different candidates showed similar numerical values.

### 5.3.2 Sensitivity analysis

An evaluation of the sensitivity analysis for the entire model is depicted by figure 5.3 which shows the figure of merit ranges for each rule set parameter series based on all land-use classes. A number of parameters can be identified that clearly have a higher impact on the simulation result than others, namely P1, P4, P7, P8, P12, P13, P14 and P15. Figure of merit values are relatively small because of the large number of forest cells that remain unaffected by the land-use change rules compared to the number of changed cells. Therefore it is also necessary to examine the contribution of single land-use classes to the overall figure of merit.

Sensitivity evaluations for a selection of land-use classes are shown in Figure 5.4. Note that the range of the values on the y-axis is larger than in figure 5.3. We observed figure of merit values between 0.0298 and 0.4553. It is obvious, that land-cover/land-use classes which cover large areas such as forest show less variation than smaller ones like coconut. Consequently, the range of the overall measure was strongly limited by the dominant forest class. The effect also occurred when testing other map comparison algorithms (e.g. kappa, results not shown here). Thus, the narrow range of the figure of merit is primarily caused by the land-cover/land-use distribution of the study region and not by a limitation of the test methodology. Different effects could be observed for

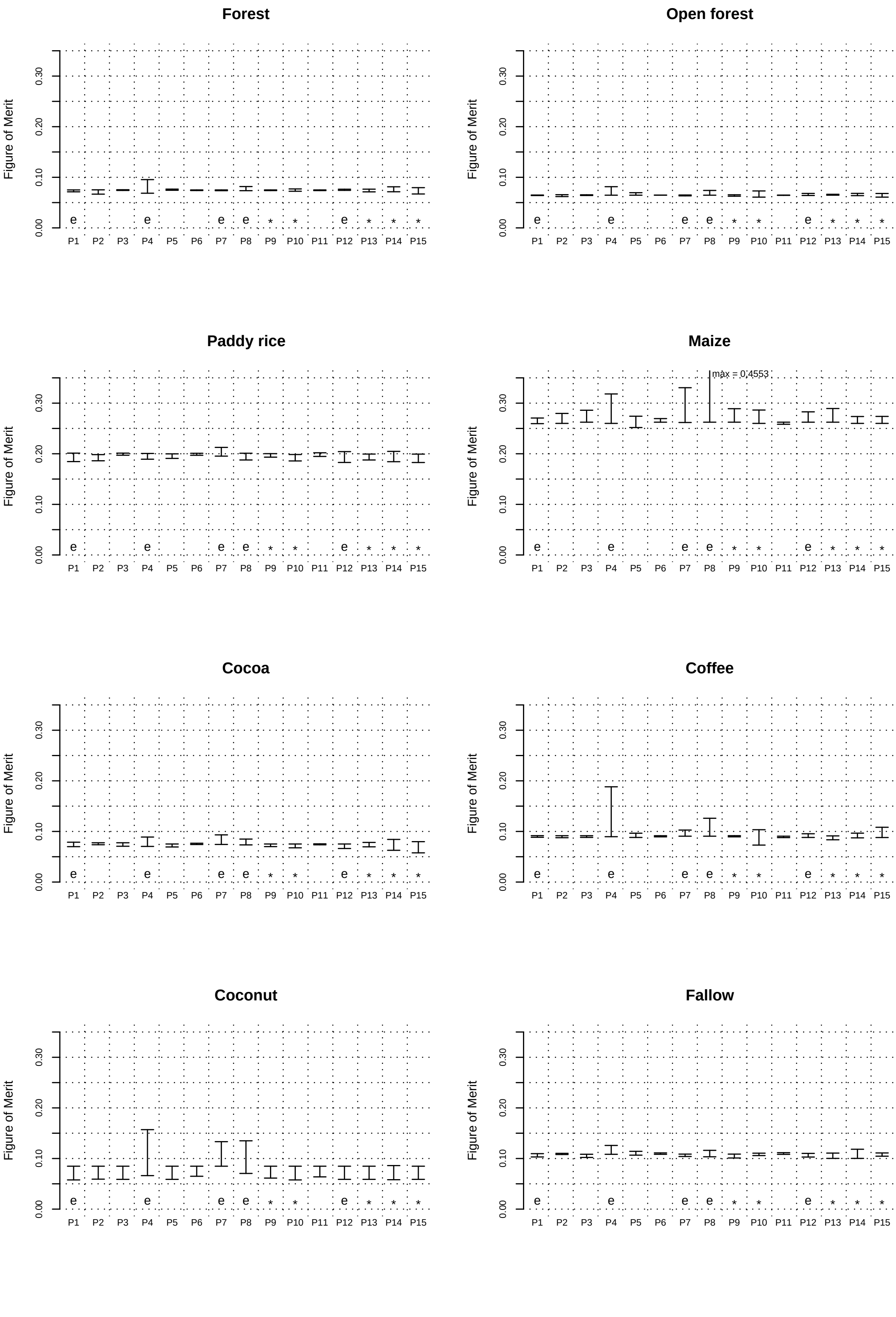


Figure 5.4: Figure of merit variations for parameters P1 to P15. Calculations are based on indicated classes with all other classes subsumed to one rest class. Parameters labeled with "c" were not considered for calibration since respective values could be derived empirically after sensitivity analysis. Values labeled with "*" were selected for calibration.

the coconut, coffee and maize crop classes which cover smaller areas. The parameters showing high variation in the map comparison series based on all classes usually are also among the ones with high variation in the single class analyses. Different impacts on different land-use classes could be observed for parameters P1 to P6. These parameters are distance thresholds used to calculate suitabilities for different land-use classes and thus directly only affect the respective class. This was especially obvious for maize, cocoa and coffee. Nevertheless, since the rule set implements a competitive approach for land-use allocation, other classes are affected, too. A good example for this effect are the classes cocoa and coffee, since most grid cells suitable for cocoa are also well suited for coffee, which explains the high variations of the cocoa class when altering the specific distance threshold for coffee. Parameters P7 to P15 affect all classes except settlement simultaneously, with the exception of P13, which affects only the suitability calculation for possibly populated cells. Variations observed for the Fallow class are relatively small, but they also correspond well with the overall variations.

Valuable information was also generated by analyzing the effect that varying selected parameters had on the figure of merit (figure 5.5). As the plots show, the values reach certain maxima or minima for specific parameter settings. Especially for parameter P4 (max. dist. between settlement and cocoa cells) this is obvious, where an extreme value for most land-use classes can be identified for a parameter value of 6000m. In particular, allocation of coffee, coconut and open forest reacts strongly to this parameter, but all other crop cells are clearly affected, too. Other distance parameters, which are not displayed in Figure 6 show similar patterns. For these cases, local and global maxima were tested for plausibility with empirical data derived from field studies like surveys or spatial analyzes of reference maps.

Parameter P9 (national park protection) is strongly influenced by distance parameters like P4. The larger the distances are, the more settlement or crop cells can be allocated inside the protected areas and the more cells are affected by the corresponding factor simulating forest protection. Consequently, simulating larger activity ranges of the local farmers via increased distance parameters (P1 to P6) caused larger variations.

Parameters for which we achieved a good agreement between the the results from the sensitivity analysis and empirical evidence (labeled with “e” in figures 5.3 and 5.4; see also parameter progression plots figure 5.5) were not calibrated. Thus, Besides parameters P16 and P17, which directly affect crop yield levels, we selected parameters P9, P10, P13, P14 and P15 for model calibration.

### 5.3.3 Overall model calibration

Figure 5.6 shows the population progression for the overall model calibration. Despite of the lower number of generations, the same pattern can be observed as for the calibration of the settlement class. Minimum and mean score values steadily improved with each generation, while increases of the maximum score value occurred discretely. The variability decreased with increasing number of generations, but in the final generation was considerably larger than the variability of the settlement parameters. This was reflected in larger differences of the numerical values of the parameters in the final generation. The best final parameterization of the best candidate was improved by 40% compared to the worst and by 8% compared to the best candidate of generation 0 respectively. Score

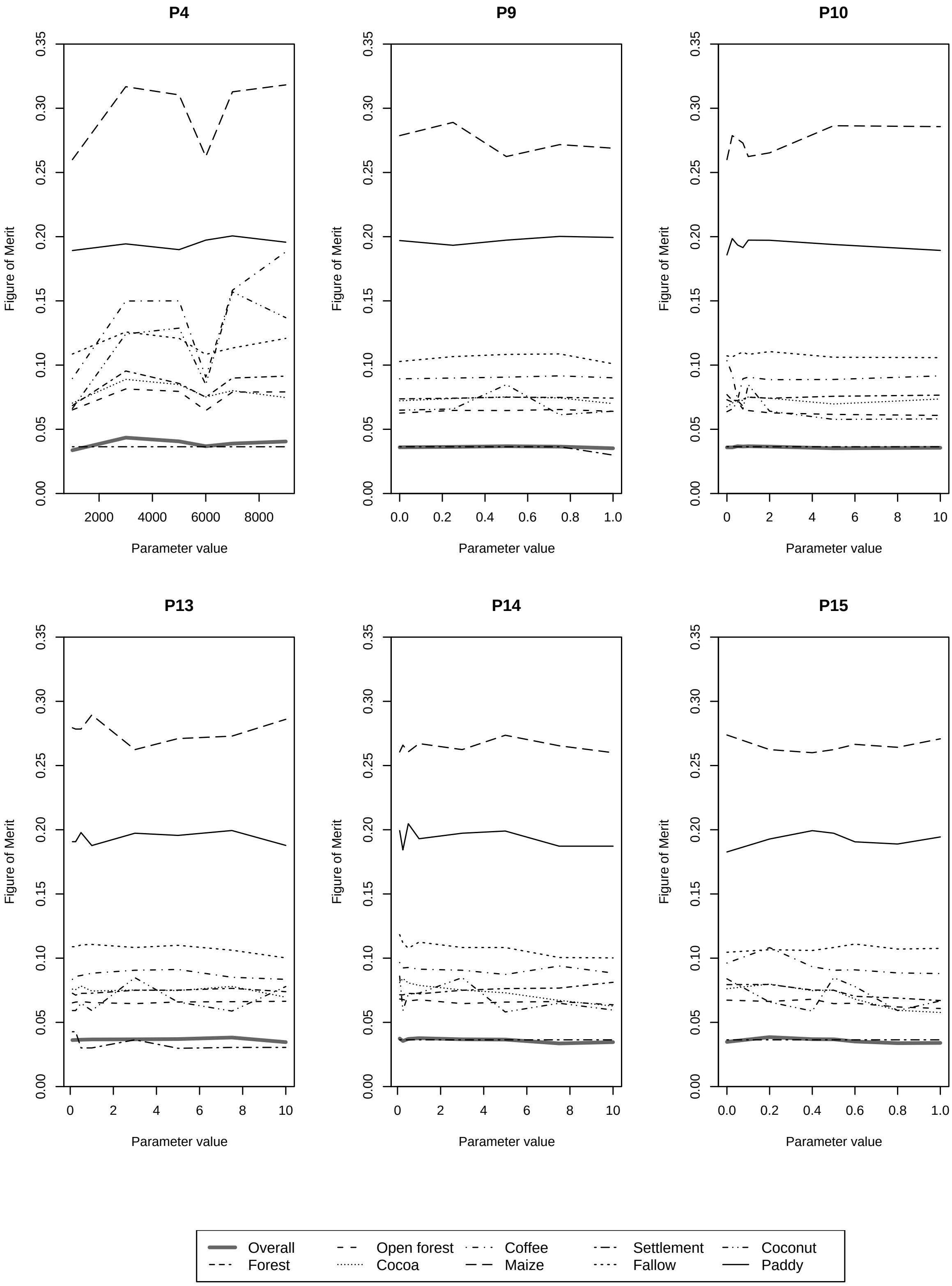


Figure 5.5: Sensitivity variations per parameter. The broad line indicates evaluation including all classes.

Table 5.2: Overview of rule set parameters selected as calibration candidates. A subset of these parameters was calibrated after the influence of each parameter on the rule set had been analyzed. The Values column lists the values that are assigned to each of the parameters during sensitivity analysis, the default value printed bold. Default distances (P1 to P6) are based on spatial analysis and survey data.

| No. | Description | Affected LU classes | Values [Unit] | Calibrated value |
|---|---|---|---|---|
| P1 | Maximum distance between settlement and paddy rice | Paddy rice | 1000, 2000, 2500, **3000**, 4000, 5000 [m] | |
| P2 | Maximum distance between water and paddy rice | Paddy rice | 1000, 2000, 2500, **3000**, 4000, 5000 [m] | |
| P3 | Maximum distance between settlement and maize | Maize | 1000, 2000, 3000, **4000**, 5000, 6000 [m] | |
| P4 | Maximum distance between settlement and cocoa | Cocoa | 1000, 3000, 5000, **6000**, 7000, 9000 [m] | |
| P5 | Maximum distance between settlement and coffee | Coffee | 1000, 3000, 5000, **6000**, 7000, 9000 [m] | |
| P6 | Maximum distance between settlement and coconut | Coconut | 1000, 3000, 5000, **6000**, 7000, 9000 [m] | |
| P7 | Minimum difference in overall suitability for one crop type compared to the current crop type of a cell to make a change of LU class effective | Crop classes | 0.01, 0.05, 0.1, **0.25**, 0.5, 0.7, 0.9 | |
| P8 | Minimum relative difference of a crop cell's current yield compared to the average district yield that forces the cell to be converted to fallow | Crop classes | 0.0, 0.05, 0.1, 0.3, **0.5**, 0.7, 0.9 | |
| P9 | Reduction factor representing the protection status of a cell (applied to cells in Lore Lindu National Park) | All classes | 0.0, 0.25, **0.5**, 0.75, 1.0 | 0.2 |
| P10 | Weight applied to the overall biophysical suitability of a cell | Crop and forest classes | 0.0, 0.25, 0.5, 0.75, **1.0**, 2.0, 5.0, 10.0 | 0.35 |
| P11 | Weight applied to the suitability concerning biomass in the process of calculating a cell's suitability for forest use | Forest, Open forest | 0.0, 0.25, 0.5, 0.75, **1.0**, 2.0, 5.0, 10.0 | |
| P12 | Weight applied to the suitability regarding the number of neighboring crop cells | Crop classes | 0.0, 0.25, 0.5, 0.75, **1.0**, 2.0, 5.0, 10.0 | |
| P13 | Resource limitation parameter applied to a cell's suitability for settlement based on area already covered by settlements | Settlement | 0.1, 0.25, 0.5, 1.0, **3.0**, 5.0, 7.5, 10.0 | 1.0 |
| P14 | Resource limitation parameter applied to a cell's suitability for crops based on area already covered by crops | Crop classes | 0.1, 0.25, 0.5, 1.0, **3.0**, 5.0, 7.5, 10.0 | 0.1 |
| P15 | Reduction factor for crop and forest types inversely related to population density of closest hamlet | Crop classes | 0.0, 0.2, 0.4, **0.5**, 0.6, 0.8, 1.0 | 0.2 |
| P16 | PRDX value influencing cocoa growth (DAYCENT sub model) | Cocoa | 0.6, 1.0, 1.4, **1.7**, 2.0, 2.4 | 0.8 |
| P17 | PRDX value influencing coffee growth (DAYCENT sub model) | Coffee | 0.4, 0.6, 0.8, **0.85**, 0.9, 1.0, 1.1, 1.2 | 0.55 |

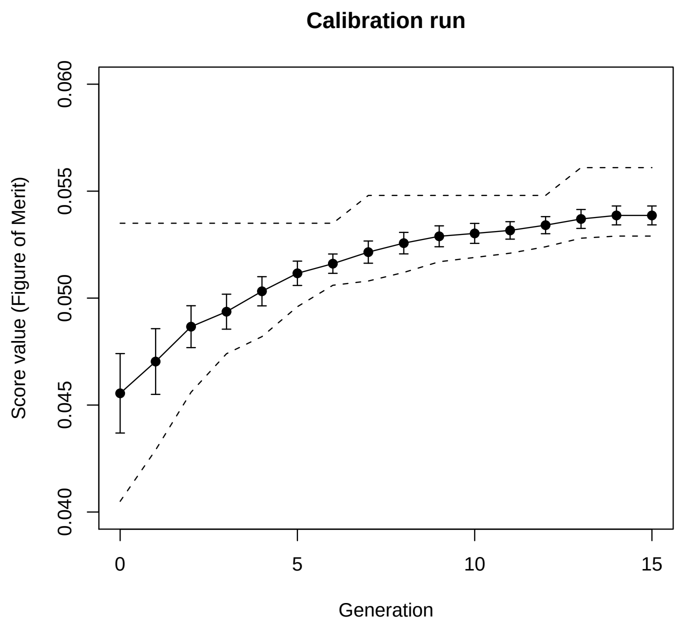


Figure 5.6: Population progression for final calibration.

values reached a maximum of 0.0561.

## 5.4 Discussion

### 5.4.1 Sensitivity analysis and parameter selection

For the sensitivity analysis we examined the influence of altering single parameter values while all other parameters remained unchanged having been set to their default values. Using this technique, we could show that the rule set reacted sensitive to varying the parameters listed in table 5.2. However, the procedure described in the past paragraphs is not capable of unveiling sensitivity due to altering a combination of parameters. For this task, a number of methods, especially Latin Hypercube sampling, are available (Saltelli et al., 2004; Muleta and Nicklow, 2005), which are not feasible for our application due to the high run time requirements. The main basis for establishing the calibration parameter set was the analysis of sensitivity ranges shown in figure 5.3, and the comparison of the progression plots shown in figure 5.5 with the available empirical information.

One of the main goals was to keep the calibration parameter set as small as possible, which was owed to the fact that we had to work with a small number of simulation runs and small population sizes. Evaluating each candidate solution in the population of different parameterizations of the calibration algorithm implied running a complete simulation. Profiling the run-time behavior unveiled that most time was consumed by the calculation of actual and potential yields of crop cells using the DAYCENT sub model. To address this problem, we developed an approach for reducing the number of required calculations. This was combined with parallel execution of DAYCENT calculations resulting in the above stated run time. Mitchell (1998) recommends populations of around 100 candidate solutions to provide a good sampling of the entire parameter space. However, the suggested population sizes are not feasible for our task, since a calibration run would last several months. This, as a consequence, lead to an increased probability of the genetic algorithm getting trapped in local maxima.

