

**A Framework for the
Semantic Representation of Energy Policies
related to Electricity Generation**



Aidid Bin Chee Tahir
New College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2011

CONTENTS

Abstract	i
Acknowledgements	iii
List of Abbreviations	iv
List of Notations and Units	v
List of Figures	xi
List of Tables	xiv
Table of Contents	xvi

Abstract

Energy models are optimisation tools which aid in the formulation of energy policies. Built on mathematics, the strength of these models lie in their ability to process numerical data which in turn allows for the generation of an electricity generation mix that incorporates economic and the environmental aspects.

Nevertheless, a comprehensive formulation of an electricity generation mix should include aspects associated with politics and society, an evaluation of which requires the consideration of non-numerical qualitative information. Unfortunately, the use of energy models for optimisation coupled with the evaluation of information other than numerical data is a complicated task.

Two prerequisites must be fulfilled for energy models to consider political and societal aspects. First, the information associated with politics and society in the context of energy policies must be identified and defined. Second, a software tool which automatically converts both quantitative and qualitative data into mathematical expressions for optimisation is required.

We propose a software framework which uses a semantic representation based on ontologies. Our semantic representation contains both qualitative and quantitative data. The semantic representation is integrated into an *Optimisation Modelling System* which outputs a model consisting of a set of mathematical expressions. The system uses ontologies, engineering models, logic inference and linear programming.

To demonstrate our framework, a *Prototype Energy Modelling System* which accepts energy policy goals and targets as inputs and outputs an optimised electricity generation mix has been developed. To validate the capabilities of our prototype, a case study has been conducted. This thesis discusses the framework, prototype and case study.

this page intentionally left blank

Acknowledgements

Funding for this research has been provided by Petroliaam Nasional Berhad (PETRONAS), the national oil company of Malaysia.

This research has been supervised by Dr. René Bañares-Alcántara, Fellow of New College, Oxford and Reader in Engineering Science, University of Oxford.

I wish to thank my parents, Dr. Sharifah Barlian Aidid and Dr. Chee Tat Siang @ Chee Tahir Abdullah, and my brother Ir. Dr. Aidil Chee Tahir.

Aidid Chee Tahir

December 2011

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
BAU	Business As Usual
CO ₂	Carbon Dioxide
CGE	Computable General Equilibrium
DAML	DARPA Markup Language
DARPA	Defence Advanced Research Projects Agency
GBP	Great Britain Pound
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IDE	Integrated Development Editor
LHS	Left Hand Side
LP	Linear Programming
MCDA	Multicriteria Decision Analysis
NO _x	Nitrogen Oxide
OWL	Web Ontology Language
PESTLE	Political, Economic, Social, Technological, Legal, Environmental
RDF	Resource Description Framework
RHS	Right Hand Side
RM	Ringgit Malaysia
RPS	Renewable Portfolio Standard
SO _x	Sulphur Oxide
SWOT	Strength, Weakness, Opportunity, Threats
USD	US Dollar
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

List of Notations and Units

Symbol	Notation	Units
c	capital cost per MW of generation capacity	GBP£ (MW) ⁻¹
i	electricity generation technology (e.g. coal, wind, solar)	
teG	total electricity generation capacity	MW
om	operation cost per MW of production	GBP£ (MW) ⁻¹
j	fossil fuel electricity generation technology (e.g. coal, gas, oil)	
teP	total electricity production	MW
$lcep$	life cycle energy payback per MW of electricity production	(MW) ⁻¹
CP	CO ₂ emission per MW of electricity production	tons (MW) ⁻¹
CpC	CO ₂ emissions per capita allowance	tons (capita) ⁻¹
pop	population size	capita
$teDp$	total electricity demand per capita	MW (capita) ⁻¹
$teGG$	total electricity generation capacity	MW
$tePP$	total electricity production	MW
$nMod$	night time electricity demand modifier	
k	heat and power generation technology	
$teHPP$	total heat production	MW
$hpMod$	heat demand modifier	MW
m	reserve electricity generation technology (i.e. gas, coal, oil, nuclear)	
$teRPP$	total reserve electricity generation demand	

Symbol	Notation	Units
$resMod$	reserve electricity modifier	
p	renewable electricity generation technology (i.e. wind, solar)	
$teREPP$	total intermittent electricity production	MW
$rewMod$	renewable electricity generation modifier	
$\dot{Q}_{combustion}$	combustion heat production rate	kJ.s^{-1}
ic	components (e.g. air, oxygen, water, carbon dioxide)	
$\dot{M}$	mass flow rate of components	kg.s^{-1}
h	enthalpy at specified temperature and pressure	kJ.kg^{-1}
$\dot{m}_{steamout}$	steam production rate	kg.s^{-1}
$h_{steamout}$	enthalpy of produced steam	kJ.kg^{-1}
$h_{boilerwaterin}$	enthalpy of boiler feed water	kJ.kg^{-1}
$\dot{Q}_{wasteheat}$	waste heat production rate	kJ.s^{-1}
$\dot{W}_{ST,out}$	net work out rate from steam turbine (power)	kJ.s^{-1}
$h_{ST,out}$	enthalpy of steam at steam turbine exit	kJ.kg^{-1}
$S_{ST,out}$	entropy of steam at steam turbine exit	$\text{kJ.}(\text{kg.K})^{-1}$
$x_{ST,out}$	vapour fraction at steam turbine exit	
$S_{ST,liq}$	entropy of liquid phase at steam turbine exit	$\text{kJ.}(\text{kg.K})^{-1}$
$S_{ST,vap}$	entropy of vapour phase at steam turbine exit	$\text{kJ.}(\text{kg.K})^{-1}$
$h_{ST,liq}$	enthalpy of liquid phase at steam turbine exit	kJ.kg^{-1}
$h_{ST,vap}$	enthalpy of vapour phase at steam turbine exit	kJ.kg^{-1}
$\dot{W}_{COOL,in}$	net work out by steam condenser	kJ.s^{-1}

Symbol	Notation	Units
$\dot{m}_{\text{boilerwaterin}}$	boiler feed water flow rate at steam condenser outlet	kg.s^{-1}
$h_{\text{COOL},\text{in}}$	enthalpy at steam condenser inlet	kJ.kg^{-1}
$h_{\text{COOL},\text{out}}$	enthalpy at steam condenser outlet	kJ.kg^{-1}
$\dot{W}_{\text{PUMP},\text{in}}$	net work in rate by boiler feedwater pump	kJ.s^{-1}
$h_{\text{PUMP},\text{out}}$	enthalpy of boiler feed water at pump discharge	kJ.kg^{-1}
$h_{\text{PUMP},\text{in}}$	enthalpy of boiler feed water at pump suction	kJ.kg^{-1}
$\dot{Q}_{\text{HEX},\text{out}}$	net heat production rate from heat exchanger	kJ.s^{-1}
$h_{\text{HEX},\text{out}}$	enthalpy of steam at heat exchanger exit	kJ.kg^{-1}
$S_{\text{HEX},\text{out}}$	entropy of steam at heat exchanger exit	$\text{kJ} \cdot (\text{kg.K})^{-1}$
$X_{\text{HEX},\text{out}}$	vapour fraction at heat exchanger	
$S_{\text{HEX},\text{liq}}$	entropy of liquid phase at heat exchanger exit	$\text{kJ} \cdot (\text{kg.K})^{-1}$
$S_{\text{HEX},\text{vap}}$	entropy of vapour phase at heat exchanger exit	$\text{kJ} \cdot (\text{kg.K})^{-1}$
$h_{\text{HEX},\text{liq}}$	enthalpy of liquid phase at heat exchanger exit	kJ.kg^{-1}
$h_{\text{HEX},\text{vap}}$	enthalpy of vapour phase at heat exchanger exit	kJ.kg^{-1}
$\dot{m}_{\text{BWTot}}$	total boiler feed water flowrate	kg.s^{-1}
$\dot{m}_{\text{BWloss}}$	boiler feed water loss rate	kg.s^{-1}
$\dot{m}_{\text{coolingwaterin}}$	cooling water in flowrate	kg.s^{-1}
$\dot{m}_{\text{CWTot}}$	total cooling water in flowrate	kg.s^{-1}
$\dot{m}_{\text{CWloss}}$	cooling water loss rate	kg.s^{-1}
$\dot{m}_{\text{demintot}}$	total demineraliser water flowrate	kg.s^{-1}
$\dot{m}_{\text{deminloss}}$	demineraliser water loss rate	kg.s^{-1}

Symbol	Notation	Units
$\dot{m}_{ROtotal}$	total reverse osmosis flowrate	kg.s ⁻¹
$\dot{m}_{ROloss}$	reverse osmosis loss rate	kg.s ⁻¹
$\dot{m}_{filtertotal}$	total filtered water flow rate	kg.s ⁻¹
$\dot{m}_{filterloss}$	filtered water loss rate	kg.s ⁻¹
$\dot{m}_{withdrawal}$	total water withdrawal rate	kg.s ⁻¹
$\dot{m}_{BWTopup}$	total boiler feed water top up rate (steam system top up rate)	kg.s ⁻¹
$\dot{m}_{CWTopup}$	total cooling water top up rate (cooling system top up rate)	kg.s ⁻¹
$\dot{m}_{consumption}$	total water consumption rate	kg.s ⁻¹
$P_{electrical,out}$	electricity generated	W, kWh.day ⁻¹
η	generator efficiency	
ΔH_c^o	enthalpy of combustion	kJ.mol ⁻¹
$\dot{m}_{fuel}$	fuel flow rate	kg.s ⁻¹
$\dot{m}_{air}$	air flow rate	kg.s ⁻¹
$MM_{fuel,c}$	molar mass of fuel required to produce enthalpy of combustion	kg.mol ⁻¹
$h_{c,in}$	enthalpy of air at combustor inlet	kJ.kg ⁻¹
$h_{c,out}$	enthalpy of flue gas at combustor outlet	kJ.kg ⁻¹
N	molar mass	kg.mol ⁻¹
$\dot{W}_{PT,out}$	net work out rate from power turbine	Kw
$h_{C,out}$	enthalpy of flue gas at combustor outlet	kJ.kg ⁻¹
$h_{PT,out}$	enthalpy of flue gas at power turbine outlet	kJ.kg ⁻¹
$h_{C,in}$	enthalpy of air at combustor inlet	kJ.kg ⁻¹

Symbol	Notation	Units
h_{atm}	enthalpy of air at compressor inlet	kJ.kg^{-1}
$\dot{W}_{WDT,out}$	net work out rate from wind turbine	J.s^{-1}
β	Betz constant	
ρ	ambient air density	kg.m^{-3}
V	wind velocity	ms^{-1}
r	wind turbine blade rotor radius	m
η_{pv}	photovoltaic generator efficiency	
τ	solar global radiation	$\text{kJ.m}^{-2}.\text{day}^{-1}$
A_{pv}	photovoltaic surface area	m^2
$\dot{Q}_{col}$	thermal heat production rate	kWh.day^{-1}
A_{col}	collector surface area	m^2
η_{col}	collector efficiency	
a_1	1st order heat loss coefficient	K^{-1}
a_2	2nd order heat loss coefficient	K^{-2}
T_m	collector mean temperature	K
T_a	collector ambient temperature	K
$\dot{W}_{WT,out}$	net work out rate from water turbine	J.s^{-1}
ρ_{wt}	water density	kg.m^{-3}
g	gravity	ms^{-2}
h	dam water tunnel height	m
Q_{water}	water flow rate	m^3s^{-1}

Symbol	Notation	Units
U_{bio}	biomass energy yield	$\text{kJ}.\text{year}^{-1}$
A_{bio}	biomass leaves or plant surface area	m^2
σ	biomass energy requirement factor	
α	process efficiency factor	
ω	absorption efficiency factor	
β	solar radiation factor	
γ	mass production efficiency	
$Yield_{plant}$	biomass energy yield	$\frac{\text{kJ.}}{(\text{hectare}.\text{year})^{-1}}$
P_{bio}	plantation area	hectare
EFB_{cv}	empty fruit bunches calorific value	$\text{kJ}.\text{kg}^{-1}$

Index of Figures

Figure 1.1.	Process flow of an optimisation model (energy model) for policy evaluation and analysis.....	5
Figure 1.2.	Relationship between the Optimisation Modelling System and Optimisation Model in this thesis.....	8
Figure 1.3.	System architecture of the Optimisation Modelling System.....	9
Figure 2.1.	Classification of languages [40] inclusive of OWL.....	20
Figure 2.2.	The stack of ontology markup languages [38].....	22
Figure 4.1.	Process flow of an optimised model for policy evaluation and analysis.	43
Figure 4.2.	Data codification paths proposed in this framework.....	59
Figure 5.1.	System architecture of the Optimisation Modelling System.....	71
Figure 5.2.	Relations between the Ontologies and Engineering Models in the Semantic Representation Module.....	74
Figure 5.3.	Relations between the top level classes in the ontology.....	76
Figure 5.4.	Capital cost and CO ₂ emission positions of selected electricity generation technologies.....	83
Figure 5.5.	Protégé screen shot for the formulation of the CO ₂ emissions constraint (Expression 4.2.1) in the Energy Model Construction Ontology of the Prototype Energy Modelling System.....	87
Figure 5.6.	Process flow chart for inference and construction of the objective function and constraints.....	92
Figure 5.7.	Process flow chart for the inference and construction of the Technology Preference Expression.....	93
Figure 5.8.	Linear programming module process flow.....	98
Figure 6.1.	System architecture of the Prototype Energy Modelling System.....	109
Figure 6.2.	User input window for energy policy goals and targets.....	111
Figure 6.3.	User input window for additional optimisation options.....	112
Figure 6.4.	Simplified information flow of the Prototype Energy Modelling System.....	113

Figure 6.5.	Relations between the ontologies and the engineering models in the Semantic Representation Module for the Prototype Energy Modelling System.....	114
Figure 6.6.	Taxonomy of energy_Policy.....	115
Figure 6.7.	Taxonomy of natural_Laws.....	116
Figure 6.8.	Taxonomy of natural_Resources.....	117
Figure 6.9.	Taxonomy of manmade_Concepts.....	119
Figure 6.10.	Taxonomy of manmade_Objects.....	121
Figure 6.11.	Protégé screen shot of the taxonomy of the Energy Scenarios Ontology.	133
Figure 6.12.	Taxonomy of the Energy Model Construction Ontology.....	136
Figure 6.13.	GUI for electricity generation plant.....	139
Figure 6.14.	Capital cost objective function.....	160
Figure 6.15.	Maximum allowable CO2 emissions supply constraint.....	161
Figure 6.16.	Minimum electricity generation demand constraint.....	162
Figure 6.17.	Linear programming optimised generation mixes in TWh for Scenario 1.....	164
Figure 6.18.	Linear programming optimised generation mixes in TWh for Scenario 2.....	165
Figure 6.19.	Optimised generation mixes in TWh for Scenario 1.....	166
Figure 6.20.	Optimised generation mixes in TWh for Scenario 2.....	167
Figure 6.21.	Two objective (multiobjective) optimised generation mixes in TWh for Scenario 2.....	168
Figure 6.22.	Optimised generation mixes in TWh for Scenario 2 using Random Probability formulation method for Technology Preference Equation.	169
Figure 6.23.	Optimised generation mixes in TWh for Scenario 2 using Random Probability formulation method for Technology Preference Equation.	169
Figure 7.1:	Proposed final electricity generation mix in MW for Scenario 1 BAU	179

Figure 7.2:	Proposed final electricity generation mix in MW for Scenario 2 EcoFriend.....	180
Figure 7.3:	Proposed final electricity generation mix in MW for Scenario 3 Nuclear.....	181
Figure 7.4.	Annualised capital cost for Scenarios 1, 2 and 3.....	185
Figure 7.5.	Annualised total cost for Scenarios 1, 2 and 3.....	186

Index of Tables

Table 4.1.	Summary of the background and requirements of the Framework.....	45
Table 4.2.	Examples for the policy terms used in this thesis.....	46
Table 4.3.	Summary of the ontology construction methodology used in this framework.....	54
Table 4.4.	Examples of the ontology terms used in this thesis.....	60
Table 5.1.	Top-level classes and the concepts represented in the ontology.....	75
Table 5.2.	Relations between the top-level classes in the ontology.....	76
Table 5.3.	Example of assessment from the comparison of generation capacity for electricity generation plants.....	77
Table 5.4.	Capital cost clustering results for electricity generation plants.....	79
Table 5.5.	Example of an axiomatic inference for a policy goal and target.....	81
Table 5.6.	Summary of results from the application of step one and step two of the six step process on Expression 5.2.1.....	86
Table 6.1.	Selected example of the relationship between manmade_Objects and manmade_Concepts and the Value Partition Ontology.....	123
Table 6.2.	Selected example of the relationship between manmade_Objects and data, value or literal.....	124
Table 6.3.	Selected example of the relationship between plant_Generation and equipment.....	125
Table 6.4.	Relationship between other_Objects and natural_Laws and natural_Resources.....	126
Table 6.5.	Value ranges for low, medium and high.....	129
Table 6.6.	Classification of electricity generation sources.....	130
Table 6.7.	Classification of electricity generation sources with respect to public health and energy supply security risk.....	132
Table 6.8.	Relationship between the ontology and the engineering model for Coal Power Plant.....	142
Table 6.9.	Relationship between the ontology and the engineering model for Gas Power Plant.....	143

Table 6.10.	Relationship between the ontology and the engineering model for Oil Power Plant.....	144
Table 6.11.	Relationship between the ontology and the engineering model for Wind Power Plant.....	145
Table 6.12.	Relationship between the ontology and the engineering model for Hydroelectric Power Plant.....	145
Table 6.13.	Relationship between the ontology and the engineering model for Solar Power Plant.....	146
Table 6.14.	Relationship between the ontology and the engineering model for Biomass Power Plant.....	147
Table 6.15.	Relationship between the ontology and the engineering model for Nuclear Power Plant.....	148
Table 6.16.	Variables and constants used in Table 6.18 and Figures 6.14, 6.15 and 6.16.....	156
Table 6.17.	List of policy goals and the selected options for Scenarios 1 and 2...	157
Table 6.18.	Technology Preference Expressions for Scenarios 1 and 2.....	158
Table 7.1.	Geographical parameters, values, and applicable electricity generation plants for Malaysia.....	173
Table 7.2.	Socio-economic parameters for Malaysia at start of case study in Year 2005.....	174
Table 7.3.	Historical and forecast population growth rates for Malaysia.	175
Table 7.4.	Historical and forecast electricity growth rates for Malaysia.....	175
Table 7.5.	Energy policy goals and the selected options for Scenarios 1, 2 and 3.	178
Table 7.6.	Comparison of selected parameters between initial condition in 2005, Scenario 1 BAU in 2035, Scenario 2 EcoFriend in 2035 and Scenario 3 Nuclear in 2035.....	183

Table of Contents

Chapter 1	Introduction.....	1
1.1	Background.....	2
1.2	Motivation.....	6
1.3	Objectives of Research.....	7
1.4	Contributions of the Research.....	10
1.5	Structure of the Thesis.....	11
Chapter 2	Literature review of Semantic Representation and Automatic Programming.....	12
2.1	Knowledge Representation and Engineering.....	12
2.1.1	Historical Background of Knowledge Representation	12
2.1.2	Ontologies and Knowledge Representation	13
2.1.3	Logic and Knowledge Representation	15
2.1.4	Application of Ontology-Based Knowledge Representation Systems.....	16
2.2	Ontologies as a Knowledge Representation Formalism.....	18
2.2.1	Introduction to Ontologies.....	18
2.2.2	The Ontology Taxonomy.....	19
2.2.3	Ontology Development.....	20
2.2.4	Ontology Construction.....	24
2.3	Automatic Programming.....	26
2.3.1	Current Approaches for Automatic Programming.....	27
2.3.2	Current Methods for Automatic Programming.....	28
2.3.3	Knowledge Representation and its Application in Automatic Programming.....	29
2.4	Conclusion.....	30

Chapter 3	Literature Review of Energy Models.....	32
3.1	Introduction to Energy Models	32
3.2	Characterisations of Energy Models.....	33
3.2.1	Purpose of Energy Models.....	33
3.2.2	Analytical Approach of Energy Models.....	35
3.2.3	Underlying Methodology	36
3.2.4	Mathematical Approach.....	38
3.3	Summary of Energy Models.....	39
3.4	Conclusions.....	40
Chapter 4	Development of the framework.....	42
4.1	Introduction to the Framework	42
4.2	Background of the Framework	44
4.3	Requirements, Technologies and Techniques of the Framework	47
4.3.1	Requirements of the Framework.....	48
4.3.2	Required Technologies and Techniques to Simulate the Tasks of the Modeller.....	48
4.4	Design Methods and Guidelines of the Framework.....	50
4.4.1	Guidelines for the Appropriate Identification of Data and Selection of Information for Codification.....	51
4.4.2	Methodology for Ontology Construction.....	53
4.4.3	Integration of Both Qualitative and Quantitative Data for Evaluation Within a Numerical Environment.....	56
4.4.4	Underlying Mathematical Approach.....	62
4.5	Summary.....	66
Chapter 5	The Framework.....	67
5.1	Specifications of the Framework.....	68
5.2	Details of the Framework.....	69

5.2.1	Semantic Representation Module.....	72
5.2.1.1	Knowledge Domain Ontology.....	75
5.2.1.2	Value Partition Ontology.....	77
5.2.1.3	Scenarios Ontology.....	80
5.2.1.4	Equation Construction Ontology.....	84
5.2.1.5	Engineering Models.....	88
5.2.2	Knowledge Inference and Expression Builder Module.....	90
5.2.3	Linear Programming Module.....	97
5.2.4	Support Module.....	101
5.2.5	Graphical User Interface Module.....	103
5.3	Steps of the Framework.....	103
5.4	Conclusion.....	104
Chapter 6	Prototype Energy Modelling System.....	105
6.1	Overview of the Prototype Energy Modelling System.....	105
6.2	Semantic Representation of the Energy Policy Knowledge Domain Using Ontologies and Engineering Models.....	114
6.2.1	Energy Policy Ontology.....	115
6.2.2	Value Partition Ontology.....	127
6.2.3	Energy Scenarios Ontology.....	132
6.2.4	Energy Model Construction Ontology.....	135
6.2.5	Engineering Models.....	137
6.3	Application of the Semantic Representation of the Energy Policy Knowledge Domain.....	149
6.3.1	Mathematical Expressions for the Sets of Linear Programming and System of Inequalities.....	150
6.3.2	Formulation of the Mathematical Expressions.....	156
6.3.3	Optimisation of Electricity Generation Mix Based on Logic Inference of a Set of Energy Policy Goals and Targets.....	163
6.4	Conclusions.....	170

Chapter 7	Case study.....	171
7.1	Introduction to the Malaysian Energy Policy.....	171
7.2	Data Configuration of the Prototype Energy Modelling System	173
7.3	Scenarios.....	176
7.4	Results.....	179
7.5	Conclusions.....	188
Chapter 8	Conclusions and Future Work.....	190
8.1	Conclusions.....	190
8.1.1	Advantages and Capabilities.....	191
8.1.2	Disadvantages and Limitations.....	193
8.2	Future Work.....	195
8.3	Contributions.....	197
8.4	Conference and Journal Papers.....	198
8.5	Final Remark.....	198
Appendix A	Rankine Steam Cycle.....	199
Appendix B	Brayton Cycle.....	207
Appendix C	Wind Turbine Model.....	210
Appendix D	Solar Photovoltaic and Concentrator Model.....	211
Appendix E	Hydroelectric Model.....	212
Appendix F	Biomass Production Model.....	213
References		214

this page intentionally left blank

Chapter 1 Introduction

This thesis introduces a new modelling system which supports policy makers by automatically converting their policy goals and targets into an optimisation model consisting of a set of equivalent mathematical expressions. The main contributions of this thesis are the development of a methodology (framework) which incorporates techniques from artificial intelligence (AI) into an optimisation system and the creation of a *Prototype Energy Modelling System*. This *Prototype Energy Modelling System* incorporates a semantic representation of the energy policy knowledge domain and utilises knowledge inference to extract information from this semantic representation. The integration of these two technologies (semantic representation and knowledge inference) enables the *Prototype Energy Modelling System* to offer two unique features. First, the *Prototype Energy Modelling System* is able to accept and evaluate both qualitative and quantitative inputs and data. Second, the *Prototype Energy Modelling System* is able to create an energy model for the optimisation of an electricity generation mix through the automatic formulation of the mathematical expressions necessary for numerical optimisation.

This chapter introduces this research project. The background, motivation, objectives and expected contributions of the project are discussed in brief. Discussion of the research background, Section 1.1, begins with the definition of modelling and models, followed by a summary of what mathematical and optimisation models are, and ends with a discussion of our perceived limitations of current optimisation models used for energy policy analysis. Section 1.2 then explains the motivation behind this research which is to address the extreme dependence of such models on the skill of the modeller. Section 1.3 defines the objectives of the project and states the hypothesis of this research which is to simulate the role of the modeller using techniques from

AI, in particular an ontology-based semantic representation. Section 1.4 closes with a summary of the expected contributions of the project which are a development framework and a *Prototype Energy Modelling System*. Section 1.5 concludes with the structure of the thesis.

1.1 Background

The word “modelling”, which originates from the Latin word *modellus*, is the activity of devising or using abstract or mathematical models [1]. Anthropologists have hypothesised that the ability to build abstract models is what gave *Homo sapiens* the competitive edge over their less developed rivals the *Homo neanderthalensis*. In the preceding years, modelling has been featured in fields as diverse as geometry and astronomy with mathematics used to describe the shapes of objects in the world and the movements of the celestial bodies.

In simplistic terms, a model is a simplified version of something that is real [1]. The level of formality, explicitness, detail and richness, and relevance of a model is dependent on the modelling objective. In building models, real world objects are replaced with simpler objects. Our knowledge of the real world is structured by the model which reduces everything to just those phenomena and aspects which are considered important. Hence, a model only describes a part of the real world phenomenon and is therefore restricted in its scope of application. Nevertheless, in spite of this simplification, models can be used to explain phenomena, make predictions and assist in decision making. In particular, mathematical based models can be used for prediction, forecast and analysis of situations even when the complete sets of laws which govern the world under analysis are not fully known [2].

Mathematical models are a class of models where real world objects are represented by mathematical objects using a formalised mathematical language. Such

models allow for precise analysis through the use of mathematical theory and algorithms. The explosive growth of computational power has resulted in the increased popularity of mathematical modelling which are often used for a special class of problems called optimisation.

Optimisation models are computer programs which determine an optimal distribution of a designated parameter through the use of mathematical methods. The strength of optimisation models lie in their use of mathematics which are grounded on principles and formulas established in academic areas such as engineering and economics. An optimisation model tries to minimise or maximise a quantity associated with a decision process (e.g. cost) by exploiting the available degrees of freedom under a set of restrictions or constraints [1]. In mathematical terms, this best value is either a global or at minimum local value. Mathematical optimisation methods guarantee an optimal solution that satisfies all constraints.

Optimisation models are used in a variety of fields from engineering to business by a variety of organisations from governments to private companies for the optimisation of a variety of resources or activities from energy utilisation to logistical arrangements. Examples of optimisation models include *energy models* for energy policy analysis and *consumption capital asset pricing models* for macroeconomics analysis. Indeed, the widespread use of optimisation models is a testament to their utility in the world today.

While the power of optimisation models are undisputed, such models do have limitations. The outputs from an optimisation model are not only based on the mathematics behind the model but also on both the data used as the basis for calculation and the skill of the modeller. Such models almost exclusively require data that are numerical and in a format compatible with both the computer system and the mathematics used. The role of the human modeller in optimisation models is critical

for successful optimisation as it is the modeller who determines the appropriate data and configuration of the model. The modeller exercises a greater influence on the optimisation results than is generally acknowledged. While this is not an unanticipated requirement of optimisation models, nevertheless, it is surprising that limited efforts have been undertaken to minimise this dependency. This is, in our opinion, an especially pressing issue for optimisation models used for policy evaluation and analysis.

The role of the modeller within the process flow of policy analysis can best be illustrated with an example for optimisation based energy models such as MARKAL, MESSAGE, and DNE-21. Energy models are tools which aid in the formulation of energy policies, the details of which are discussed in Chapter 5. Figure 1.1 illustrates the role of both the model and the modeller within the scheme of energy policy evaluation. The process begins with policy makers who formulate a set of energy policy goals, instruments and targets. A group of modellers subsequently translates these policies into inputs compatible with an energy model. The modellers use their best judgement to ensure that the translated inputs are appropriate and reflective of the policies formulated by the policy maker. Similarly, outputs from the model, usually in the form of numbers and graphs, require translation and interpretation by both the modellers and policy makers to validate the optimisation results. Subsequent to the review, a report may be then authored based on these model outputs.

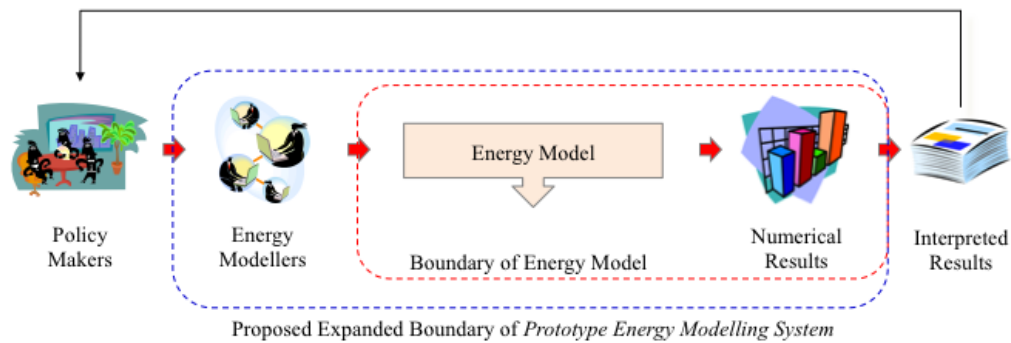


Figure 1.1. Process flow of an optimisation model (energy model) for policy evaluation and analysis.

While this process flow is a simplistic representation (e.g. there is not always a distinction between policy maker and modeller), nevertheless what can be accepted is the fact that the modellers have a distinct and indispensable role in the process. This requirement for modellers disconnects the policy makers from the energy model itself. This disconnect can reduce the efficiency of the overall process flow. Additional time is required for the modellers to comprehend the inputs from the policy makers. Even more importantly, communication problems could result in the misinterpretation of the inputs from the policy makers by the modellers which would require a restart of the modelling process. It would therefore be advantageous if the role of the modellers could be incorporated into the modelling system.

1.2 Motivation

It is the expressed interest of this research to bridge the gap between policy makers and the optimisation model (energy model) itself. The aim of this research is to expand the boundaries of optimisation models used for policy evaluation and analysis with particular emphasis given to that of energy models. In Figure 1.1, the boundary of a current energy model is marked in red while the proposed expanded boundary of the *Prototype Energy Modelling System* (which is explained further in Section 1.3) is marked in blue.

To expand the boundaries of optimisation models, a greater part of the function and associated tasks of the modeller may be semi-automated. Such an expansion in capability will improve the accessibility of optimisation models to the policy maker whose main interest lies with the effects caused by policies rather than the intricacies of the data and the model itself.

To achieve this shift, it is necessary to place an emphasis on policy goals as the inputs to the optimisation model and the effects of such policy goals on the outputs from the optimisation model. These policy goals can be either qualitative or quantitative. In general, politics and social policy goals are qualitative goals while economics and environmental goals are quantitative goals. The optimisation model must be able to accommodate both qualitative and quantitative inputs and data with minimal human intervention.

Indeed, to simulate the role of the modeller, the system must offer three capabilities. First, it must codify the knowledge relevant to the domain where the optimised model is used (e.g. energy policy knowledge domain for the optimisation of electricity generation resources). Second, the system must be able to evaluate both the qualitative and quantitative data contained within the knowledge base in order to execute a correct translation of the policy maker's intentions which are expressed as

policy goals. Third, the system must formulate, as inputs to the optimisation phase of the model, a set of mathematical expressions that is representative of these policy goals.

The capabilities outlined here represent a significant departure from that of current optimisation based energy models which exclusively require numerical inputs and are dependent on the modeller to provide the necessary energy model configuration. Where non numerical inputs are present, the energy model relies on the skills and knowledge of the modeller to translate such inputs into their numerical equivalence. The vision of the policy maker, as defined by the energy policy goals, must be converted by the modeller into mathematical equivalents fit for input into the energy model.

1.3 Objectives of Research

It is the hypothesis of this research that it is possible to develop a modelling system which supports policy makers by automatically converting their policy goals and targets into an optimisation model consisting of a set of equivalent mathematical expressions. In this thesis, this software tool is referred to as an *Optimisation Modelling System*. In the context of energy models, this requires the creation of a *Prototype Energy Modelling System* which can encompass some of the function and associated tasks of the modeller.

The extension proposed simulates some of the thought processes behind some of the functions of the modeller through the use of AI techniques. The thought process that can be simulated is that of information consideration while the task that can be automated is the configuration of the energy model. Computationally speaking, the *Prototype Energy Modelling System* must be able to accept and evaluate both qualitative and quantitative input and data. Subsequently, the *Prototype Energy*

Modelling System must be able to automatically create an *Energy Model* by automatically formulating the required mathematical expressions for optimisation. This *Energy Model* must then be able to optimise an energy mix.

The key emphasis here is that the hypothesis proposes a modelling system which operates at a level that is above that of optimisation models (Figure 1.2). The proposed modelling system *encompasses* optimisation models but it is not the focus of the system. The *Prototype Energy Modelling System* therefore has two main outputs. First, it must create an *Energy Model*. Second, the *Energy Model* must output an optimised energy mix.

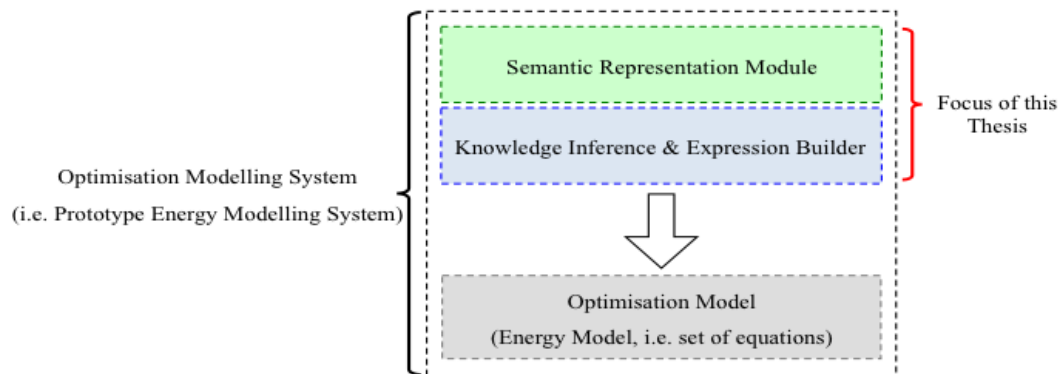


Figure 1.2. Relationship between the *Optimisation Modelling System* and *Optimisation Model* in this thesis.

To validate the hypothesis, this research aims to formulate a new methodology (framework) for the development of an *Optimisation Modelling System* which can be used for policy evaluation and analysis. This framework proposes the integration of semantic representation and knowledge inference within an optimisation model. The selected technique for semantic representation is through the use of a formalism method known as ontologies while the selected technique for knowledge inference is

through the use of description logic. The ontologies also integrate a mathematical-based set of engineering models. It is envisioned that this combination will result in a user friendly system that recommends, in the case of energy models, an electricity generation mix which fulfils the energy policy goals of energy security, energy affordability and environmental responsibility among others.

Figure 1.3 illustrates the system architecture of the *Optimisation Modelling System* that has been developed using the proposed framework. The prototype has modules that are similar to that of other optimisation models which are highlighted in grey. The prototype also has additional modules which are the *Semantic Representation* and the *Knowledge Inference and Expression Builder* modules that are highlighted in blue. New proposals and concepts within the prototype are highlighted in red. As the discussion in this thesis progresses, additional details will be revealed in Figure 5.1, and Figure 6.1.

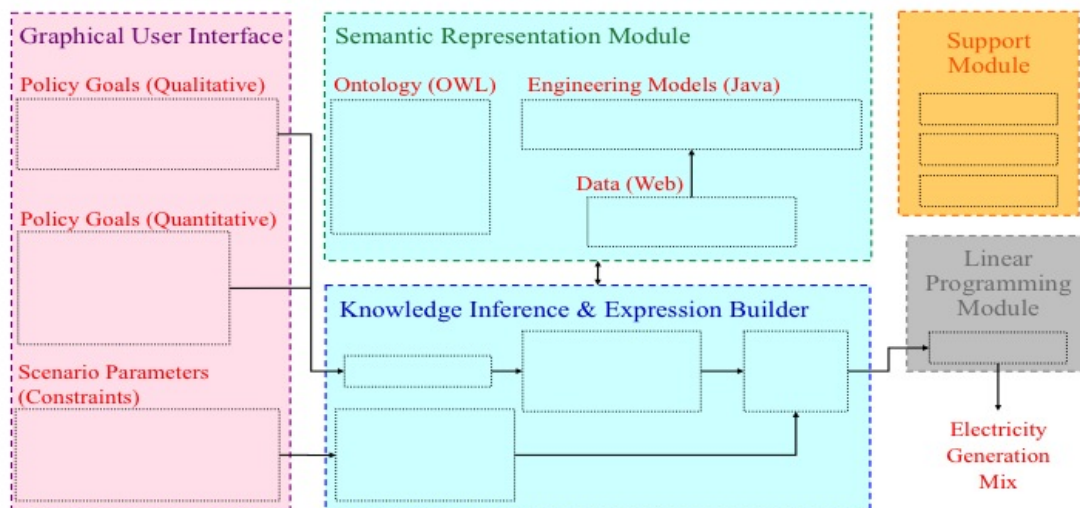


Figure 1.3. System architecture of the *Optimisation Modelling System*.

To successfully develop the framework for the *Optimisation Modelling*

System, the challenges in the list below had to be met:

1. Accurate representation of the knowledge domain using an ontology-based semantic representation method.
2. Accurate interpretation of the inputs, which are expressed as policy goals and targets, from the policy maker, into mathematical expressions.
3. Automatic formulation of a set of mathematical expressions necessary for numerical optimisation.

The successful completion of this research project hinges on the proper resolution of these challenges. The methods by which this is achieved are discussed in Chapters 4 and 5.

1.4 Contributions of the Research

The key contributions of this research are a framework and a prototype modelling system for energy policy.

The framework documents the methods by which the incorporation of AI techniques within an optimisation system has been achieved. It delineates how ontologies are used to represent the semantics of the knowledge domain and describes how this knowledge is then evaluated through the use of description based logic inference. The evaluated knowledge is then used to automatically formulate mathematical expressions for use within the optimisation model. The framework is discussed in Chapters 4 and 5.

A Prototype Energy Modelling System which has been constructed on the basis of the framework is discussed in great detail in Chapter 6 with an accompanying case study in Chapter 7.

To recapitulate, the contributions of this research are:

- A framework which outlines the integration of a semantic representation and

knowledge inference with an optimisation model used for policy evaluation and analysis.

- An ontology-based semantic representation of the energy policy knowledge domain using OWL with description logic.
- A *Prototype Energy Modelling System* which accepts both qualitative and quantitative energy policy goals as inputs and outputs an electricity generation mix that complies with the specified policy goals.
- An energy policy case study of Malaysia with regards to electricity generation mixes.

1.5 Structure of the Thesis

This thesis introduces the framework and development details of a *Prototype Energy Modelling System*. Chapter 1 (this chapter) introduces the project. Chapter 2 reviews the literature of semantic representation with emphasis on ontologies as a knowledge representation technique and automatic programming. Chapter 3 reviews the literature of energy models. Chapter 4 discusses the background while Chapter 5 outlines the proposed framework. Chapter 6 documents the details of the *Prototype Energy Modelling System* while Chapter 7 illustrates a case study for Malaysia. Chapter 8 concludes with a brief discussion of the advantages, capabilities, disadvantages, and limitations of the framework.

Chapter 2 Literature review of Semantic Representation and Automatic Programming

The purpose of this chapter is to review the literature relevant to the research work. It provides the reader with an understanding of the main techniques and concepts from AI employed in this project. Section 2.1 begins with a background on the field of knowledge representation within computer science, describes the role of ontologies and logic within this field and reviews the application areas of knowledge representation. Section 2.2 introduces ontologies with an explanation of the ontology hierarchy, a description of the variants of the ontology language OWL and a review of the application of ontologies in current research. Section 2.3 introduces automatic programming, a concept where a system interacts with a user in the natural terms of a subject domain and then automatically synthesises a computer program that meets the requirements specified by that user. Section 2.4 concludes this chapter.

2.1 Knowledge Representation and Engineering

Plato defines knowledge as the justified true belief which requires three conditions: that something is true, that someone believes it is true, and that the particular person's belief is indeed justified. The term 'representation' has different meanings and example of which is a notation together with an interpretation of the notation [3].

2.1.1 Historical Background of Knowledge Representation

Interest in knowledge representation began in the 1970s with the development of knowledge-based systems which sought to transfer human knowledge into the computers [4]. This transfer methodology assumed that knowledge elements are

codified and all that is required is the collection of information and implementation within a computer system. However it soon became evident that this assumption was flawed due to the important role of tacit knowledge which is knowledge known to the subject matter expert and not easily written or transferable to another person [5]. This leads to a paradigm shift from that of a transfer approach to that of a modelling approach. This in turn leads to the development of knowledge engineering as one discipline which seeks to integrate knowledge into computer systems in order to solve complex problems which would normally require a high level of human expertise.

Knowledge engineering seeks to construct computer models with the aim of realising problem-solving capabilities comparable to but with no intention to replace that of a domain expert. It is not intended to create a cognitive equivalent of a human domain expert, but rather to create a model which offers similar problem solving results in the area of the concern [6].

The advent of distributed information exchange through the Internet and reusable code within an interoperable multi-platform software has resulted in the further evolution of knowledge engineering into ontology engineering. In ontology engineering, the “ontology” is the key technology concept used for the construction of knowledge-intensive systems [7]. While an ontology is only one of several forms of knowledge representation technologies available, nevertheless it is fast becoming a popular, feasible and efficient solution [8,9] for the implementation of knowledge representation systems.

2.1.2 Ontologies and Knowledge Representation

The term ontology itself originates from philosophy and dates from the time of Aristotle who sought to classify the objects in the world through how and where they are employed as a method to describe their existence [4]. In a sense, an ontology tries

to answer the questions "*What is a being ?*" and "*What are the features common to all beings ?*" [10].

This definition of ontology fits well within the realm of AI which deals with reasoning about the world using models. Within the AI community, the term ontology has come to refer to a wide range of formal representations from taxonomies and hierarchical terminology vocabularies to detailed logical theories describing a domain [11]. In practice however, there are numerous different viewpoints [10] and hence definitions of what an ontology is [4].

One of the earliest definitions by Neches *et al.* [12] is "an ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary". Another definition, from a knowledge representation perspective, by Issa [13] is "ontologies can be considered to be a form of knowledge representation in which the terminologies have been structured to capture the concepts being represented precisely enough to be processed and interpreted by people and machines without any ambiguity". Ontologies have also been characterised by Noy *et al.* as [11] "An ontology is an explicit specification of the concepts in a domain and the relations among them which in turn provides a formal vocabulary for information exchange". From an ontological engineering project standpoint, Schreiber *et al.* states that "An ontology is an explicit, partial specification of a conceptualization that is expressible as a meta-level viewpoint on a set of possible domain theories for the purpose of modular design, redesign and re-use of knowledge-intensive system components" [14]. And last but not least, the term ontologies can also refer to a set of computer technologies for knowledge representation that allows for rapid and accurate storage, retrieval and inference capabilities.

Ontologies can be modelled using various modelling techniques or paradigms

and can be implemented in various kinds of languages [15]. Nevertheless, because ontologies should be as formal as possible, the formalism methods most often used are based on logic. The classical paradigm which is grounded in classical logic and the Datalog paradigm which is based on non classical logic are two examples of ontology paradigms [16]. Indeed, the use of logic which enjoys widespread familiarity and is grounded using mathematics [17] is one of the reasons for the popularity of ontologies.

2.1.3 Logic and Knowledge Representation

As logic is a formal method for reasoning, many concepts that can be verbalised can also be translated into symbolic representations that closely approximate the meaning of these concepts. These symbolic structures can then be manipulated in programs to deduce various facts akin to automated reasoning. In addition, logic offers the only formal approach to reasoning that has a sound theoretical foundation. This is important for automated reasoning as the inferred results should be correct, reproducible and logically sound [17].

There is nevertheless considerable debate over the appropriate logic type necessary for knowledge representation. The objective of using logic, which is the production of a structured expression that the receivers or interpreters cannot interpret differently from the author's intention, requires a trade-off between expressiveness and unambiguity [17].

Logics can be divided into two broad categories: classical and non-classical. Classical logic is composed of two types [17]; propositional and first order predicate logic. Propositional logic can be considered as a simplified form of first order predicate logic. In classical logic, we only consider the quantifiers ‘every’, ‘all’, and ‘some’. Predicate logic is by far the most important technique for knowledge

representation. Indeed, description logic, which is the logic used by a number of ontology languages, is essentially a decidable fragment of first order predicate logic [18].

Non-classical logics, examples of which are fuzzy logic and non-monotonic logic, provide additional flexibilities that are not available within classical logic. Fuzzy logic allows for the representation of the vague expressions of a human expert through the use of more general quantifiers such as 'more or less', 'very', 'few', and so forth. This increases the range of the usage of logic for real world representation.

Non-monotonic logic allows for the assumption of incomplete knowledge which is useful in applications where most of the facts are known, and it is, therefore, reasonable to assume that if a proposition cannot be proven, it is false. The basic idea of non-monotonic logic is that a new bit of knowledge can be derived from generally accepted premises unless a counter instance is explicitly proven [17]. This is of use for real world representation.

Despite limitations, classical first order predicate logic has been the most successful. While the narrow scope of predicate logic, due to its inflexible structure, can limit its practical use for the representation of natural languages, for most other representations it has proven to be extremely useful. Significant tools for knowledge representation have been developed with an emphasis towards first order predicate logic. These include reasoners which can be used for knowledge inference.

2.1.4 Application of Ontology-Based Knowledge Representation Systems

The current application of ontologies has focused on the representation and manipulation of semantics in computer software. Ontologies have been applied in the areas of chemical process modelling [19], decision making in pharmaceutical product development [20], automated regulatory compliance in pharmaceutical manufacturing

[21], improved interoperability between Geographical Information Systems (GIS) and Computer-Aided Design (CAD) [22], knowledge sharing for functional design knowledge on production systems [23], and process operations mapping in an oil-refinery plant [7,24]

While the encoded knowledge domain can be in any specific subject area, ontologies have seen particular extensive use in the life sciences and environmental science domain [25,26]. Examples of ontologies used in engineering modelling are the *OntoCAPE Ontology* developed under the IMPROVE and COGents project [27] and the *Plant Ontology* [7,23,24].

OntoCAPE is [19] a formal large scale ontology that represents the chemical process engineering knowledge domain. The domain covers the major engineering activities of design, construction and operations of chemical plants. Applications of *OntoCAPE* include that for the automatic selection of software components for process modelling [28], computer-aided modelling of chemical processes [29], and the semantic annotation of the contents of electronic documents related to the process engineering domain such as flowsheets and material properties databases [19].

Plant Ontology [8] is a collection of ontologies (i.e. task, domain) that represents the equipment, materials and operating processes used in chemical plants. This *Plant Ontology*, (called *Device Ontology* in [24] and *Extended Device Ontology* in [23]) when combined with a *Functional Ontology* [30], can be used for the functional representation of a chemical plant. Applications include the development of a knowledge management software called SOFAST (Sumitomo Osaka-University Function Analysis and Systematization Tool) [23].

The use of ontologies for knowledge representation within the area of energy models is a new development. The *STS ontology* and *SynCity ontology* are ontologies designed to improve the interoperability between some agent-based modelling

systems [31]. The *STS ontology* is designed for the modelling of socio-technical infrastructure systems while the *SynCity ontology* is used for modelling urban energy systems [32]. To the best of our knowledge however, ontologies have not been applied to optimisation based energy models of the type reviewed in Chapter 5. For such energy models, only rule based expert systems have been developed [33].

2.2 Ontologies as a Knowledge Representation Formalism

This section discusses ontologies as a technique for knowledge representation. It introduces the concept of ontologies, describes what an ontology taxonomy is, explains how this taxonomy is coded and concludes with a review of the different construction taxonomies.

2.2.1 Introduction to Ontologies

From a computer science perspective, the strength of an ontology lies in its ability to encode knowledge in a manner that can be communicated and shared between people and software tools [6,34,35]. Ontologies achieve this through the representation of the semantics of a knowledge domain with all associated terms, interconnections and inference rules [36].

While the term ontology is widely used to refer to a representation concept, there are in fact several different types of ontologies. These ontologies share some underlying idea of capturing explicitly static knowledge about some domain but also vary considerably. It is therefore useful to distinguish between the generality levels of ontologies [7]. These are domain ontologies (to capture knowledge from a particular domain), generic ontologies (which are valid across several domains), application ontologies (which contain knowledge necessary for modelling a particular domain), and representational ontologies (which do not commit to any particular domain).

2.2.2 The Ontology Taxonomy

All ontologies however, share something in common and that is a hierarchy. An ontology hierarchy is a formal description of a knowledge domain using classes, properties to describe the various features and attributes of the classes, and instances [37]. The ontology hierarchy is constructed by defining classes of concepts, their taxonomy, the possible relations between the concepts, and the axioms (logic statements) for those relations. It is their combination that allows for a semantic representation, i.e. the capture of the meaning of words and statements.

A class represents a category of concepts that share a set of properties [34]. *Power_stations*, for example, can be a class which represents a category for all facilities that generate power. Class taxonomies are defined with the use of the subclass relation. A class is a subclass of another class if every member of the subclass is also a member of the superclass [34]. This 'is-a' relation facilitates automatic inheritance of a primitive or subclass and its associated semantics [7]. A *Hydroelectric_dam*, for example, can be a subclass of *Power_stations*. In addition, an instance is a specific instantiation of a class and is used to represent a specific object. The *Hoover_dam*, for example, can be an instance of the *Hydroelectric_dam* class.

There are two kinds of properties: object and datatype. An object property represents the relation between classes. For example, *generate* can be an object property that relates the class *Hydroelectric_dam* to the class *Electricity*. A datatype property is a pointer from a class (or instance) to its numerical or Boolean value. Axioms are used to further specify the relationships between classes. Examples of datatype properties are *hasValue* and *lessThan*.

Axioms are used to define the specific relationship between concepts using the properties. An axiom is based on a proposition or sentence that is always true. Axioms are the most powerful means of representing knowledge in ontologies, and are often

used to represent knowledge that cannot be represented with the other primitives of the language [38].

It is also sometimes helpful to think of the ontology taxonomy by differentiating two components: TBox and ABox [21]. The TBox section describes terminology, i.e., the ontology in the form of concepts and roles. The ABox contains assertions about individuals using the concepts in the ontology. Concepts or classes describe sets of individuals while roles or properties describe the relations between individuals.

2.2.3 Ontology Development

To develop ontologies, a computer language must be selected that has the ability to represent information about objects and how such objects are interrelated. In addition, the languages must include the ability to specify a vocabulary which both describes the object and also formally defines it in a format that facilitates automated reasoning [39]. Indeed there are many ontology languages available for selection as shown in Figure 2.1, which is based on a figure in [40] with the inclusion of OWL.

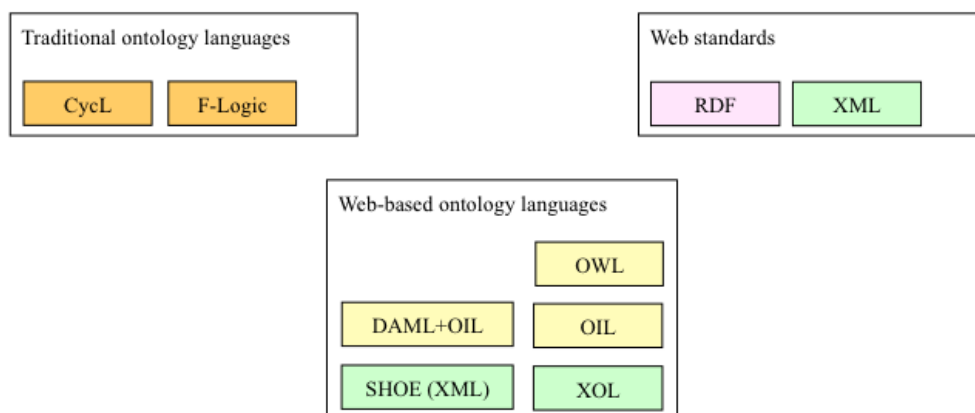


Figure 2.1. Classification of languages [40] inclusive of OWL.

Early approaches, in particular for the deployment of ontologies as a knowledge representation within the semantic web, saw the development of the Knowledge Interchange Format (KIF), F-Logic, Dublin Core and CycL [39]. KIF is a formal language for the interchange of knowledge among disparate computer programs which was developed under the Defence Advanced Research Projects Agency (DARPA) [41]. F-Logic is a full fledged logic formalism that has a model-theoretic semantics and a sound and complete proof theory [42]. Dublin Core is widely adopted initiative that aims to facilitate the electronic resource discovery on the web through standardisation in the vocabulary terms used to describe the resources [39]. CycL is a declarative and expressive language similar to first-order predicate calculus that has been used to develop CYC, a large ontology which contains formalised representation of a vast quantity of fundamental human knowledge (e.g. heuristics, rules of thumb, face) for reasoning about objects and events of everyday life [43].