Our two basic criteria for calibration parameter selection turned out to be not entirely sufficient, thus additional aspects were considered. Further constraints for calibration parameter selection are given by the rule set semantics. It might appear promising to select both parameters P4 (max. dist. Between settlement and cocoa cells) and P15 (crop suitability reduction based on population density) as the ones with the strongest impact on model outcome. However, the sensitivity progression analysis for P4 (see figure 5.5) showed a clear maximum for most classes at 6000m, a result which was supported by empirical evidence from spatial analyzes of the reference maps of 1981 and 2002. Parameter P1 (max. dist. between settlement and paddy cells) was not calibrated due to the same reasons as for P4. Parameter P12 (crop neighbor weight) also seemed to be a good candidate, but test simulations had shown that high values of P12 favored the formation of artificial cell clusters of identical land-use, clearly not corresponding with land-use patterns of the 1981 and 2002 reference maps. Parameter P9 (protection factor) was causing lower variations with respect to the objective function for the entire area. However, P9 is the trigger for determining the protection status of the national park, which is of specific interest for our analysis and could not be determined empirically. P10 (weight applied to biophysical suitability) is also of specific interest, since it weights the biophysical suitability factors against the socio-economic factors and thus provides important information about the contribution of each factor group to the simulated land-use decisions in the suitability calculations.

### 5.4.2 Model calibration

For both model calibration steps, the GA continuously improved the initial set of candidate solutions. Although we were forced to work with small GA population sizes to create results in a reasonable amount of time, the population progression plots showed the desired behavior. Score mean values continuously increased, which indicates that for each generation the algorithm was able to improve the scores of each population. Globally improved candidate solutions with a new maximum score value were found for discrete generations. Due to the underlying random processes in the GA, it is not predictable, at which generations such improvements occur, which implies that a high maximum score may even be reached in an early state of the calibration process.

Working with small populations of candidate solutions results in increasing danger of getting an inefficient sampling of the entire search space, i.e. getting stuck in local maxima. To avoid this, we kept the search space as small as possible by reducing its dimension to a minimum, i.e. by selecting the smallest possible number of parameters for calibration. Additionally, through the adoption of a two-step calibration approach we were able to split a search space of high dimension into two separate smaller search spaces by considering settlement-related parameters in the first and all other parameters in the subsequent calibration step. This strategy of narrowing the parameter space has already been successfully deployed for the highly parameterized Patuxent Landscape Model (Boumans et al., 2001), although in this case a search space of 128 dimensions and not run time considerations have been the reason for the reduction.

Of great importance for the GA optimization result is the method of establishing the initial population of candidate solutions. Prior to the selection, we defined sensible parameter ranges and increments. The potential parameter values were assigned with equal probability. Since increment definitions allowed five to ten different values per parameter this further limited the number of possible solutions and yielded a higher probability of obtaining a well distributed sampling of the search space. For a GA-based calibration application that used a similar population size (18 individuals), Goldstein (2003) proposed to compose the initial population of fixed "stratified", partial random and pure random solutions to ensure a good sampling while allowing a higher variety of parameter values. A different strategy to create an initial population with an unlikely perspective to get trapped in a local optimum can be applied if there are already one or more known candidate solutions with a high score value. Starting from such solutions, the initial population can be composed of similar clones forcing a more directed search process (Seppelt and Voinov, 2002).

The high run time demands made it necessary to terminate the GA after a predefined number of generations. Termination was defined to happen after 30 generations for settlement class calibration and after 15 generations for the final calibration. Further calibration samples, however, showed, that the GA-based calibration procedure is capable of reproducing this level of maximum score values in this relatively small number of steps, which supports our assumption of GAs being suited as calibration procedure for complex land-use models. Additionally, we observed that final populations were relatively homogeneous and repeatedly showed multiple instances of identical candidate solutions. This, in turn, reduced the chance of finding new better solutions since high scoring identical solutions are preferably selected for crossover, an observation also made by Goldstein (2003). Thus, we can safely assume that after 15 generations, our GA population already reached a relatively stable state. Further alteration of these identical solutions beyond the final generation can be caused by the mutation operation.

GA literature proposes to use threshold or convergence criteria to decide whether to terminate a GA (Goldberg, 1989; Mitchell, 1998). For our land-use modeling application it was not possible to define the threshold for which we regard a parameter set to be good enough. In contrast to that, our strategy was to obtain the best possible result under the given restrictions. The same strategy has been applied by Goldstein (2003)and Muleta and Nicklow (2005) who also terminated the GA after a fixed number of generations. After 15 generations, the performance of our model with a figure of merit of 0.0561 was comparable to the Pontius et al. (2007) performance review of land-use models. A relatively small figure of merit had to be expected, since the observed net land-use change in our research are is comparatively small, and models applied in regions where a large fraction of the land is subject to change tend to have larger predictive accuracy, due the smaller fraction of unchanged land where simulation errors can occur (Pontius et al., 2007).

All score values and further decisions are based on the use of the figure of merit map comparison measure as objective function. However, it leaves uncertainties that are propagated into the final calibration result. Visual comparison of the simulated map using the best parameter set found and the 2002 reference map, still unveiled a number of errors. Such errors can either be the allocation of certain land-use types at the wrong location, or the creation of land-use patterns different from the ones found in the reference map. Errors regarding land-use patterns, potentially could be reduced using a more pattern-based map comparison method. However, one can anticipate, that such a measure may comprise drawbacks in other aspects. Straatman et al. (2004) faced a comparable problem and also identified deficits for the calibration of models based on constrained cellular automata. Currently, no generally agreed measure for all of the before mentioned purposes exists, especially when dealing with categorical maps (Kuhnert et al., 2005).

The SITE calibration component is designed as a feature to support automated calibration of LUCC model rule sets. A similar approach involving GA optimization of a parameter set with antecedent analysis of parameter sensitivity has been introduced by (Muleta and Nicklow, 2005) for the SWAT distributed watershed model highlighting the necessity of automation. Nevertheless, our study also showed that there are certain limitations to automation. Especially for establishing the calibration parameter set it is not sufficient to define criteria e.g. with respect to model sensitivity, research targets or empirical information. Specific knowledge of the underlying application rule set and model behavior is required.

We emphasize the generic nature of the SITE calibration component which is directly usable for other land-use modeling applications. SITE is a framework designed to run different CA-based land-use modeling applications. With this feature, we go one step further than most other applications by making automated calibration functionality available to a multitude of LUCC modeling applications.

Other optimization methodologies than GAs can easily be integrated into the SITE calibration component, e.g. simulated annealing (SA). In a study of spatial planning problems it could be shown that SA could produce similar results as GAs (Stewart et al., 2004), although SA consumed more CPU time. Another alternative is to use Monte Carlo sampling, which would be feasible for projects with lower CPU time demands. This has been done for calibrating of the SLEUTH urban growth model (Xian et al., 2005). Nevertheless, for the latter, only the implementation on a large computer cluster resulted in reasonable calibration times. Potential for improvement of GA-based calibration can be seen in a modification of the classical GA as proposed by Ndiritu and Daniell (2001) for the calibration of a rainfall-runoff model. By using a fine-tuning strategy involving search range reduction, hill-climbing for a search range shift to promising regions of search and independent subpopulation searches coupled with shuffling, the authors were able to effectively locate global optima for their test projects. Obviously, this approach holds certain potential for our study due to its improved capability of finding the global optimum.

## 5.5 Conclusions

The calibration of complex land-use models like the Indonesia case study presented in this work can be described as an n-dimensional optimization problem. For such models a generic calibration methodology is provided by the SITE framework. Using a GA as search procedure we were able to achieve adequately good calibration results in reasonable time compared to the Pontius et al. (2007) performance review of land-use models. In addition, we were able to resume that GAs are well suited as underlying calibration methodology given the limitations one has to face when working with complex land-use models, especially high run time requirements. It could be shown, that automated calibration of a land-use model as a process is possible. Nevertheless, the selection of parameters to calibrate requires specific knowledge of the application and cannot be generalized as the optimization process itself.

The applicability of other optimization procedures like hill climbing or simulated annealing needs to be assessed. The SITE framework architecture is designed for further extensions with respect to optimization procedures. The adoption of map comparison procedures focusing on specific aspects like land-use patterns will probably yield different results than more general methods. Which measure to use is highly dependent on the actual modeling task. For our task, the figure of merit measure described above was able to deliver satisfying map comparison values which also corresponded with our visual interpretations of simulated and reference maps. As for optimization procedures, the SITE framework is open for extensions regarding new map comparison methodologies.

Generity is a central design goal of the SITE framework. The possibility to combine GA calibration methodology (or other algorithms) with different map comparison methods is one aspect of this goal. This study demonstrated the suitability of the calibration component and a GA-based methodology implemented in the SITE framework to calibrate a complex land-use change model.

# 6 A study on linking deforestation scenarios to pollination services and economic returns

The ecological and economic consequences of rain forest conversion and fragmentation for biodiversity, ecosystem functioning and ecosystem services like protection of soils, water retention, pollination, or biocontrol are poorly understood. In human-dominated tropical landscapes, forest remnants may provide ecosystem services and act as source for beneficial organisms immigrating into adjacent annual and perennial agro-ecosystems. In this study, we use empirical data on the negative effects of increasing forest distance on both pollinator diversity and fruit set of coffee to estimate future changes in pollination services for different land use scenarios in Sulawesi, Indonesia. Spatially explicit land-use simulations demonstrate that depending on the magnitude and location of ongoing forest conversion, pollination services are expected to decline continuously and thus directly reduce coffee yields by up to 18%, and net revenues per hectare up to 14% within the next two decades (compared to average yields of the year 2001). Currently, forests in the study area annually provide pollination services worth 46 Euros per hectare. However, our simulations also revealed a potential win-win constellation, in which ecological and economic values can be preserved, if patches of forests (or other natural vegetation) are maintained in the agricultural landscape, which could be a viable near future option for local farmers and regional land use planners.

## 6.1 Introduction

Forest conversion, agricultural expansion and infrastructure extension have transformed landscapes throughout the world, resulting in biodiversity loss and threatened ecosystem services (Chapin et al., 2000; Balmford et al., 2002; Geist and Lambin, 2002; MEA, 2005). During recent years the concept of ecosystems services is increasingly used to describe different categories of benefits for human society attainable from natural and human influenced ecosystems (Costanza et al., 1997; Daily et al., 1997). Land-use changes may affect important ecosystem services, but consequent effects on societies' benefits are poorly understood. While goods like crop yields are tangible and can easily be measured, an economic quantification of ecosystem services like air quality, carbon storage, water retention, biological control and pollination is difficult. However, ecosystem services have been recently estimated to be worth several trillion Euros (Costanza et al., 1997), and already the direct benefits from tropical forests can exceed those obtained from converted habitats (Balmford et al., 2002).

Globally, important crops like coffee as well as many other cultivars (Free, 1993), benefit from pollination services of nearby forests or other natural habitats, which provide forage and nesting space for pollinators (Kremen et al., 2002; Roubik, 2002b; Klein et al., 2003a; De Marco and Coelho, 2004; Ricketts, 2004). Pollination represents a basic ecosystem service with an estimated economic benefit between 90 billion and 160 billion Euro at the global scale (Costanza et al., 1997; Kearns et al., 1998). Thus, it is of great interest to understand how future land-use changes might affect ecologically and economically important functions provided by natural forests.

Policy scenarios are a valuable tool to evaluate the consequences of socioeconomic and environmental drivers on land use and land use change at different spatial scales (Alcamo et al., 1998; Lambin et al., 2001; Alves, 2002; Van Jaarsveld et al., 2005). While spatially explicit simulation models have been employed to quantify policy scenarios in terms of land cover / land use change (Irwin and Geoghegan, 2001; Verburg et al., 2004), expanding such scenarios to assess the impacts on biodiversity and ecosystem services has been rarely realized, partly due to the limited availability of empirical data that quantify the relationship between land use patterns and ecosystem services.

Here, we use recent results on the effects of forest distance on the fruit set of coffee, to evaluate future risks to coffee yields and farmer's revenues under different land use scenarios. Pollinator availability improved fruit set and crop yield of highland coffee (*Coffea arabica L.*) between 15 and 50% (Roubik, 2002a). With increasing forest distance pollinator diversity and fruit set of coffee is significantly reduced (Klein et al., 2003a,b; De Marco and Coelho, 2004; Ricketts et al., 2004). Here, we used the results of Klein et al. (2003a,b) and a novel spatially explicit land use model to analyze effects of different scenarios of future land use and land cover change on pollination services and related yields of small-scale coffee agroforestry systems in Sulawesi, Indonesia. Additionally, we estimated the economic value of the pollination services of present forests and evaluated the potential future reduction of the net revenues of coffee farmers.

## 6.2 Methods

### 6.2.1 Study region

The study was carried out in Central Sulawesi, Indonesia, at the north-eastern border of the Lore Lindu National Park (LLNP) at an elevation between 500m and 1600m. The study region belongs to the broad valley of the Palolo river, which forms the central part, while the northern and southern parts are shaped by moderate to steep slopes. The climate is tropical humid with 2000mm rainfall per year and an annual mean temperature of 23°C. As shown in Figure 6.1, the southern part of the research area is situated within the LLNP. The protected forest close to the border is already partly converted into agricultural land and small coffee and cocoa plantations, which are owned by farmers living in villages adjacent to the LLNP. Presently, coffee (almost exclusively *Coffea arabica*) is grown mostly between 600 and 800 m.a.s.l., and is frequently intercropped with cacao (*Theobroma cacao L.*). Agricultural crops, dominated by paddy rice are preferably grown in the valley, i.e. the lower, plain areas, but maize, upland rice and coffee are also grown on adjacent slopes.

### 6.2.2 Pollination services at risk

The scenarios presented in this study are based on the results of field experiments carried out by Klein et al. (2003a,b,c) in the same region. The studied systems are typical small-scale coffee agroforestry systems with a variety of shade trees. Highland coffee, *Coffea arabica*, is potentially wind-pollinated but recent studies show that fruit set is significantly

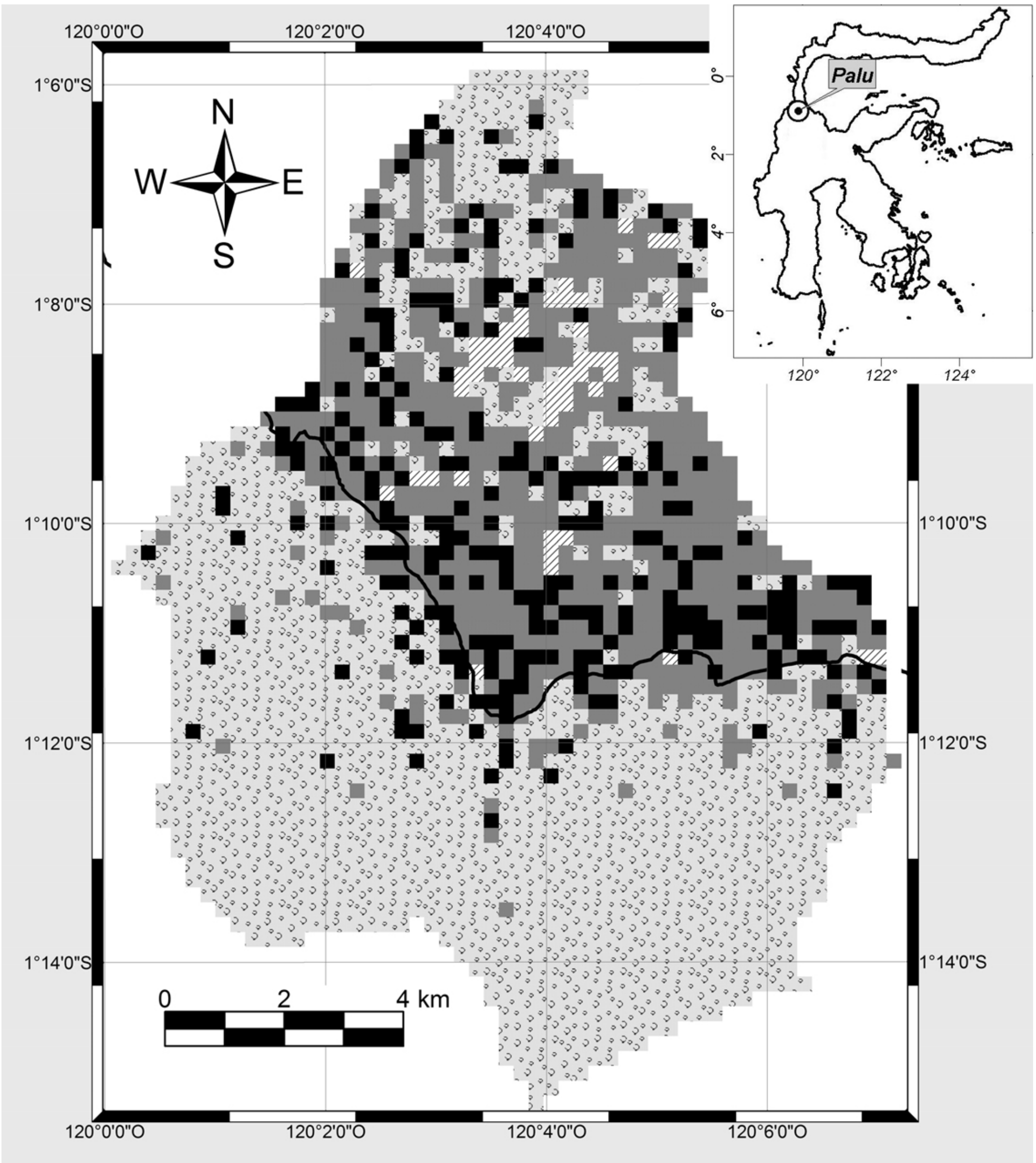


Figure 6.1: Gridded land-use/land-cover map of the research area south of the city of Palu in Sulawesi (black rectangle in the small map). Legend: Light grey with tree symbols: primary and secondary forest; medium grey: agriculture; black: coffee plantations; hatched area: hamlets and villages; black line: road and LLNP boundary. Note that the entire area south of the border line belongs to LLNP.

enhanced by insect pollination (Roubik, 2002a; Klein et al., 2003a). In a regional context the diversity of pollinators in coffee agroforestry systems was determined by the distance to the natural forest margin and additionally by local management practices (Klein et al., 2003b,c). Most interestingly, both the species richness of coffee pollinators and the fruit set of highland coffee declined with increasing forest distance (Klein et al., 2003c). Thus, coffee pollination services are at risk due to ongoing forest conversion in the study region.