More recent approaches include the Extensible Markup Language (XML), Resource Description Framework (RDF), Ontology Inference Layer (OIL) and DARPA Agent Markup Language (DAML) as shown in Figure 2.2. XML is a markup language for documents and is used to encoded structured information about the contents or parts within a document (e.g. headings, titles, fonts, and graphics) [44]. Markup languages [45] are computer languages that utilise words or symbols to describe the identity or function of parts of a document an example of which is the Hypertext Markup Language (HTML) that is used for pages on the World Wide Web (WWW or the Web). RDF is a standard (i.e. data model or data structure) to add formal semantics to the web and is used to describe resources in order to allow interoperability between web applications that exchange machine-understandable information [46]. The RDF specification includes the RDF Schema (RDF-S), a

vocabulary description language that defines the words used to describe the resources [46]. XOL and SHOE (XML) are ontology-exchange languages that have similar functions to RDF. OIL is a method of representing information on the web in such a way that its meaning (semantics) is machine-accessible [36]. OIL combines the modelling primitives of frame-based systems (e.g. class or frames and properties or slots) with the expressive power of description logics to describe semantics and the syntax structure of markup languages which are part of the web standards [47]. DAML is a US Government sponsored project aimed at providing the foundation for the evolution of the semantic web. DAML consist of two portions, the ontology language DAML+OIL which is built on top of RDF and RDFS with its root in description logics and an inference language DAML-L which expresses constraints and adds inference rules [39]. HTML, SHOE, XOL have not been reviewed here given their similar function to XML and RDF.

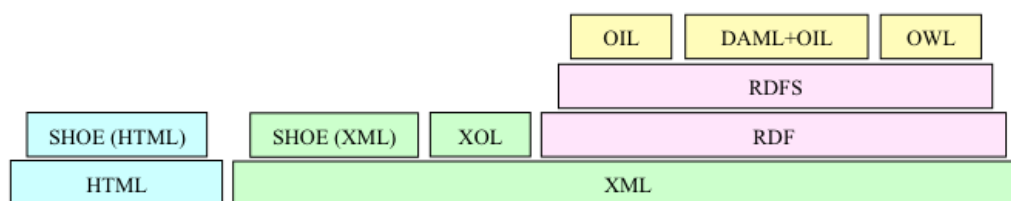


Figure 2.2. The stack of ontology markup languages [38].

The explosive growth of the web and the expansion of ontologies have required standardisation. In 1994, the World Wide Web Consortium (W3C) was established as the main international standards organisation for the Web. One of the efforts of the W3C has been the development of a new ontology language called the OWL Web Ontology Language [48].

OWL was developed by the W3C Web Ontology Working Group as a language to represent information about categories of objects and how objects are interrelated and can also represent information about the objects themselves which is the sort of information that is often thought of as data [49].

OWL has been developed to fit into the semantic web vision of a stack of languages including XML and RDF (Figure 2.2). OWL adopts and extends the fact stating ability of RDF and the class and property structuring capabilities of the RDF scheme [33]. OWL also retains compatibility with existing languages including SHOE, OIL and DAML+OIL.

OWL can declare classes and organise these classes into a hierarchy [25]. Using axioms, the classes can be specified as logical combinations of other classes or as enumerations of specified objects. OWL can also declare properties and organise these properties into a hierarchy [49]. The properties can be provided with domains (which is a class or instance) and ranges (which is a class, instance or literal). The properties can be specified as either transitive, symmetric, functional or as the inverse of another property. OWL can express which instances belong to which classes and what property values are associated with it.

As OWL is designed with the specific intention of knowledge representation using ontologies within a semantic web, it has become the preferred language. Three flavours of OWL languages (OWL Lite, OWL DL and OWL Full) are available and they differentiate in terms of expressivity. OWL DL is a version of OWL with decidable inference that can be written in a description logic manner. OWL Lite, which is generally considered to be a syntactic subset of OWL DL, uses a simpler syntax and has a more tractable inference capability. OWL Full provides extensive upward compatibility at both the semantic and syntactic level to RDFS but can result in non resolvable inferences [49,39].

OWL-DL in particular is based on description logics, which is a family of class-based (concept-based) knowledge representation formalisms characterised by the use of various constructors to build complex classes from simpler ones with an emphasis on decidability of key reasoning problems and the provision of sound, complete and (empirically) tractable reasoning services [49]. The key feature of description logics is the use of formal languages with well-defined semantics. Description logic is essentially a decidable fragment of first-order predicate logic [18] whose language is so formed to be practical for modelling purposes with good computational properties such as decidability. For details on OWL, the reader is encouraged to refer to [49].

To develop an ontology, OWL has a variety of publicly available tools for editing and syntax checking [50] such as Protégé [51,52] and SWOOP [53]. Protégé is a free, open-source Integrated Development Editor (IDE) developed by Stanford University [51]. Protégé provides a graphical user interface for ontology editing and knowledge acquisition that allows the user to concentrate on domain models at the conceptual level without having to know the syntax of the language on which the model is built upon. It allows a developer to focus on the concepts and relationships in the domain and the facts about them that need to be expressed [11].

2.2.4 Ontology Construction

Ontology construction is the act of constructing an ontology taxonomy with an 'is-a' hierarchy of concepts with attributes, values and relations. The information about these classes and their relations to each other, as well as constraints on attribute values for each class, are captured through axioms [7].

Ontology construction has been the subject of several research proposals since the late 1990s [38]. These proposals have been born out of ontology development

projects which have seen the development of the Cyc Ontology [43], Enterprise Ontology [54], TOVE (Toronto Virtual Enterprise) Ontology [55], and Esprit KACTUS Ontology [14,56], SENSUS Ontology [57]. and Chemicals Ontology [58].

Several papers on the methodologies for ontology development have been published. These include Lenat *et al.* [43], Ushold and King [54], Gruninger and Fox [55], Bernaras *et al.* [56], Swartout *et al.* [57], Fernandez-Lopez *et al.* [58], Staab *et al.* [9] and Noy *et al.* [59]. Descriptions, comparisons and evaluation of these methodologies have been published by Fernandez-Lopez and Gomez-Perez [38,60,61,62].

Lenat *et al.* [43] proposes a construction method which consist of three phases: the manual codification of articles and documents, the manual codification of new information using tools and the automatic codification of new information using tools. Ushold and King [54] propose a four step process: identify the purpose of the ontology, build it, evaluate it and document it. In both these methods, one of the main sources of knowledge for codification is from that of a domain expert.

Gruninger and Fox [55] propose an approach that is philosophically different in that knowledge for codification flows not from a domain expert but rather from problems that the ontology wishes to solve. The approach consists of three phases: define the requirements of the ontology based on the problems encountered, define the terminology of the classes, properties and instances in the ontology and specify the definitions and constraints of the terminologies.

Bernaras *et al.* [56] proposes a bottom up construction method that focuses on application development with continuous expansion of the ontology as each new application is created [61]. The created ontology starts as a very specific domain ontology that evolves into a general ontology. This three step method consists of the specification of the application, the preliminary design of the relevant top-level

ontological categories and the refinement and restructuring of the ontology.

Swartout *et al.* [57] in contrast, proposes a top-down approach that focuses on deriving domain specific ontologies from huge ontologies [38]. This five step approach is: identify the seed (key domain) terms, manually link the seed terms to SENSUS, add paths to the root of SENSUS, add new domain terms and complete the hierarchy [59].

Fernandez-Lopez *et al.* [58] propose a life cycle approach in Methontology. The approach has three broad activities which are project management, ontology development and ontology support [38,62]. The development phase has five tasks which are specification, conceptualization, knowledge acquisition, integration, and implementation [58].

Staab *et al.* [9,63] proposes the On-To-Knowledge methodology which focuses on ontologies for knowledge management and consists of five phases: feasibility study, kickoff, refinement, evaluation and maintenance.

Noy *et al.* [59] recommends five easy steps for ontology building which are to define the classes, arrange the taxonomy, create the properties, fill the property values, and instantiate the instances.

2.3 Automatic Programming

The goal of generating programs in an automatic way (i.e. Automatic Programming) has been sought since the first time programmers came face to face with the difficulties of programming. In the ideal world of automatic programming, a user would need to just define what was expected from the program (i.e. the requirements) and the program would be automatically generated by the computer without assistance from a programmer.

Another motivation of automatic programming is the assumption that many

computer users have no interest in doing their own programming. To these users, computation is just a tool used to solve a problem in a domain that is of interest to them. Such users would benefit from a system could produce programs for them based on communications using the natural terms, concepts and styles used in the problem domain. Conventional programming however is unable to cater to such a need [64].

Thus, most automatic programming systems attempt to perform, without human intervention, all of the transformation required to generate a working implementation from an initial specification of user requirements [65]. Unfortunately, achieving this goal has proven to be extremely difficult.

2.3.1 Current Approaches for Automatic Programming

As the goal for a general purpose user oriented fully automatic programming system is not possible, current approaches involve some form of a compromise. The three compromises that are currently practised are bottom-up, narrow domain and assistant [66].

The bottom-up approach sacrifices end-user orientation. It starts at the programmer's level and pushes the threshold of automation upwards. This threshold has been raised from machine-level to high-level languages and the current goal is to raise the threshold further to very high level languages. The evolution of machine code to assembler languages to high level languages is a good example of this approach.

The narrow domain approach sacrifices general purpose. The focus of the domain is narrow enough that it is possible to construct an automatic program generator which communicates directly with the user. An example of this approach, in our opinion, is that of a domain specific automatic programming for oil well logging

developed by Barstow [67].

The assistant approach sacrifices full automation. It instead provides assistance with programming through various tools such as intelligent editors and online documentation aids. Current research in this approach focuses on improvement in both the integration between the various tools and the level of assistance provided by each individual tool. An example of such a system, in our opinion, would be KBMoSS, a knowledge based processing modelling support system that synthesises a model. This model consists of a set of mathematical equations based on a step by step description of the chemical process (e.g. process flow configuration, thermodynamic state of the fluids) necessary to create a process flow diagram [68].

2.3.2 Current Methods for Automatic Programming

Current automatic programming systems work by mapping domain specific terms from the user into implementation specific terms (i.e. programming commands). Several implementation methods are available for this, among which are procedural, deductive, and transformational [66].

In a procedural method, a special-purpose program that gets the right results from the inputs is written. Examples of programs using the procedural method are compilers.

In a deductive method, the logical relations between the inputs and outputs specified by the user are deduced [69,70]. A theorem proving algorithm then attempts to synthesise a program that is guaranteed to satisfy this logical relationship. An early example of a program using this method is PROW [71]. PROW accepts the specification of the program in the language of predicate calculus, determines the algorithm for the program and then produces a LISP program which is an implementation of this algorithm.

In the transformational method, a sequence of transformations is applied to convert the inputs, which are written in a very high level language, into a low level language for implementation [66]. An example of this system, in our opinion, is that developed by Perkins *et al.*, where the prototype system automatically generates a mathematical model to describe the dynamic behaviour of a process system from purely physical descriptions in the English language [72].

2.3.3 Knowledge Representation and its Application in Automatic Programming

Although transformation methods are usually employed to address this problem, they cannot be employed if the gap between the specification and the actual implementation is too wide. For automatic programming to work, the end user oriented automatic programming systems must be domain experts [66].

For automatic programming systems to function, implicit knowledge of the subject domain is required [67]. The system must have implicit knowledge of two domains: one of the subject domains to enable interpretation of the inputs from the user and another of programming methods to enable the synthesis of the computer program. In addition, because such programs attempt to answer complex problems, the use of straight forward programming techniques to write the programs is insufficient. Sophisticated AI techniques such as the knowledge representation of the task domain can play a major role in helping a machine to cope with this complexity [64].

This need for a knowledge domain, encoded in some machine usable form, thus suggests a way to try to build automatic programming systems [73]. Among those researchers who have attempted to develop this approach is Barstow [64,67,74]. The PECOS system is a rule based system which synthesises programs from abstract

specifications through the use of a knowledge base which contains programming rules [74]. Barstow has also outlined an experimental framework for a knowledge domain approach to automatic programming [64] which was followed with the development of an automatic programming system for oil well logging [67].

The specific use of ontologies as a knowledge representation method for automatic programming has also been developed. In [75], Borgida *et al.* discusses the use of ontologies to assist in the requirement stage of software development. Two examples of systems using knowledge representation are by Nakayama *et al.* [76] and Uschold [77]. In [76], Nakayama *et al.* describes an automatic programming system which generates control programs for steel plants. In [77], Uschold discusses a prototype system which allowed an ecologist to create models of the objects and processes in a real or imagined ecological system which was then converted into executable code in FORTRAN or Prolog.

The use of ontologies and furthermore semantics within automatic programming represents an exciting future which can truly bring in a world where application development opens up to the masses. As Uschold [73] boldly asserts, the distinction between user and developer and between conventional and semantic computing will blur in the future with all information systems being ontology driven.

2.4 Conclusion

For this project, a semantic representation of the energy policy knowledge domain has been developed using ontologies. The resulting ontology taxonomy is then combined with engineering models (implemented in Java) to complete the knowledge representation of the energy policy knowledge domain.

The semantic representation forms part of an *Optimisation Modelling System*. The aim of the system is to use knowledge inference to automatically create energy

models which comprise a set of mathematical expressions for optimisation. This created model is based on inputs from the policy maker. In this regard, the *Optimisation Modelling System* is a form of automatic programming in so far as it accepts high level inputs and then automatically creates a computer program without further intervention from either the policy maker or computer programmer. The approach of this system is a narrow domain knowledge based approach in the area of policy and the target users for communication are policy makers.

Given the aim and scope of the project, certain technologies and approaches were better suited than others for implementation in the system.

For semantic representation using ontologies, OWL was selected over all other languages. Cyc (CycL) and F-Logic were not selected because of our preference for an open world assumption in order to facilitate the codification of incomplete information [16]. KIF was not selected because of its language structure, which is cumbersome to use for ontology modelling due to the fact that the original intention of KIF was for information exchange between machines [62]. OIL+DAML and RDF-S were not selected as OWL represents the evolution (and in fact is built upon) of these languages and is the current recommended standard.

Of the three approaches to automatic programming, in our opinion, the narrow domain approach is the most attractive and relevant to this project. Given that the target users of our system have little interest in programming, our system should have minimal programming requirements. Both the bottom-up approach which still involves programming at higher level and the assistant based tool system which sacrifices automation, would not have fulfilled the requirements of our system. Hence, the narrow domain approach was the logical choice.

In conclusion, it is our view that OWL and the narrow domain approach represents the best choices for use in this project.

Chapter 3 Literature Review of Energy Models

The purpose of this section is to review the literature relevant to the *Energy Model* generated by the *Prototype Energy Modelling System*. The chapter discusses the literature relevant to energy models, a type of computer software that is used to simulate the technological, economic and environmental effects of energy supply. We begin with an introduction, followed by model characterisations and conclude with a summary which places into context the *Energy Model* generated by the *Prototype Energy Modelling System*.

3.1 Introduction to Energy Models

Energy models are useful tools which simulate the technological, economic and environmental effects of energy supply. Models such as MESSAGE [78,79,80], MARKAL [81,82], and DNE-21 [83] are used by energy planners for medium to long term energy system planning, policy analysis and scenario development [79,80]. The strength of energy models as optimisation tools is derived from their solid economic foundations which are based on complex mathematical expressions that can process numerical data and provide numerical solutions related to economics and the environment. Energy models can also be used as simulation tools based on their detailed accounting of the energy system.

There are, arguably, four milestones in the application of energy models for economic and policy analysis. Originally developed in the 1960s for specialised studies such as oil price prediction, these models gained prominence in the early 1970s. In 1972, the Club of Rome published the 'Limits of Growth', a study through modelling which sought to explore the limits of human growth within the context of limited resources. In 1974, the US Federal Energy Administration launched Project

Independence, an effort to develop and implement a national energy policy in the United States which featured the use of energy policy models for policy analysis [84]. In 1984, IIASA published 'Energy in a Finite World' by IIASA [85] which explored energy scenarios backed up with extensive use of optimisation models.

The contribution and value of energy models in energy policy analysis is undeniable but has not been without controversy. The performance and value of the models used in 'Energy in a Finite World' have been contested by Keepin [86,87]. At issue was the claim that the outputs of such models were effectively prescribed informally in the input data without any change by the model and that the results of the models were highly sensitive to perturbations in input data and hence lacked robustness [86,87]. In a rebuttal, Halafe and Rogner [2], noted that the main purpose of an energy model was to describe in a consistent manner scenarios of evolution which were conceivable but not necessarily actual futures. Energy models are, in principle, able to investigate the technological, economic and environmental issues and then formulate long-term solutions which meet both supply constraints and end-user demands at the lowest possible cost [82].

3.2 Characterisations of Energy Models

Energy models can be characterised in several ways and indeed papers where model classification and comparisons have been made have been authored by Grubb *et al.* [88], Urban *et al.* [89], van Beeck [90], and Connolly *et al.* [91]. In this review, four model characteristics are selected for discussion which are the purpose, analytical approach, underlying methodology, mathematical approach.

3.2.1 Purpose of Energy Models

The purpose of an energy model is an important characteristic as such models

are often developed to address specific questions and are therefore only suitable for the purpose that they were designed for [90]. A model can have either a general or specific purpose. General energy models are used to explore the future of energy while specific models are used to estimate, analyse and determine the impact of energy supply and demand [90].

General energy models are used for three purposes: to predict or forecast the future (forecasting), to look back from the future to the present (backcasting) and to explore the future (scenario analysis) [90,91]. Forecasting is a method for the estimation of future trends from the extrapolation of historical data and its value is highly dependent on the accuracy of the forecast [92]. Backcasting is a method to determine the areas for attention which starts from the construction of a desired future and follows with an analysis of the required changes to accomplish the desired future; the intention of the analysis is to highlight the different implications to the different futures chosen not on the basis of likelihood but on the basis of the criteria selected to form the desired future [92]. Scenario analysis is a technique which explores the future through the construction of alternative plausible futures and the comparison of such futures to a business as usual condition [93]. These alternative futures consist of a series of events that collectively forms a scenario. The scenarios themselves are based on a consistent set of plausible assumptions. Plausibility is key requirement of the constructed scenarios; while events in the scenarios may have a low probability of occurrence, nevertheless the events must be believable and the scenarios must be grounded in logic [94].

Specific energy models provide three kinds of outputs. Energy demand models focus on the estimation of energy demand as a function of other variables such as population, income and energy prices [90]. Energy supply models focus on the technical aspects of energy systems and whether supply can meet demand [90].

Energy impact models focus on the social, economic and environmental impacts caused by changes to energy systems and policy instruments or measures [90].

3.2.2 Analytical Approach of Energy Models

Energy models can be classified into one of three analytical approaches which are top-down, bottom-up and hybrid. A top-down approach uses macroeconomics with no explicit technology representation while a bottom-up approach models technology in explicit detail with an objective function set to minimise the costs of serving an exogenous energy demand [89,95,96]. A hybrid approach uses a combination of both top-down and bottom-up approaches.

The top-down models start with an economic model using prices and elasticity as economic indices and present the relations among energy consumption, production and economic indices internally while bottom-up models focus on the final consumption of energy based on actual energy use and the ways energy services are performed.

For a long-term forecast, a top-down world model is based on market equilibrium that forecasts world economic activity and the idea that change is indispensable. Among the many advantages of "bottom-up models", one important aspect is that their results can be interpreted clearly because they are based on detailed descriptions of changes in human activities and technologies. When introducing new policies, these bottom-up models, with their tangible results and explicability, are indispensable for explaining the directions and effects of policies to politicians [97]. Top-down models tend to underestimate the opportunities for low-cost efficiency improvements because the model often ignores whole categories of gains that are either too small to be represented or trapped beneath a price barrier or [88,98].

Bottom-up models have been developed in two directions. One is for

analysing more efficient technologies and their combinations by focusing on the supply and conversion of energy. MARKAL is an example of such a model. The other is to calculate how changes in the lifestyles of each sector influence energy demand by focusing on demand and consumption of energy in an attempt to determine the most efficient combination of energy technologies. MESSAGE is an example of such a model. Bottom-up models tend to overestimate the potential of efficiency improvements because they neglect various 'hidden' costs and constraints that limit the uptake of apparently cost-effective technologies [88,98].

Hybrid models are tailor made combinations of top-down and bottom-up models which serve a specific purpose, have specific data requirements and provide specific outputs [89]. MARKAL-MACRO is a hybrid model which combines MARKAL with MACRO, a top down macro-economic model [99,100] while MESSAGE-MACRO is a hybrid model which combines MESSAGE with MACRO [80].

3.2.3 Underlying Methodology

Energy models can be classified into one of three common underlying methodologies which are equilibrium, optimisation and simulation [89]. Several other methodologies are available, however since their use in energy models is limited, these methodologies are not reviewed here.

Economic equilibrium models are models which consider the energy sector as part of the overall economy for the medium to long terms. These models focus on the interrelations between the energy sector and the rest of the economy. The models do not concentrate on energy specifically but on the economy as a whole, of which energy is only a small part of the model. Indeed there is considerable debate on the classification of these models as energy models. There are two kinds of equilibrium

models which are macro-economic and economic equilibrium.

Macro-economic models simulate the entire economy and the interactions between the sectors for the short and medium terms. These models adopt a top-down approach and are based on macro-economic theories. MACRO is an example of a macro-economic model.

Economic equilibrium models, sometimes referred to as resource allocation models or computable general equilibrium (CGE) models, are bottom up models in either general or partial equilibrium. General equilibrium models implement a balance between supply and demand in all markets of the economy while partial equilibrium models implement equality between supply and demand in a specific market only. General equilibrium models are particularly concerned with the conditions which allow for simultaneous equilibrium in all markets, as well as the determinants and properties of such an economy-wide set of equilibriums [90]. Partial equilibrium models only focus on equilibrium in parts of the economy, such as the equilibrium between energy demand and supply. The mathematical differences between partial and general equilibriums are discussed in Beenstock [101]: a CGE model constructs the behaviour of economic agents based on microeconomic principles while a partial equilibrium model (i.e. dynamic optimisation model) minimises the total cost of an energy system [82].

Optimisation models are used to optimise energy investment decisions by finding an optimised solution which fulfils a set of given inputs [89]. The optimised solution represents the best outcome for a stated objective function within the scope defined by the constraints [90]. To achieve the solution, optimisation models often assume perfect markets and optimal consumer behaviour, factors which do not exist in reality [95]. To address this shortfall, such models can be connected to economic equilibrium models. Linear programming is often the selected technique for

optimisation [90].

Simulation models are descriptive models based on a logical representation of a system aimed at reproducing a simplified operation of the system. The simulation is static if it represents the operation of the system in a single time period and dynamic if the output in the current period is affected by evolution over several periods. These models are often used for scenario analysis. A disadvantage of simulation models is their tendency to be extremely complex [90].

3.2.4 Mathematical Approach

For mathematical based models, a further distinction with regards to their approach is possible. Linear programming, mixed-integer programming, dynamic programming or a combination of these techniques are commonly used. Other possible, but not in widespread use techniques include fuzzy logic which will not be discussed here.

Linear programming is a technique for finding a solution which either maximises or minimises an objective function subject to constraints with all relationships in the equations expressed in linearized terms. Linear programming was first introduced by L. Kantorovich in 1939 [102]. Subsequently, Dantzig *et al.* developed the Simplex Method to provide a systematic and computationally efficient process for solving general linear programming problems [102].

Linear programming is a very simple technique which requires little mathematical knowledge and can provide quick results. As a result, linear programming is a popular method that is used in a number of optimisation type energy models including MARKAL [81,103], MESSAGE [78], and DNE-21 [83].

Linear programming however does have a disadvantage in that all equations in the set must be linear (or piecewise) functions [88]. In addition, the produced

solutions can be very sensitive to variations in input parameters; small changes to the input parameters can lead to large variations in the produced solution (the 'big bang' problem) [88]. Furthermore, the most attractive variables are selected first before consideration is given to other variables. For example, in a resource optimisation problem, the cheaper resource is consumed up to its limits before consideration of other alternatives is given [90]. In energy optimisation models for example, this problem is usually dealt with through the introduction of constraints to limit the maximum introduction of technologies and hence forces the models towards a 'simulation' model. [104]. In addition, Messner *et al.* [78] has developed a stochastic version of MESSAGE to address this issue.

3.3 Summary of Energy Models

A selected set of energy models are reviewed here for completeness of the chapter.

MARKAL or MARKet ALlocation is a bottom-up linear programming model developed under the Energy Technology Systems Analysis Programme (ETSAP) of the International Energy Agency (IEA) [81]. MARKAL contains the entire energy system from imports and domestic production thru to end-user technologies and energy service demands [105]. MARKAL is used in a number of countries including the United States, United Kingdom, Australia, Canada, Japan, Malaysia, New Zealand and Thailand [103,105].

MESSAGE (an acronym for Model for Energy Supply Strategy Alternatives and their General Environmental impact) is a systems engineering optimisation model that is used for medium to long term energy planning, energy policy analysis and scenario development [78]. MESSAGE was developed by the International Institute for Applied Systems Analysis (IIASA) and has been used for various report including

the IPCC Special Reports on Climate Change [106]

HOMER is a micropower design tool developed by the National Renewable Energy Laboratory and is available for free download [91]. HOMER simulates and optimises stand-alone and grid-connected power systems with any combination of power generation sources.

RETScreen is a decision support tool developed by the Natural Resources Canada and is available as a free download [107]. RETScreen can be used to evaluate energy production and savings, costs, emission reductions and financial viability for renewable energy and energy efficient technologies.

Invert is a simulation tool developed by the Energy Economics Group at the Vienna University of Technology [108]. Invert supports the design of efficient promotion schemes for renewable and efficient energy technologies [91]. Invert simulates national energy-systems and can include all thermal generation sources except for nuclear power and all renewable sources except for wave and tidal [91].

3.4 Conclusions

Despite modelling differences, current energy models are powerful programs that have proved to be useful as policy simulation tools. However, their complexity and input data requirements are both a strength and a weakness [89,109]. These models require users with advanced skills, subject expertise and extensive training. These models also often require data to be expressed in either monetary or quantitative terms. Examples of projects which have attempted to address the complexity of models include the work by Hung *et al.*[33]. The MEDEE-S system was developed because it was felt that prevalent energy systems were too technical resulting in a system from which it was difficult to obtain results and that was difficult to evaluate in general. In order to assist planners, a rule-based MEDEE-S system was

developed and attached onto an energy model.

For classification purposes, the *Energy Model* generated by the *Prototype Energy Modelling System* presented in this document can be considered to be a simple bottom-up model with perfect foresight and without an attached economic model. It is emphasised, however, that the overall complexity of the generated *Energy Model* is much lower than that of traditional energy models and that the simple supply and demand simulation lacks the details of an economic equilibrium model.

The general purpose, analytical approach and underlying methodology of the general *Energy Model* were decided after due consideration of the literature. In our opinion, scenario based analysis represents the method of choice because it incorporates the uncertainties associated with the future of which both forecasting and backcasting cannot, on their own, account for [100]. In addition, we have selected the bottom up approach because of our desire to represent technology in an engineering sense. Furthermore, in adherence to the spirit of engineering design, which is about the optimisation of parameters, we hold that optimisation is most appropriate method for the allocation of energy resources.

The next chapter discusses the framework on which the *Prototype Energy Modelling System* and the created *Energy Model* based on.

Chapter 4 Development of the framework

This chapter and the next chapter discuss the framework used in this research to achieve the proposed objectives outlined in Chapter 1. This framework uses a combination of ontologies, engineering models, linear programming and description logic inference to develop an *Optimisation Modelling System* which considers both qualitative and quantitative inputs and data to automatically formulate mathematical expressions.

The chapter covers the first part of the framework which is the ground work on which the framework was developed. Section 4.1 discusses the concept of a framework. Section 4.2 provides a background to the problem that this framework intends to address, which is the dependence on a modeller, and explains how the basic requirements of the framework have been determined through a detailed analysis of the tasks performed by the modeller. Section 4.3 outlines the requirements of the framework while Section 4.4 discusses the design methodology that has been adopted for the framework.

4.1 Introduction to the Framework

As discussed in Chapter 1, this research has been motivated by the desire to expand the boundaries of an *Optimisation Modelling System* which encompasses a greater portion of the task and function of the modeller (Figure 4.1). The aim of the research is to improve the accessibility of models to the policy maker whose main interest lies with the effects caused by policies; it therefore follows that an energy model should have an emphasis on policy goals as the input with some form of automatic translation. Current optimisation energy models which utilise mathematical techniques such as linear programming require significant numerical inputs. Hence a

modeller is required to translate the intention of the policy maker, as defined by a set of policy goals, into mathematical equivalents fit for input into the energy model.

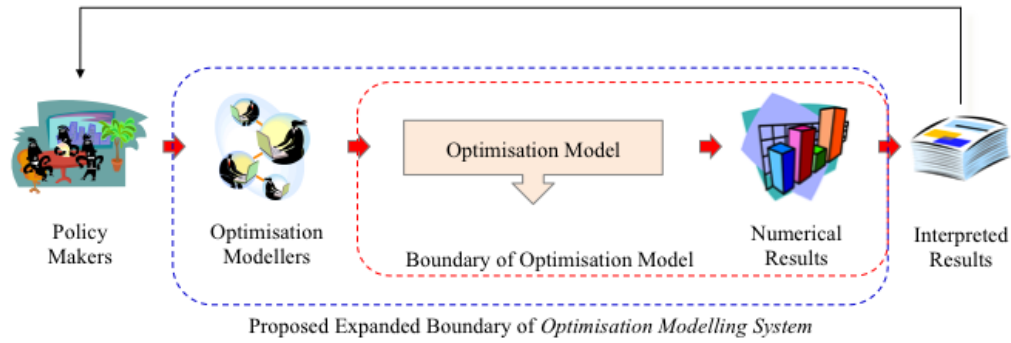


Figure 4.1. Process flow of an optimised model for policy evaluation and analysis.

As part of this research is to address this reliance on the modeller, a framework has been developed. A framework is a basic structure of a concept which provides an organised approach towards problem solving [111]. In the context of computer software development, a framework is an abstract design and implementation for an application in a given problem domain [112]. As a programming concept, a framework should contain a blueprint which defines the main components of the software and their respective functions, inputs and outputs, within the system architecture.

For this project, the framework documents the methods by which the application of a semantic representation using ontologies, description logic and logic inference techniques within an optimisation model for policy evaluation and analysis has been made. It delineates how ontologies are used to represent the semantics of the policy knowledge domain. It also describes how the knowledge that is inferred through the use of description-based logic inference can be used within the prototype

model.

The rest of this chapter discusses the ground work which forms the basis for the development of the framework.

4.2 Background of the Framework

To achieve the aim of this research, which is to mimic the role of the modeller, it is necessary to identify the current tasks of the modeller within the process of policy analysis using optimisation. Subsequently, an understanding of the complexities and requirements of each task is necessary in order to determine the computer techniques that can be used to mimic the role of the modeller.

With reference to Figure 4.1, it is evident that the main function of the modeller is to act as a bridge between the policy maker and the model. As a bridge, the main tasks of the modeller are to translate the policy goals (as inputs) from the policy maker into inputs for the optimisation model. Furthermore, this task is in fact a combination of five smaller subtasks which are to *understand*, *evaluate*, *relate*, *convert*, and *configure*. Table 4.1 illustrates these tasks.

Table 4.1. Summary of the background and requirements of the Framework.

Background of the Framework			Requirements of the Framework					The Framework	
Role (Modeller)	Subtasks (Modeller)	Subtask details (Modeller)	Required Functionality (Computer)	General Method (Computer Science)	Specific Technique (Computer Science)	Design Methods and Guidelines	Modules	Submodules	
Translate goals from the policy maker into inputs for the computer model	<i>Understand</i> the goals from the policy maker	Topic of the goals must be defined (e.g. policy) Type of goals can be either qualitative or quantitative	Store and retrieve qualitative and quantitative data Store data as information	Knowledge representation	Ontologies with Description Logic Query	Data identification and information selection Ontology construction Qualitative and quantitative data integration and evaluation	Semantic Representation Module	<i>Knowledge Domain Ontology</i> Engineering Models <i>Value Partition Ontology</i>	
	<i>Evaluate</i> the goals from the policy maker in the context of optimisation model analysis	Context of the goals must be specified (e.g. target group)	Information search						
	<i>Relate</i> the goals from the policy maker to the inputs required for optimisation	Relationships of the goals (e.g. policy instruments)	Information processing					<i>Scenarios Ontology</i> <i>Equation Construction Ontology</i>	
	<i>Convert</i> the goals into mathematical expressions for the optimisation model	Specific actions must be defined (e.g. constraint modification)	Information conversion and utilisation	Knowledge inference and procedural programming	Java based procedural programming for mathematical kernel interface	Mathematical approach	Knowledge Inference and Equation Builder Linear Programming Module		
	<i>Configure</i> the optimisation model	Specific actions	Mathematical optimisation	Procedural programming					

First, to *understand* the goals from the policy maker the modeller must be familiar with both the topic of application and the type of information which forms the input. For this research, the topic of application which is considered is that of policies such energy policy. Within policy, the types of information which form the input are either qualitative or quantitative policy goals. An example of a quantitative energy policy goal with a target is the reduction of CO₂ emissions to four tons per capita while that of a qualitative type energy policy goal is the enhancement of energy supply security.

Second, to *evaluate* the goals from the policy maker, the modeller must appreciate the context in which the inputs are applied. For policies, this context is called a policy variable which is the real world object that is the focus of the policy. An example of a policy variable is the electricity generation plants which are the object of focus for an energy policy goal of CO₂ emissions reduction to four tons per capita per year. Within an optimisation model, the policy variables are usually the variables of the mathematical expressions. Table 4.2 summarises the policy terms used in this thesis.

Table 4.2. Examples for the policy terms used in this thesis.

Terms	Example
Policy Goal	CO ₂ emission reduction
Policy Target	4 tons per capita per year (CO ₂ emissions)
Policy Variable (variable)	Electricity generation plant
Policy Instrument	Renewable Portfolio Standard

Third, to *relate* the goals from the policy maker to specific variables within an optimisation model, the modeller must determine how the inputs influence the policy variable. The mechanism by which the policy variable is influenced is referred to as a policy instrument. An example of an energy policy instrument in use in European Union (EU) member states is the quota obligation system which is also referred to as the Renewable Portfolio Standard (RPS) [113]. This obligation system supports renewable energy generation by establishing the amount or proportion of electricity supply that must be from renewable sources. Such an obligation system can in turn support the reduction of CO₂ emissions to four tons per capita per year.

Fourth, to *convert* the goals from the policy maker to inputs for the optimisation model, the modeller must either formulate new or modify existing mathematical expressions within the mathematical module. These mathematical expressions can be either the objective function, inequalities or equalities which represent policy goals or policy constraints.

Fifth, the modeller will have to *configure* the optimisation model to accept the new or modified mathematical expressions.

With the role, subtasks and subtask details of the modeller completely defined, the required computer functions can be identified. These can then be used as the basis for the requirements of the framework. The next section discusses this.

4.3 Requirements, Technologies and Techniques of the Framework

The main requirement of the framework is to mimic the five tasks of the modeller discussed in the previous section. For each task, a similar computer functionality is identified and forms the part of the requirement of the system. Once these requirements are determined, the basic class of technologies and the specific implementation technique to achieve these functionalities can be selected. This

section discusses these issues.

4.3.1 Requirements of the Framework

To simulate the *understand* subtask of the modeller, the computer must be able to store and retrieve data. In addition, the method of storage must be compatible for both qualitative (words) and quantitative (numbers) data storage. Furthermore, processing of the data into information should also form part of the data structure.

To simulate the *evaluate* and *relate* subtasks of the modeller, the computer must be able to retrieve the information using search criteria. These search criteria would be based on the context of the input from the policy maker. In addition, the collected information must be modelled into knowledge.

To simulate the *covert* and *configure* subtasks of the modeller, the retrieved information must be further processed to enable the automatic formulation of mathematical expressions for optimisation. The formulated equations should incorporate the retrieved information and reflect accurately the inputs of the policy maker. In addition, the mathematical model should allow for automatic reprogramming since changes to the code would be necessary to support these converted expressions. For example, if the policy maker decides to replace capital cost with total cost as the objective function for optimisation, the mathematical model should accept the inputs for the full set of cost values (i.e. capital, operational, maintenance and fuel/material) and automatically reprogram the objective function to account for this change.

4.3.2 Required Technologies and Techniques to Simulate the Tasks of the Modeller

These identified computer functionalities form the basic requirements of the

framework from which the general computer technologies can be selected. To do this, it is convenient to group the five functionalities into three set of tasks.

The first set of three tasks to *understand*, *evaluate* and *relate* can be fulfilled by methods from AI, in particular knowledge representation. The developed system should employ a method of codification of the knowledge associated with the domain.

The *convert* task can be fulfilled through a combination of knowledge inference and procedural programming (a programming paradigm that is based on the concept of the procedure call where routines, subroutines, methods or functions are executed in a particular step in order for the program to reach the desired state [114]). In particular, the system should utilise the qualitative and quantitative data contained within the knowledge base in order to achieve a fair translation of the policy maker's intentions.

The *configure* task can be achieved through a combination of functional programming and a mathematical kernel. The system should formulate, as parts of the model, a set of mathematical expressions that is representative of the inputs from the policy maker.

Of the methods available for knowledge representation, this framework has chosen to use ontologies. Its philosophy of describing objects in the world through their function and relation to each other is particularly relevant to the subject domain of this research. As exhibited in Section 6.2.1, Figure 6.10, ontologies allows for a clear and simple representation of the energy supply chain which forms a critical part of the energy policy knowledge domain.

OWL has been chosen as the implementation paradigm because of its design which is particularly suited for the representation of semantics. The fact that OWL allows relations of subclass to be inferred rather than be explicitly mentioned and implements an open world assumption makes it the paradigm of choice for this

project. In addition, the establishment of OWL as a standard by the W3C ensures that the ontologies developed will receive widespread support among the software developer community. The availability of established editor tools such as Protégé makes OWL development easy for this project. Furthermore, OWL-DL, which uses description logic, has widespread support from logic reasoner, which is a key component of this framework.

Linear programming has been selected as the mathematical technique because of its ease of implementation. The fact that linear programming can be implemented through commands in 'off the shelf' commercial mathematical packages makes it an attractive choice. In addition, linear programming has an established use within optimisation modelling because of its efficient algorithm in finding an optima value. As discussed in Section 5.2.4, a number of energy models of the type similar to the *Energy Model* produced by the *Prototype Energy Modelling System* use linear programming.

With the selection of ontology, OWL, and linear programming as the techniques for this framework, certain challenges which required further design decisions must be made. These decisions are discussed in the rest of this chapter.

4.4 Design Methods and Guidelines of the Framework

To achieve the aim of the project using the selected computer science methodologies and techniques, certain design methods and guidelines must be adopted in order to facilitate the implementation and to address the limitations of the chosen techniques. This section discusses the four main design methods and guidelines that are employed in the framework.

As this project involves knowledge representation, a key aspect is how to select from the vast amount of data and information related to the subject domain,

those pertinent for codification. As codification of the complete subject domain is not feasible, subsection 4.4.1 provides some guidelines to assist with the selection of only those data and information important to this application.

The use of ontologies as the knowledge representation technique requires adoption of a construction methodology to ensure consistency in the ontology taxonomy. Subsection 4.4.2 discusses the method used in the framework which is based on the combination of three previous ontology construction guidelines.

Since mathematical optimisation is the adopted solution method for this framework, it is useful to outline a method by which the qualitative data and information stored within the ontology can be converted into inputs compatible for numerical optimisation. Subsection 4.4.3 explains how through the use of Multicriteria decision analysis (MCDA) results from the literature, the qualitative data aspects in the ontology can be associated to quantitative values.

The method of implementation of the mathematical optimisation itself must also be selected from the various techniques that are available after consideration of the need to allow for automatic modification of the mathematical expressions that shall be used for optimisation. Subsection 4.4.4. describes the selection of the unstructured approach using Java and a commercial solver as the preferred method because of the flexibility afforded.

4.4.1 Guidelines for the Appropriate Identification of Data and Selection of Information for Codification

In principle, it is impractical to author a procedure which determines the content of a knowledge domain. Indeed, it is entirely at the discretion of the knowledge domain expert to judge what information should or should not be incorporated into the representation. Nevertheless, because the data that can be

codified is potentially large, we have found it helpful to employ two principles to assist with the codification process. These are (i) the need to codify both the core and application knowledge and (ii) the use of situational analysis tools to assist in the extraction of information.

To ensure a successful simulation of the knowledge of the modeller, it is necessary to appreciate the fact that there are two knowledge areas that must be codified; the core knowledge domain and the application knowledge domain. The core knowledge is the knowledge that the modeller needs in order to understand the inputs from the policy maker. The application knowledge is the knowledge that the modeller needs in order to evaluate and relate the inputs from the policy maker. As an example, for the *Prototype Energy Modelling System* (Chapter 6), energy policy is the core knowledge domain while the knowledge about the formulation of the set of linear programming expressions for the optimisation of electricity generation mixes is the application knowledge.

To assist with the extraction of the information from a knowledge base, this framework proposes the use of situational analysis techniques. Situational analysis is the systematic collection and evaluation of past and present political, economic, social, technological, legal, and environmental data which is aimed at (i) the identification of internal and external forces that may influence the organisation's performance and choice of strategies and (ii) an assessment of the organisation's current and future strengths, weaknesses, opportunities and threats [115]. A situational analysis is most often done through the combined use of both the PESTLE and SWOT analysis tools. PESTLE, a mnemonic for political, economic, social, technological, legal and environmental, is a model developed by Francis J. Aguilar based on an earlier ETPS model to account for the external factors in a business [116]. PESTLE has been used for energy policy studies to determine obstacles and external factors

affecting policy instruments [117]. SWOT, a mnemonic for strengths, weaknesses, opportunities and threats, is a model introduced by Urick and Orr based on the SOFT work of Albert Humphrey for the identification of internal factors in a business [118].

4.4.2 Methodology for Ontology Construction

Ontology construction is the act of constructing an ontology taxonomy with an *is-a* hierarchy of concepts with attributes, values and relations. The information about these classes and their relations to each other, as well as constraints on attribute values for each class, are captured through axioms [7].

The interest with ontology model construction stems from the consequences that occur as a result of its construction. An ontology is an approximation of reality and as such its accuracy is highly dependent on its construction methodology. As ontology construction is a cyclic process which continues to evolve as the ontology expands to represent a greater proportion of knowledge, its representation accuracy could change considerably. Further compounding this is the fact that ontology construction is a highly subjective work and very dependent on the subjective interpretation of the knowledge engineer or modeller [4]. Because of this, it is useful for an ontology to be constructed based on an evolved or established methodology.

For this framework, the proposed ontology construction methodology combines three construction methodologies that have been published in the literature and reviewed in Chapter 2. Table 4.3 illustrates how these three methodologies have been combined. Three methodologies are proposed in order to encompass the steps from the start to the end of the ontology construction as shown in the rows of Table 4.3 and from the broad to details of the ontology construction as shown in the columns of Table 4.3

Table 4.3. Summary of the ontology construction methodology used in this framework.

	Ontology Construction Detail Overall -----> Detail		
Construction Methodology from Literature	Ushold and King [54]	Cyc [43]	Ontology 101 [59]
Start -----> Finish Ontology Construction Steps	Identify Purpose		
	Construct	Manual Codification	Define Classes
			Construct Taxonomy
			Define Properties
			Assign Values
			Instantiate
		Semi Automatic Codification	
		Automatic Codification	
	Evaluate		
	Document		

The broad and overall construction steps of the ontology follow the methodology proposed by Ushold and King [54] which comprises four activities: (i) identify the purpose of the ontology, (ii) build the ontology, (iii) evaluate the ontology, and (iv) document the ontology. The detailed construction of the ontology follows the steps of the Cyc project [43] which has three phases: (i) manual codification of articles, pieces of information (e.g. common sense knowledge) and data by hand, (ii) semi-manual codification of new information using a combination of tools and by

hand, and (iii) automatic codification of information without human intervention.

The manual codification phase of the ontology is based on the steps outlined by Noy *et al.* [59] which has five steps: (i) define the classes, (ii) arrange the classes into a taxonomic hierarchy, (iii) define the properties and describe the allowed values for these properties, (iv) fill the values for the properties, and (v) create the knowledge domain by defining the instances in the classes, filling specific property value information and providing relations between the classes (using axioms). In addition, it is recommended that the constructed ontology should be easily understandable, maintainable, and modular as per the recommendations specified in [8].

An example of an ontology constructed by combining these three methodologies is available in Section 6.2.1 which describes the *Energy Policy Ontology*. This representation, which is part of the *Prototype Energy Modelling System*, (Chapter 6) uses manual codification for the energy policy knowledge domain. In addition, the prototype energy model also has a web crawler component (Section 5.2.4.I) within its support module that can assist in the semi-manual codification of new information. Furthermore, the support module has a data clustering component (Section 5.2.4.II) which allows for the automatic codification of information and data already endogenous to the system.

In the case of the *Prototype Energy Modelling System*, the purpose of the ontology is stated in Section 6.1, the construction details of the ontology are provided in Section 6.2, and the application of the ontology is presented in Section 6.3.

4.4.3 Integration of Both Qualitative and Quantitative Data for Evaluation Within a Numerical Environment

As stated Section 4.2, a requirement of this framework is to develop a program that simulates the *understand* and *evaluate* tasks of the modeller. To support this requirement, Section 4.4.1 addressed how the pertinent knowledge for codification can be identified while Section 4.4.2 addressed how the identified knowledge can be arranged into an ontological taxonomy.

In this section, the issue of how to link, relate or integrate the qualitative data with quantitative data and vice versa in the knowledge domain is addressed. This function is a necessary part of the framework as the understanding and evaluating of both qualitative and quantitative data is done by the modeller in the *understand* and *evaluate* phase. In particular, data in the policy knowledge domain will be of qualitative and quantitative types while inputs, in the form of criteria and parameters from the policy makers are often qualitative rather than quantitative.

Qualitative (text) and quantitative (numerals) data and inputs are two different ways of describing objects in the world. However, while their objective, which is to describe objects, is similar, their integration into a single system for evaluation is difficult because of their different natures and operators. The nature of qualitative data as alphabets use a different set of operator rules for addition; alphabets are added together to form words which in turn are added together to form sentences using grammatical rules. On the other hand, the nature of quantitative data as numbers uses a different set of operator rules for addition, subtraction, multiplication and division; numbers are added, subtracted, multiplied and divided using mathematical rules.

Several methods where both qualitative and quantitative data are integrated within a single system exist. MCDA for decision making and fuzzy logic for control systems are but two examples of such integration.

One of the methods to quantitatively define qualitative data, parameters and criteria is centred on multicriteria decision analysis (MCDA). In MCDA studies, the researcher forms a quantitative and qualitative conclusion through the use of decision analysis tools such as AHP or ELECTRE. AHP uses pairwise comparisons along with a semantic and ratio scale to assess the decision maker's preference [119] while ELECTRE uses an outranking approach that tests comparisons between different alternatives according to several criteria [120].

Other methods available to quantitatively define qualitative parameters include lexical analysis [121] and quasi-statistics [122]. Lexical analysis consists of the calculation of word frequencies in a body of text in order to identify key words or themes of interest. In an article, the key words or themes could have a high frequency of occurrence and hence its relative importance can be numerically determined [88]. Quasi-statistics, a technique advocated by Becker and Geer [89], measures the frequency and distribution of phenomena that are derived from qualitative field data. For further details and examples of lexical analysis and quasi-statistics, the reader is encouraged to read articles [121,122].

As an analytical method for computer based systems, quasi-statistics offer several distinct advantages. First, quasi-statistics can be used to help identify major problems, concepts or indicators in field data. Second, quasi-statistics can be used to evaluate the plausibility of tentative hypotheses. Third, quasi-statistics can be combined with lexical analysis to provide automated processing of information and data. It is therefore unsurprising that quasi-statistics is a popular method for computer based qualitative analysis. Such computer programs often entail a database consisting of relatively short text items collected via a structured protocol in a format suitable for lexical analysis [121,122].

To develop a method for the integration of qualitative data, criteria and

parameters for numerical analysis, this framework plans to utilise, where possible, current methods of qualitative and quantitative analysis. Thus this framework recommends that emphasis be given to the manual codification of MCDA literature which contains the numerical equivalent of qualitative data. In addition, this framework proposes the use of automatic codification of information endogenous to the system using data clustering to establish the relationship between qualitative information and quantitative data. Furthermore, the framework proposes, although not implemented, the use of semi-automatic codification of qualitative data using the web crawler. The web crawler could be equipped with a lexical analysis and quasi-statistics subroutine in order to analyse the literature information available on the web. For this project however, this 'web crawler semi-automatic codification' path (see path 4 below) has not been implemented as the required lexical analysis and quasi-statistics software are already available. In addition, since the area of automatic data codification and ontology construction is being conducted by other researchers as reviewed in Section 4.4.3.III, the better option would be to incorporate the results of such research at some point in the future.

Figure 4.2 illustrates the data codification paths proposed in this framework. Four codification paths have been identified to accommodate the different categories of data to be codified (i.e. text, numerals, MCDA). These paths are:

- I. a manual then automatic codification path for quantitative data (path 1)
- II. a manual then automatic codification path for MCDA data (path 2)
- III. a manual then automatic codification path for qualitative data (path 3)
- IV. a semi-automatic then automatic codification path for qualitative data (path 4)

The manual then automatic codification paths have been implemented in the *Prototype Energy Modelling System* described in Chapter 6. The semi-automatic codification path using lexical analysis and quasi-statistics (in grey) has not been

implemented.

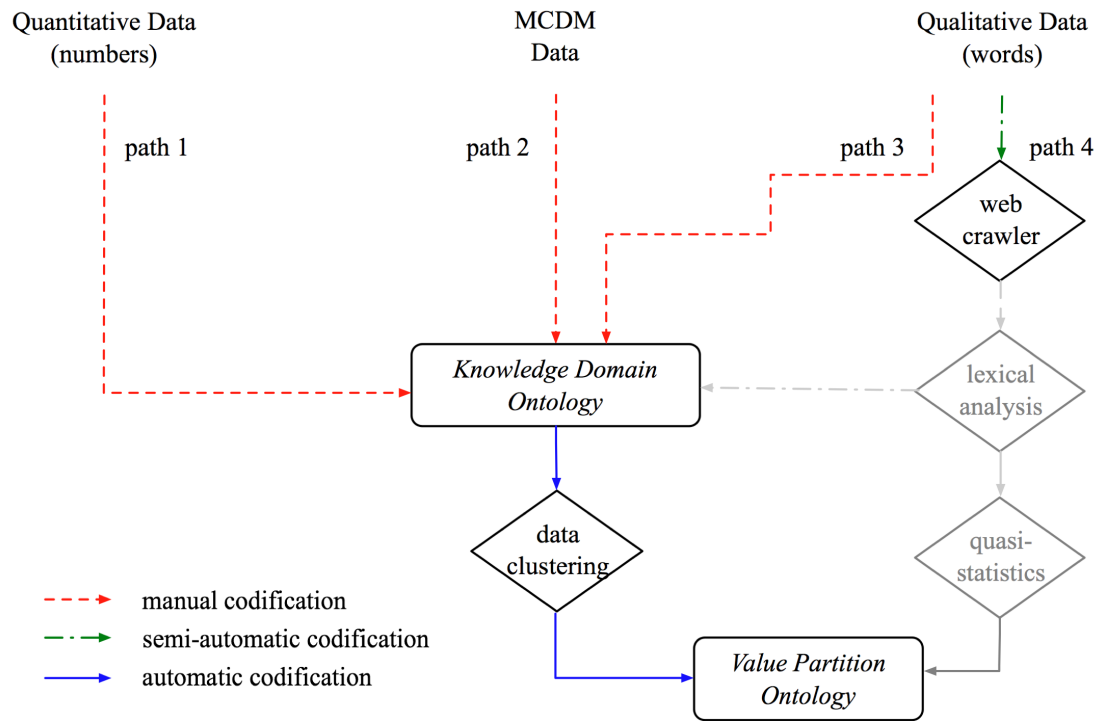


Figure 4.2. Data codification paths proposed in this framework.

I. Codification Path for Quantitative Data (Path 1)

Quantitative data (numbers) from the literature are manually codified into the *Knowledge Domain Ontology*. This manual codification involves association of the data with a property of a specific instance in the *Knowledge Domain Ontology* (Section 5.2.1.1). Subsequently the data for the similar property for instances of a similar class are compared to each other and clustered using the Least Euclidean Method (see Section 5.2.4 for an explanation on data clustering and Least Euclidean Method) and into the categories of *none*, *low*, *medium* and *high*. These categories, which can be expanded or reduced depending on the amount of data to be categorised,

provide a measure of qualitative equivalence and are automatically codified into the *Value Partition Ontology* (Section 5.2.1.2). Table 4.4 provides an example of the data, property, instance and classes that are compared and stored.

Table 4.4. Examples of the ontology terms used in this thesis.

Terms	Example
Data	500MW, 40MW
Property	Electricity Generation
Instance	Coal Power Plant, Wind Power Plant
Class	Electricity Generation Plant

II. Codification Path for MCDA Data (Path 2)

MCDA data from the literature is manually codified into the *Knowledge Domain Ontology*. Similar to codification path 1 for quantitative data, this manual codification involves association of the data with a property of a specific instance in the ontology. Subsequently, data clustering is done and the categorised results are codified into the *Value Partition Ontology*.

III. Codification Path for Qualitative Data (Path 3 and 4)

Qualitative data (text) from the literature or web can be either manually or semi-automatically codified into the *Knowledge Domain Ontology*. The manual codification path involves association of the data with a property of a specific instance. This association is determined by the programmer or subject

matter expert who has been assigned to codify the data. Subsequently, the data regarding the similar property for instances of a similar class are compared to each other and clustered using by the programmer using his or her best judgement. The results are then manually codified into the *Value Partition Ontology*. Computerised codification using the data clustering proposed in Section 5.2.4. is not possible as the data is in the form of text.

It is also possible to implement a semi-automatic codification path for qualitative data. Significant research in this area using a combination of a web crawler, text processing software, and learning algorithms are being conducted by researchers such as Maedche, Celjaska, and Balakrishna. Maedche and Staab describes a system which uses an association rule algorithm to detect relations between concepts from text [123]. The concepts are automatically codified into the ontology at the appropriate level of abstraction. Celjaska and Vargas-Vera describes a system which can semi-automatically populate an ontology based on supervised learning. The system uses a natural language processor, a dictionary, and an information extraction tool [124]. Balakrishna *et al.* discusses a general methodology to semi-automatically create domain ontologies from textual resources [125].

For the *Prototype Energy Modelling System*, a web crawler has been created which can be used as an interface for incorporation of the technologies and methods discussed in the literature. However, the integration of such technologies remains an exercise for a future research team.

A detailed explanation of the *Knowledge Domain Ontology* is provided in Section 5.2.1.1 while examples for the prototype modelling systems is illustrated in Sections 6.2.1. A detailed explanation of the *Value Partition Ontology* is provided in

Section 5.2.1.2 with examples illustrated in Sections 6.2.2. Further information on the web crawler, as implemented in the *Prototype Energy Modelling System* is available in Section 5.2.4.I.