We used these empirical results to analyze the potential effects of future land use change on the spatial distribution of pollination services. The effect of forest distance on the fruit set of coffee can be expressed as:

$$F = a + b\sqrt{D} \tag{6.1}$$

where $F$ is the fruit set of coffee in %, $a$ and $b$ are highly significant fitted constants ($a = 85.22$ and $b = -0.64$; $p < 0.005$; $R^2 = 0.34$), and $D$ is the distance of the coffee plantation to the forest in meters (based on data from Klein et al. (2003c)).

The range of equation 6.1 is limited by the maximum fruit set of 85.2% at the forest edge (20 bee species), and the minimum fruit set of 60.4% in 1,500m distance (3 bee species) that is similar to fruit set of treatments with only wind pollination (Klein et al., 2003a). Thus, we do not expect further reductions of fruit set at larger distances and consequently limited the potential yield reductions in the scenarios to the above-mentioned range. Therefore, in the gridded spatial data sets, in which every 250m pixel was evaluated using a Geographic Information System, distances larger than 1500m were set to 1500m (for the calculations of coffee yields). Coffee yields were assumed to be proportional to the fruit set of coffee and were expressed as a fraction of the yields harvested in 2001. This assumption is also supported by the positive correlation between fruit weight and fruit set (Roubik, 2002b). Furthermore, the frequency of coffee-flower visiting bees could be directly related to the coffee yield per ha measured as number and weight of harvested bags filled with fresh coffee berries (Dorthe Veddeler, unpublished data). There was no effect of forest distance on rates of coffee bean infestations by the coffee berry borer *Hypothemus hampei* (Alexandra-Maria Klein, unpublished data).

### 6.2.3 Land-use change model and policy scenarios

The simulation experiments were carried out with the new SImulation of Terrestrial Environments model (SITE model), recently developed at the University of Kassel. In this paper land use change is simulated in a spatially explicit fashion on a 250m grid in yearly time steps. The core of the model consists of a "Generalized Cellular Automata" approach. In the land use model, the allocation of land use is based on biophysical suitability (climate, soils, topography), allocation factors (e.g. distance to the next river, land use on adjacent cells, preferred walking distance to field), demography (population growth rate, migration) and the land use strategies of farmers (e.g. moderate or high intensity agriculture, forest use) and other restrictions like the protection of the LLNP area. A detailed description of the SITE model is given in chapter 3. Simulation experiments were calculated for the period 2002 - 2021, and were based on the land cover / land use distribution of 2001. The land cover in the year 2001 was derived from a LANDSAT ETM+ scene (path 114 / row 61) of 24th August 2001 classified by Haertel et al. (2002).

Four different scenarios of land use change were generated, based on assumptions about demographic trends, local and regional policies, socio-economic development, and land use strategies of local smallholders (Table 6.1). Human population growth was calculated according to the United Nations database (Indonesia: high scenario; (UN, 2004)). The differences between the low and high UN scenarios were not very pronounced in terms of land use change within the next two decades (data not shown here), because regional demographic trends are dominated by immigration to the comparatively less densely populated rural areas of Sulawesi ($< 20\,\text{inhabitants/km}^2$). Per capita demand for agricultural land was estimated from 1980 - 2000 village statistics and compared against the LANDSAT ETM+ scene of 24th August 2001 (Haertel et al., 2002). The calculated value of

Table 6.1: Land use strategies and driving forces of land use change: historical / present situation and four policy scenarios. Note that the most important drivers of each scenario are underlined.

| | Present situation (1980 -2001) | Business as Usual (*BaU*) | Agricultural Pogress (*AgPro*) | High Migration (*HiMig*) | Forest Enchroachment (*ForEnc*) |
|---|---|---|---|---|---|
| *Policy* | | | | | |
| Rural immigration[a] | Continued | Continued | Continued | Doubled | Doubled; no control |
| Economic stimulation[b] | Low | Low | High | Low | Low |
| Protection of LLNP[c] | Imperfect protection | Protection (50%) | Protection (50%) | Protection (50%) | No protection |
| *Agricultural activities* | | | | | |
| Subsistence crops | Self sufficiency | Self sufficiency | Self sufficiency production for market | Self sufficiency | Self sufficiency |
| Cash crops[d] | Moderate intensity | Moderate intensity | Increased intensity | Moderate intensity | Moderate intensity |
| Forest use (logging) | For subsistence mainly | For subsistence | For subsistence | For subsistence | Commercial |

[a]Migration was/is either controlled via the official transmigration porgram, which affects only a small fraction of migrants in Central Sulawesi, or in the case of spontaneous migration at the village level via access to land and other resources, which is negotiated between representatives of the villagers and the newcomers.

[b]Improved market access via improved infrastructure and increased access to credits (Maertens, 2003).

[c]In the scenarios presented here, the protection of LLNP forest grid cells was simulated as a 50% reduction in suitability for agricultural crops.

[d]The spatial extension of coffee plantations was kept stable in all land use change scenarios, in order to enable the comparison to the present situation.

0.23 ha/capita was consistent with FAO (2004) estimates for Indonesia (0.21 ha/capita) for the same year.

We developed the following four scenarios:

- In the "Business as Usual" (*BaU*) scenario, the present demographic trend (1970 - 2001) of a high, but decreasing population growth rate and considerable rural immigration is continued, as well as local policies to protect the LLNP with moderate efficiency. LLNP protection is simulated by reducing the suitability for agricultural and residential use of protected grid cells by 50%. Local farmers are assumed to continue cultivating their fields with the same labor intensity and frequency of field visits as before.
- The "Agricultural Progress" (*AgPro*) scenario is based on economic measures suggested by Maertens et al. (2003), the same demographic assumptions and the continued protection of the National Park. In contrast to the other scenarios, agricultural land use practices of farmers are supposed to be more labor intensive, resulting in preferred smaller walking distances to the fields due to (i) more frequent field visits and (ii) a higher investment of labor per unit agricultural area (weighing factor: 10). This process of agricultural intensification has been taken into account by simulating a slightly reduced amount of agricultural area cultivated per capita (initial value: 0.23 ha; -1% per capita per year).
- The "High Migration" (*HiMig*) scenario differs from the previous ones in the increased immigration rate (1.5 fold), caused by economic problems and high population pressure elsewhere in the archipelago. All other assumptions are the same as in the *BaU* scenario.
- The "Forest Encroachment" (*ForEnc*) scenario was developed for the evaluation of the consequences of rapid deforestation. *ForEnc* was inspired by recently increased forest conversion activities in the nearby Dongi-Dongi area, an illegal forest invasion during which more than 2,200 hectares of protected forest were lost within only 3-4 years (Erik, 2002; Erasmi et al., 2004). In this scenario, we assume the complete failure of regional protection policies, resulting in increased logging and forest conversion activities in protected and non-protected forests. Due to the unsustainable forest use (e.g. conversion of forest on steep land unsuitable for agriculture), the per capita land demand in the *ForEnc* scenario is slightly increasing (+1.5% per capita per year), which is equivalent to approximately 250m$^2$ of additional demand for land per household per year (Table 6.1).

### 6.2.4 Economic evaluation of pollination services

The economic contribution of forests to the production of coffee beans was calculated as average yield increase per hectare of forest. Only the forests within the 1500m forage distance of bees were used for the valuation of pollination services.

Additionally, the net revenues of coffee farmers were estimated for the four policy scenarios, based on prices, fixed and variable costs of 2001 (see Table 6.2). According to the National Statistics Bureau of Indonesia (BPS, 2003), average coffee yields in Central Sulawesi were 468 kg/ha both in 2001 and 2002, which is close to coffee yields reported in

Table 6.2: Average net revenues of coffee farmers in 2001. All values besides the last line are given in Indonesian Rupiahs (IDR).

| | Unit | Unit cost (IDR) | Quantity | IDR/ha |
|---|---|---|---|---|
| *Costs* | | | | |
| Annualized establishment costs[a] | | | | 54,915 |
| Cleaning | labor days | 10,000 | 4.0 | 40,000 |
| Pesticides | | | | 5,900 |
| Harvest coffee | labor days | 10,000 | 6.0 | 60,000 |
| Transport | | | | 18,000 |
| Processing | | | | 751,809 |
| Total costs | | | | 930,625 |
| *Revenues* | | | | |
| Coffee yield (green beans) | kg/ha | 10,515 | 478.5 | 5,031,340 |
| Net revenues, IDR 2001 | | | | 4,100,715 |
| Net revenues, Euro 2001 | IDR → Euro | | 8,835.981 | **464.09 Euro** |

[a]Total establishment costs are IDR 570,000 for a productive period of 15 years.

our household survey in 2000/2001 (581 kg/ha; see (Zeller et al., 2002) for a description of the survey methodology), and about half of the average value of 891 kg/ha reported for the same period for entire Indonesia (FAO, 2004). We calculated the mean producer price of 1998 - 2003 for highland coffee paid to coffee farmers in Indonesia, which was 1.19 Euros per kg green beans (Organization, 2004).

## 6.3 Results

### 6.3.1 Land-use change scenarios

The process of forest conversion within and outside the LLNP continued in all scenarios. Continued population growth resulted in the increase of residential and agricultural areas, while forest areas decreased respectively (Table 6.3). Due to differences in scenario assumptions concerning immigration and land use strategies, forest conversion ranged between 4% and 44%, which is equivalent to deforestation rates between 0.18%/yr and 2.20%/yr. Additionally, large differences were observed in the simulated distribution of land use / land cover types. The *AgPro* scenario was the only one, in which the present distribution of patches of primary and secondary forests within the agriculturally dominated landscape – mainly the broad valley bottom – was maintained (primary and secondary forests were assumed to be comparable in their capacity to support pollinators). In the *BaU*, *HiMig* and *ForEnc* scenarios, the lowland forests completely disappeared and the northern boundary of the LLNP forests was shifted southward. In the *ForEnc* scenario (see Figure 6.2 for the land use / land cover change sequence), in which increased logging activities were simulated, forest conversion both on the northern and southern slopes of the central valley were strongest, followed by the *HiMig* scenario. In the *BaU* and *AgPro* scenarios, the southern forests remained almost untouched, while in the *BaU* scenario

Table 6.3: Land use / land cover in 2001 and 2021. Note that the spatial extend of coffee plantations was kept stable in all scenarios (1656 ha).

| Start year / scenario | Residential area [ha] | Agricultural area [ha] | Forest area [ha] | Mean deforestation rate [%/year] |
|---|---|---|---|---|
| Start year 2001 | 294 | 3,506 | 7,975 | 0.60[a] (1972-2001) |
| *BaU* 2021 | 438 | 4,525 | 6,813 | 0.73 |
| *AgPro* 2021 | 438 | 3,650 | 7,688 | 0.18 |
| *HiMig* 2021 | 538 | 5,256 | 5,981 | 1.25 |
| *ForEnc* 2021 | 538 | 6,775 | 4,463 | 2.20 |

[a]The rate was reported by Erasmi et al. (2004) for the entire Lore Lindu region (~7,500 $km^2$).

a clear decrease occurred in the forest areas in the northern part of the Palolo valley (Figures 6.3 and 6.4).

### 6.3.2 Pollination

The spatial analysis revealed the lowest average distance of 157 m between forest areas and coffee plantations in the year 2001. All plantations were located within the 1500m bee forage zone (max. distance $\leq$ 1000m), and profited from pollination services. After 20 simulated years, distances between forests and coffee sites increased in all land use / land cover change scenarios in the order *AgPro* - *BaU* - *HiMig* - *ForEnc*. In consequence, fruit set of coffee (initially 80% in 2001) decreased in all scenarios (Figure 6.5). The decrease in pollination services presented a dichotomy between the *AgPro* and *BaU* scenarios on one hand, and the *HiMig* and *ForEnc* scenarios on the other hand. In the latter, decrease in pollination was strongest in the first few years, during which the forest patches remaining in the valley, close to many of the coffee plantations, were converted (see Figure 6.2, land use after 5 years). Further forest conversion mostly on the northern and southern slopes, where less coffee plantations were located, had a less pronounced effect on the fruit set of coffee (Figure 6.5). After 20 years, increased pollination limitation resulted in a mean fruit set of only 67% in the *HiMig* and of 66% in the *ForEnc* scenarios. In the other two scenarios the forest plots in the valley were either preserved (AgPro), resulting in minor reductions (mean fruit set of 79%), or disappeared more gradually (*BaU*) causing increasing reductions of coffee yields over the entire simulation period of 20 years and a final mean reduction to 70% fruit set. Note that we assumed the same pollinator diversity for all forest patches, irrespective of patch size, because the relation between patch size and pollinator diversity is unknown for tropical forest fragments.

### 6.3.3 Economic evaluation of pollination services

Based on the mean producer price of 1.19 Euro/kg green coffee beans, the total value of the 2001 coffee harvest within the study area (1656ha coffee plantations) was 1.1 million Euros. Primary and secondary forests in the 1,500 meter forest zone surrounding the coffee plantations, summed up to 5659 hectares. Thus, in the year 2001, the pollination service ensured by neighboring forests translated into an economic benefit of 258,000 Euro for the coffee farmers, which is equivalent to 46 Euro per hectare “pollination” forest. To

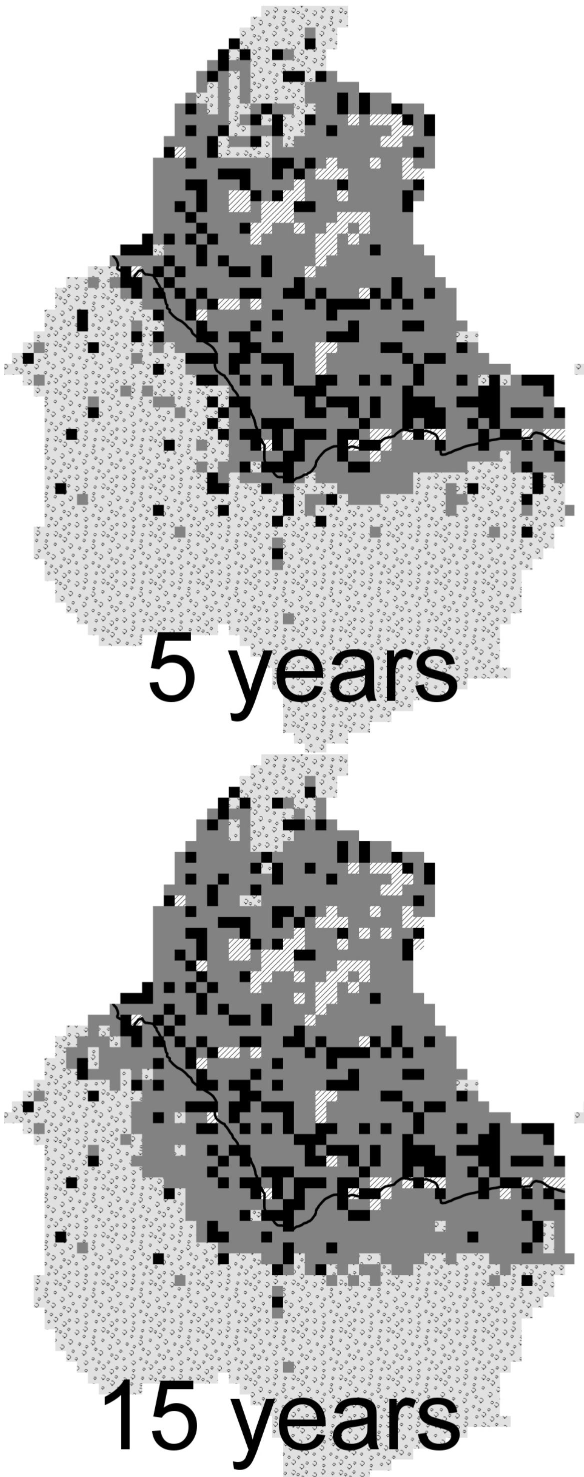


Figure 6.2: Policy scenarios of land use / land cover change. Simulation of the *ForEnc* scenario; land use change after 5, 10, 15 and 20 years. See Figure 6.1 for the legend.