4.4.4 Underlying Mathematical Approach

This subsection discusses the mathematical approach for the framework. Several issues were considered in the development of the mathematical approach. These are:

- I. Identification of the objective of the system
- II. Selection of the mathematical technique which supports the objective
- III. Selection of the mathematical kernel
- IV. Application considerations for the selected mathematical technique

I. Identification of the Objective of the System

The objective of the system is to provide a normative decision analysis method. In practice this involves determining the action that best achieves a desired goal or objective. In mathematical terms this involves seeking the minimisation or maximisation of a function by selecting values within a permissible set.

II. Selection of the Mathematical Technique which supports the Objective

Several mathematical methods are available to support normative economic decision analysis. Stochastic programming, linear programming, weighting objectives and goal programming are examples of available techniques. Linear programming is the method that is currently used in the

type of energy models that this project intends to extend. (see Section 3.2.4)
This framework therefore proposes the use of linear programming.

Linear programming itself consists of an objective function and a set of constraints. In decision terms, the objective function represents the effects (either maximising or minimising) of all available actions on the optimisation of the value of the object while the constraints reflect the limitations of the available actions. While linear programming is often used for the optimisation of a single objective function, multiple objective function optimisation is also possible.

III. Selection of the Linear Programming Kernel

To implement a linear programming kernel, several methods are available. The three most popular methods are:

- Unstructured. An unstructured approach uses a computer programming language (e.g. FORTRAN, C or Java) to combine the model and the data which produces either an input file for a commercial solver or a custom solution algorithm within the overall program.
- Table or spreadsheet. A table or spreadsheet approach uses templates within a spreadsheet program to provide a mathematical structure for data input and solution output.
- Algebraic Modelling. An algebraic modelling approach uses a special languages (e.g. GAMS [126] or AMPL [127]) to implement mathematical optimisation using a mathematical engine.

In this framework, the unstructured approach using an 'off the shelf' mathematical kernel such as Mathematica [128,129,130], Maple [130,131] or Matlab [132] is preferred. These kernels are of a proven design, provide

reliable numerical solutions and offer good computational performance. In addition, these kernels allow program access using the Java computer language. This is a critical factor as Java is also the language for access to other components of the *Optimisation Modelling System* (i.e. the logic reasoners and the OWL Ontology). The ability to use Java as the single platform language greatly simplifies the integration between the mathematical kernel, logic reasoner and ontology representation.

IV. Application considerations for the selected Mathematical Technique

The ability of the linear programming technique to find either the minima or maxima value is a feature that requires careful consideration when applied in the context of policy analysis. Policy analysis often requires a trade-off between the quantifiable benefits (e.g. economic, environment) and the non quantifiable benefits (e.g. politics, social). Hence, while the value from linear programming is mathematically most optimum, it may not be the preferred policy solution. An example of this is when a cost centric approach is taken for an energy policy analysis related to electricity generation mixes. The linear programming solution (where cost minimisation is the single objective function) will maximise utilisation of the cheapest electricity generation technology. In reality, the preferred generation mix could be one that is more expensive but also more politically acceptable and socially desirable.

To facilitate the need for trade offs and flexibility in optimisation of the objective function, this framework proposes a two step calculation approach. In the first step, a pure linear programming solution (optimum) is obtained from a mathematical expression set consisting of an objective function and a

set of selected constraints. The optimum from this step represents ideal conditions. In the second step, a solution is obtained for a system of inequalities consisting of the set of complete constraints, a modified objective function which uses the optimum from the first step and a mathematical expression which reflects the objectives of the user. A feature of this second step is the option (in Java) to control the magnitude of deviation from the optimum found in the first step. This allows the user to determine the level of compromise the system should consider (i.e. what is the "willingness to pay" of society in excess of the cheapest solution). In addition, this second step contains a stochastic option which is akin to a Monte Carlo type optimisation approach. Further details of this two step approach is outlined in Section 5.2.3.

For completeness, it is noted that the principle mathematical technique in both steps is linear programming. This framework differentiates between the linear programming and system of inequalities steps only because of (i) the different Mathematica commands used in the first and second steps and (ii) the method to allow for control of the magnitude of deviation from the optimum step has been implemented in Java.

In conclusion, linear programming is the proposed mathematical technique for use in the *Optimisation Modelling System* because of its established use in energy models. In addition, the unstructured approach using an 'off the shelf' mathematical kernel has been selected because it allows the integration of all components of the *Optimisation Modelling System* using a single computer language. To address the limitations of linear programming, a two step calculation approach will be employed. The details of this entire module called the *Linear Programming Module* are discussed in the next chapter.

4.5 Summary

This Chapter has discussed the ground work which forms the foundation of the framework for development of an *Optimisation Modelling System*. The background of the problem that the framework intends to address and the basic requirements that must be fulfilled to address the problem has been covered. In the next section, the details of the framework is presented.

Chapter 5 The Framework

As discussed in Chapter 1, to simulate the role of the modeller, the system must offer three general capabilities. First, the system must codify the knowledge associated with the domain. Second, the system must utilise the qualitative and quantitative data contained within the knowledge base in order to achieve a fair translation of the policy maker's intentions. Third, the system must formulate, as part of the model, a set of mathematical expressions that is representative of the policy maker's inputs.

In addition, in Chapter 4, these capabilities were refined into five specific computer functionalities. The technologies and techniques which would fulfil these functionalities were then identified. Subsequently, some design methods and guidelines that were necessary to facilitate the implementation of these technologies were discussed.

In this chapter, the architecture of the framework is discussed. Section 5.1 specifies the scope and specifications of the framework. Section 5.2 discusses the implementation details of the framework using the system architecture of the software that embodies the framework as a blueprint. Section 5.3 provides a brief summary of the steps required to develop the *Optimisation Modelling System* envisioned in this framework. Section 5.4 offers conclusions to the chapter.

The main contribution of the methodology is a new framework for the development of an *Optimisation Modelling System*. The framework combines ontologies, engineering models, linear programming and description logic inference to expand the boundaries of optimisation modelling to enable the consideration of both qualitative and quantitative input and data for the automatic formulation of mathematical expressions. While it is arguable that models with such features are

already in use, these models are produced manually.

5.1 Specifications of the Framework

The scope of this framework is to outline the development steps for an *Optimisation Modelling System* that accepts and stores both qualitative and quantitative inputs and data from the user (policy maker). The user inputs (policy goals) are then converted into their mathematical equivalent for numerical optimisation to determine an optimum solution.

The specification of the framework is to develop an *Optimisation Modelling System* which offers the following features:

- An easy to use modelling system which places an emphasis on the main drivers (e.g. factors or information) that form the scenarios. These drivers may be either qualitative or quantitative. The target user of the program is the policy maker.
- An ontology-based semantic representation to store the knowledge domain of the policy in question. The representation shall have data storage and retrieval with logic-based knowledge inference capabilities.
- The ability to infer both quantitative and qualitative data from the knowledge base and translate this inference into a set of mathematical expressions representative of the intentions of the user. An example of this is a set of linear programming expressions which optimises an electricity generation mix (variables) based on the minimisation of production costs (objective function) within the constraints imposed by the Kyoto Protocols (scenario constraints).
- The ability to solve this set of mathematical expressions and provide an output.

5.2 Details of the Framework

This section explains the details of the framework necessary to develop the *Optimisation Modelling System*. It begins with an explanation of the basic parameters that are required to define the capabilities of the prototype model (i.e. *Prototype Energy Modelling System*). Then a detailed description of the framework is provided through a discussion of the five main components of the *Optimisation Modelling System*.

Development of the *Optimisation Modelling System* starts with a specification of both the modelling system and the model that is to be produced:

1. Intention or Purpose. The purpose of both the modelling system and the created model to form the basis for its specifications. For example, the purpose of the *Prototype Energy Modelling System* is to automatically create an *Energy Model* which reflects the inputs from the user while the purpose of the *Energy Model* is to optimise for the lowest cost of electricity supply that meets demand.
2. Scope or Coverage. The scope defines the overall performance and breadth of model created by the modelling system. Examples of scope would be geographical, sectoral or temporal. For the *Prototype Energy Modelling System*, the created *Energy Model* has a country based geographical scope.
3. Inputs and Outputs. The inputs and outputs of the modelling system, which are the defined boundaries of the programs can be either qualitative or quantitative. They are, for the inputs, the main factors for optimisation and for the outputs, the results of the optimisation. In the *Prototype Energy Modelling System* for example, the inputs should be the energy policy goals while the output should be mathematical expressions which form an *Energy Model*. This *Energy Model* accepts as inputs the scenario parameters and outputs an electricity generation mix reflective of the goals. Figure 6.2 shows an example of the energy policy

goals and scenario parameters inputs while Section 6.3.2 and Section 6.3.3 provide examples of the mathematical expressions and generation mix outputs.

4. **Data Requirements.** The data requirements and assumptions of the program are dictated by the purpose, scope and type of model. For the *Prototype Energy Modelling System*, an example of the required data is the energy policy knowledge domain. For the *Energy Model*, examples of the required data include energy demand, population size and expected economic growth.

To develop a modelling system with the above specifications, this framework proposes a five module configuration, as illustrated in Figure 5.1:

1. *A Semantic Representation Module.* This module stores information and data related to both the knowledge domain (core knowledge such as energy policy) and the optimisation domain (application knowledge) using ontologies and mathematical models. In addition, this component is supported by a data clustering software for data classification and a web crawler software for knowledge domain acquisition. See Section 6.3 for an example.
2. *A Knowledge Inference and Expression Builder module.* This module facilitates the extraction of knowledge from the ontologies through the use of logic reasoners. The extracted knowledge is used to automatically formulate a set of mathematical expressions. See Section 6.3.1 and 6.3.2 for examples of mathematical expressions.
3. *A Linear Programming Module.* This module solves the resulting set of mathematical expressions. Although multiple solution methods are available, a combination of linear programming and a system of inequalities solver is the method that was selected. See Section 6.3.3 for examples of optimised solutions
4. *A Graphical User Interface (GUI).* This module allows for user inputs (e.g. scenario parameters, policy goals) and system outputs. See Figure 6.2 for an

example.

5. A *Support Module*. This module contains support programs such as data clustering algorithms, web crawlers and graphing subroutines.

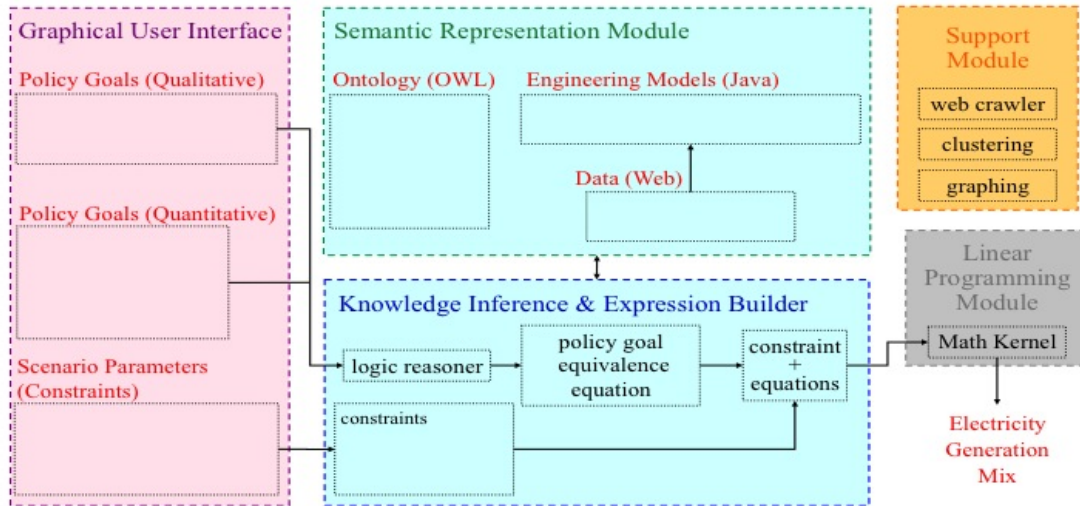


Figure 5.1. System architecture of the *Optimisation Modelling System*.

The design of this system architecture is founded on the principles of a clear separation of functions and enhanced modularity. Component modularity allows for the modification, enhancement and replacement of the modules with minimal impact on the system. It allows for the rapid adoption of newer versions of the mathematical kernels (e.g. Mathematica [128,129,130], Maple [130,131], Matlab [132]) and logic reasoners (e.g. FACT++ [133], Pellet [134]). The separation of modules by function is a widely recommended software design principle, especially for conventional intelligent systems, where the knowledge base contains the domain knowledge and is separate from the inference engine or reasoner [15].

The next subsections of this chapter present the detailed steps for constructing the modules in the framework. They are presented through an explanation of the main

features of each module.

5.2.1 Semantic Representation Module

This framework proposes a semantic representation module which consists of four ontology taxonomies with links to engineering models. Representation of the core knowledge is formed by two ontologies with support from engineering models while representation of the application knowledge is formed by two ontologies.

The ontologies which represent the core knowledge are:

1. *Knowledge Domain Ontology*. This ontology represents the knowledge domain of the policy in question (e.g. energy policy) and is supported by engineering models (e.g. Brayton cycle). Its function is to enable the program to simulate the *understand* task of the modeller. Section 5.2.1.1. details the top level hierarchy of this ontology while Section 5.2.1.5 details the engineering models.
2. *Value Partition Ontology*. This ontology stores a representation of the results from comparisons between data (e.g. 800 MW, 5MW) associated with instances (e.g. coal power plant, wind power plant) in the knowledge domain for a particular property (e.g. electricity generation capacity). For example, the electricity generation capacity of coal power plants, when compared to wind power plants is higher (higher is the comparison result). Its function is to facilitate conversion between qualitative (e.g. high, low) and quantitative (e.g. 800MW, 5MW) data in support of the program's capability to simulate the *understand* task of the modeller. Section 5.2.1.2 discusses the concept of this ontology in detail.

The ontologies which represent the application knowledge are:

1. *Scenarios Ontology*. This ontology stores the knowledge (e.g. advancement of clean technologies) used to develop plausible scenarios (e.g. an environmentally

conscious future). Its function is to enable the program to simulate the *evaluate*, *relate* and *convert* tasks of the modeller. Axioms within the ontology simulate representations of policy instruments (e.g. clean technology subsidies) which through description logic inference using a reasoner, are used to infer the policy variables (e.g. clean coal installations). Section 5.2.1.3 discusses the concept of this ontology in greater detail while Section 5.2.2 explains how the inferred knowledge is used to formulate mathematical expressions representative of the intentions of the policy maker.

2. *Equation Construction Ontology*. This ontology stores the axioms necessary for the automatic formulation of mathematical expressions such as the objective function and constraints. Its function is to enable the program to simulate the *modify* task of the modeller. Section 5.2.1.4 discusses the concept of this ontology while Section 5.2.2 explains how the inferred results are used to formulate both the objective function and constraints for optimisation.

Figure 5.2 illustrates, in no particular sequence, the ontologies and engineering models which form this module. As illustrated, these ontology taxonomies are connected to the *Knowledge Domain Ontology* via object properties while the engineering models are connected using Java.

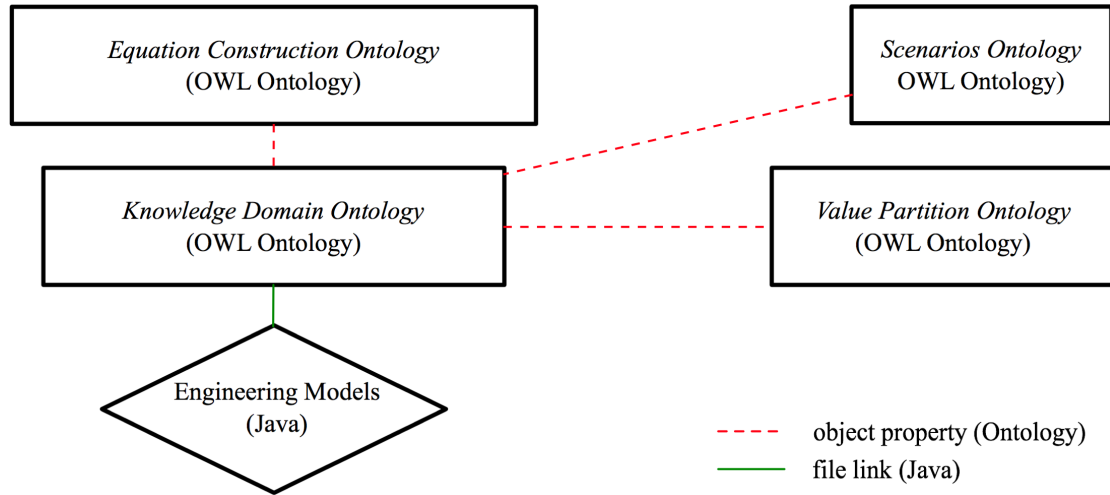


Figure 5.2. Relations between the Ontologies and Engineering Models in the *Semantic Representation Module*.

This codification arrangement allows for the separation between the core and application knowledge. In principle, it is possible to replace both the *Scenarios Ontology* and *Equation Construction Ontology* with description logic queries; this framework however, recommends that ontologies in OWL be used in-lieu of description logic queries embedded in Java. Embedded Java code would be cumbersome to implement and even more difficult to modify if the required queries change. In comparison, queries stored as axioms within ontologies are easily followed by the program user, simple to implement and easy to modify if required.

The ontologies in this module, when combined with the other modules, allow the system to provide qualitative and quantitative evaluation and facilitate inference for automatic expression formulation without interference from the user. The next subsections discuss the development details of these four ontologies (Sections 5.2.1.1 to 5.2.1.4) and engineering model (Section 5.2.1.5).

5.2.1.1 Knowledge Domain Ontology

This framework proposes that this ontology represents the semantics of the knowledge domain in question. Examples of such knowledge domains is that of energy policy.

To represent this knowledge, this framework proposes that the representation of the world be divided into four classes based on concepts classified according to two principles: (i) whether the concepts are either created or attributable to humans or whether the concepts exist originally in nature and (ii) whether the concepts are abstract or physical. Based on the combination of these two principles, four top-level classes are defined as shown in Table 5.1.

Table 5.1. Top-level classes and the concepts represented in the ontology.

Class	Concept Represented
<i>natural_Laws</i>	Pre-existing in nature and of abstract type (e.g. combustion process)
<i>natural_Resources</i>	Pre-existing in nature and of physical type (e.g. water, air)
<i>manmade_Concepts</i>	Attributable to humans and of abstract type (e.g. politics)
<i>manmade_Objects</i>	Attributable to humans and of physical type (e.g. gas power generation plant)

To complement these top-level classes, this framework proposes a set of object properties to relate these four main classes to each other. Table 5.2 lists these object properties while Figure 5.3 illustrates these relations. An example of the *Knowledge Domain Ontology* is presented in Section 6.2.1 for energy policy.

Table 5.2. Relations between the top-level classes in the ontology.

Class Relation	Object Property (Selected Example)	Concept Represented (Selected Example)
<i>manmade_Objects</i> to <i>natural_Resources</i>	<i>carries</i>	Represents the interaction between machines and resources (e.g. lorry carries coal)
<i>manmade_Objects</i> to <i>natural_Laws</i>	<i>utilises</i>	Represents the utilisation of thermodynamic laws by machines (e.g. boiler utilises combustion)
<i>manmade_Objects</i> to <i>manmade_Concepts</i>	<i>hasState</i>	Represents the fact that machines have a human defined state (e.g. boiler hasState cost)
<i>manmade_Concepts</i> to <i>natural_Resources</i>	<i>has_Value</i>	Represents the fact that human defined states have impacts on resources (e.g. cost hasValue money)
<i>manmade_Concepts</i> to <i>natural_Laws</i>	<i>encourages</i>	Represents the fact that human defined concepts effect the utilisation of engineering processes (e.g. low emission laws discourages combustion)
<i>natural_Laws</i> to <i>natural_Resources</i>	<i>consumes</i>	Represents the utilisation of natural resources by engineering processes (e.g. combustion consumes fuel)

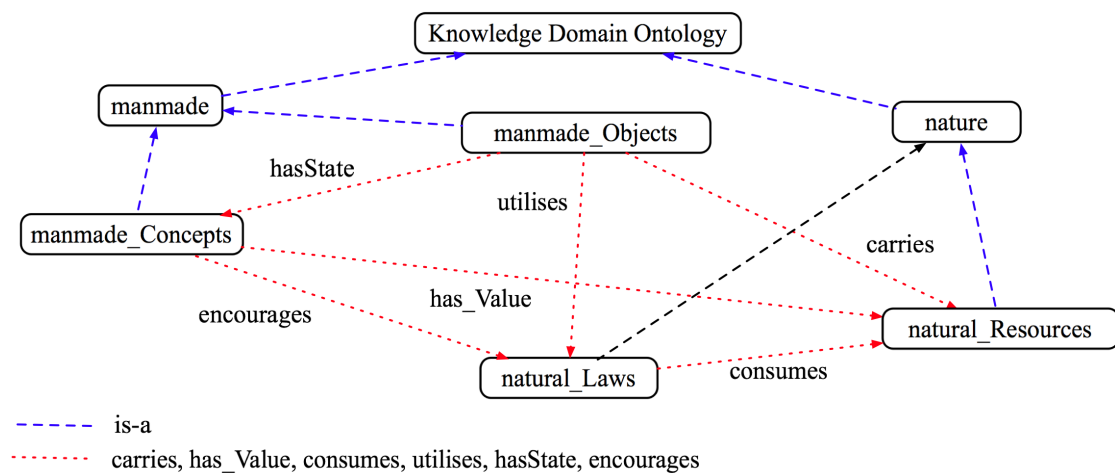


Figure 5.3. Relations between the top level classes in the ontology.

5.2.1.2 Value Partition Ontology

This framework proposes that this ontology represents the assessments from comparisons between data (e.g. 500MW, 40MW) associated with instances (e.g. coal power plant, wind power plant) in the knowledge domain for a particular property (e.g. electricity generation capacity). This comparison of data associated with combinations of instances of a particular class (e.g. electricity generation plants) for a particular criterion (e.g. electricity generation capacity) is useful as it provides a qualitative assessment (e.g. high, low) of quantitative data. Thus, for example, in a comparison between the coal and wind power plants, the qualitative assessment for coal would be *high* while for wind would be *low* for the comparison criteria of electricity generation capacity. The *high* and *low* assessments are stored in the *Value Partition Ontology*. Table 5.3 summarises this example

Table 5.3. Example of assessment from the comparison of generation capacity for electricity generation plants.

<i>Knowledge Domain Ontology</i>				<i>Value Partition Ontology</i>
Class	Instance	Property / Criteria	Quantitative Data	Qualitative Assessment
<i>electricity generation plant</i>	<i>coal power plant</i>	<i>electricity generation capacity</i>	500MW	High
	<i>wind power plant</i>		40MW	Low

The assessment stored in this ontology supports its function which is to facilitate conversion between qualitative and quantitative data in support of the program ability to simulate the *understand* task of the modeller. This conversion capability is necessary given that while inputs from the policy maker can be in both

qualitative and quantitative terms, inputs into the modelling system can only be in quantitative terms.

The assessments stored in this ontology are from the comparisons made in the data clustering phase of the codification process for data as mentioned in Section 4.4.3 while the details for automatic clustering is described in Section 5.2.4. The data of the instances (e.g. coal power plant, wind power plant) for a particular property (e.g. capital cost) are clustered either automatically or manually in groups of low, medium or high. Table 5.4 illustrates an example of the clustered results for capital cost for electricity generation plants used in the *Prototype Energy Modelling System*.

Table 5.4. Capital cost clustering results for electricity generation plants.

<i>Knowledge Domain Ontology</i>		<i>Value Partition Ontology</i>		
Instance	Quantitative Data	Qualitative Assessment For Property		
electricity generation plant	Capital Cost¹ (GBP per MW)	Low	Medium	High
Coal	867	x		
Gas	580 - 584	x		
Oil	1022 - 1131	x		
Biomass	1853		x	
Nuclear	2683		x	
Wind	929		x	
Solar	52273			x
Hydroelectric	4000		x	
¹ Capital cost in this context is the incurred capital cost which is capital cost * 1/(plant capacity factor). For example, while the cost for a 1MW solar plant is 5750 GBP, the incurred capital cost to guarantee 1MW of supplied electricity from a solar power plant is 52273 GBP as a solar power plant has a plant capacity factor of 11%. Capital cost data taken from UK MARKAL Model Documentation [105]				

This ontology therefore represents (i) the qualitative categories for data clustering, (ii) the value ranges of the respective categories (e.g. low, medium, high), (iii) the results of the data clustering, and (iv) the association between the qualitative categories and the scenarios ontology. The qualitative categories used are related to politics, economics, social issues and the environment. More extensive examples of this *Value Partition Ontology* are available in Section 6.2.2.

5.2.1.3 Scenarios Ontology

This framework proposes this ontology to represent the main drivers for the future scenarios for which the system is intended to provide an optimised solution. It is an inherent assumption in the development of this system that the user always seeks to optimise the designated resource within the context of a scenario (refer to Chapter 7 for examples of the scenario used for the case study). These scenario drivers which frame the scenario are presented as a combination of policy goals and policy targets to the user as shown in Figure 6.2 which is the GUI for the *Prototype Energy Modelling System*.

A scenario driver is thus a combination of both the policy goal and policy target. For example, the minimisation of capital cost is one of the policy goals for energy policy. To convert this policy goal into a scenario driver the policy target (i.e. low, medium, high) must also be specified. The user therefore frames the scenario through selection of both the policy goal and the policy target. The need to reduce future CO₂ emissions, the requirement to minimise capital and production costs and the urgency to increase energy security are some examples of policy goals and targets that can be used to develop scenario drivers and subsequent scenarios.

The function of this ontology is to enable the system to formulate a mathematical expression called the *Technology Preference Expression*, which reflects the intention of the user as defined by the selected scenario drivers. The *Technology Preference Expression* is an inequality made up of policy variables that have been arranged in an order of preference to provide maximum support to the selected policy goals and targets. In the ontology, these scenario drivers are represented as axioms within the classes. These axioms allow the modelling system to determine the appropriate policy variables that can best support the policy goal and policy target. The policy variables or instances are inferred through a logic reasoner

from the *Knowledge Domain Ontology*. Table 5.5 illustrates, as an example, the selected results for the axiomatic inference if the significant (policy target) reduction of CO₂ emissions (policy goal) is defined as a scenario driver in the Figure 6.2 GUI. The axiom, which is programmed as a logic statement in the ontology, has been described in English for readability.

Table 5.5. Example of an axiomatic inference for a policy goal and target.

Policy Goal and Target	Axiom Description (in English Equivalent)	Policy Variable	Inferred
Significant reduction of CO ₂ emissions.	Which electricity generation plants can support a significant reduction in CO ₂ emissions?	Wind	Yes
		Coal	No
		Solar	Yes
		Hydroelectric	Yes
		Oil	No
		Gas	No
		Biomass	Yes
		Nuclear	Yes

It is important to remember that the scenario drivers are not representations of the policy instruments. Scenario drivers must be developed for a specific ontology based on its underlying logical relationship to the policy goal, policy target and policy instrument. Both the drivers and their effects are described logically using axioms. This use of logic also ensures plausibility, a key requirement of the scenarios.