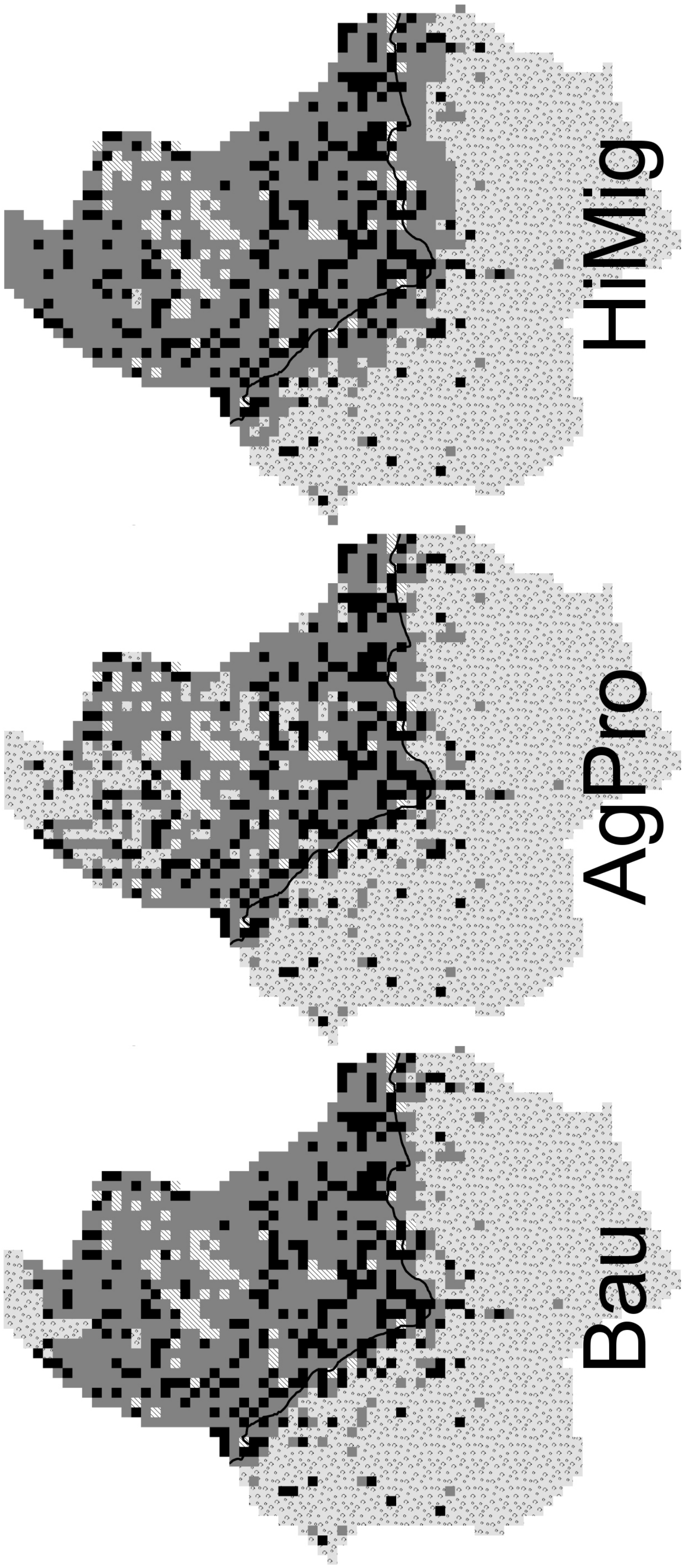


Figure 6.3: Policy scenarios of land use / land cover change. Simulation of the *BaU*, *AgProg* and *HiMig* scenarios; land-use change after 20 years. See Figure 6.1 for the legend.

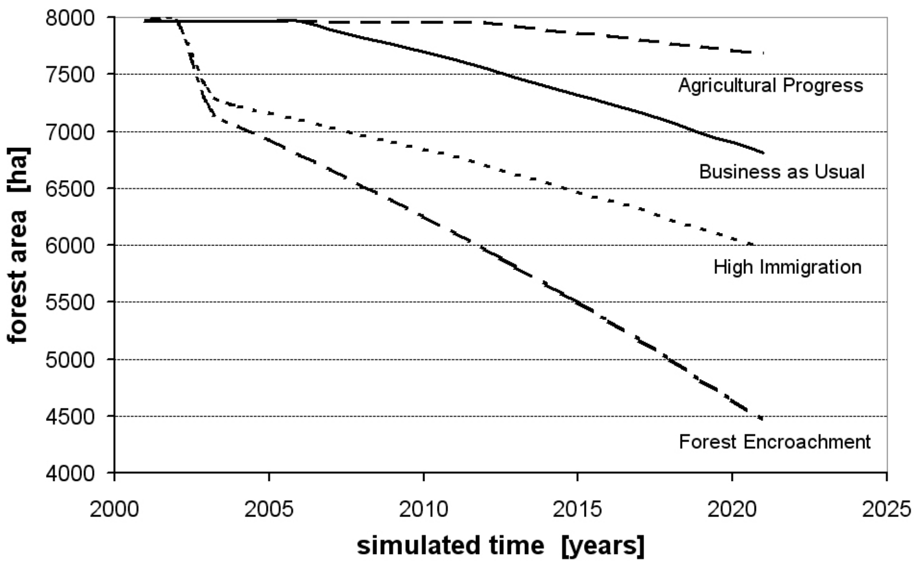


Figure 6.4: Change of forest cover 2001 - 2021. Change of forest cover simulated in four policy scenarios for the period 2001-2021.

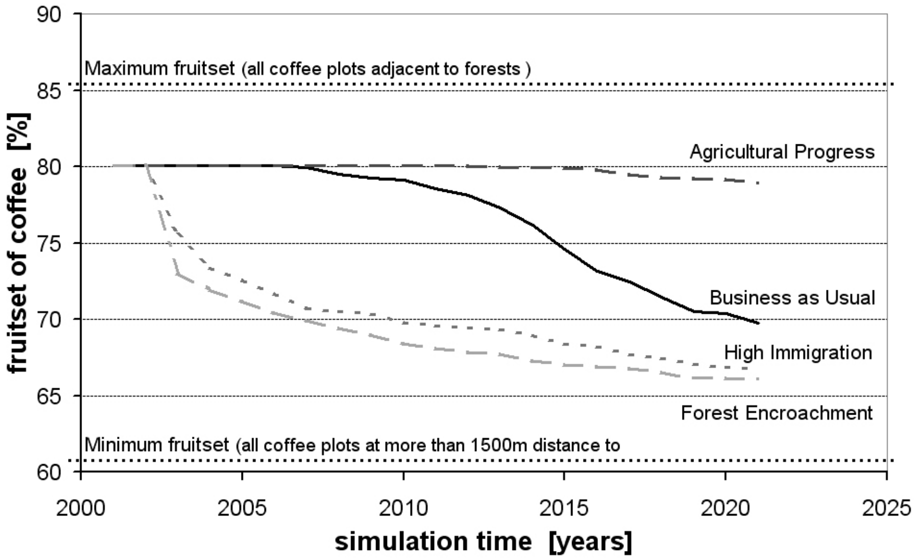


Figure 6.5: The fruit set of coffee, as affected by land-use / land-cover change, simulated in four policy scenarios for the period 2001 - 2021. The calculations are based on equation 6.1, which is derived from field work of Klein et al. (2003a) in the same region. The authors reported 85.2% as the maximum fruit set at the forest border, while the minimum fruit set was 60.4% (distance to the nearest forest $\geq$ 1500m). The coffee growing area was kept constant throughout the simulation experiments.

assess the potential economic value lost due to future reductions in pollination services, we calculated crop yields per ha coffee plantation, based on the four different land use scenarios. The dichotomy observed for the fruit set, between the *AgPro* scenario on one hand, and the *BaU*, *HiMig* and *ForEnc* scenarios on the other hand, was replicated in the calculations of the net revenues for coffee production over the next two decades. All net revenues decreased during the simulations and revealed cumulative reductions between

Table 6.4: Cumulative losses of net revenues of all coffee farmers, simulated in four policy scenarios. Cumulative losses of net revenues were calculated as differences between revenues with continued high pollination rates, as observed in 2001 (18,836,000 Euro; 1656ha for the period 2001 - 2021), and reduced coffee yields simulated in four policy scenarios.

| | *BaU* | *AgPro* | *HiMig* | *ForEnc* |
|---|---|---|---|---|
| Cumulative losses [Euro] | 850,310 € | 60,757 € | 2,360,139 € | 2,660,008 € |
| Cumulative losses [%] | 4.5% | 0.3% | 12.5% | 13.8% |

8 and 101 Euro per hectare coffee after 20 years (changes over time are identical to the pattern in Figure 4). Cumulative losses of net revenues over 20 years ranged between 0.3% in the AgPro scenario and 13.8% in the *ForEnc* scenario (Table 6.4).

## 6.4 Discussion

The quantification and valuation of ecosystem services is poorly understood (Kremen et al., 2004; Steffan-Dewenter et al., 2005), and to date only a few estimates for the many services provided by natural systems have been published (Costanza et al., 1997; Balmford et al., 2002). Over the last decade pollination services, which are vital for many agricultural production systems, have received increasing attention, covering biological, economic and land use aspects (Roubik, 2002a; Klein et al., 2003b,c; Kremen et al., 2004; Ricketts et al., 2004). However, we know of only a few other studies, applying a combination of field research and spatially explicit model simulations, to assess the potential ecological and economic impacts of land use change on ecosystem services (Guo et al., 2000; Van Jaarsveld et al., 2005). In this study, we use different patterns and spatial coverages resulting from land use change simulations to calculate changes of pollination services provided by tropical rainforests and to assess the economic value of both pollination services and the natural habitat providing it.

### 6.4.1 Policy scenarios and land-use and land-cover change

The SITE model was developed for regional land use change studies using a rule based approach. Rule based models include available expert knowledge and explicitly represent causal relationships (e.g. in our study derived from the household and village surveys, statistics and satellite imagery). In consequence, such models are well suited for a broad range of scenarios, while empirical or regression models should only be used for scenarios, which are similar to the initial conditions under which the regressions were derived (Priess et al., 2001; Verburg et al., 2002). The scenario assumptions employed in this paper are based on socio-economic developments occurring in Indonesia and other tropical regions, where the expansion of crop production and agroforestry hits the forest frontier. It has been shown by Geist and Lambin (2002) that the underlying proximate causes for the resulting deforestation are manifold, and vary considerably between regions or countries. For this study we generated four scenarios of contrasting policy measures, demographic trends, and farmer's land use practices. It has been shown for our study region that population growth including migration (Maertens, 2003) and the availability of land suitable for agriculture (Kreisel et al., 2004) are major factors explaining present and historical land use, and can be considered major drivers of land use change in the near future. The policy scenarios generated for this study are moderate modifications of existing and historical socio-economic trends and farmer's preferences and land use strategies. We avoided extremely optimistic or pessimistic scenarios (e.g. "Great Transition" by Raskin et al. (2002); or "Barbarization" of the GSG (2004)) with respect to economic developments, policy measures or consumption of natural resources. Nevertheless, the resulting extend of land use and land cover change, as well as land use patterns, are highly contrasting and have strong effects on natural and agro-ecosystems, and respective pollination services.

### 6.4.2 Pollination at risk

Forests, interspersed in an agricultural landscape, continually provide ecologically and economically valuable pollination services. The importance of adjacent or nearby natural vegetation for the pollination of commercially important crops, has repeatedly been reported (Kremen et al., 2002; Klein et al., 2003a,b; De Marco and Coelho, 2004; Kremen et al., 2004; Ricketts et al., 2004). The beneficial effect of pollinating insects, which can reach up to 41%-enhanced fruit set in Indonesia (Klein et al., 2003a,c), may vary between regions and between different crops (Free, 1993; Kearns et al., 1998; Roubik, 2002a; De Marco and Coelho, 2004; Kremen et al., 2004; Ricketts et al., 2004). In this study, adjacent forests and forest patches provided near optimal pollination services in the base year 2001, while our simulations indicate that this ecosystem service can be expected to decline in the coming decades due to land use and cover changes. In the *ForEnc* and the *HiMig* scenarios the historic rates of forest conversion of 0.6%/yr (Erasmi et al., 2004) were exceeded, but on the other hand were slower or comparable to recent nearby forest transformations in Sulawesi and in other forest frontier areas in the tropics (Shriar, 2001; Geist and Lambin, 2002). However, due to the small foraging radius of bees (Gathmann and Tscharntke, 2002), not only the extent, but also the location of forest conversion affects pollination services. In Sulawesi, in only 375m distance from forests, 50% of the beneficial effect of pollination is lost, which means that average fruit set drops from 85 to 73%, (see equation 6.1).

Patch area of remaining forest might also influence species richness and abundance of pollinator communities as larger habitat fragments in many cases show higher species richness and density (Fahrig, 2003). However, concrete studies on species-area relationships for pollinators in tropical forest fragments are lacking. Therefore, we could not include effects of patch area in our model suggesting that the potential loss of pollination services might be even larger than estimated in our simulations. However, the minimum area of forest patches in the model was 6.25 ha due to the grid length of 250m. Considering the nonlinear relationship between habitat area and species richness (Fahrig, 2003) we assume that these fragments supported a significant fraction of total pollinator diversity and that the effect of patch area on pollinator availability was less significant than isolation distance. The generality of our simulations is supported by similar effects of nearby natural vegetation on crop yields in several independent studies in different tropical and temperate regions (Kremen et al., 2002; Klein et al., 2003a,b,c; Kremen et al., 2004; Ricketts et al., 2004). In consequence, all future land use policies leading to a purely agricultural landscape without persisting forest patches or other natural habitats, will face a severe reduction in pollination. Alternatively, appropriate local management of agroforestry systems providing continuously available pollen resources and nesting sites could increase diversity of bees in agricultural habitats (Klein et al., 2003b), but such appealing strategies of land use management to conserve pollination services are not yet developed. It is a realistic assumption that other ecosystem services might be lost in parallel, and this will result in associated additional negative ecological and economic impacts (Balvanera et al., 2005). There also might be some loss of ecosystem "disservices" e.g. pests, potentially counterbalancing some of the negative impacts.

### 6.4.3 Economic evaluation

According to Maertens et al. (2003) and Van Rheenen et al. (2003) the farmers in the study region typically are involved both in food-crop production, mainly paddy and upland rice as well as in agroforestry, planting small plots with coffee or cocoa. Income from crop production on average are accounting for 44% of the household income (Schwarze, 2004). In the year 2001, 80% of the population in the study area (Palolo district) was engaged in coffee production, managing on average 0.5ha of coffee plantations. Many of the 3440 coffee-growing households operate close to or below the poverty line of 520 Euros per year for a typical household of five persons (Van Rheenen et al., 2003). Thus, the income of the majority of the rural population in Palolo would be directly affected by the expected 0.3% - 13.8% reduction in net revenues from coffee sales. Notably, the lowest losses of ecological and economic values were estimated in the *AgPro* scenario, in which we simulated policies assumed to reduce forest conversion via controlling migration, intensification, diversification and improving market access and off farm opportunities. The before mentioned policies are ambiguous with respect to their impact on tropical rainforest conversion; both beneficial and detrimental effects of such policies have been reported from the tropics (Angelsen and Kaimowitz, 1999; Fearnside, 1997b; Murniati et al., 2001; Geist and Lambin, 2002; Maertens et al., 2003) depending on accessibility of land resources.

During recent years, Indonesia established a stable position as the world's fourth most important coffee producer. In some parts of the country the Indonesian government still encourages the installation of new coffee plantations. Studies from the island of Sumatra revealed even an increase of coffee growing areas (as part of rural development plans) in spite of low producer prices, causing further forest conversion including unintended deforestation of an adjacent national park (O'Brien and Kinnaird, 2003). If we assume a comparable trend in the study area in Sulawesi, future land use dynamics would translate into a constellation similar to the *ForEnc* scenario presented in this study, or even more severe forest conversion. Moreover, the further conversion of forest leads to a reduction in yields on the already existing coffee plots due to reduced pollination. This might again urge many farmers to clear additional forest or to convert their extensively managed and ecologically sustainable agroforestry systems to annual cropping systems, which might be more profitable in the short term. However, small-scale coffee plantations in Indonesia and elsewhere, usually are located on sloping terrain with inherent high erosion risks, and thus are much less suited for annual cropping than for the currently still dominating coffee and cocoa agroforestry systems. The estimates for the economic value of the forest, but also the potential financial losses of farmers in Sulawesi, were calculated based on the mean coffee price of 1998 - 2003, which is at the low end of the range of coffee prices

Table 6.5: Relations between coffee yields, pollination services for coffee framers and estimated minimum forest cover.

| Mean distance between forest and coffee plantations [m] | Coffee yield [%] | Pollination service [%][a] | Minumum forest cover [%][b] |
|---|---|---|---|
| 150 | 91 | 69 | 8 |
| 400 | 85 | 48 | 2 |

[a]Pollination service at the forest boundary in 0m distance is 100%.
[b]Assuming 1ha forest patches in regular triangular spacing.

reported by the International Coffee Organisation (2004). Our estimate of 46 Euro per hectare strongly contrasts with estimates of 315 Euro per hectare by Ricketts et al. (2004). The discrepancy simply reflects the fact that their estimate is based on a forest area to coffee area ratio of 0.3, while in our study the ratio was 3.4, due to the large forest area of the nearby LLNP. In both studies the minimum area for providing pollination (and other ecosystem) services is not known, but based on equation 1 and a regular triangular spacing of forest patches, we can calculate the number of forest patches per $km^2$, which are needed to provide pollination services for coffee farmers in Sulawesi (Table 6.5). In the study region the 1,656ha of coffee plantations are distributed over ~5,000 ha of land, translating into a minimum forest cover of 400ha, considering 70% pollination services as satisfactory. The resulting minimum forest area to coffee area ratio would be 0.25, which is similar to the value reported by Ricketts et al. (2004). The corresponding value of annual pollination services would be 470 Euro per hectare, thus exceeding the estimate of Ricketts et al. (2004) by ~50%.

In case of increasing coffee prices, the economic value of the forests' pollination services would also increase and might well become more attractive than other forest uses or forest conversion, especially when considering that (a) coffee is only one of many tropical crops depending on pollinators and (b) pollination represents just one out of many services provided by forests (Kremen et al., 2000; Pattanayak and Kramer, 2001; Balmford et al., 2002). As an additional effect, potential losses of net revenues would also increase in line with coffee prices, if deforestation continues (given that costs and producer prices follow a similar trend).

Smallholder coffee production is common in Indonesia (Van Rheenen et al., 2003; O'Brien and Kinnaird, 2003) and other parts of the tropics , especially Central America and Africa (Nair, 2000; Shriar, 2001). In consequence, the results of this study are applicable in other coffee growing regions, where coffee is produced under similar conditions. At the global scale, it is unknown how many of the 10.4 million hectares of coffee plantations (FAO, 2004) are managed by smallholders in forest margin areas, and how many of the 25 million people depending on coffee production are small-scale coffee producers. Based on our field and simulation studies in Indonesia, we can safely assume that the average deforestation rate of 1.57% per year for tropical forests (Geist and Lambin, 2002), on a global scale causes considerable losses in net revenues of small-scale coffee farmers. However, considering the large number of agricultural crops potentially affected by reduced pollination (Free, 1993; Steffan-Dewenter et al., 2005) the total economic losses may be considerably higher.

Improving the livelihood of a growing rural population without destroying the forest ecosystems, which provide the foundation for many of the activities of smallholders, remains one of the most challenging (scientific and societal) tasks. Future changes of land use will not automatically threaten (protected) forests and revenues of rural households. We identified and simulated policy measures and land use strategies, which might aid to identify and explore sustainable options for politicians and farmers in the real world. Our simulations clearly indicate that much of the ecological and economic value can potentially be conserved for future decades, if forest patches are preserved in the agricultural landscape due to changed attitudes of farmers and land use planners. The integration of impact analysis and spatially explicit land use modeling provides a novel tool to quantify the ecological and economical impacts of land use change scenarios, additionally providing the foundation for upscaling ecosystem-service estimates from the regional to larger spatial scales.