Another key consideration in development of the scenario drivers is that of trade-offs which is a key aspect of policy development. The implementation of a

policy always results in winners and losers. A reason why optimisation programs are used in policy analysis is to determine the best trade-off in achieving the policy goals. As scenario drivers are based on policy goals, these drivers should also reflect this trade-off. In this ontology, the trade-offs are reflected at two levels: at the policy goal and policy target levels.

The scenario drivers should, where possible and within reason, contradict one another. Thus most desirable outcome for an optimisation problem is to optimise for a set of scenario drivers where each driver results in the inference of a different set of policy variables. This reflects the real world situation of trade-offs and allows for variation in the possible solutions which meets the overall sum of all of the policy goals and targets.

Examples of scenario drivers that contradict each other are that of CO₂ emissions and capital cost. As Figure 5.4 illustrates, the most desirable outcome (the area in the upper right quarter of the figure) would be for both the minimisation of CO₂ emissions and minimisation of capital costs. It is however impossible to achieve both goals with the same set of electricity generation technologies. A trade-off is therefore required between the two goals and an optimisation of the electricity generation mix is required to find the best compromise.

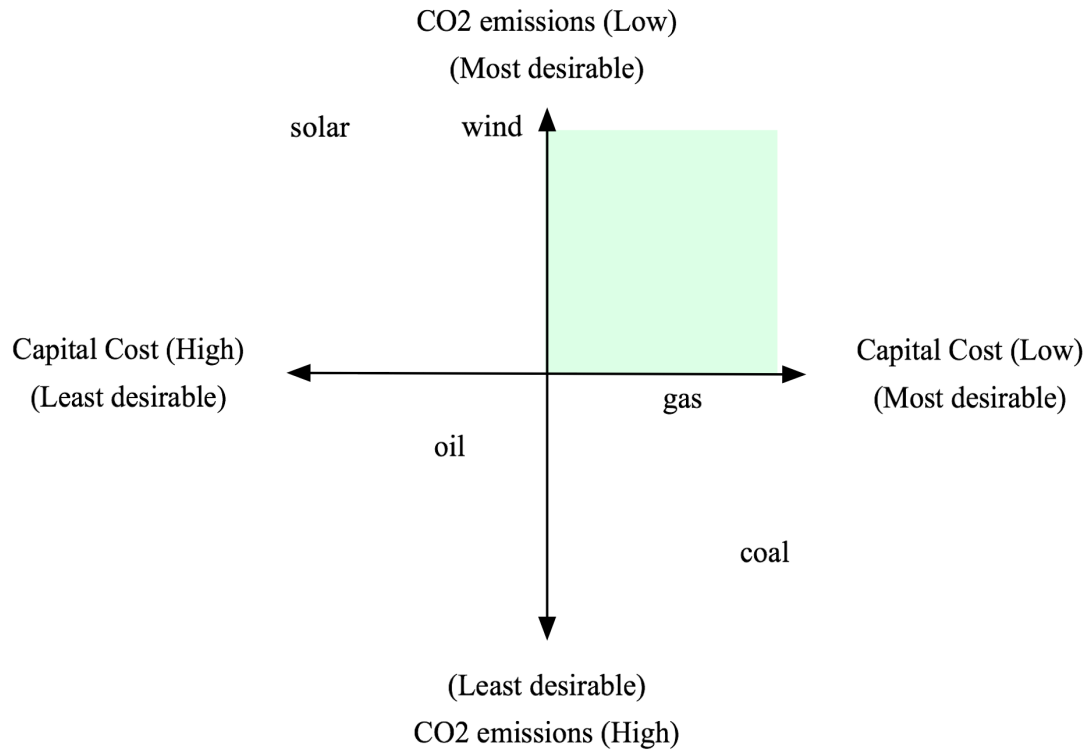


Figure 5.4. Capital cost and CO₂ emission positions of selected electricity generation technologies.

For ontology representation purposes, this framework proposes that the scenario drivers be categorised into the topics of politics, economics, social, technological, legal and the environment. In addition, it is recommended for ease of ontology readability, that each combination of policy goal and target be represented by an axiom. Hence, the *Prototype Energy Modelling System* has 27 axioms to represent the nine policy goals with their three policy targets.

As mentioned earlier, the function of this ontology is to enable the system to formulate a mathematical expression called the *Technology Preference Expression*. For each selected policy goal and target, the system will infer a set of policy variables which is then recorded. All of these policy variables are then summed. The variables with higher number of inferences are assumed to be more preferred by the user than

variables with lower number of inferences. The technology preference expression is then formulated based on this principle.

Further details of both the inference mechanism and its role in the formulation of the *Technology Preference Expression* are described in Section 5.2.2.II while Figure 5.7 illustrates the flow process of the formulation. An extensive set of *Technology Preference Expression* examples is available in Section 6.3.2.I.

5.2.1.4 Equation Construction Ontology

The function of this ontology is to allow for the automatic formulation of the objective function and the constraints which form the set of linear programming expressions and the set of system of inequalities for optimisation. The premise on which this ontology is based is that all variables, constants (multipliers) and values (production) in an equation or mathematical expression have a logical relationship.

For this ontology to work, it must codify the variables, constants, values and logical relationship of the expression. To ensure the completeness of the codification, this framework proposes a six step process: (i) the purpose of the expression is identified, (ii) the expression is broken down into its variables, constants (multipliers) and values (product), (iii) the logical basis of the expression is identified, (iv) an axiom to enable inference of the variables and identification of the constants is made, (v) the commands to either load from the ontology or calculate the constants and values are programmed and (vi) the commands to automatically synthesise the expression is programmed.

To better illustrate this six step process, an example of its application on Expression 5.2.1 is discussed.

$$\sum_{i=1}^n numPwr_i * CO2Pwr_i \leq CO2pCap * pop$$

where

numPwr = number of electricity generation plant (plant) (5.2.1)

i = electricity generation technology (e.g. coal)

CO2Pwr = CO₂ emissions per electricity generation plant (tons/plant)

CO2pCap = CO₂ emissions per capita allowance (tons/capita)

pop = population size (capita)

The purpose of Expression 5.2.1 is to protect the environment by limiting CO₂ emissions from the electricity generation sector (policy goal). The maximum allowable CO₂ emissions are derived from the total population multiplied by a CO₂ per capita value (policy target). The electricity generation technologies (policy variables) which must be included in this expression are identified by *i*. Bearing in mind that Expression 5.2.1. forms part of a set of constraints, the unknown variable that is to be solved is the number of electricity generation plants for each electricity generation technology. The constant is the amount of CO₂ emitted for each electricity generation plant. The value is the maximum allowable CO₂ emissions.

Table 5.6 summarises the results from application the first two steps of the six step process in Expression 5.2.1.

Table 5.6. Summary of results from the application of step one and step two of the six step process on Expression 5.2.1.

Steps		Results
Identify purpose of expression		Limit CO ₂ emissions
Break expression down	Variables	Number of electricity generation plants (<i>numPwr</i>) for each electricity generation technology (<i>i</i>)
	Constants	CO ₂ emissions per electricity generation plant (<i>CO2Pwr</i>) for each electricity generation technology (<i>i</i>)
	Value	Maximum allowable CO ₂ emissions (<i>CO2pCap * pop</i>)

For an electricity generation technology to be constrained by a CO₂ emissions limit, it must be employ a chemical process such as fossil fuel combustion which emits CO₂. This logical argument can be used as the basis of forming an axiom within the ontology. This axiom should also identify the constant necessary to calculate the amount of emitted CO₂. Figure 5.5 illustrates an example of this axiom (highlighted in green) which is part of the ontology contained within the *Prototype Energy Modelling System*.

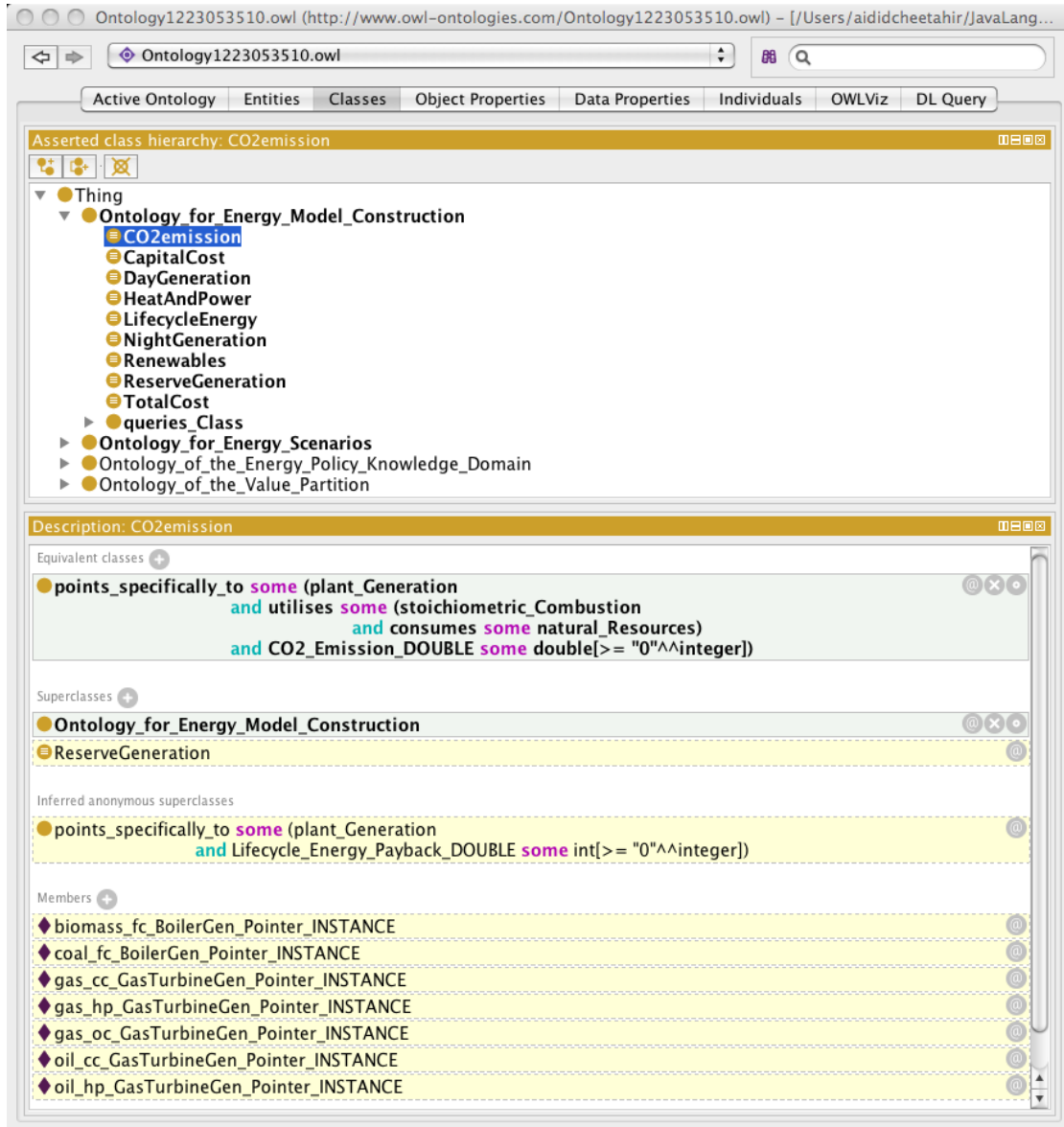


Figure 5.5. Protégé screen shot for the formulation of the CO₂ emissions constraint (Expression 4.2.1) in the *Energy Model Construction Ontology* of the *Prototype Energy Modelling System*.

In plain English, this axiom states that any electricity generation technology (*plant_Generation*) which uses the combustion process (*stoichiometric_Combustion*) that consumes fuel (*natural_Resources*) will emit a quantity of CO₂ which is greater than zero (*double*[$\geq "0"^\wedge\wedge\text{integer}$]). Figure 5.5 also shows the inference results of the

axiom. The electricity generation technologies that are applicable to Expression 5.2.1 are *biomass*, *coal*, *gas* and *oil*.

The fifth step involves the programming of instructions to load or calculate the constants and value. For this example, the CO₂ emissions per electricity generation plant are stored as data and linked to the respective generation technology using the data property *CO2_Emission_DOUBLE*. The value, on the other hand, is calculated during the optimisation phase of the calculations and therefore not stored in the ontology.

The last step of this six step process is the automatic synthesis of expression using standard computer commands which link the constants, variables and values together. For this example, the result of the synthesis is shown in Figure 6.15.

It must be noted that the computer commands for the fifth and sixth steps are not part of the ontology. The commands are instead a component of the *Knowledge Inference and Expression Builder Module*. Further details on that module is available in Section 5.2.2.

5.2.1.5 Engineering Models

The principle behind this module is predicated on the fact that ontologies alone are insufficient to provide a complete representation of a knowledge domain which needs to be expressed in mathematical terms. While ontologies provide excellent representation of concepts (e.g. text, words) and processes (e.g. business processes), they provide a limited representation of mathematical based processes (e.g. the material and energy balances representing chemical and mechanical processes). The capabilities of ontologies to represent mathematical models are mostly limited to numbers, cardinalities and value partitions.

The mathematical models in this module are intended to complete the

ontological representation of the knowledge domain. The models provide details of the process through the mathematical simulation of the processes themselves. The models represent the relevant energy and mass balances of the electricity generation processes in steady state conditions; real world losses are accounted for through coefficients while non steady state conditions are not considered. Simulation results from the models are stored in the ontology as data properties associated with the class representation of the respective processes. This data can be updated from within the *Optimisation Modelling System* by running new simulations as and when necessary.

The development of the engineering models can be broken down into several steps:

1. Identification of the classes and instances in the ontology which require models
2. Determination of the relevant mathematical expressions and engineering processes
3. Specification of the boundaries of the models and the assumptions of the processes
4. Implementation of the models using Java and use of the Mathematica Kernel for calculations
5. Testing of the models for accuracy and correctness

The completed models are indirectly linked to the ontological taxonomies through the use of text pointers. These file pointers are text annotations in the ontology class that contains the Java class (file) name of the engineering model. Java commands can be used to read the annotations and load the correct engineering model. A direct link from the files which contain the models to the ontology class or instance is not currently possible as no available Application Programming Interface (API) has been developed.

5.2.2 Knowledge Inference and Expression Builder Module

For a problem to be optimised, it must first be represented with a set of mathematical expressions. The set of expressions required are the objective function, the constraints and the *Technology Preference Expression*. A key requirement of this framework in order to simulate the *convert* task of the modeller is the automatic formulation of mathematical expressions based on the user inputs.

The main function of this module is to facilitate the logic inference of knowledge from the ontology and to use the inferred knowledge to develop expressions representative of the user's intentions. To achieve this function, this module has two components:

- Inference Engine. A Java program that uses an off the shelf description logic reasoner such as FACT++ [133], Pellet [134] or Hermit [135]. While, in principle, any reasoner can be used, this framework recommends the use of the FACT++ reasoner. FACT++, is an open source logic reasoner developed in the University of Manchester which provides extensive support for reasoning with instances, user-defined data-types and ontology debugging. FACT++ infers results from description logic based queries made to it. Its ability to interface with Java, Pellet and the OWL-API was a central criterion in its selection for this project.
- Equation Builder. A Java program that acts as a translator. It translates the inference results (e.g. policy variables) from the logic reasoner into expressions for use by the mathematics kernel of the linear programming module.

I. Formulation of the Objective Function and Constraints

This framework proposes the formulation of the objective function and the constraints through queries to the *Knowledge Domain Ontology* using axioms embedded in the *Equation Construction Ontology*. For both, the objective function (e.g. minimisation of capital cost) and constraints (e.g. required amount of generated electricity), the inferred knowledge (policy variables) are used as the variables (e.g. coal power plant) of the expression while the constants (e.g. electricity generation capacity) of the variables are loaded from the ontology by reading the literal (numbers) of the selected data property. The values are either loaded from the ontology, calculated using Java commands or inputted directly by the user. The equation builder constructs these expressions into a format compatible for use with the Linear Programming solver. For the objective function, the formulated expression can either be that of a minimum or maximum function. Figure 5.6 illustrates the process flow for the construction. Section 6.3 details the available objective functions in the *Prototype Energy Modelling System* while Figure 6.14 demonstrates the automatic formulation of the capital cost objective function.

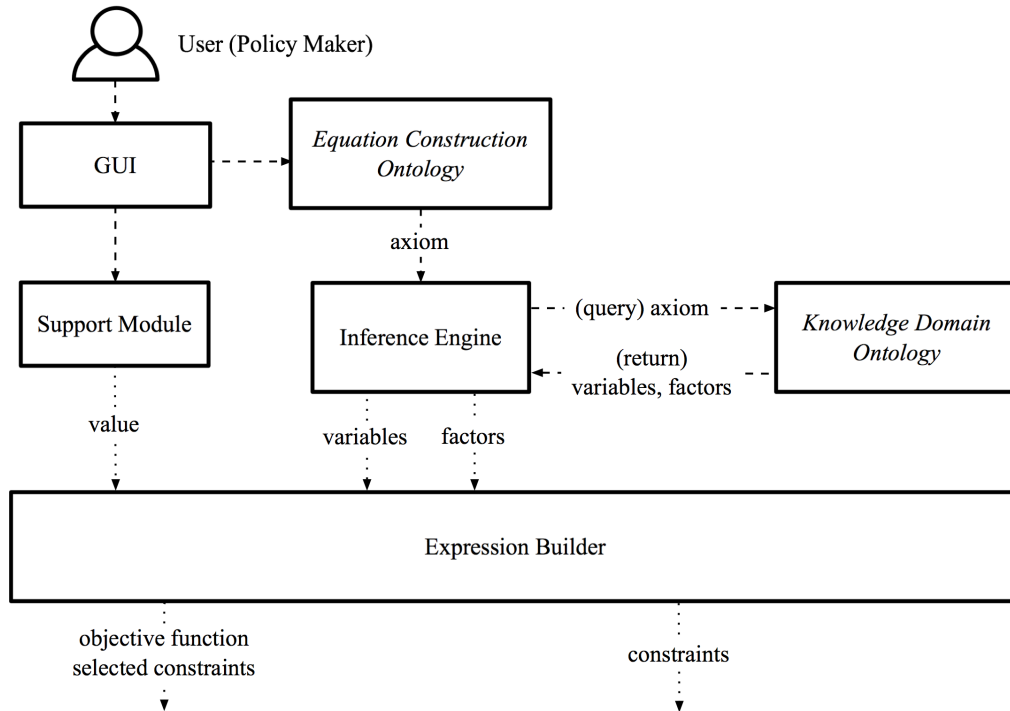


Figure 5.6. Process flow chart for inference and construction of the objective function and constraints.

The selected constraints which are used in the linear programming step of the optimisation (explained in Section 4.4.4.IV) consist of constraints that are exogenous to the optimisation and independent of the selected scenario drivers. For the *Prototype Energy Modelling System*, an example of a selected constraint is that of electricity demand due to the current population and industrialisation state. The complete set of constraints which are used in the system of inequalities generation step (explained in Section 4.4.4.IV) consists of all the constraints represented in the *Equation Construction Ontology*. Figure 5.6 illustrates the process flow for the construction. Section 6.3 details both the selected constraints and the complete set of constraints used in the *Prototype Energy Modelling System* while Figures 6.15 and 6.16 demonstrate

the automatic formulation of the supply and demand constraints.

II. Formulation of the Technology Preference Expression

This framework proposes the formulation of the *Technology Preference Expression* through the combined query of the *Knowledge Domain Ontology* and the *Value Partition Ontology* using axioms embedded in the *Scenarios Ontology*. The inferred knowledge (policy variables) is used to construct the *Technology Preference Expression* which reflects the scenario drivers selected by the user. Figure 5.7 illustrates the process flow for the construction.

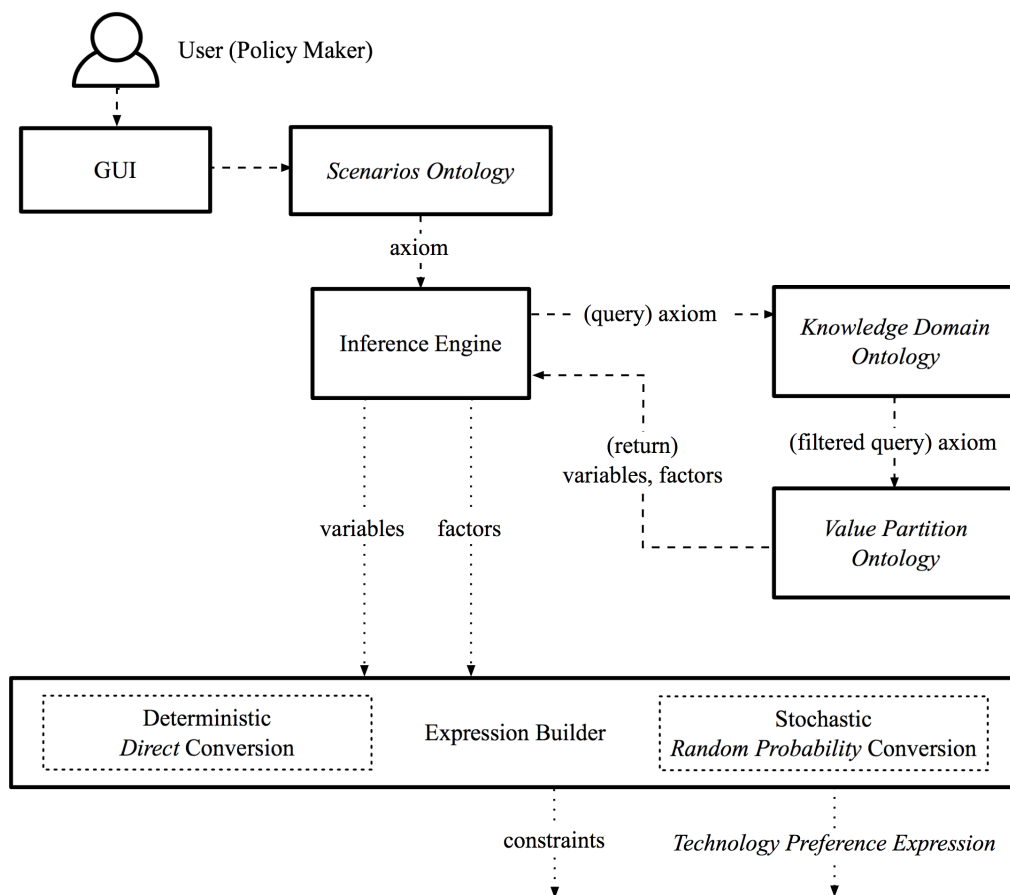


Figure 5.7. Process flow chart for the inference and construction of the *Technology Preference Expression*.

The principle for the construction of the *Technology Preference Expression* is based on the concept of quasi-statistics. The basic idea is that the more a variable is inferred (due to the aims declared by the user), the greater its preference or importance should be. A numerical summation is used to track the number of times a variable is inferred. These variables with the greatest number of occurrences will be placed at the head of a sorted list, while the variables with the lowest number of occurrences will be placed at the tail of the same sorted list. This sorted list is then translated into the *Technology Preference Expression*. Two formulation methods are available: *Direct* which is a deterministic process and *Random Probability* which is a stochastic process.

In the *Direct* formulation method, the equation builder constructs the mathematical inequality based strictly on the sorted list. The assumption used to construct this mathematical inequality is that the preference order for the variables is absolute.

In the *Random Probability* formulation method, the sorted list is used to generate a probability distribution curve. The size of the area for each variable under this probability curve is based on its order in the sorted list. Variables with a higher order will have a larger area. A random number generator is then used to generate a new set of random numbers and a function calculates the total number of numbers which are within each of the areas of the curve. Variables with the greatest number of numbers will be placed at the head of a new sorted list while variables with the lowest number of numbers will be placed at the tail of the new sorted list. This new sorted list is then converted into a mathematical equivalence.

Both the *Direct* and *Random Probability* formulation methods have

advantages. The *Direct* method allows the user to directly relate the decisions made when framing the scenarios to the final output of the program. The consistent and reproducible results are a critical aspect of optimisation using computer models. The *Random Probability* conversion allows for the user to introduce an element of uncertainty in the process. Given that in reality, the relationship between a policy goal, its target and the affected policy variables is not absolute, the conversion attempts to simulate this relationship using randomisation. This element of randomness highlights the fact that several combinations of policy variables could fulfil the same policy goals and targets. For example, the goal of CO₂ emissions reduction can be achieved using any combination of generation mix consisting of wind, solar, hydroelectric, biomass and nuclear. Indeed, in a democratic country with an open economy, the final generation mix is uncertain given the freedom of choice that people have. The *Random Probability* conversion simulates this uncertainty in the formulation of the *Technology Preference Expression* and as such allows the user to explore different solution options. Furthermore, as the *Random Probability* formulation method relies on several iterations, its solution is more 'robust' when compared to that of the *Direct* formulation method. In the *Prototype Energy Modelling System*, five sets of *Technology Preference Expression* are automatically formulated during the *Random Probability* formulation. In principle however, the number of sets to be formulated can be increased with additional programming.

On the other hand, both the *Direct* and *Random Probability* formulation methods have disadvantages. The reliance of the *Direct* formulation method on complete certainty can result in a situation where a small advantage for a particular electricity generation technology results in an

absolute advantage in mathematical terms which is not reflective of the actual intent of the policy maker. For example, if coal were to have one occurrence while gas were to have two occurrence, it would be a reasonable assumption to expect that coal would contribute 33% of total electricity generation. However, because the inequality states that gas must be higher than coal, it is possible in a LP solution for coal to contribute anywhere from 66% to 99% of the total electricity generation. With a *Random Probability* formulation method the inflexible nature inherent in both the *Direct* option and in the linear programming solutions is addressed. However, the *Random Probability* results are only useful if several iterations are executed (Monte Carlo simulations). Given that for each iteration, the *Technology Preference Expression* will change, the results of each iteration is not reproducible. Hence, it is an average of the results from hundreds of iterations that should be used. To complete such an optimisation however requires significant computational resources. Thus, for the *Prototype Energy Modelling System*, only five iterations are executed. To increase this number of iterations, additional programming is required.

For both formulation methods, an extensive set of examples of *Technology Preference Expression* based on the selected scenario drivers (energy policy goals) is available in Section 6.3.2. In addition, examples of the different solutions resulting from different *Technology Preference Expressions* are exhibited in Section 6.3.3, in particular in Figures 6.19 and 6.20.

5.2.3 Linear Programming Module

The function of this module is to solve the set of linear programming expressions and system of inequalities formulated by the inference engine and equation builder. As described in Section 4.4.4, a two step calculation process is employed to determine the optimum solution. This is necessary to address the fact that linear programming can converge to a local minima or maxima.

The module therefore is made up of two solvers, a *Linear Programming Solver* and a *System of Inequalities Solver* as illustrated in Figure 5.8. Both solvers utilise a mathematical kernel with an interface to interact with the other modules in the program. In the prototype modelling system developed using this framework, Mathematica has been selected as the mathematical kernel while Java is used as the interface language. Mathematica is a computational software program which supports procedural programming and can be used to solve linear equations [130]. Code written in the Java computer language interfaces with the Mathematica Kernel via an API provided by Wolfram Research.

The two step process flow of this module starts with the execution of the linear programming solver and ends with the complete execution of the system of inequalities solver. As explained in Section 4.4.4.III, both modules use linear programming as the principle mathematical method; differentiation is made in this framework because of the different software implementation methods in Java, different commands used in Mathematica and different constraints for the set of mathematical expressions.

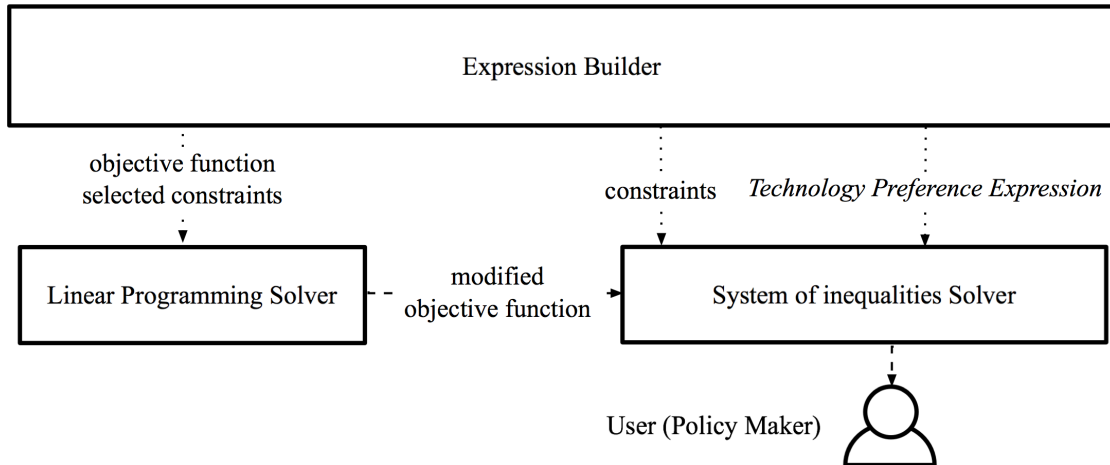


Figure 5.8. *Linear programming module* process flow.

I. Linear Programming Solver (First Step)

The function of this solver is to implement the first step of the calculation process. It uses a linear programming method to solve an objective function with selected constraints. The aim of this step is to obtain the optimum value of the objective function under ideal conditions. This optimum value is then used as an initial estimate for the second step. For example, in the optimisation of an electricity generation mix to meet demand with the minimisation of capital cost as the objective function, the optimum capital cost value will be that of the cheapest combination of available generation technologies which are in general fossil fuel based electricity generations. This optimum mix can be achieved because the minimisation of capital cost is the priority irrespective of other policy goals (e.g. CO₂ emissions reduction) or constraints (e.g. energy efficiency). Only required constraints (i.e. electricity demand) rather than optional constraints (e.g. energy efficiency) are selected.

Hence the optimum is achieved under ideal conditions. Section 6.3.1 lists the objective function and selected constraints used for the *Prototype Energy Modelling System*.

II. System of Inequalities Solver (Second Step)

The function of this solver is to implement the second step of the calculation process. It uses a system of inequalities solver to solve a system of inequalities which consist of constraints, a modified objective function and the *Technology Preference Expression*. The modified objective function comprises the objective function of the first step on the left hand side (LHS), an inequality operator, and the optimum value from the first step on the right hand side (RHS). This optimum value is used because it represents a good starting point from which to find a new optimum value for the system of inequalities. In addition, the required solution time for the new optimum can be reduced as the entire solution space does not need to be searched. Section 6.3.1 lists the modified objective function, *Technology Preference Expression* and constraints used for the *Prototype Energy Modelling System*.

A loop subroutine is executed to allow the system to search for a new optimum solution by incrementing or decrementing the RHS value of the modified objective function. If the objective function is minimisation (e.g. minimise costs), the loop increments while if the objective function is maximisation (e.g. maximise energy efficiency), the loop decrements.

This incrementing or decrementing loop is necessary as the new optimum value on the RHS, which originates from the optimum value found in the first step, is a value under 'ideal conditions' where not all of the constraints have been considered. When used as part of the second step, where the

complete set of constraints is considered, often, a feasible solution cannot be found unless some relaxation of this value is allowed. The magnitude of increment or decrement is adjustable by the user from the GUI.

Variants of this two step process flow, for the formulation of the *Technology Preference Expression* and optimisation of multiple objective functions, are also implemented in this framework.

- Formulation of *Technology Preference Expression*. As explained in Section 5.2.2.II, if the user selects a *Random Probability* conversion of the *Technology Preference Expression*, the second step will undergo five iterations in order to formulate five different *Technology Preference Expressions*.
- Optimisation of multiple objective functions. The user can select to optimise more than one policy goal and target. For the *Prototype Energy Modelling System*, the user can select to both minimise the financial cost (i.e. capital or total cost) and maximise the energy payback (i.e. life cycle energy payback) simultaneously. In this variant, the first step is executed twice: once for each of the objective functions in order to obtain the respective optimum values. The two modified objective functions (see Section 6.3.1.II, Expressions 6.4.6, 6.4.7 and 6.4.8) are incorporated into the system of inequalities in the second step. The second step loops and for each loop either increments (if the objective function is to minimise) or decrements (if the objective function is to maximise) both optima values until a feasible solution is found. The magnitude of increments and decrements for each loop, the number of loops before each increment or decrement and the optima which is to be incremented or decremented first can be adjusted by the user in Java.

For the *Prototype Energy Modelling System* (Chapter 6), the Mathematica commands 'LinearProgramming' and 'FindInstance' are used by the Linear Programming solver and System of Inequalities Solver for the first and second steps respectively. Java commands are used to export the expressions to and import the results from the mathematical kernel. Java commands are also used to implement both the first and second steps of the calculation. Examples of the optimisation from the linear programming module are exhibited in Section 6.3.3, in particular in Figures 6.17 and 6.18.

5.2.4 Support Module

The *Support Module* encompasses four components which are a *Web Crawler*, a *Clustering Subroutine*, a *Graphing Subroutine* and the rest of the Java code necessary to integrate the modules into a coherent single program. The *Web Crawler* and *Clustering Subroutine* support the knowledge codification process as explained in Section 4.4.3 and illustrated in Figure 4.2. The *Graphing Subroutine* is used to display results from the linear programming module.

I. Web Crawler

A *Web Crawler* is a program which browses the Web in a systematic and automated manner. Its function, envisioned in this framework, is to locate information on the Web that can be used to support the knowledge domain. The crawler searches for articles relevant to the knowledge domain based on key words. Potential websites where such articles could reside include those of new organisations and research institutes. The key words used are taken from the ontology and represent the qualitative aspects being measured. It is proposed, as a future exercise that the web crawler be used as the interface

point for the incorporation of the techniques for automatic information discovery and ontology population developed by other researchers as mentioned in Section 4.4.3.III

II. Data Clustering Subroutine

The function of the *Data Clustering Subroutine* is to categorise quantitative data into qualitative categories of none, low, medium and high. These categories, which can be expanded or reduced depending on the amount of data to be categorised, provide a measure of qualitative equivalence.

This categorisation of data is achieved through the use of data clustering. Data clustering is the unsupervised classification of data into groups or clusters [136]. The technique of data clustering used is the Least Euclidean Method. In data clustering, a simple distance measure like the Euclidean distance can be used to reflect either the dissimilarity or similarity between two patterns [136]. In the Least Euclidean Distance method, the clustering of the data is such that the distances between the data is minimised.

As explained in Section 4.4.3 (Figure 4.2), the data stored in the *Knowledge Domain Ontology* is clustered and the results from the clustering are stored in the *Value Partition Ontology*. The subroutine reads the numerical values for each of the designated classes, clusters the data and then stores their clustered results into the value partition. Examples of the clustering results using this subroutine are shown in Section 6.2.2, Table 6.5.

III. Graphing Subroutine

The function of the *Graphing Subroutine* is to display the results from the optimisation. It uses the JfreeChart API [137] and is written in Java.

The *Support Module* also holds additional Java code that integrates the five main modules (i.e. *Semantic Representation Module*, *Knowledge Inference and Expression Builder*, *Linear Programming Module*, *Graphical User Interface*, *Support Module*) into a single coherent program. These additional code allows for ontology loading and saving, logic reasoner activation, simple mathematical calculations and file operations. The existence of these minor codes is mentioned in this paragraph for completeness of documentation. No examples are provided given that examples of such code can be found in other publications and via the internet.

5.2.5 Graphical User Interface Module

The main function of this module is to enable changes in the simulation parameters of the engineering models, the selection of the scenario drivers, the selection of scenario parameters and the display of outputs from the program. The Java Swing API has been selected because of its cross platform capability and consistent visuals across multiple operating systems. Furthermore, the Java Swing API is supported by the Netbeans IDE with automatic code generation and modification. Examples of the Graphical User Interface used in the *Prototype Energy Modelling System* is shown in Figures 6.2, 6.3 and 6.13.

5.3 Steps of the Framework

The steps proposed in this framework to develop an ontology-based semantic representation optimisation model for policy evaluation are summarised here:

1. Determine the specifications and parameters
2. Confirm the modelling system design
3. Develop the modelling system based on five main modules:

- 3.1. Semantic representation module
- 3.2. Knowledge inference and expression builder module
- 3.3. Linear programming module
- 3.4. Support module
- 3.5. Graphical user interface
- 4. Compile and verify the modelling system using test procedures
- 5. Validate the created model with case studies

5.4 Conclusion

This chapter has discussed the details of a framework for the development of an *Optimisation Modelling System*. The method to define the knowledge base, the design philosophy of the ontology and engineering models, the technique to use ontologies and mathematical clustering as a method for qualitative and quantitative data processing, and the use of logic inference to assist in the formulation of expressions for both linear programming and a system of inequalities optimisation are the main topics that have been discussed.

To demonstrate the framework, a *Prototype Energy Modelling System* has been developed. The next chapter discusses this system.

Chapter 6 **Prototype Energy Modelling System**

A *Prototype Energy Modelling System* has been developed to validate the proposed framework. This chapter discusses the implementation details of this prototype and the evidence to support its claimed performance. The emphasis of the discussion shall be on the uniqueness of this prototype which resides in its semantic representation.

Section 6.1 begins with an overview of the system and follows with a description of the specification, architecture and program flow. Section 6.2 explains in detail the contents of the *Semantic Representation Module*. Section 6.3 demonstrates the two main results of the prototype which are the automatic creation of an *Energy Model* and the ability of the created model to be optimised for an electricity generation mix. Section 6.4 offers some conclusions to the chapter.

6.1 Overview of the Prototype Energy Modelling System

A *Prototype Energy Modelling System* for energy policy evaluation and analysis has been developed. The prototype uses a narrow domain based approach with energy policy as the knowledge domain and policy makers in the area of energy policy analysis as the target users. The main feature of this prototype is that it combines the advantages of a mathematical model, whose strength lies in numerical data manipulation, with those of a semantic representation, whose strength lies in the manipulation of non-numerical information. This feature, which is achieved through the combined use of semantic representation, knowledge inference and linear programming, allows the *Prototype Energy Modelling System* to automatically create a custom *Energy Model* consisting of objective functions, constraints and mathematical expressions reflective of the scenario framed by the user.

The created *Energy Model* is a bottom up linear programming model with perfect foresight. It optimises for the lowest cost of energy supply which meets demand over a period of 20 to 30 years at the country level. Designed for energy policy makers, the purpose of the energy model is to explore future electricity generation mixes based on scenarios formed by the implementation of energy policies goals and targets.

A unique feature of the *Prototype Energy Modelling System* is its emphasis on energy policy goals and policy targets as inputs for the energy model. The policy goals and targets which encompass the areas of politics, economics, society and the environment are of both types, qualitative and quantitative. To offer this feature, the *Prototype Energy Modelling System* has addressed two key impediments: (i) that political and social issues are often expressed in a non-numerical fashion and hence their integration into the numerical environment of energy models is difficult and (ii) that the translation of policy goals into mathematical expressions for use in energy models must be explicit and based on some form of rationale.

The *Prototype Energy Modelling System* is able to accept qualitative and quantitative based energy policy goals and targets as inputs because of its integration to an ontology-based semantic representation. This semantic representation of energy policy is used for two purposes: (i) to translate a set of energy policy goals into a set of logic queries which is then used to determine the preferred electricity generation technologies (policy variables), and (ii) to assist in the formulation of the mathematical expressions to be used in the numerical solution of the mathematical model to propose an electricity generation mix. Within this set up, the user has only to specify the scenario drivers together with other supplemental scenario inputs such as energy consumption, electricity demand profiles, socio-economic growth rates and industrialisation state. The *Prototype Energy Modelling System* then selects the

appropriate mathematical expressions, chooses the related energy data sets and automatically creates the *Energy Model*. The created *Energy Model* then outputs an electricity generation mix that is appropriate for the formulated scenario.

Our proposal to use a symbolic knowledge representation within energy modelling is not without precedence. Hung *et al.* describes a Knowledge Based System incorporated into MEEDES, an energy demand forecast model using VP-Expert [33]. To the best of our knowledge however, ontologies have never been applied to an optimisation based energy model for the purposes intended in this modelling system.

In terms of size and complexity, the *Prototype Energy Modelling System* consists of approximately 70 Java classes, 200 Java methods and 20000 lines of Java code. The ontology section of the program contains more than 200 subclasses, 70 properties, 450 axioms and 280 instances. Approximately 40MB of disk storage and 50MB of runtime memory are required.

The specifications of the *Prototype Energy Modelling System* are as follows:

- An easy to use system that places the emphasis on policy goals. The main inputs are the policy goals which may be either qualitative or quantitative.
- The ability to automatically formulate a set of mathematical expressions consisting of the *Technology Preference Equation* (which represents the selected energy policy goals), an objective function for optimisation, and constraints which represent the parameters of the case study.
- A created *Energy Model* that outputs an optimised electricity generation mix which meets the policy goals and targets selected by the policy maker.

The architecture of the *Prototype Energy Modelling System* is as described in Chapter 5. The five modules of the prototype are:

- A *Semantic Representation Module* which represents the energy policy knowledge domain through the use of ontologies and engineering models. The module consists of four ontologies which are the *Energy Policy Ontology*, *Value Partition Ontology*, *Energy Scenarios Ontology* and *Energy Model Construction Ontology*. In addition, a mathematical-based set of engineering models, which simulate available plant technologies, supports the *Energy Policy Ontology*. The ontologies have been coded in the OWL Web Ontology Language (OWL) [48] using Protégé [50,51] while the engineering models have been coded in the Java computer language.
- A *Knowledge Inference and Expression Builder Module* which facilitates knowledge inference from the ontology for the construction of a set of mathematical expressions. This is achieved through the use of description logic queries as determined by the selected energy policy goal options and the FACT++ logic reasoner [133]. The equation builder is built using Java and translates the output into commands for Mathematica.
- A *Linear Programming Module* which obtains an optimised electricity generation mix by solving a set of mathematical expressions comprising an objective function, energy supply and demand constraints, and the *Technology Preference Expression*. The details of the *Technology Preference Expression*, which is a mathematical expression of the scenario drivers selected by the user, has been described in Section 5.2.2.II. The module is programmed in the Java computer language and is solved by the Mathematica symbolic computational software [130].
- A *Graphical User Interface (GUI)* which allows for user input and display output. Written in Java, the module utilises the Swing widget toolkit

JfreeChart API [137]. Figure 6.2 shows the user input window. Selectable options include the energy policy goals and their policy targets, societal state to reflect the energy consumption per capita, current population level and growth rate, maximum CO₂ emission in tons per capita, GDP growth rate and the electricity generation plant decommissioning rate.

- A *Support Module* which contains a web crawler, a data clustering component, a graphing component and other parts of the program that allow for the interoperation of the modules.

Figure 6.1 illustrates the system architecture of the *Prototype Energy Modelling System* together with the main inputs, data requirements and technology of each module.

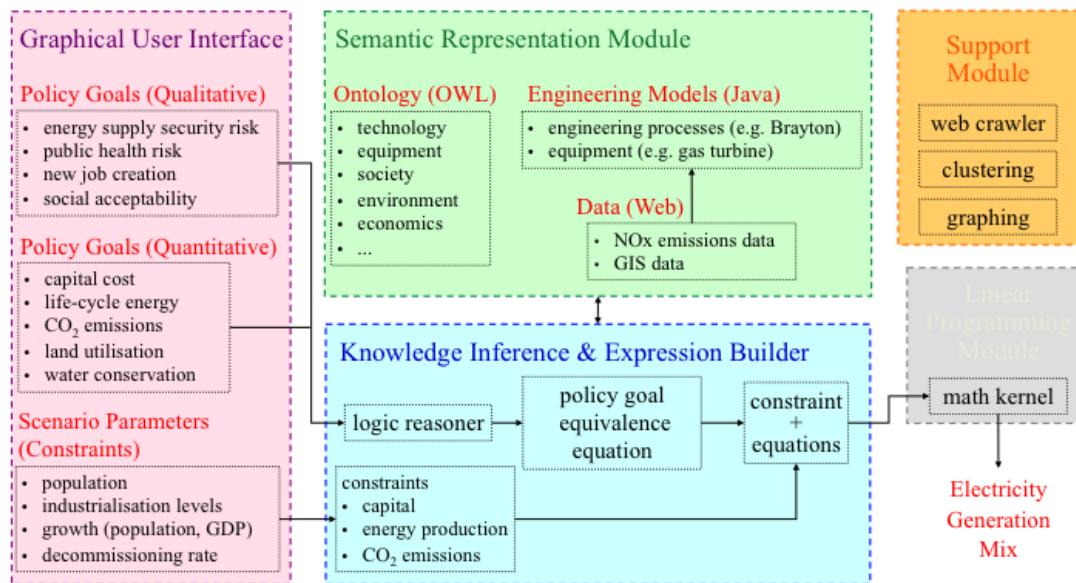


Figure 6.1. System architecture of the *Prototype Energy Modelling System*.

The information flow in the prototype is as follows. Once the prototype is started, the user is presented with a list of energy policy goals and targets applicable at the country level via a *GUI* as shown in Figure 6.2. Currently, nine policy goals have

been programmed with three policy targets for each goal. A weights input is also provided to allow for greater emphasis or even deactivation of a policy goal. Additional policy goals and targets can be included into the system through additional code and the modification of the ontology. Further details of these goals are discussed in Section 6.2.3.

The user can also specify other scenario parameters to effect a case study of the country of interest. Current population, population growth rates for 10 year periods, current energy per capita use (based on societal state), energy per capita growth rates for 10 year periods, maximum CO₂ emission per capita for electricity generation and expected plant decommissioning rates are the additional parameters that can be specified from the *GUI*.

The screenshot shows a software interface for setting energy policy goals. It includes sections for Societal State (with radio buttons for Least Developed, Developing, Newly Industrialized, Industrialised, and Msia/UK), Economic Goals (with priority levels for cost and payback), Environmental Goals (with levels for CO2 emissions, land, and water), Social Goals (with priority levels for health, jobs, and acceptability), Political Goals (with levels for energy security), and Simulation Options (with checkboxes for Total Cost, Dual Objective, Random Probability, and Reserve Generation). It also features sliders for Population and CO2 emissions per capita, and a table for 10-year period rates for GDP and population growth.

Figure 6.2. User input window for energy policy goals and targets.

The *GUI* in Figure 6.2 is also where the user specifies the various options for optimisation such as the simultaneous optimisation of two objective functions (multiobjective optimisation) and the *Direct* or *Random Probability* formulation methods for the *Technology Preference Expression* which has been explained in Section 4.2.2.II.

In addition, the user can also specify other essential parameters to reflect the current energy condition in the country of interest. Current electricity generation mix, remaining fossil fuel reserve life, electricity generation reserve margins, variation

between day and night electricity generation and maximum generation from intermittent sources are among the parameters that can be specified from a second *GUI*. The reserve margins allow the user to set the percentage of excess generation capacity which is important given that the system does not consider the import of electricity during shortfalls. Figure 6.3 shows the second *GUI* panel.

Consumption Profile (100.00%)

0049 Coal Industrial	0000 Coal Commercial	0000 Coal Agricultural
0813 Gas Industrial	0577 Gas Commercial	2061 Gas Residential
0573 Oil Industrial	0000 Oil Commercial	0000 Oil Agricultural
0647 Elec Industrial	0539 Elec Commercial	0627 Elec Residential
	0000 Elec Agricultural	0473 Elec Transportation

Generation Profile (100%)

00 Solar Power	37 Coal Power
02 Wind Power	02 Biomass Power
36 Gas Power	03 Hydro Power
02 Oil Power	18 Nuclear Power

Generation Reserve Profile (100%)

50 Coal Power Reserve
20 Gas Power Reserve
20 Oil Power Reserve
10 Nuclear Power Reserve

Reserve (100%)

20 % Above Demand
--

Maximum Renewable Contribution (100%) *current totalled

25 Wind Power	15 Solar Power
--	---

Other Parameters (100%)

20 -% Night Electricity	50 +% Day Electricity
--	--

Figure 6.3. User input window for additional optimisation options.

Once the selected scenario drivers and simulation parameters have been fixed, the user activates the modelling system process by pressing the *Solution* button. The modelling system associates the selected policy goals and targets with a set of axioms

for use as description logic queries by a logic reasoner. The reasoner infers from the ontology the technological solutions which support the selected scenario drivers. The necessary policy variables, constants and values for the construction of additional mathematical expressions are loaded from the ontology. The formulated mathematical expressions are fed into the *Linear Programming Solver* and solved for an electricity generation mix. Examples of the mathematical expressions which form the *Energy Model* together with the optimised results are discussed in Section 6.3. Figure 6.4 summarises the information flow of the *Prototype Energy Modelling System*.

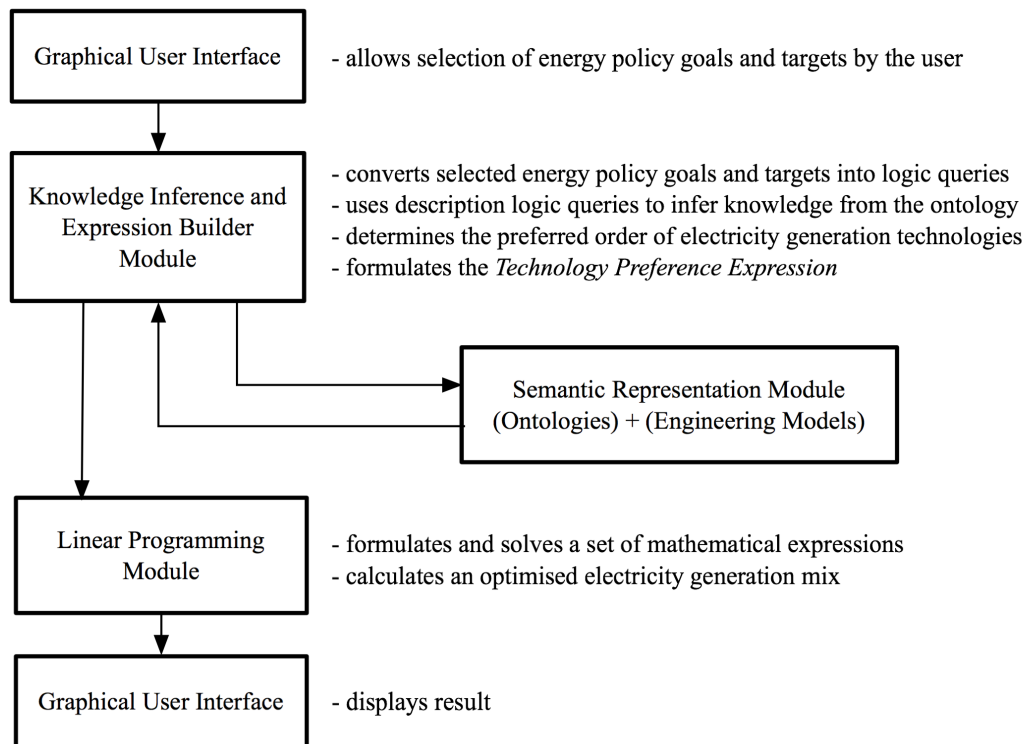


Figure 6.4. Simplified information flow of the *Prototype Energy Modelling System*.

6.2 Semantic Representation of the Energy Policy Knowledge Domain Using Ontologies and Engineering Models.

This section discusses the representation details of the energy policy knowledge domain contained in the *Semantic Representation Module*. As discussed in Section 4.4.1, the codified knowledge consist of the core knowledge and the application knowledge.

The core knowledge is stored in *Energy Policy Ontology* and the *Value Partition Ontology*. The application knowledge is stored in the *Energy Scenarios Ontology* and the *Energy Model Construction Ontology*. In addition, engineering models are used to support the *Energy Policy Ontology*. Figure 6.5 illustrates the relations between the ontologies and engineering models in this module.

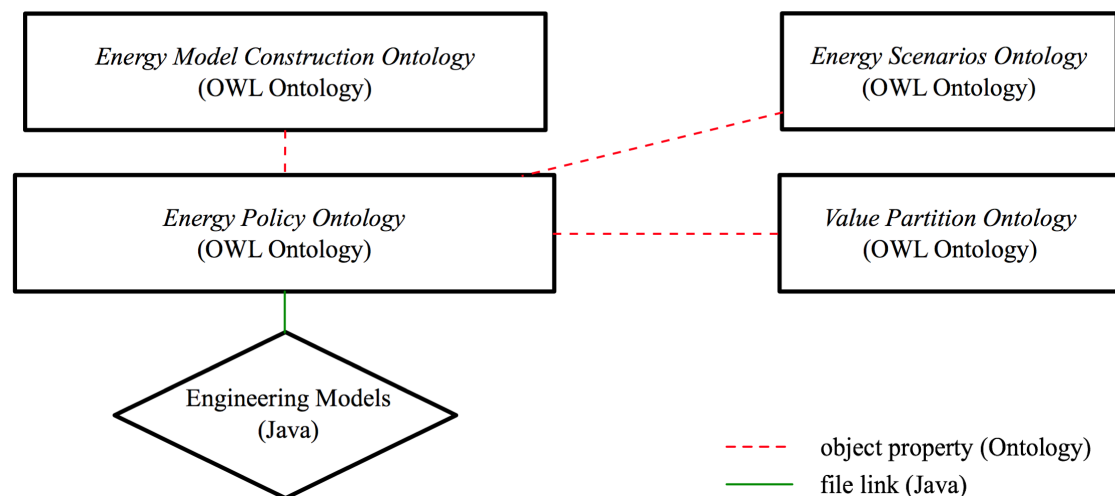


Figure 6.5. Relations between the ontologies and the engineering models in the *Semantic Representation Module* for the *Prototype Energy Modelling System*.

6.2.1 Energy Policy Ontology

This ontology represents the core knowledge associated with electricity generation which is, in our view, an energy supply chain that centres on the transportation of energy and its transformation from one form to another. This energy supply chain involves mining, collecting, transporting, transforming, transmitting, consuming, and utilising the energy. The energy medium itself often changes from that of a fuel at the source to work (useful energy) at the destination.

Ontologically, this supply chain is represented as a collection of manmade objects (e.g. electricity generation plants) which utilise natural resources (e.g. natural gas) through the use of natural laws (e.g. stoichiometric combustion) and is influenced by manmade concepts (e.g. production costs). As proposed in the framework, the ontology (Figure 6.6) is built around two main classes, *nature* and *manmade*, with four subclasses, *natural_Laws*, *natural_Resources*, *manmade_Concepts* and *manmade_Objects*.