# 7 Synthesis

As the main result of this thesis, the SITE (Simulation of Terrestrial Environments) generic framework for integrated land-use modeling on the regional scale was designed and implemented. The basic goal is to gain new insights into interactions between land use and its biophysical and socio-economic environment. Therefore, the SITE capabilities for model integration and coupling were of specific importance. According to its first application in the context of the interdisciplinary STORMA research project, the framework needed to be able to simulate possible future changes at tropical rainforest margins to serve as decision basis for management actions. In this chapter, a summary of achievements will be provided. Furthermore, a short outlook on potential future developments and research questions will be given.

## 7.1 Summary of findings

### 7.1.1 Framework development

In this study, the design and implementation details of SITE were introduced and discussed with respect to their contribution to research on land-use dynamics. SITE was planned as an integrative tool for interdisciplinary search projects. This application scenario implied a number of specific requirements, among which the capability of integration was the most important one. Besides that, additional requirements could be identified, e.g. generic applicability, integration of calibration and model test functionality or high usability of the system and communicability of simulation results. A review of existing modeling frameworks revealed that none of them could match all of our requirements. Particular emphasis during the implementation was laid on a component-based architecture and use of the object-oriented programming paradigm. The system was designed to be expandable, thus enabling long-term usability and the possibility to integrate further developments.

To allow model integration, a generic interface for the coupling of models was implemented. This interface enables the feedback of modeling results to the calling instance. In addition, it supports parallel processing, provided that this is allowed by the modeling methodology. In addition to the model coupling interface, integration of sub models is also possible at the level of the SITE application scripting language.

SITE provides a generic modeling platform by separating general modeling functionality from the specification of actual model semantics (modeling applications). For the implementation of modeling applications, SITE resembles and enhances the concept of domain-specific languages by using an established and widely used scripting language (Python) for the actual implementation of land-use models. The functionality of the scripting language was extended to match land-use modeling requirements. With this approach, no fixed guidelines for specific modeling methodology are made, thus providing a maximum of flexibility. Integration of models is possible in this generic context.

Unlike other land-use modeling frameworks, SITE integrates functionality to automatically calibrate models. Calibration in SITE is understood as finding an optimal or adequate solution for a freely definable parameter set based on an objective function that is provided via a SITE component housing a collection of map comparison algorithms. Typically, calibration is performed using historical land-use maps as reference. The SITE calibration component is designed to contain an arbitrary number of optimization algorithms that can be freely combined with map comparison algorithms acting as objective function. The system design also include all integrated models simultaneously in the calibration process.

SITE implements an explicit representation of quantified scenarios. Model semantics and scenario data are two separate instances, thus simulations can be performed based on an arbitrary combination of model and scenario. Scenarios in SITE are handled interactively. It is possible to stop a simulation run at a predefined step and to evaluate if simulation targets are likely to be reached. Depending on the outcome of that analysis, it is possible to edit scenario parameters (e.g. management parameters). Thus, interaction of policy makers can be simulated. Conceptually, interactive scenarios establish a feedback loop to the model driving forces. This way, a major limitation in the field of scenario analysis could be overcome.

Although model development requires programming knowledge, SITE is open to be used by scientists from a wide variety of disciplines as it can be operated via a graphical user interface. It is also possible to expose the parameterization of a model via the GUI, enabling users to simply change parameters without having to edit the model code. All changes made by users via the SITE GUI are automatically recorded, thus guaranteeing reproducibility of simulation results.

The SITE framework was designed to overcome limitations of previous approaches. The entirety of innovations make it a valuable tool in the interdisciplinary field of land-use modeling, especially due to its high degree of integration it provides for components of the land sysrtem. SITE has been applied in case studies (see chapters 4 and 6) in the context of the collaborative research center “Stability of Rainforest Margins in Indonesia” (STORMA, SFB 552). At present, applications for an Indian and a Mongolian region are being developed.

### 7.1.2 Study on land-use dynamics and socio-environmental impacts

In a first case study using SITE, land-use dynamics and selected socio-environmental impacts were examined for a research area in Central Sulawesi, Indonesia. SITE was used to simulate historical land-use changes in the period 1981 to 2002. In addition, a simulation for the same time period was conducted using a scenario that did not incorporate migration in its population dynamics and crop demands. We analyzed land-use change, economic impacts and trace gas emissions under both scenarios.

For the case study, a new land-use model was developed. It explicitly simulates twelve land-use classes, including 5 different crop types (paddy rice, maize, cocoa, coffee, coconut). The land use model consists of two major compartments: suitability analysis (SUIT) and allocation (ALLOC). In SUIT, suitability maps for all land-use classes were calculated based on multicriterial analysis. These suitability maps serve as decision basis for land-use class allocation in the ALLOC module. In ALLOC, we implemented a hierarchy for assigning land use classes. The settlement level is followed by crop and forest use levels. Current and potential crop yields are incorporated in the decision process. They are calculated by the DAYCENT model which we integrated into the model structure. In this process, also changes in land-use intensities have been regarded which so far is unique in land-use modeling. The amount of fertilizer applied to crop areas serves as proxy for land-use intensity.

In this study, we showed that SITE is capable of processing relevant driving forces and simulating major impacts on the socio-environmental systems of the research area. Simulations showed that land-use dynamics in the study region are mainly characterized by a strong expansion of agricultural land at the expense of natural resources (forest). The increased cultivation of cash crops, in particular cocoa, indicates an improvement of the livelihood of the rural population. This was notably obvious for the historical simulation. However, the economic achievements can be considered unsustainable as they are based on forest conversion and the increased reallocation of crop areas into unproductive fallow land. Therefore, slowing down the consumption and conversion of forest resources and regulation of access to land can be seen as major challenges in Central Sulawesi. In the context of the STORMA project, it is planned to communicate the results from this study to local policy-makers.

### 7.1.3 Automated model calibration

The importance of model calibration is reflected in the SITE framework by the integration of calibration methodology. This methodology can be used for all models implemented in the SITE framework. It enables automated calibration based on algorithms that are capable of finding an optimal or adequate solution for a set of model parameters depending on an objective function. In the current implementation, SITE provides a genetic algorithm (GA) as optimization heuristic. Nevertheless, the SITE calibration component allows simple integration of additional methodologies. As an objective function, different map comparison algorithms, implemented in an additional SITE component, can be selected. As with the calibration component, the system is prepared for simple integration of new map comparison procedures.

In this study, we used SITE to calibrate the STORMA land-use model (chapter 4). The first task was the selection of calibration parameters. We chose parameters for which (i) values were not known from empirical evidence and (ii) altering parameter values resulted in a significant difference of the simulation result. The latter criterion was checked by performing a sensitivity analysis, comparing simulation results using the figure of merit map comparison measure. Figure of merit was also chosen as objective function in the subsequent calibration process (period 1981 to 2002), where it was used to assess similarities between the 1981 to 2002 simulation results and a 2002 reference map.

The rule set allocation hierarchy and less demanding run-time requirements allowed a two-step calibration approach, first deriving parameters for the settlement level. For the final calibration, which was performed using parallel DAYCENT instances to minimize run-time requirements, we selected seven parameters. During the calibration runs, the genetic algorithm could significantly improve solutions throughout its sequence of iterations. The calibrated model was compared to a study by Pontius et al. (2007). We found that the quality could well compete with comparable models.

In the study we could prove, that an effective automated calibration of models within the SITE framework is possible. Nevertheless, it is not possible to automate rule set parameter selection, since this is based on expert-knowledge of the specific modeling application run by SITE.

### 7.1.4 Linking deforestation scenarios to pollination services and economic returns

In another SITE application we evaluated and quantified how pollination services and economic returns of coffee are affected by deforestation in a study area in Central Sulawesi (a subset of the study area described in chapter 4). We used empirical data on the negative effects of increasing forest distance on both pollinator diversity and fruit set of coffee. Four different deforestation scenarios were applied to estimate future changes in pollination services. For the quantitative analysis, we integrated an empirical model that calculates the percentage of coffee fruit set dependent from the distance to next closest patch of natural forest.

Our land-use simulations demonstrated that depending on the magnitude and location of persistent forest conversion, pollination services can be expected to decline continuously. As a consequence of that, coffee yields are directly reduced by up to 18% and net revenues per hectare up to 14% within the next two decades as compared to average yields of the year 2001. Currently, forests in the study area annually provide pollination services of the equivalent of 46 Euros per hectare.

Besides uncovering the threat of decreasing economic returns, our simulations revealed another interesting fact that can be interpreted as a potential win-win situation. Both ecological and economical values can be preserved, if patches of forests or other natural vegetation remain in the agricultural landscape. This could be a future option for local farmers and regional land use planners.

## 7.2 Outlook

In its current state, the SITE model has proven to be a valuable tool in the field of land-use modeling. In addition to the case studies described in this thesis, SITE is already being applied in other research projects, e.g. for regional land-use modeling in Mongolia ("Integrated Water Resource Management for Central Asia: Model Region Mongolia", http://www.iwrm-momo.de) and India.

Nonetheless, there is large potential for further developments and improvements. Since SITE has been designed expandable, users can expect to be able to rapidly take advantage from new developments, while at the same time long-term usability is ensured. New features that are currently being implemented are a closer coupling to databases and the integration of tools supporting the analysis of simulation results. Mid- to long-term improvements could be e.g. the establishment of a graphical model builder on top of the SITE scripting language or the integration of parameterizable model building blocks.

Besides the large number of imaginable developments and improvements on the SITE system side, the framework will play an important role in the development of a generalized regional land-use model that is applicable and parameterizable for a variety of world regions. The development of a generalized regional land-use model is a long-term task due to a large number of remaining open research questions. SITE can support this process significantly by providing the ideal platform for development, analysis and test of model prototypes.

# Bibliography

Abdullah, F., Al-Badrahnih, L., 2000. Application of a rainfall-runoff model to three catchments in Iraq. Hydrological Sciences Journal - Journal des Sciences Hydrologiques, 45, 13–25.

Achard, F., Eva, H.D., Stibig, H.J., Mayaux, P., Gallego, J., Richards, T., Malingreau, J.P., 2002. Determination of deforestation rates of the world's humid tropical forests. Science, 297, 999–1002.

Adhiati, M.A.S., Bobsien, A. (eds), 2001. Indonesia's Transmigration Programme: An Update. http://dte.gn.apc.org/ctrans.htm.

Agarwal, C., Green, G.M., Grove, J.M., Evans, T.P., Schweik, C.M., 2002. A Review and Assessment of Land-Use Change Models: Dynamics of Space, Time, and Human Choice. U.S.D.A. Forest Service, Northeastern Research Station. GTR NE-297. Newton Square, PA.

Agarwal, D.K., Silander, J.A.Jr., Gelfand, A.E., Dewar, R.E., Mickelson, J.G.Jr., 2005. Tropical deforestation in Madagascar: analysis using hierarchical, spatially explicit, Bayesian regression models. Ecological Modelling, 185, 105–131.

Akiyama, T., Nishio, A., 1996. Indonesia's cocoa boom: hands-off policy encourages smallholder dynamism. Policy Research Working Papers, No 1580, The World Bank. Washington D.C., USA.

Alcamo, J., 2002. Three issues for improving integrated models: uncertainty, social science, and legitimacy. In: Gethmann, C.F., Lingner, S. (eds). Integrative Modellierung zum Globalen Wandel. pp 3–14. Springer, Berlin, Heidelberg, New York.

Alcamo, J., Leemans, R., Kreilemann, E., 1998. Global Change Scenarios of the 21st Century – Results from the IMAGE 2.1 model. Elsevier Science. Oxford.

Alcamo, J., Endejan, M., Kaspar, F., Rösch, T., 2001. The GLASS model: a strategy for quantifying global environmental security. Environmental Science and Policy, 4, 1–12.

Alcamo, J., Kok, K., Busch, G., Priess, J.A., 2006. Searching for the Future of Land: Scenarios from the Local to Global Scale. In: Lambin, E.F., Geist, H.J. (eds). Land-Use and Land-Cover Change. pp 137–155. Springer.

Alves, D.S., 2002. Space-time dynamics of deforestation in Brazilian Amazonia. International Journal of Remote Sensing, 23, 2903–2908.

Angelsen, A., Kaimowitz, D., 1999. Rethinking the Causes of Deforestation: Lessons from Economic Models. World Bank Research Observer, 14, 73–98.

Argent, R.M., 2004. An overview of model integration for environmental applications – components, frameworks and semantics. Environmental Modelling & Software, 19, 219–234.

Baker, W.L., 1989. A review of models of landscape change. Landscape Ecology, 2, 111–133.

Balmford, A., Bruner, A., Cooper, P., Costanza, R., Farber, S., Green, R.E., Jenkins, M., Jefferiss, P., Jessamy, V., Madden, J., Munro, K., Myers, N., Naeem, S., Paavola, J., Rayment, M., Rosendo, S., Roughgarden, J., Trumper, K., Turner, R.K., 2002. Economic reasons for conserving wild nature. Science, 297, 950–953.

Balvanera, P., Kremen, C., Martinez-Ramos, M., 2005. Applying community structure analysis to ecosystem function: examples from pollination and carbon storage. Ecological Applications, 15, 360–375.

Balzert, H., 2000. Lehrbücher der Informatik. Bd.1. Software-Entwicklung: Lehrbuch der Software-Technik. Spektrum, Akad. Verlag. Heidelberg, Berlin.

Batty, M, 1997. Cellular Automata and urban form: A primer. Journal of the American Planning Association, 63, 266–274.

Berger, T., 2001. Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy. Agricultural Economics, 25 (2), 245–260.

Bockstael, N.E., 1996. Modeling economics and ecology: the importance of a spatial perspective. American Journal of Agricultural Economics, 78, 1168–1180.

Boomgard, P.., 1993. Upliftment down the drain? Effects of welfare measures in late colonial Indonesia. In: Dirkse, J.-P. (ed). Development and social welfare: Indonesia's experiences under the New Order. pp 247–253. Springer, Leiden.

Boumans, R.M., Villa, F., Costanza, A., Voinov, A., Voinov, H., 2001. Non-spatial calibrations of a general unit model for ecosystem simulations. Ecological Modelling, 146, 17–32.

Bousquet, F., Bakam, I., Proton, H., LePage, C., 1998. Cormas: common-pool resources and multiagent systems. Lecture Notes in Artificial Intelligence, 1416, 826–837.

BPS, (ed), 1971. Sulawesi Tengah dalam angka tahun 1970. Palu, Indonesia.

BPS, (ed), 1981. Registrasi penduduk propinsi Sulawesi Tengah akhir tahun 1980. Palu, Indonesia.

BPS, (ed), 2001. Statistics Indonesia. URL: http://www.bps.go.id.

BPS, (ed), 2003. Statistics Indonesia. URL: http://www.bps.go.id.

BPS, (ed), 2004. Statistics Indonesia. URL: http://www.bps.go.id.

Brovkin, V., Ganopolski, A., Claussen, M., Kubatzki, C., Pethoukov, V., 1999. Modelling climate response to historical land cover change. Global Ecology and Biogeography, 8, 509–517.

Brown, D.G., Walker, R., Manson, K., S.and Seto, 2004. Modeling land use and land cover change. In: Gutman, G., Janetos, A.C., Justice, C.O., Moran, E.F., Mustard, J.F., Rindfuss, R.R., Skole, D.L., Turner, B.L., Cochrane, M.A. (eds). Land Change Science: Observing, Monitoring and Understanding Trajectories of Change on the Earth's Surface. pp 395–409. Springer, New York.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M., 1998. Pattern-orientierte Software-Architektur: Ein Pattern-System. Addison-Wesley-Longman. Bonn, Paris.

Candau, L., 2000. Calibrating a cellular automaton model of urban growth in a timely manner. In: Parks, B.O., Clarke, K.M., Crane, M.P. (eds). Proceedings of the 4th International Conference on Integrating Geographic Information Systems and Environmental Modelling: Problems, Prospects, and Needs for Research. University of Colorado, Boulder.

Carr, D., Barbieri, A., Pan, W., Iravani, H., 2006. Agricultural change and limits to deforestation in Central America. In: Brower, F., McCarl, B.A. (eds). Agriculture and climate beyond 2015. pp 91–107. Springer, Dordrecht.

Chapin, F.S., Zavaleta, E.S., Eviner, V.T., Naylor, R.L., Vitosek, P.M., Reynolds, H.L., Hooper, D.U., Lavorel, S., Sala, O.E., Hobbie, S.E., Mack, M.C., Diaz, S., 2000. Consequences of changing biodiversity. Nature, 405, 234–242.

Charney, J., Stone, P.H., 1975. Drought in the Sahara: A biogeophysical feedback mechanism. Science, 187, 434–435.

Chen, M., Norman, R.J., 1992. A framwork for integrated CASE. IEEE Software, 9, 18–22.

Clark, J.D., 1992. Modeling and simulating complex spatial dynamic systems: a framework for application in environmental analysis. Simulation Digest, 21, 9–19.

Clarke, K.C., Gaydos, L.J., 1998. Loose-coupling a cellular automaton model and gis: long-term urban growth prediction for San Francisco and Washington/Baltimore. International Journal of Geographical Information Science, 12, 699–714.

Clarke, K.C., Gaydos, L.J., Hoppen, S., 1996. A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area. Environment and Planning B, 24, 247–261.

Cooper, V.A., Nguyen, V.T.V., Nicell, J.A., 1997. Evaluation of global optimization methods for conceptual rainfall-runoff model calibration. Water Science and Technology, 36, 53–60.

Costanza, R., d'Arge, D., Bruner, A., Cooper, P., De Groot, R., Grasso, M., Hannon, B., Limburg, K., Naeem, S., O'Neill, R.V., Paruela, J., Raskin, R.G., Sutton, P., Van Den Belt, M., 1997. The value of the world's ecosystem services and natural capital. Nature, 387, 253–260.

Costanza, R., Duplisea, D., Kautsky, U., 1998. Ecological modeling and economic systems with STELLA. Ecological Modelling, 110, 1–4.

Daily, G.C., Alexander, S., Ehrlich, P., Goulder, L., Lubchenco, J., Matson, P., Mooney, H.A., Postel, S., Schneider, S.H., Tilman, D., Woodwell, G.M., 1997. Ecosystem services: benefits supplied to human societies by natural ecosystem. Issues in Ecology, 2, 1–16.

De Marco, P., Coelho, F.M., 2004. Services performed by the ecosystem: forest remnants influence agricultural cultures' pollination and production. Biodiversity and Conservation, 13, 1245–1255.

De Vasconcelos, M., Zeigler, B., 1993. Discrete-event simulation of forest landscape response to fire disturbance. Ecological Modelling, 65, 177–198.

Eastman, J.R., Weigen, J., Kyem, P., Toledano, J., 1995. Raster Procedures for Multiobjective Land-Use Planning. Photogrammetric Engineering and Remote, 65, 177–198.

Endejan, M., 2003. Entwicklung einer Software-Architektur für Systeme zum integrierten simulationsbasierten Assessment des globalen Wandels. PhD thesis. University of Kassel. Kassel, Germany.

Engelen, G., Geertman, S., Smits, P., Wessels, C., 1999. Dynamic GIS and Strategic Physical Planning: A Practical Application. In: Stillwell, J., Geertman, S., Openshaw, S. (eds). Geographical Information and Planning. Advances in Spatial science. Springer.

Erasmi, S., Priess, J.A., 2007. Satellite and survey data: a multisource approach to study land-cover/land-use change in Indonesia. In: Dickmann, F. (ed). Geovisualisierung in der Humangeographie, Kartographische Schriften, Vol. 3. pp 101–114. Kirschbaum Verlag.

Erasmi, S., Twele, A., Ardiansyah, M., Malik, A., Kappas, M., 2004. Mapping deforestation and land cover conversion at the rainforest margin in Central Sulawesi, Indonesia. EARSeL eProceedings, 3, 388–397.

Erik, W., 2002. Walhi award torn up in protest at Lore Lindu illegal logging. The Jakarta Post, Oct. 26, 2002. Jakarta, Indonesia.

Etter, A., McAlpine, C., Wilson, K., Phinn, S., Possingham, H., 2006. Regional patterns of agricultural land use and deforestation in Colombia. Agriculture, Ecosystems & Environment, 114, 369–386.

Fahrig, L., 2003. Effects of habitat fragmentation on biodiversity. Annual Review of Ecology, Evolution and Systematics, 34, 487–515.

Fall, A., Fall, J., 2001. A domain-specific language for models of landscape dynamics. Ecological Modelling, 141, 1–18.

FAO, (ed), 2004. FAOSTAT data. Food and Agriculture Organization of the United Nations. URL http://apps.fao.org.

FAO, UNEP (ed), 1999. Terminology for Integrated Resources Planning and Management. Food and Agriculture Organization / United Nations Environmental Programme. Rome, Italy / Nairobi, Kenia.

Farooqui, K., Logrippo, L., Meer, J.de, 1995. The ISO Reference Model for open distributed processing: an introduction. Computer Networks and ISDN Systems, 2, 1215–1229.

Fearnside, P.M., 1997a. Transmigration in Indonesia: Lessons from its Environmental and Social Impacts. Environmental Management, 21, 553–570.

Fearnside, P.M., 1997b. Biodiversity as an environmental service in Brazil's Amazonian forests: risks, value and conservation. Environmental Conservation, 26, 305–321.

Filippi, J.-B., Bisgambiglia, P., 2004. JDEVS: an implementation of a DEVS based formal framework for environmental modelling. Environmental Modelling & Software, 19, 261–274.

Fischer, G., Sun, L.X., 2001. Model based analysis of future land-use development in China. Agriculture, Ecosystems & Environment, 85, 163–176.

Free, J.B., 1993. Insect pollination of crops. Academic Press. London, UK.

Frelich, L.E., Lorimer, C.G., 1991. A simulation of landscape-level stand dynamics in the northern hardwood region. Journal of Ecology, 79, 223–234.

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. Boston, San Francisco, USA.

Gao, Q., 1996. Dynamic modeling of ecosystems with spatial heterogeneity – A structured approach implemented in Windows environment. Ecological Modelling, 85, 241–252.

Gathmann, A., Tscharntke, T., 2002. Foraging ranges of solitary bees. Journal of Animal Ecology, 71, 757–764.

Geist, H.J., Lambin, E.F., 2001. What Drives Tropical Deforestation. LUCC Report Series, 4. LUCC International Project Office. Louvain-la-Neuve.

Geist, H.J., Lambin, E.F., 2002. Proximate Causes and Underlying Driving Forces of Tropical Deforestation. BioScience, 52, 143–150.

GLP, (ed), 2005. Science Plan and Implementation Strategy. IGBP Report No. 53 / IHDP Report No. 19. IGBP Secretariat, Stockholm.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley. Reading, MA, USA.

Goldstein, N.C., 2003. Brain vs. Brawn – Comparative strategies for the calibration of a Cellular Automata-based Urban Growth Model. Proceedings of the 7th International Conference on GeoComputation.

GSG, (ed), 2004. Global Scenario Group of the Stockholm Environmental Institute, Boston Branch. URL http://www.gsg.org.

Guo, Z.W., Xiao, X.M., Li, D.M., 2000. An assessment of ecosystem services: water flow regulation and hydroelectric power production. Ecological Applicatons, 10, 925–936.

Haertel, M., Erasmi, S., Maertens, M., Faust, H., Birner, R., Waltert, M., Priess, J.A., 2002. Land Use Map of the Palu Region. In: Maertens, M. (ed). Ecological Modeling of Agricultural Land-Use Patterns in Forest Frontier Areas. Theory, Empirical Assessment and Policy Implications for Central Sulawesi, Indonesia. Disseration, University of Göttingen.

He, H.S., Larsen, D.R., Mladenoff, D.J., 2002. Exploring component-based approaches in forest landscape modeling. Environmental Modelling & Software, 17, 519–529.

Heistermann, M., Müller, C., Ronneberger, K., 2006. Land in sight? Achievements, deficits and potentials of continental to global scale land-use modeling. Agriculture, Ecosystems & Environment, 114, 141–158.

Hilferink, M., Rietveld, P., 1999. Land use scanner: an integrated gis model for long term projections of land-use in urban and rural areas. Journal of Geographical Systems, 1, 155–177.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press. Ann Arbor, MI, USA.

Houghton, R.A., Boone, R.D., Melillo, J.M., Palm, C.A., Woodwell, G.M., Myers, N., Moore, B., Skole, D.L., 1985. Net flux of carbon dioxide from tropical forest in 1980. Nature, 316, 617–620.

Hubacek, K., Sun, L., 2001. A scenario analysis of China's land use and land cover change: incorporating biophysical information into input output modeling. Structural Change and Economic Dynamics, 12, 367–397.

Irwin, E.G., Geoghegan, J., 2001. Theory, data, methods: developing spatially explicit economic models of land use change. Agriculture, Ecosystems & Environment, 85, 7–23.

Ishizuka, S., Tsuruta, H., Murdiyarso, D., 2002. An intensive field study on $CO_2$, $CH_4$, and $N_2O$ emissions from soils at four land-use types in Sumatra, Indonesia. Global Biogeochemical Cycles, 16 (3).

Itami, R., 1994. Simulating spatial dynamics: cellular automata theory. Landscape and Urban Planning, 30, 24–47.

Jenerette, G.D., Wu, J., 2001. Analysis and simulation of land-use change in the central Arizone-Phoenix region, USA. Landscape Ecology, 16, 611–626.

Kaimowitz, D., Angelsen, A., 1998. Economic Models of Tropical Deforestation: A Review. Center for International Forestry Research. Bogor, Indonesia.

Kearns, C.A., Inouye, D.W., Waser, N.M., 1998. Endangered mutualism: The conservation of plant-pollinator interactions. Annual Review of Ecology and Systematics, 29, 83–112.

Keil, A., Birner, R., Zeller, M., 2007. Potentials to reduce deforestation by enhancing the technical efficiency of crop production in forest margin areas. In: Tscharntke, T., Leuschner, C., Guhardja, E., Zeller, M. (eds). The Stability of Tropical Rainforest Margins: Linking Ecological, Economic and Social Constraints of Land Use and Conservation. pp 391–416. Springer, Berlin, London, New York.

Kirkpatrick, S, Gelatt, C.D.Jr., Vecchi, M.-P., 1983. Optimization by simulated annealing. Science, 220, 671–680.

Klein, A.M., Steffan-Dewenter, I., Tscharntke, T., 2003a. Fruit set of highland coffee increases with the diversity of pollinating bees. Proceedings of the Royal Society of London B, 270, 955–961.

Klein, A.M., Steffan-Dewenter, I., Tscharntke, T., 2003b. Pollination of *Coffea canephora* in relation to local and regional agroforestry management. Journal of Applied Ecology, 40, 837–845.

Klein, A.M., Steffan-Dewenter, I., Tscharntke, T., 2003c. Bee pollination and fruit set of *Coffea arabica* and *C. canephora* (Rubiaceae). Maerican Journal of Botany, 90, 153–157.

Klenner, W., Kurz, W.A., Webb, T.M., 1997. Projecting the spatial and temporal distribution of forest ecosystem characteristics. In: Proceedings GIS 97. pp 418–421. GIS World Inc., Ft. Collins, Colorado, USA.

Klug, W., Graziani, G., Grippa, G., Pierce, D., Tassone, D., 1992. Evaluation of long range atmospheric transport models using environmental radioactivity data from the Chernobyl accident: The ATMES report. Elsevier. London, UK.

Krauchi, N., 1995. Application of the model FORSUM to the Solling spruce site. Ecological Modelling, 83, 219–228.

Kreisel, W., Weber, R., Faust, H., 2004. Historical Impacts on Use and Management of Natural Resources in the Rainforest Margins of Central Sulawesi. In: Gerold, G., Fremerey, M., Guhardja, E. (eds). Land Use, Nature Conservation and the Stability of Rainforest margins in Southeast Asia. pp 39–65. Springer, Berlin, New York.

Kremen, C., Niles, O., Dalton, M.G., Daily, G.C., Ehrlich, P.R., Fay, J.P., Grewal, D., Guillery, R.P., 2000. Economic Incentives for Rain Forest Conservation Across Scales. Science, 288, 1828–1832.

Kremen, C., Williams, N.M., Thorp, R.W., 2002. Crop pollination from native bees at risk from agricultural intensification. Proceedings of the National Academy of Science, 99, 16812–16816.

Kremen, C., Williams, N.M., Bugg, R.L., Fay, J.P., Thorp, R.W., 2004. The area requirements of an ecosystem service: crop pollination by native bee communities in California. Ecology Letters, 7, 1109–1119.

Kruyt, A.C., 1938. De West-Toradjas op Midden-Celebes. 4 vols. Noord-Hollandsche Uitgevers-Maatschappij. Verhandelingen der Koninklijke Nederlandsche Akademie van Wetenschapen te Amsterdam, Afdeeling Letterkunde, Nieuwe Reeks, Deel XL. Amsterdam, The Netherlands.

Kuhl, F., Weatherly, R., Dahmann, J., 1999. Creating Computer Simulation Systems: An Introduction to the High Level Architecture. Prentice Hall PTR. Upper Saddle River, NJ, USA.

Kuhnert, M., Voinov, A., Seppelt, R., 2005. Comparing Raster Map Comparison Algorithms for Spatial Modeling and Analysis. Photogrammetric Engineering & Remote Sensing, 71, 975–984.

Lambin, E.F., 1997. Modelling and monitoring land-cover change processes in tropical regions. Progress in Physical Geography, 21, 375–393.

Lambin, E.F., Geist, H.J., 2003. Regional Differences in Tropical Deforestation. Environment, 45, 22–36.

Lambin, E.F., Baulies, X., Bockstael, N.E., Fischer, G., Krug, T., Leemans, R., 2000a. Land-use and land-cover change (LUCC), implementation strategy. IGBP Report 48, IHDP Report 10. IGBP, IHDP. Stockholm, Bonn.

Lambin, E.F., Rounsevell, M.D.A., Geist, H.J., 2000b. Are agricultural models able to predict changes in land-use intensity? Agriculture, Ecosystems & Environment, 82, 321–331.

Lambin, E.F., Turner, B.L., Geist, H.J., Agbola, S.B., Angelsen, A., Bruce, J.W., Coomes, O.T., Dirzo, R., Fischer, G., Folke, C., George, P.S., Homewood, K., Imbernon, J., Leemans, R., Li, X., Moran, E.F., Mortimore, M., Ramakrishnan, P.S., Richards, J.F., Skanes, H., Steffen, W., Stone, G.D., Svedin, U., Veldkamp, T.A., Vogel, C., Xu, J., 2001. The causes of land-use and land-cover change: moving beyond the myths. Global Environmental Change, 11, 261–269.

Lambin, E.F., Geist, H.J., Lepers, E., 2003. Dynamics of Land-Use and Land-Cover Change in Tropical Regions. Annual Review of Environment and Resources, 28, 205–241.

Lambin, E.F., Geist, H.J., Rindfuss, R.R., 2006. Local processes with global impacts. In: Lambin, E.F., Geist, H.J. (eds). Land-Use and Land-Cover Change. pp 1–8. Springer.

Lau, L., Young, R.A., McKeon, G., Syktus, J., Duncalfe, F., Graham, N., McGregor, J., 1999. Downscaling global information for regional benefit: coupling spatial models at varying space and time scales. Environmental Modelling & Software, 14, 519–529.

Li, T.M. (ed), 1999. Transforming the Indonesian uplands: marginality, power and production. Routledge. London, UK.

Li, X., Gar-On Yeh, A., 2001. Calibration of cellular automata by using neural networks for the simulation of complex urban systems. Environment and Planning A, 33, 1445–1462.

Lindenschmidt, K.E., Rauberg, J., Hesser, F.B., 2005. Extending uncertainty analysis of a hydrodynamic-water quality modelling system using high-level architecture (HLA). Water Quality Research Journal of Canada, 40, 59–70.

Liu, Y., Phinn, S.R., 2001. Developing a cellular automaton model of urban growth incorporating fuzzy set approaches. In: Pullar, D.V. (ed). Proceedings of the 6th International conference on GeoComputation. University of Queensland, Brisbane, Australia.

Lopez, A., 2004. Population movements, environmental change and social conflicts in the Brazilian Amazon. In: Unruh, J.D., Krol, M.S., Kliot, N. (eds). Environmental Change and its Implications for Population Migration. pp 145–163. Kluwer, Dordrecht.

Lorek, H., Sonnenschein, M., 1998. Object-oriented support for modelling and simulation of individual-oriented ecological models. Ecological Modelling, 108, 77–96.

Maertens, M., 2003. Economic Modeling of Agricultural Land-Use Patterns in Forest Frontier Areas. Theory, Empirical Assessment and Policy Implications for Central Sulawesi, Indonesia. PhD thesis. University of Göttingen. Göttingen, Germany.

Maertens, M., Zeller, M., Birner, R., 2003. Does Technical Progress in Agriculture Have a Forest Saving or a Forest Clearing Effect. In: Gerold, G., Fremerey, M., Guhardja, E. (eds). Land Use, Nature Conservation and the Stability of Rainforest Margins in Southeast Asia. pp 179–197. Springer, Berlin, NewYork.

Manson, S.M., 2001. Simplifying complexity: a review of complexity theory. Geoforum, 32, 405–414.

Mas, J.F., Puig, H., Palacio, J.L., Sosa-López, A., 2004. Modelling deforestation using GIS and artificial neural networks. Environmental Modelling & Software, 19, 461–471.

Matthews, K.B., Hutchins, M.G., Hill, G., 2006. Bridging the Design-Use Gap for DSS in Environmental Policy and Practice. In: Voinov, A., Jakeman, A.J., Rizzoli, A.E. (eds). Proceedings of the IEMSS Third Biennial Meeting: Summit on Environmental Modelling and Software. International Environmental Modelling and Software Society, Burlington, USA.

Maxwell, T., Costanza, R., 1997. A language for modular spatio-temporal simulation. Ecological Modelling, 103, 105–113.

McNeill, J., Alves, D., Arizpe, L., Bykova, O., Galvin, K., Kelmelis, J., Migot-Adholla, S., Morrisette, P., Moss, R., Richards, J., Riebsame, W., Sadowski, F., Sanderson, S., Skole, D., Tarr, J., Williams, M., Yadav, S., Young, S., 1994. Toward a Typology and Regionalization of Land-Cover and Land-Use Change: Report of Working Group B. In: Meyer, W.B., Turner, B.L. (eds). Changes in Land Use and Land Cover. pp 55–72. Cambridge University Press.

MEA, (ed), 2005. Ecosystems and Human Well-Being: Synthesis. Island Press. Washington D.C., USA.

Mendelsohn, R., Dinar, A., 1999. Climate change, agriculture and developing countries: does adaptation matter? World Bank Research Observer, 14, 277–293.

Messina, J.P., Walsh, S.J., 2001. Morphogenesis: modeling landuse and landcover dynamics in the Ecuadorian Amazon. Plant Ecology, 156, 75–88.

Meyer, W.B., Turner, B.L., 1994. Changes in Land Use and Land Cover: A Global Perspective. Cambridge University Press. Cambridge.

Miller, J.H., 1998. Active Nonlinear Tests (ANTs) of Complex Simulation Models. Management Science, 44, 820–830.

Minar, N., Burkhart, R., Langton, C., Askenazi, M., 1996. The Swarm simulation system: A toolkit for for building multi-agent simulations. http://www.swarm.org.

Mitchell, M., 1998. An Introduction to Genetic Algorithms. MIT Press. Cambridge, MA, USA.

Mladenoff, D.J., Host, G.E., Boeder, J., Crow, T.R., 1996. Landis: a spatial model of forest landscape disturbance, succession and management. In: Goodchild, M., Steyaert, L.T., Parks, B.O. (eds). GIS and Environmental Modelling. pp 175–179. GIS World Books.

Monserud, R.A., Leemans, R., 1992. Comparing global vegetation maps with the Kappa statistic. Ecological Modelling, 62, 221–233.

Muleta, M.K., Nicklow, J.W., 2005. Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. Journal of Hydrology, 306, 127–145.

Murniati, , Garrity, D.P., Gintings, A.N., 2001. The contribution of agroforestry systems to reducing farmers' dependence on the resources of adjacent national parks: a case study from Sumatra, Indonesia. Agroforestry Systems, 52, 171–184.

Nair, P.K.R., 2000. An Introduction to Agroforestry. Kluwer. Dordrecht, The Netherlands.

Ndiritu, J.G., Daniell, T.M., 2001. An Improved Genetic Algorithm for Rainfall-Runoff Model Calibration and Function Optimization. Mathematical and Computer Modelling, 33, 695–706.

Neill, C., Steudler, P.A., Garcia-Montiel, D.C., Melillo, J.M., Feigl, B.J., Piccolo, M.C., Cerri, C.C., 2005. Rates and controls of nitrous oxide and nitric oxide emissions following conversion of forest pasture in Rondonia. Nutrient Cycling in Agroecosystems, 71, 1–15.

Neilson, J., 2007. Global markets, farmers and the state: sustaining profits in the Indonesian cocoa sector. Bulletin of Indonesian Economic Studies, 43, 227–250.

O'Brien, T.G., Kinnaird, M.F., 2003. Caffeine and Conservation. Science, 300, 587.

Oliva, R., 2003. Model calibration as a testing strategy for system dynamics models. European Journal of Operations Research, 151, 552–568.

OMG, (ed), 1999. OMG Unified Modeling Language - Specification V.1.3 (June 1999). Object Mangement Group.

Organization, International Coffee (ed), 2004. Historical data, producer prices for arabica coffee in Indonesia 1993-2003. URL http://www.ico.org/index.htm.

Otten, R.H.J.M., 1989. The Annealing Algorithm. Kluwer Academic Publishers. Boston, MA, USA.

Ottermann, J., 1974. Baring high-albedo soils by overgrazing: A hypothesized desertification mechanism. Science, 86, 531–533.

Oxley, T., McIntosh, B.S., Winder, N., Mulligan, M., Engelen, G., 2004. Integrated modelling and decision-support tools: a Mediterranean example. Environmental Modelling & Software, 19, 999–1010.

Parker, D.C., Manson, S.M., Janssen, M.A., Hoffmann, M.J., Deadman, P., 2003. Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review. Annals of the Association of American Geographers, 93, 314–337.

Parton, W.J., Hartman, M., Ojima, D., Schimel, D., 1998. DAYCENT and its land surface submodel: description and testing. Global and Planetary Science, 19, 35–48.

Pattanayak, S.K., Kramer, R.A., 2001. Worth of watersheds: a producer surplus approach for valuing drought mitigation in Eastern Indonesia. Environment and Development Economics, 6, 123–146.

Percivall, G. (ed), 2002. The OpenGIS Abstract Specification – Topic 12: OpenGIS Service Architecture. Open Geospatial Consortium. Wayland, MA, USA.

Perica, S., Foufoula-Georgiou, E., 1996. Model for the multiscale disaggregation of spatial rainfall based on coupling meteorological and scaling descriptions. Journal of Geophysical Research, 101, 26347–26361.

Pijanowski, B.C., Pithadia, S., Shellito, B.A., Alexandridis, K., 2005. Calibrating a neural network-based urban change model for two metropolitan areas of the Upper Midwest of the United States. International Journal of Geographical Information Science, 19, 197–215.

Pontius, R.G.Jr., 2000. Quantification Error Versus Location Error in Comparison of Categorial Maps. Photogrammetric Engineering & Remote Sensing, 66, 1011–1016.

Pontius, R.G.Jr., 2002. Statistical Methods to Partition Effects of Quantity and Location During Comparison of Categorial Maps at Multiple Resolutions. Photogrammetric Engineering & Remote Sensing, 68, 1041–1049.

Pontius, R.G.Jr., Pacheco, P., 2004. Calibration and validation of a model of forest disturbance in the Western Ghats, India 1920-1990. GeoJournal, 61, 325–334.

Pontius, R.G.Jr., Schneider, L.C., 2001. Land-cover change model validation by an roc method for the Ipswich watershed, Massachusetts, USA. Agriculture, Ecosystems & Environment, 85, 239–248.

Pontius, R.G.Jr, Cornell, J.D., Hall, C.A.S., 2001. Modeling the spatial pattern of land-use change with GEOMOD2: application and validation for Costa Rica. Agriculture, Ecosystems & Environment, 85, 191–203.

Pontius, R.G.Jr., Boersma, W., Castella, J.-C., Clarke, K., De Nijs, T., Dietzel, C., Zengquiang, D., Fotsing, E., Goldstein, N., Kok, K., Koomen, E., Lippitt, C.D., McConnell, W., Pijanowski, B., Pithadia, S., Sood, A.M., Sweeney, S., Trung, T.N., Veldkamp, A.T., Verburg, P., 2007. Comparing the input, output, and validation maps for several models of land change. Annals of Regional Science, in press.

Power, C., Simms, A., White, R., 2001. Hierarchical fuzzy pattern matching for the regional comparison of land use maps. International Journal of Geographical Information Science, 15, 77–100.

Priess, J.A., De Koning, G.H.J., Veldkamp, A., 2001. Assessment of interactions between land use change and carbon and nutrient fluxes. Agriculture, Ecosystems & Environment, 85, 269–279.

Pullar, D., 2004. SimuMap: a computational system for spatial modelling. Environmental Modelling & Software, 19, 235–243.

Purbopuspito, J., Veldkamp, E., Brumme, R., Murdiyarso, D., 2006. Trace gas fluxes and nitrogen cycling along an elevation sequence of tropical montane forests in Central Sulawesi, Indonesia. Global Biogeochemical Cycles, 20 (3).

Rahman, J.M., Seaton, S.P., Cuddy, S.M., 2004. Making frameworks more usable: using model introspection and metadata to develop model processing tools. Environmental Modelling & Software, 19, 275–284.

Raskin, P., Banuri, T., Gallopin, G., Gutman, P., Hammond, A., Kates, R., Swart, R. (eds), 2002. Great Transition: The Promise and Lure of the Times Ahead. A Report of the Global Scenario Group. SEI PoleStar Series Report no. 10. Boston, Stockholm Environmental Institute.

Recknagel, F., 2001. Applications of machine learning to ecological modelling. Ecological Modelling, 146, 303–310.

Renner, S.C., Voigt, M., Markussen, M., 2006. Regional deforestation in a tropical montane cloud forest in Alta-Verapaz, Guatemala. Ecotropica, 12, 43–49.

Ricketts, T.H., 2004. Tropical forest fragments enhance pollinator activity in nearby coffee crops. Conservation Biology, 18, 1262–1271.

Ricketts, T.H., Daily, G.C., Ehrlich, P.R., Michen, C.D., 2004. Economic value of tropical forest to coffee production. PNAS, 101, 12579–12582.

Roubik, D., 2002a. The value of bees to the coffee harvest. Nature, 417, 708.

Roubik, D., 2002b. Africanized honeybees and coffee yields. In: Pollinating bees. The conservation link between agriculture and nature. Proceedings of the Workshop on the Conservation and Sustainable Use of Pollinators in. Sao Paulo, Brazil.

Rubiano, J., 2000. Land use change in tropical hillsides: the influence of pattern on process. Advances in Environmental Monitoring and Modelling, 1, 61–79.

Ruf, F., Yoddang, , 2001. Cocoa migrants from boom to bust. In: Gerard, F., Ruf, F. (eds). Agriculture in crisis: people, commodities and natural resources in Indonesia. pp 97–156. Montpellier, Richmond.

Sagan, C., Toon, O.B., Pollack, J.B., 1979. Anthropogenic albedo changes and the Earth's climate. Science, 206, 1363–1368.

Sala, O., Chapin, F., Armesto, J., Berlow, E., Bloomfield, J., Dirzo, R., Huber-Sanwald, E., Huenneke, L., Jackson, R., Kinzig, A., Leemans, R., Lodge, D., Mooney, H., Oesterheld, M., Poff, N., Sykes, M., Walker, B., Walker, M., Wall, D., 2000. Biodiversity - global biodiversity scenarios for the year 2100. Science, 287, 1770–1774.

Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M., 2004. Sensitivity Analysis in Practice. A Guide to Assessing Scientific Models. John Whiley & Sons. Chichester, England.

Schaldach, R., 2004. Räumliche Modellierung der Dynamik von Landnutzung in Hessen und ihrer Wirkung auf die biologische Kohlenstoffspeicherung. PhD thesis. University of Kassel. Kassel, Germany.

Schaldach, R., Alcamo, J., 2006. Coupled simulation of regional land use change and soil carbon sequestration: A case study from the state of Hesse in Germany. Environmental Modelling & Software, 21, 1430–1446.

Schneider, L.C., Pontius, R.G., 2001. Modeling land-use change in the Ipswich watershed, Massachusetts, USA. Agriculture, Ecosystems & Environment, 85, 83–94.

Schürmann, G., 1995. The evolution from open systems interconnection (OSI) to open distributed processing (ODP). Computer Standards & Interfaces, 17, 107–113.

Schulze, T., Strassburger, S., Klein, U., 1999. Migration of the HLA into civil domains: Solutions and prototypes for transportation applications. Simulation, 73, 296–303.

Schwarze, S., 2004. Determinants of income generating activities of rural households: a quantitative study in the vicinity of the Lore Lindu National Park in Central Sulawesi, Indonesia. PhD thesis. Institute of Rural Development, University of Göttingen. Göttingen, Germany.

Schwarze, S., Zeller, M., 2005. Income diversification of rural households in Central Sulawesi, Indonesia. Quaterly Journal of International Agriculture, 44, 61–73.

Seppelt, R., Voinov, A., 2002. Optimization methodology for land use patterns using spatially explicit landscape models. Ecological Modelling, 151, 125–142.

Serneels, S., Said, M.Y., Lambin, E.F, 2001. Land cover change around a major east African wildlife reserve: the Mara Ecosystem (Kenya). International Journal of Remote Sensing, 22, 3397–3420.

Shriar, A.J., 2001. The Dynamics of Agricultural Intensification and Resource Conservation in the Buffer Zone of the Maya Biosphere Reserve, Petén, Guatemala. Human Ecology, 29, 27–48.

Snyder, J.P., 1987. Map Projections – A Working Manual. U.S. Geological Survey Professional Paper 1395. United States Government Printing Office. Washington, D.C., USA.

Steffan-Dewenter, I., Potts, S.G., Packer, L., 2005. Pollinator diversity and crop pollination services at risk. Trends in Ecology and Evolution, 20, 651–652.

Stehfest, E., 2005. Modelling of global crop production and resulting $N_2O$ emissions. PhD thesis. University of Kassel. Kassel, Germany.

Stehfest, E., Bouwman, A.F., 2006. $N_2O$ and NO emission from agricultural fields and soils under natural vegetation: summarizing available measurement data and modeling of global annual emissions. Nutrient Cycling in Agroecosystems, 74, 207–228.

Stewart, T.J., Janssen, R., Van Herwijnen, M., 2004. A genetic algorithm approach to multiobjective land use planning. Computers & Operations Research, 31, 2293–2313.

STORMA, (ed), 2003. Stability of Rainforest Margins. Application for continued funding of the SFB 552. University of Göttingen. Göttingen, Germany.

Stéphenne, N., Lambin, E.F., 2001. A dynamic simulation model of land-use changes in Sudano-sahelian countries of Africa (SALU). Agriculture, Ecosystems & Environment, 85, 145–161.

Straatman, B., White, R., Engelen, G., 2004. Towards an automatic calibration procedure for constrained cellular automata. Computer, Environment and Urban Systems, 28, 149–170.

Sumner, N.R., Fleming, P.M., Bates, B.C., 1997. Calibration of a modified SFB model for twenty-five Australian catchments using simulated annealing. Journal of Hydrology, 197, 166–188.

Thyer, M., Kuczera, G., Bates, B.C., 1999. Probabilistic optimization for conceptual rainfall-runoff models: A comparison of the shuffled complex evolution and simulated annealing algorithms. Water Resources Research, 35, 767–773.

Tscharntke, T., Leuschner, C., Zeller, M., Guhardja, E., Bidin, A. (eds), 2007. Stability of Tropical Rainforest Margins – Linking Ecological, Economic and Social Constraints of Land Use and Conservation. Springer. Berlin, Heidelberg.

Turner, B.L., 1994. Local faces, global flows: the role of land use and land cover in global environmental change. Land Degradation and Rehabilitation, 5, 71–78.

Turner, B.L., Skole, D., Sanderson, S., Fischer, G., Fresco, L., Leemans, R., 1995. Land-Use and Land-Cover Change Science/Research Plan. Joint Publication of the International Geosphere-Biosphere Programme (Report No. 35) and the Human Dimensions of the Global Environmental Change Programme (Report No. 7). Royal Swedish Academy of Sciences, Stockholm.

Turner, B.L., Villar, S.C., Foster, D., Geoghegan, J., Keys, E., Klepeis, P., Lawrence, D., Mendoza, P.M., Manson, S., Ogneva-Himmelberger, Y., Plotkin, A.B., Salicrup, D.P., Chowdhury, R.R., Savitsky, B., Schneider, L., Schmook, B., Vance, C., 2001. Deforestation in the southern Yucatan peninsular region: an integrative approach. Forest Ecology and Management, 154, 353–370.

Twele, A., Kappas, M., Lauer, J., Erasmi, S., 2006. The effect of stratified topographic correction on land cover classification in tropical mountainous regions. In: Kerle, N., Skidmore, A. (eds). Remote Sensing: From Pixels to Processes. Proceedings of the ISPRS/ITC-Midterm Symposium. Enschede, The Netherlands.

UN, (ed), 2004. Population Division of the Department of Economic and Social Affairs of the United Nations Secretariat, World Population Prospects: 2002 Revision and World Urbanization Prospects: The 2001 Revision. http://esa.un.org/unpp.

Van Delden, H., Luja, P., Engelen, G., 2007. Integration of multi-scale dynamic spatial models of socio-economic and physical processes for river basin management. Environmental Modelling & Software, 22, 223–238.

Van Jaarsveld, A.S., Biggs, R., Scholes, R.J., Bohensky, E., Reyers, B., Lynam, T., Musvoto, C., Fabricius, C., 2005. Measuring conditions and trends in ecosystem services at multiple scales: the Southern African Millennium Ecosystem Assessment (SAfMA) experience. Philosophical Transactions: Biological Sciences, 360, 425–441.

Van Rheenen, T., Elbel, C., Schwarze, S., Nuryartono, N., Zeller, M., Sanim, B., 2003. Enchroachments on Primary Forests: Are They Really Driven by Despair? In: Gerold, G., Fremerey, M., Guhardja, E. (eds). Land Use, Nature Conservation and the Stability of Rainforest Margins in Southeast Asia. pp 198–213. Springer, Berlin, New York.

Veldkamp, A., Fresco, L.O., 1996. CLUE-CR: an integrated multi-scale model to simulate land use change scenarios in Costa Rica. Ecological Modelling, 91, 231–248.

Veldkamp, A., Lambin, E.F., 2001. Predicting land-use change. Agriculture, Ecosystems & Environment, 85, 1–6.

Verburg, P.H., Koning, G.H.J.de, Kok, K., Veldkamp, A., Bouma, J., 1999. A spatial explicit allocation procedure for modelling the pattern of land use change based upon actual land use. Ecological Modelling, 116, 45–61.

Verburg, P.H., Soepboer, W., Veldkamp, A., Limpiada, R., Espaldon, V., Mastura, S.S.A., 2002. Modeling the Spatial Dynamics of Regional Land Use: The CLUE-S Model. Environmental Management, 30, 391–405.

Verburg, P.H., Schot, P.P., Dijst, M.J., Veldkamp, A., 2004. Land use change modelling: current practice and research priorities. GeoJournal, 61, 309–324.

Villa, F., Costanza, R., 2000. Design of multi-paradigm integrating modelling tools for ecological research. Environmental Modelling & Software, 15, 169–177.

Voinov, A., Costanza, R., Wainger, L.A., Boumans, R., Villa, F., Maxwell, T., Voinov, H., 1999. Patuxent landscape model: integrated ecological economic modeling of a watershed. Environmental Modelling & Software, 14, 473–491.

Wall, B.M., 1996. A Genetic Algorithm for Resource-Constrained Scheduling. PhD thesis. MIT. http://lancet.mit.edu/ga.

Wassenaar, T., Gerber, P., Verburg, P.H., Rosales, M., Ibrahim, M., Steinfeld, H., 2007. Projecting land use changes in the Neotropics: The geography of pasture expansion into forest. Global Environmental Change, 17, 86–104.

Watson, R.T.E., Noble, I.N., Bolin, I.N., Ravindranath, N.H., Verardo, D.J., Dokken, D.J., 2000. Land-Use, Land-Use Change and Forestry – A Special Report of the IPCC. Cambridge University Press. Cambridge.

Weber, R., Faust, H., Schippers, B., Mamar, S., Sutarto, E., 2007. Migration and Ethniticy as Cultural Impact Factors on Land Use Change in the Rainforest Margins of Central Sulawesi, Indonesia. In: Tscharntke, T., Leuschner, C., Guhardja, E., Zeller, M. (eds). The Stability of Tropical Rainforest Margins: Linking Ecological, Economic and Social Constraints of Land Use and Conservation. pp 392–417. Springer, Berlin, London, New York.

Weimar, J.R., 1997. Simulation with cellular automata. Logos Verlag. Berlin, Germany.

Weitz, A.M., Linder, E., Frolking, S., Crill, M., Keller, M., 2001. $N_2O$ emissions from humid tropical agricultural soils: effects of soil moisture, texture and nitrogen availability. Soil Biology & Biochemistry, 33, 1077–1093.

Wenderholm, E., 2005. Eclpss: a Java-based framework for parallel ecosystem simulation and modeling. Environmental Modelling & Software, 20, 1081–1100.

White, R., Engelen, G., 1997. Cellular automata as the basis of integrated dynamic regional modelling. Environment and Planning B – Planning and Design, 24, 235–246.

White, R., Engelen, G., 2000. High-resolution integrated modelling of the spatial dynamics of urban and regional systems. Computer, Environment and Urban Systems, 24, 383–400.

White, R., Engelen, G., Uljee, I., 1997. The use of Constrained Cellular Automata for high-resolution modelling of urban land-use dynamics. Environment and Planning B – Planning and Design, 24, 323–343.

Whitley, D., 2001. An overview of evolutionary algorithms: practical issues and common pitfalls. Information and Software Technology, 43, 817–831.

Winter, S., 2000. Location similarity of regions. ISPRS Journal of Photogrammetry & Remote Sensing, 55, 189–200.

Wolf, J., Bindraban, P.S., Luijten, J.C., Vleeshouwers, L.M., 2003. Exploratory study on the land area required for global food supply and the potential global production of bioenergy. Agricultural Systems, 76, 841–861.

Wolfram, S., 1984. Universality and Complexity in Cellular Automata. Physica D, 10, 1–35.

Wong, C.K., Wong, S.C., Tong, C.O., 1998. A new methodology for calibrating the Lowry model. Journal of Urban Planning and Development – ASCE, 124, 72–91.

Wong, S.C., Wong, C.K., Tong, C.O., 2001. A parallelized genetic algorithm for the calibration of the Lowry model. Parallel Computing, 27, 1523–1536.

Woodwell, G.M., Hobbie, J.E., Houghton, R.A., Melillo, J.M., Moore, B., Peterson, B.J., Shaver, G.R., 1983. Global deforestation: Contribution to atmospheric carbon dioxide. Science, 222, 1081–1086.

Wu, F., 1998. GIS-based simulation as an explanatory analysis for space-time processes. Journal of Geographical Systems, 1, 199–218.

Xian, G., Crane, M., Steinwand, D., 2005. Dynamic modeling of Tampa Bay urban development using parallel computing. Computers & Geosciences, 31, 920–928.

Zeller, M., Schwarze, S., Van Rheenen, T., 2002. Statistical sampling frame and methods used for the selection of villages and households in the scope of the research program on Stability of Rainforest Margins in Indonesia (STORMA). STORMA Discussion Paper Series No 1. Bogor, Indonesia: Universities of Göttingen and Kassel, Germany, Institut Pertanian Bogor and Universitas Tadulako, Indonesia. URL http://www.storma.de/DPS/pdf/SDP1.pdf, 31, 920–928.

# A System/application domain interface documentation

The SITE system/application domain interface is technically realized by extending the Python scripting language used to implement SITE application rule sets to also include the specific data structures used by SITE to represent the simulation environment. In the following a detailed reference on the language extensions established is provided. Three groups of objects can be discriminated: Object directly passed to the application domain by the application, objects that are managed by those passed objects and are made accessible, and object that can be instantiated independently in the application domain. To enhance readability of Python application rule sets, the Python classes and class methods do not necessarily have the same name as their C++ counterparts which in some cases might appear rather cryptic.

## Base objects passed to the SITE application domain

These are the instances of the classes *Grid* and *DynInfo* that are the arguments of the basic Python functions *Initialize()* and *SimulationStep()* which are called by the SITE system domain. Both instances provide access to a number of other objects, representing simulation data structures (*Grid* object as representative of static aspects in a simulation) and dynamic components (*DynInfo* object).

**Methods of class *Grid***

| Method | Arguments | Return value |
|---|---|---|
| GetSizeX() | | Number of grid columns |
| GetSizeY() | | Number of grid rows |
| SetGeoreferene() | Upper left x and y coordinate<br>Cell resolution | |
| GetCell() | x/y-coordinates | Cell object |
| GetAttr() | Attribute name | Attribute object |
| ApplyAttrChanges() | | |
| SetInitState() | | |
| CreateClusterLayer() | Layer name | Layer object |
| GetClusterLayer() | Layer name | Layer object |

The Python *Grid* class is a representation of the underlying simulation grid one the application side. It provides access to all necessary data structures that can be manipulated by a rule set developer. Beside the simulation grid itself such data structures are single cells (accessible by their grid coordinates), attribute descriptor objects (accessible by attribute name) and thematic layer objects (also accessible by their name). The grid itself only offers limited possibilities of manipulation. Application can specify their location using the *SetReoreference()* method. With the methods *ApplyAttrChanges()* and *SetInitState()* attribute values for all grid cells can be influenced directly. The first method forces all attribute values that are subject to change to actually change their values (the C++ counterpart of this method is called automatically after each simulation step), the latter one changes all attribute values to their initial value (i.e. their value at time of their creation) and thus can be regarded as a reset of the simulation grid.

**Methods of class *DynInfo***

| Method | Arguments | Return value |
|---|---|---|
| CreateExternModel() | Model name | Extern model object |
| SetScenario() | | |
| GetScenario() | | |
| SetTimeSeriesExportPath() | Path | |
| ExportTimeSeries() | Attribute name | |

The *DynInfo* objects basically serves as carrier for objects dealing with dynamic aspects of a simulation run. It provides access to objects representing external models (method *CreateExternModel()*), model drivers (scenarios) and simulation time-dependent output (time series).

## Objects accessible via base objects

Classes described in this paragraph provide the access to selected parts of the underlying modeling data structures. All instances of these classes are created by the SITE system domain and are accessed through the *Grid* and *DynInfo* objects passed to the application script entry functions. It is not possible to instantiate these classes on the application side.

**Methods of class *Cell***

| Method | Arguments | Return value |
|---|---|---|
| HasAttr() | Attribute name | TRUE or FALSE |
| SetAttr() | Attribute name | |
| | Attribute value | |
| | Immediate effect flag | |
| SetAttrNoData() | Attribute name | |
| | Immediate effect flag | |
| IsAttrNoData() | Attribute name | TRUE of FALSE |
| GetAttr() | Attribute name | Attribute value |
| X() | | Cell x coordinate |
| Y() | | Cell y coordinate |

*Cell* objects are representatives of single grid cells. Methods of the cell class entirely deal with the access and the altering of cell attribute values, where the location access methods *X()* and *Y()* are wrappers of the *GetAttr()* function accessing the cell coordinates. Cell attributes can be of value NODATA (represented by -9999 to be compliant with standard GIS software). This value can be checked and set by specific methods (*IsAttrNoData()* and *SetAttrNoData()*). All methods that set an attribute value require a so called immediate effect flag of boolean type as argument. By evaluating this flag, the method triggers whether a new attribute value is valid directly after its termination. In case of the immediate effect flag being of value FALSE, the new attribute value will be set after a call to the *Grid* method *ApplyAttrChanges()* or its C++ counterpart in the system domain.

All *Cell* objects are contained by the *Grid* object which provides a method to access them by specifying their coordinates. However, the usual way to access grid cell is to use one of the different iterator objects described below.

**Methods of class *Attr***

| Method | Arguments | Return value |
|---|---|---|
| SetColor() | Category number | |
| | 8bit value for red portion | |
| | 8bit value for green portion | |
| | 8bit value for blue portion | |
| | Category name | |
| SetRandomColor() | | |
| InitHistogram() | Value of lowest histogram class | |
| | Histogram class width | |

Class *Attr* provides meta data on cell attributes. Its C++ counterpart is the class *AttrDscr* (component *SmltnEnvironment*). For each attribute available, the respective instance can be retrieved yb specifying the attribute's name. The interface of this class deals with attribute statistics and special configurations for category attributes (especially land use classes). These latter configurations mainly serve visualization purposes. By default, attribute values are assumed to be continuous and visualized using a color gradient from green (lowest value) to red (highest value). For category attributes it is advisable though not technically required to assign colors and category names to the different attribute values that can occur. This is done using the member function *SetColor()*.

Histograms are created and kept up to date for each attribute as part of the attribute statistics. Without explicit configuration using method *InitHistogram()*, the system assumes that histograms have a minimum class of value 0 and a class width of 1. Using these two parameter as method arguments, histograms for selected attributes can be customized.

**Methods of class *Layer***

| Method | Arguments | Return value |
|---|---|---|
| ClusterByEqualAttrValue() | | Configurable cluster algorithm object |
| ClusterByClasses() | | Configurable cluster algorithm object |
| ClusterConditional() | | Configurable cluster algorithm object |
| DoClustering() | | |
| GetClstr() | Cluster ID | Cell cluster object |

Class *Layer*, instances of it accessible via the *Grid* object by a unique name, provides functionality to aggregate similar grid cells to cell clusters (objects of class *Clstr*). The definition of similarity depends on the application and needs to be done by the rule set developer. The SITE framework provides three different algorithms to define similarity and assign grid cell to cell clusters. Objects representing these clustering algorithms can be accessed through a *Layer* object with the methods *ClusterByEqualAttrValue()*, *ClusterByClasses()* and *ClusterConditional()*. These objects provide specific interfaces to configure the represented algorithms. Calling the *DoClustering()* method starts the actual clustering process based on the underlying clustering rules and is used for both initial clustering and cluster update.

Existing cell clusters can be accessed with method *GetClstr()* by specifying the unique ID assigned during the clustering process. However, cell clusters of a thematic layer are typically traversed using a specific iterator (see below).

**Methods of class *Clstr***

| Method | Arguments | Return value |
|---|---|---|
| GetSignature() | | Unique integer value assigned to the cell cluster |
| GetBestCell() | | Cell that is representative for the cell cluster |

The actual cell clusters are represented by class *Clstr*. The interface provides access to the unique cluster id and a grid cell that is representative for the cluster.

**Methods of class *ClstrAlgorithmEqualAttrVal***

| Method | Arguments | Return value |
|---|---|---|
| SetClusterAttr() | Attribute name | |

This cluster algorithm aggregates all grid cells that have the same value for the attribute specified by method *SetClusterAttr()*. To ensure that the algorithm works properly, only categorial attributes should be used. In the resulting cell clusters, the category value is used as the unique cluster ID.

**Methods of class *ClstrAlgorithmClassify***

| Method | Arguments | Return value |
|---|---|---|
| CreateClasses() | Attribute name | |
| | Minimum class | |
| | Maximum class | |
| | Class width | |
| | Flag: TRUE for categorial values, otherwise FALSE | |

This cluster algorithm performs the aggregation of grid cells based on classification rules specified by method *CreateClasses()*. An arbitrary number of attributes can be used. Therefore this method needs to be called repeatedly passing the respective attribute name and classification information. It is also possible to define classification schemes for one attribute that do not have equal class spacing. To do this, the *CreateClasses()* method can be called repeatedly using the same attribute name, but different values for class minimum, class maximum and class width.

**Methods of class *ClstrAlgorithmConditional***

| Method | Arguments | Return value |
|---|---|---|
| CreateClasses() | Node ID | |
| | Attribute name | |
| | Minimum class | |
| | Maximum class | |
| | Class width | |
| | Flag: TRUE for categorial values, otherwise FALSE | |
| AddConditionXXX() | Node ID | |
| | Attribute name | |
| | Comparison value | |
| AddConditionElse() | Node ID | |
| GoToNode() | NodeID | |

This classification algorithm is similar to the previous one (ClstrAlgorithmClassify) since it allows the specification of classification schemes. Additionally, it gives the possibility to define additional classification paths if a cells gets a specific classification. Therefore it allows to define condition´s using the family of *AddConditionXXX()* methods (conditions available are: equality, greater than, less than, inequality). Using the algorithm interface, the rule set developer creates a conditional tree structure consisting of classifications and subsequent conditional nodes. Each is assigned a unique node ID. The method GoToNode() is used to access a specific conditional node.

**Methods of class *Scno***

| Method | Arguments | Return value |
|---|---|---|
| QueryPart() | Time series name<br>Key 1<br>...<br>Key n | Scenario value |
| GetParameter() | Parameter name | Parameter value |

An object of class *Scno* represents a scenario used for a SITE simulation run. To combine a rule set to be executed with a scenario is obligatory. Using method *QueryPart()*, the rule set developer can access scenario time series, where the part name is the name of database table holding the respective time series data. To get the correct value, a unique key must be specified (e.g. composed of simulation step, district ID, etc). The method returns a value of type double, which is the scenario value stored in the underlying database table for the given key. Independent rule set parameters that are part of a scenario are accessed using the *GetParameter()* method by specifying their name.

**Methods of class *ExternModel***

| Method | Arguments | Return value |
|---|---|---|
| SetInputVariable() | Variable name | |
| SetOutputVariable() | Variable name | |
| SetInputVrblValues() | Cell object | |
| InitCalculation() | Job ID | |
| SetRefCellId() | ID of cell for which model is invoked | |
| SetInfo() | Information contained in string | |
| SetIntermediate() | Flag defining intermediate model behavior | |
| Configure() | | |
| Execute() | | |
| HasResult() | Job ID | TRUE or FALSE |
| GetResult() | Parameter name<br>Job ID | Result value |
| CleanUp() | | |

The *ExternModel* class exposes the functionality to establish, configure and run calculation of simulation data using a third-party sub model. Calculations are based on single cells. The interface is identical to the external model interface introduced in section 3.4.5. Since th SITE sub model interface is designed to integrate point models and process modeling job parallelized if more than one server instance is available, it is necessary to first configure all job before starting the actual processing using the *Execute()* method. *Execute()* returns after all jobs have been processed. At this point, all results are stored in the *ExternModel* object and can be accessed via the *GetResult()* method by specifying the respective job IDs.

## Iterators

The SITE Python language extensions also include a number of classes that can be instantiated directly inside an application rule set script. Such objects are basically helpers to access specific parts of the underlying simulation infrastructure like iterators or algorithms that could be as well implemented as part of the application script but are significantly more efficient that way. Those algorithms are referred to as update operations.

All iterator classes expose the same interface providing a method *Frst()* to set the iterator to the first element of the underlying set, a method *Next()* to step to the next (valid) element and a method *Cont()* which can be used to check whether all elements have been traversed or not. These methods can be included in the standard python *while* loop. The current element of the set being traversed can be accessed using the method *Crnt()*. However, instances of the exposed iterator classes cannot be used in Python *for* loops. In contrast to the C/C++ *for* statement, the Python counterpart requires a traversable collection of objects that fulfill the Python standard of iterators. Python has its own concept of iterators which uses a special exception thrown as soon as the traversal is complete. This concept is not compliant with iterators in C++. Therefore it is recommendable to wrap the exposed classes by iterator classes programmed in Python to match the Python requirements and thus enable the use of iterators in Python *for* loops.

**Methods of class *GridItr***

| Method | Arguments | Return value |
|---|---|---|
| Constructor | Grid object to traverse | |
| Frst() | | |
| Next() | | |
| Cont() | | TRUE or FALSE |
| Crnt() | | Cell object |

One basic procedure in application rule sets is to traverse all cells of the underlying simulation grid and perform operations on them. The *GridItr* (grid iterator) class provides the functionality to traverse the simulation grid. To create an iterator object, the *Grid* object as representative for the simulation grid needs to be passed to the constructor.

**Methods of class *CellNghbItr***

| Method | Arguments | Return value |
|---|---|---|
| Constructor | Center cell object<br>Grid object | |
| Frst() | | |
| Next() | | |
| Cont() | | TRUE or FALSE |
| Crnt() | | Cell object |

The *CellNghbItr* (cell neighbor iterator) class implements functionality to iterate over all direct neighbor cells of a specified center cell. Since Moore neighborhood is assumed, the collection to be traversed consists of at eight grid cells at most depending on the location of the center cell in the simulation grid.

**Methods of class *LayerItr***

| Method | Arguments | Return value |
|---|---|---|
| Constructor | Layer object | |
| Frst() | | |
| Next() | | |
| Cont() | | TRUE or FALSE |
| Crnt() | | Clstr object |

This iterator traverses all *Clstr* (cell aggregate) objects of a previously defined thematic layer. To create an object of this class, the Layer object housing the requested cell clusters has to be passed to the constructor.

**Methods of class *ClstrItr***

| Method | Arguments | Return value |
|---|---|---|
| Constructor | Clstr object | |
| Frst() | | |
| Next() | | |
| Cont() | | TRUE or FALSE |
| Crnt() | | Cell object |

Cell clusters as collections of similar cells are traversed using the iterator class *ClstrItr*. The cluster object of interest is the argument passed to the constructor upon iterator object creation.

## Update operations

Update operations are operations applied to all cell of the simulation grid of a simulation. Although they could as well be implemented on the application side using the Python language, it is favorable to create respective language extensions due to performance reasons. This way, high performance C++ code is invoked from Python. The SITE system provides two such operations.

**Methods of class *UpdtOprtnNghbCount***

| Method | Arguments | Return value |
|---|---|---|
| Constructor | Target attribute name<br>Source attribute name | |

*(continued on next page)*

**Methods of class *UpdtOprtnNghbCount*** – continued

| Method | Arguments | Return value |
|---|---|---|
| | Condition value | |
| SetTargetAttr() | Attribute name | |
| SetSourceAttr() | Source attribute name | |
| SetConditionVal() | Condition value | |
| Execute() | Grid object | |

This update operation calculates the number of neighbor objects fulfilling a specific criterion for each object on the simulation grid. The criterion is specified by defining a target attribute and an appropriate attribute value. The number of neighbors that meet this criterion is stored in the source attribute. An example would to calculate the number of neighbor cells of land use class water for each cell. Configuration can either be done by passing these three arguments to the constructor or by calling the single configuration methods provided by the class interface. The actual calculation is started with the *Execute()* method.

**Methods of class *UpdtOprtnDistCalc***

| Method | Arguments | Return value |
|---|---|---|
| Constructor | Target attribute name | |
| | Reference attribute name | |
| | Source attribute name | |
| | Condition value | |
| SetTargetAttr() | Attribute name | |
| SetSourceAttr() | Source attribute name | |
| SetReferenceAttr() | Reference attribute name | |
| SetConditionVal() | Condition value | |
| Execute() | Grid object | |

Using this update operation it is possible to calculate for each grid cell the distance to the closest grid cell fulfilling a specific criterion (e.g. the distance to the closest cell of land use class water). I addition the algorithm sets the cell ID of the closest cell as second attribute value. In the nomenclature of this update operation, the calculated distance is referred to as the target attribute while the ID of the closest cell is the reference attribute. The criterion after which it is decided to which cells the distance is calculated is defined by specifying a source attribute together with a condition value. The underlying algorithm is of complexity $O(n)$ and thus also works efficiently on large simulation grids.