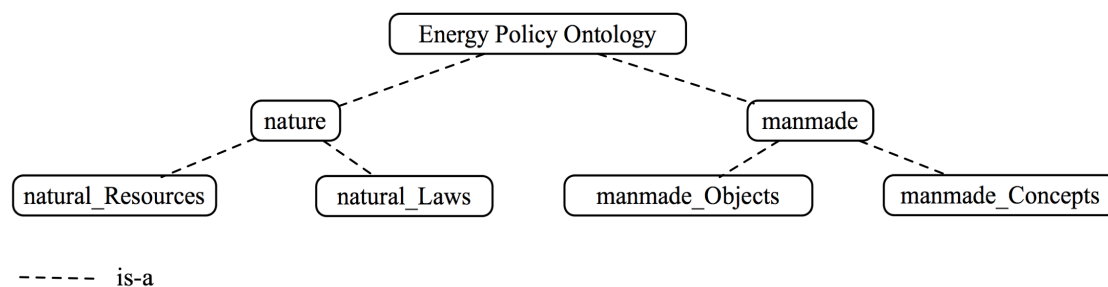


Figure 6.6. Taxonomy of *energy_Policy*.

The class *natural_Laws* (Figure 6.7) contains some of the laws of physics and chemistry relevant to energy policy. Electricity generation, energy equilibrium, fuel combustion, and mechanical work are among the laws represented using the subclasses *electricity_Generation_Process*, *zeroth_Law*, *stoichiometric_Combustion*

and *net_Work_Out*. Furthermore, the ontology contains the properties *consumes*, *generates* and *requires* which are used to represent the relationship between these processes and the required or generated resources which are represented by the class *natural_Resources*. Using axioms, a complete description logic statement of this relationship is made. For example, the steam generation process which requires water and generates steam, is represented by the axiom '*IHOV_steam_Generation requires water and generates steam*' (IHOV stands for latent heat of vaporisation). In addition, a number of these subclasses have links to Java classes in order provide mathematical support. The property *Java_Class_STRING* indicates the link between the instance and the Java class which contains the mathematical expressions of the particular engineering process.

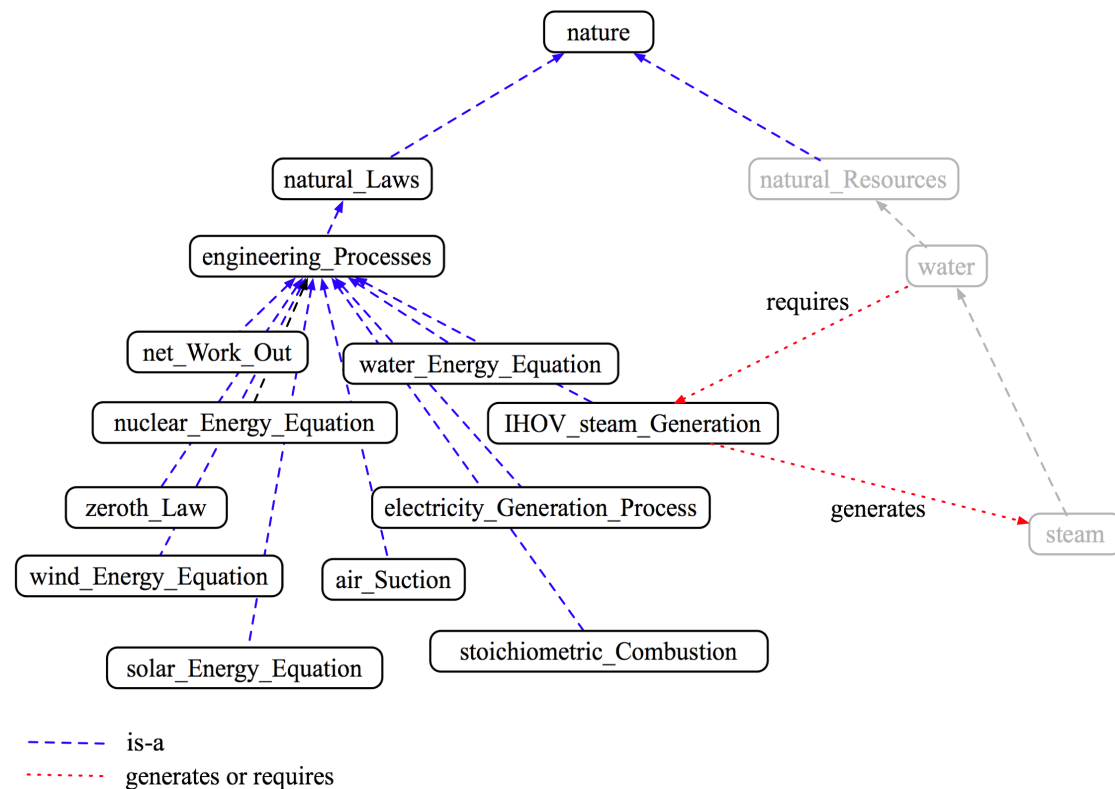


Figure 6.7. Taxonomy of *natural_Laws*.

The class *natural_Resources* (Figure 6.8) represents the elements and chemical compounds that are consumed or produced in the process of energy generation. Water, air, coal, oil, carbon dioxide and oxygen are examples of substances represented within this class. The class also represents, in some cases, the states of the substance in question. An example of this is steam, which is water in the vapour phase.

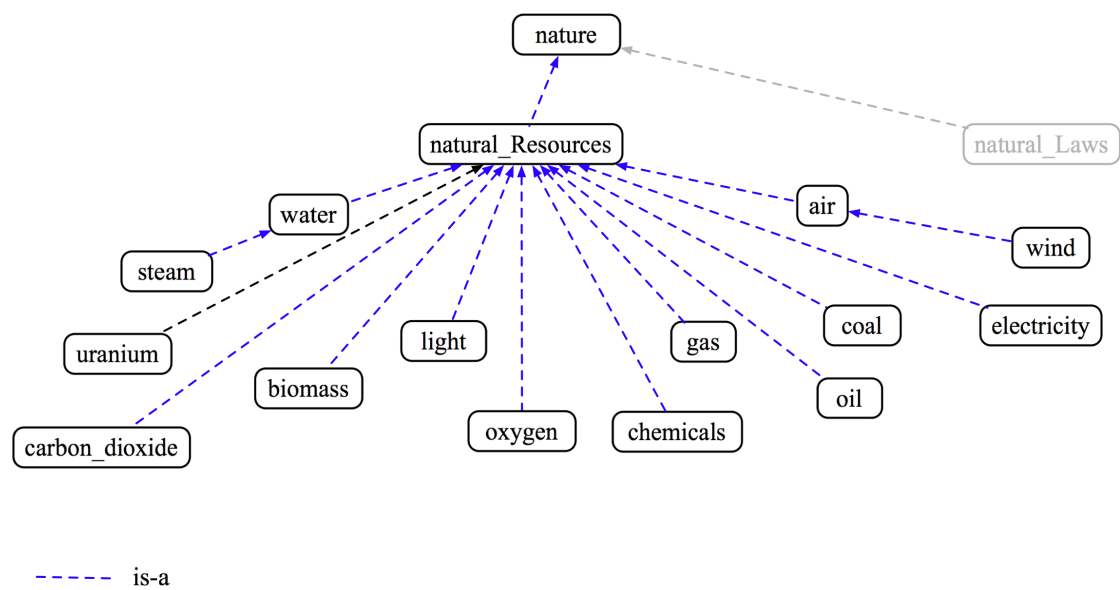


Figure 6.8. Taxonomy of *natural_Resources*.

The class *manmade_Concepts* covers the concepts related to politics, economics, society and the environment (Figure 6.9).

Politics address the issue of energy supply security which itself is a combination of both geopolitical and infrastructural risks. The geopolitical risk combines both source dependence and transit dependence risks while the infrastructure risk combines both facility dependence and structural risks [138]. The criteria used to assess the risk levels associated with oil diversification [139], the information used to determine the appropriate portfolio approach for energy security

[140] and the results of the comparative risk assessments of energy systems [141], were all used as the basis for the determination of geopolitical and infrastructure risks and their evaluation in the value partition. The values in the partition are based on our best judgement which was, in turn, supported by the literature.

Economics cover the cost of electricity generation and the life cycle energy payback of the energy process. The full cycle of the electricity generation plant is considered by the energy payback value, which is the ratio of total energy produced during that system's normal lifespan to the energy required to build, maintain and fuel the system [142,143]. The classes *capital_Cost* and *lifecycle_Energy* represent these two concepts respectively. Capital costs and total cost data were obtained from the UK MARKAL model documentation [105]. Life-cycle energy payback data were taken from Gagnon [142] for all energy sources and generation technologies.

Environmental issues are related to land utilisation, water conservation and CO₂ emissions. Land and water use for an electricity generation plant can be of special concern for electricity generation technologies such as hydroelectric power, while the CO₂ emissions from an electricity generation plant are restricted by regulations [144,145]. The classes *land_Requirement*, *water_Consumption* and *CO2_Emission* represent these concepts respectively. Land requirement values were sourced from Chatzimouratidis and Pilavachi [146]. Water consumption and CO₂ emission values were calculated using the engineering models.

Social aspects cover both the desirable and undesirable consequences of energy utilisation. The net creation of new jobs from either the commercialisation of new renewable energy sources or the continued development of current technologies [147,148] is a desirable aspect while the externalised costs of the energy system, such as increased pulmonary complications attributable to NO_x emissions [142,149,150], are represented in the system.

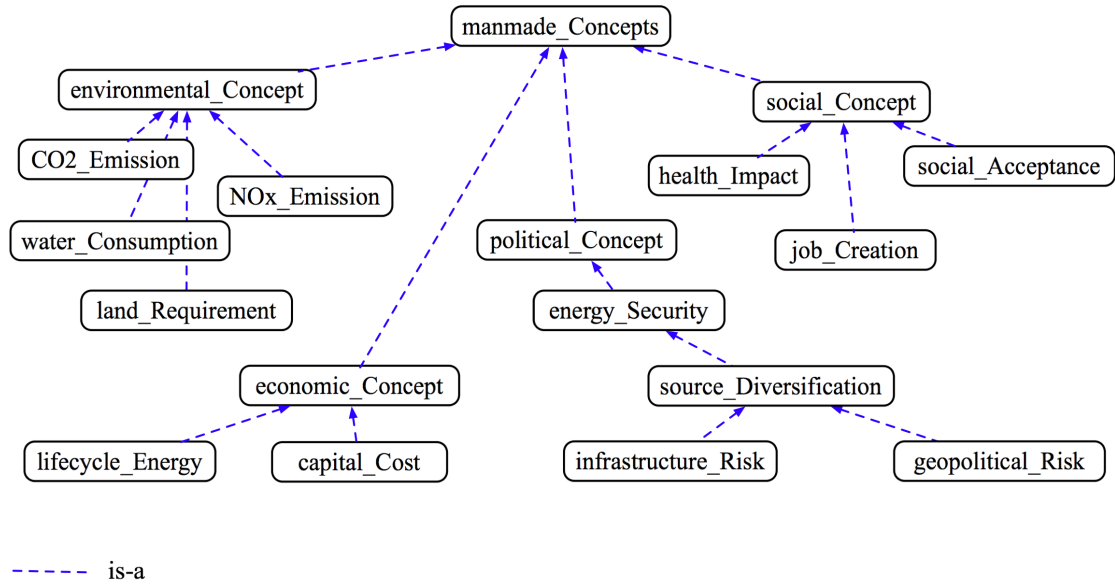


Figure 6.9. Taxonomy of *manmade_Concepts*.

The class *manmade_Objects* represents the electricity generation infrastructure (Figure 6.10). It comprises *energy_Chain_Objects* which stores the large facilities (e.g. generation plants) associated with the electricity generation supply chain and *other_Objects* which stores the small equipment (e.g. turbines) associated with electricity generation.

The *energy_Chain_Objects* represents the facilities for fossil fuel extraction (e.g. oil platform), fossil fuel transportation (e.g. pipelines), electricity generation (e.g. gas power plant), electricity transmission (e.g. electrical power lines) and electricity consumption (e.g. lights). The classes *upstream_Facilities*, *transportation_Vehicles*, *plant_Generation*, *transmission_Equipment*, and *enduser_Consumer* are used to represent these facilities. The relations between the classes are represented using the object properties *transports_via*, *delivers_to*, *connected_to*, and *transmits_to*.

The most extensive representation is the subclass *plant_Generation*, which contains eight subclasses for the representation of the coal, oil, gas, biomass, solar,

wind, hydroelectric and nuclear electricity generation plants with the assumption of on-site fuel treatment. Each plant representation is constructed in detail using axioms to relate to the main equipment used such as electricity generators, gas turbines, steam turbines and auxiliary boilers which are stored in *other_Objects*. The object properties *consist_of* connects the major equipment and components to the *plant_Generation* subclass. Data details such as capital cost, generation capacity, plant availability, plant capacity factor and CO₂ emissions are also stored and associated to the plant subclass via data properties. Furthermore, the generation plants are complemented by engineering models based on engineering processes (e.g. combined cycle) using mathematical expressions (e.g. stoichiometric combustion) with the incorporation of relevant data from the literature (e.g. air ambient temperature, which is sourced from a Geographical Information System [149]). Links are made from the generation plant level to the engineering models where a more detailed representation is required such as in the case of a Brayton cycle or a Rankine cycle.

The subclass *other_Objects* stores the equipment and other objects indirectly related to electricity generation. At the equipment level, each equipment is represented through the combination of its subcomponents where possible using the property '*consist_Of*'. A gas turbine engine, for example, is represented through its components (i.e. compressor, combustor, power turbine) using three axioms: '*gas_Turbine consist_Of compressor*', '*gas_Turbine consist_Of combustor*', and '*gas_Turbine consist_Of power_Turbine*'. In addition, to represent the connection between the energy infrastructure and its fuel resources the properties *carries* and *utilises* are used by the subclasses of *other_Objects* to relate to the subclasses of *natural_Resources* and *natural_Laws* respectively. The axioms '*steam_Generator utilises some IHOV_steam_Generation*' and '*ship carries some coal*' for example, represents the fact that a boiler utilises some steam generation process and that a ship

can be used to carry coal.

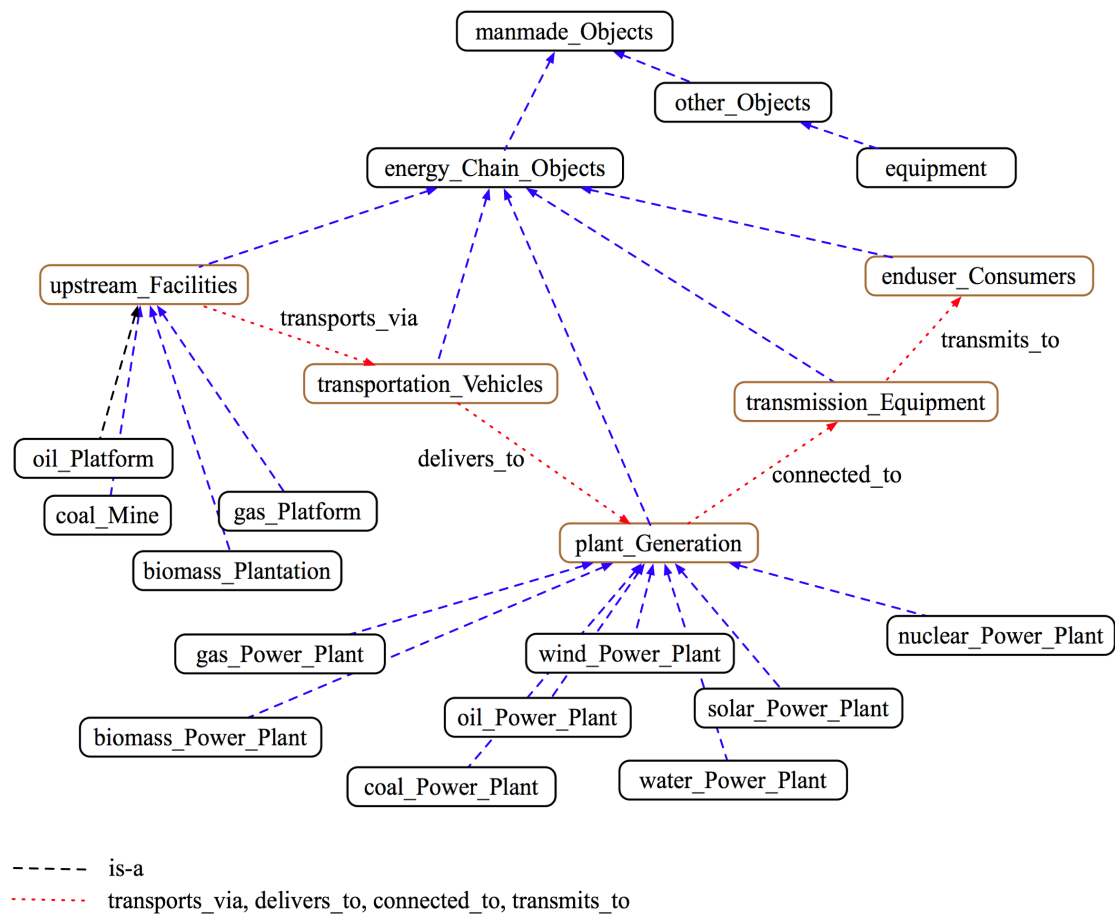


Figure 6.10. Taxonomy of *manmade_Objects*.

The class *manmade_Objects* uses the object property *hasState* to link the classes under *plant_Generation* (bottom of Figure 6.10) with the concepts under *manmade_Concepts* (Figure 6.9). In turn, the property *has_Value* links the classes under *manmade_Concepts* and the assessments stored in the *Value Partition Ontology*. The relationships between these classes are represented by axioms such as, for example, '*coal_Power_Plant hasState CO2_Emissions has_Value high*'.

The classes *manmade_Objects* also uses the object property *utilises* to link the classes under *plant_Generation* with the laws under *natural_Laws*. The relationships

between these classes are represented by axioms, of which '*coal_Power_Plant consist_Of coal_Steam_Boilers utilises stoichiometric_Combustion*' is an example.

The classes *manmade_Objects* also use data properties to link the classes under *plant_Generation* to specific data (i.e. values or literals). Examples of these data properties are *CO2_Emission_DOUBLE* and *Capital_Cost_DOUBLE* (the term DOUBLE is used because the stored data are double precision numbers). The relationships between these classes and the data are represented by axioms such as, for example, '*coal_Power_Plant CO2_Emission_DOUBLE 840*' (tons per hour per specific generation capacity of the plant).

Tables 6.1 thru 6.4 summarises these relationships through selected examples:

- Table 6.1 illustrates the relationship between the *Knowledge Domain Ontology* and *Value Partition Ontology*.
- Table 6.2 illustrates the link between classes and data within the *Knowledge Domain Ontology*.
- Table 6.3 illustrates the relationship between *energy_Chain_Objects* and *other_Objects*.
- Table 6.4 illustrates the relationship between *manmade_Objects* and *natural_Processes* and *natural_Resources*.

Table 6.1. Selected example of the relationship between *manmade_Objects* and *manmade_Concepts* and the *Value Partition Ontology*.

<i>Energy Policy Ontology</i>				<i>Value Partition Ontology</i>
Subclass of <i>manmade_Objects</i>	Object Property	Subclass of <i>manmade_Concepts</i>	Property	Instance
coal_Power_Plant	hasState	CO2_Emissions	has_Value	High
wind_Power_Plant	hasState	CO2_Emissions	has_Value	Low

Table 6.2. Selected example of the relationship between *manmade_Objects* and data, value or literal.

<i>Energy Policy Ontology</i>		
Instance of coal_Power_Plant, Subclass of manmade_Objects	Data Property	Data, Value, Literal
coal_Power Plant_INSTANCE	CO2_Emission_DOUBLE	840 (In tons/hour per 961MW Plant)
coal_Power Plant_INSTANCE	Electricity_Production_DOUBLE	961 (In MW per Plant)

Table 6.3. Selected example of the relationship between *plant_Generation* and *equipment*.

<i>Energy Policy Ontology</i>		
Subclass of <i>plant_Generation</i> , Subclass of <i>energy_Chain_Objects</i> , Subclass of <i>manmade_Objects</i>	Object Property	Subclass of <i>equipment</i> , Subclass of <i>other_Objects</i> , Subclass of <i>manmade_Objects</i>
coal_Power_Plant	consist_Of	coal_Steam_Boiler
coal_Power_Plant	consist_Of	electricity_Generator

Table 6.4. Relationship between *other_Objects* and *natural_Laws* and *natural_Resources*.

<i>Energy Policy Ontology</i>				
Subclass of other_Objects, subclass of manmade_Objects	Object Property	Subclass of natural_Laws	Object Property	Subclass of natural_Resources
coal_Steam_Boiler	utilises	stoichiometric_Combustion	consumes	coal
coal_Steam_Boiler	utilises	stoichiometric_Combustion	requires	air
electricity_Generator	utilises	electricity_Generation_Process	generates	electricity

6.2.2 Value Partition Ontology

The *Value Partition Ontology* stores the classification assessment of data related to objects in the *Energy Policy Ontology*. The classification assessments are based on the policies declared in the energy policy goals and targets. For each energy policy goal, the value partition stores three clusters which define the three policy targets. In principle however, the number of clusters and thus the number of levels can be expanded if the user so wishes.

The storage of the classification assessment of data in this ontology allows the program to associate quantitative values to qualitative categories. Quantitative values taken from the MCDA literature are the primary source of data used for classification. However, some values have been calculated using the engineering models. The classification of the data has followed *path 1* and *path 2* of the data codification paths illustrated in Figure 4.2.

An example of an energy policy goal that is stored in this ontology is the maximisation of life cycle energy payback. Three policy targets which are *low*, *medium*, and *high* are available based on the data clustering results of the values published by a paper by Gagnon [142]; the values between 0-5 have been categorised as low, 6-33 as medium and 34-280 as high.

The ontology currently stores seven sets of categorised value ranges. The seven sets are for capital cost, life cycle energy payback, CO₂ emissions, land utilisation, water consumption, new job creation and social acceptance.

The quantitative energy policy goals primarily encompass the topics of economics and the environment. The data for these goals are capital and total costs [105], life cycle energy payback [142], and land utilisation of electricity generation plants [146]. The water consumption of electricity generation technologies and the CO₂ emissions of electricity generation processes were calculated using the

engineering models. In addition, the consideration of the potential future reduction of CO₂ emissions was based on data published in the literature [149].

The qualitative energy policy goals are typically those related to society and politics. Literature data on the social acceptance of electricity generation technologies [146] and on the creation of new jobs from the implementation of new generation technologies [147] were used. Two additional energy policy goals, i.e. public health and energy security, are also stored in this ontology but are classified through other means as described later in this section.

The value ranges used to categorise a particular value as *low*, *medium* or *high* is shown in Table 6.5.

Table 6.5. Value ranges for low, medium and high.

Parameter	Unit	Value Ranges		
		Low	Medium	High
Capital Cost	GBP per MW at plant capacity factor	445 - 1022	1853 - 4000	52273
Life Cycle Energy Payback		3 - 5	6 - 16	34 - 280
CO ₂ Emissions	tons per hour per plant	0 - 88	287 - 453	824
Land Utilisation	km ² per 1000MW	25	350	1000 - 7500
Water Consumption	tons per hour per plant	0 - 183	2007 - 5575	12647
New Job Creation	new employees per 500MW	110	140 - 170	210 - 270
Social Acceptance	MCDA value	2.76 - 6.8	15.9 - 21.6	29.3 - 32

Given the value ranges in Table 6.5, the classification of the electricity generation technologies, based on their literature or calculated values, are listed in Table 6.6.

Table 6.6. Classification of electricity generation sources.

Parameter	Classification of Generation Sources		
	Low	Medium	High
Capital Cost	Gas	Coal Biomass	Nuclear Oil Solar Hydroelectric Wind
Life Cycle Energy Payback	Biomass Coal Gas Oil Solar	Nuclear Wind	Hydroelectric
CO ₂ Emissions	Biomass Gas Nuclear Solar Hydroelectric Wind	Oil	Coal
Land Utilisation	Coal Gas Oil	Nuclear Solar Wind	Biomass Hydroelectric
Water Consumption	Wind	Coal Gas Oil Solar	Biomass Hydroelectric Nuclear
New Job Creation	Coal Gas Oil Hydroelectric Nuclear	Solar Wind	Biomass
Social Acceptance (MCDA)	Coal Nuclear Wind * (UK) Oil	Biomass Gas Oil * (Msia) Hydroelectric Wind* (Msia)	Solar Wind

Changes to the classification which are inconsistent with the reference literature are indicated with an asterisk (*). These changes have been made because in our opinion, the MCDA data does not reflect the situation in the target countries of the case studies (i.e. the United Kingdom and Malaysia) where the *Prototype Energy Modelling System* will be tested. For social acceptance, the MCDA numbers from the literature indicate that oil has low and wind has high social acceptance rates. However, a literature review from the United Kingdom would refute the assertion of wind as a widely accepted source of energy [151]. In addition, oil as represented in the *Prototype Energy Modelling System*, forms part of IGCC installations due to the current situation in Malaysia where natural gas shortages have resulted in the need to use distillates as a fuel substitute [152,153].

In addition to the seven sets of categorised range values shown in Table 6.5, this ontology also stores the classification for public health risks and energy security levels associated with each electricity generation source (see Table 6.7). These classifications have been compiled based solely on our understanding of the literature and follows *path 3* of the data codification paths illustrated in Figure 4.2.

For public health, as measured by NO_x and SO₂ emissions from the electricity generation sector, over 80% of such emissions originate from coal fired generation while approximately 15% are from oil and gas fired generation [154]. In addition, while it is true that nuclear plants do not emit NO_x and SO₂, and have high safety standards [155], the damage caused by a nuclear release when an does accident occur, as illustrated by events in Chernobyl, Fukushima and Three Mile Island, warrants a high risk assessment to public health.

For energy security risk, measured both by geopolitical and infrastructural risks, renewable sources have low risks in both categories. Renewable sources are normally located in the country of use and the generation facilities are scalable and

thus have minimal project failure risks [156]. Coal has small energy supply risks given their widespread availability in most countries [157] while natural gas is highly dependent upon the geographical area [157,158,159]. Oil represents the highest risk of energy supply given its concentration in certain areas of the world [158,159] and with most of these countries currently facing geopolitical instability [160]. Nuclear technology is subjected to strict supply control to avoid weapons proliferation [161].

Table 6.7. Classification of electricity generation sources with respect to public health and energy supply security risk.

Parameter	Classification of Generation Sources		
	Low	Medium	High
Public Health Risk	Biomass Solar Hydroelectric Wind	Gas Oil	Coal Nuclear
Energy Supply Security Risk	Biomass Coal Hydroelectric Solar Wind (UK) *	Nuclear Gas (Msia / UK) *	Oil Wind (Msia) *

6.2.3 Energy Scenarios Ontology

The function of this ontology is to provide a description logic translation of the drivers used for scenario development. The scenarios are framed by the identified energy policy goals and policy targets. This selection is then translated into logic queries for the knowledge inference from the *Energy Policy Ontology*. The inferred policy variables are then used to formulate the *Technology Preference Expression*. Figure 6.11 is the Protégé screen shot of the ontology hierarchy for the *Energy*

Scenarios Ontology.

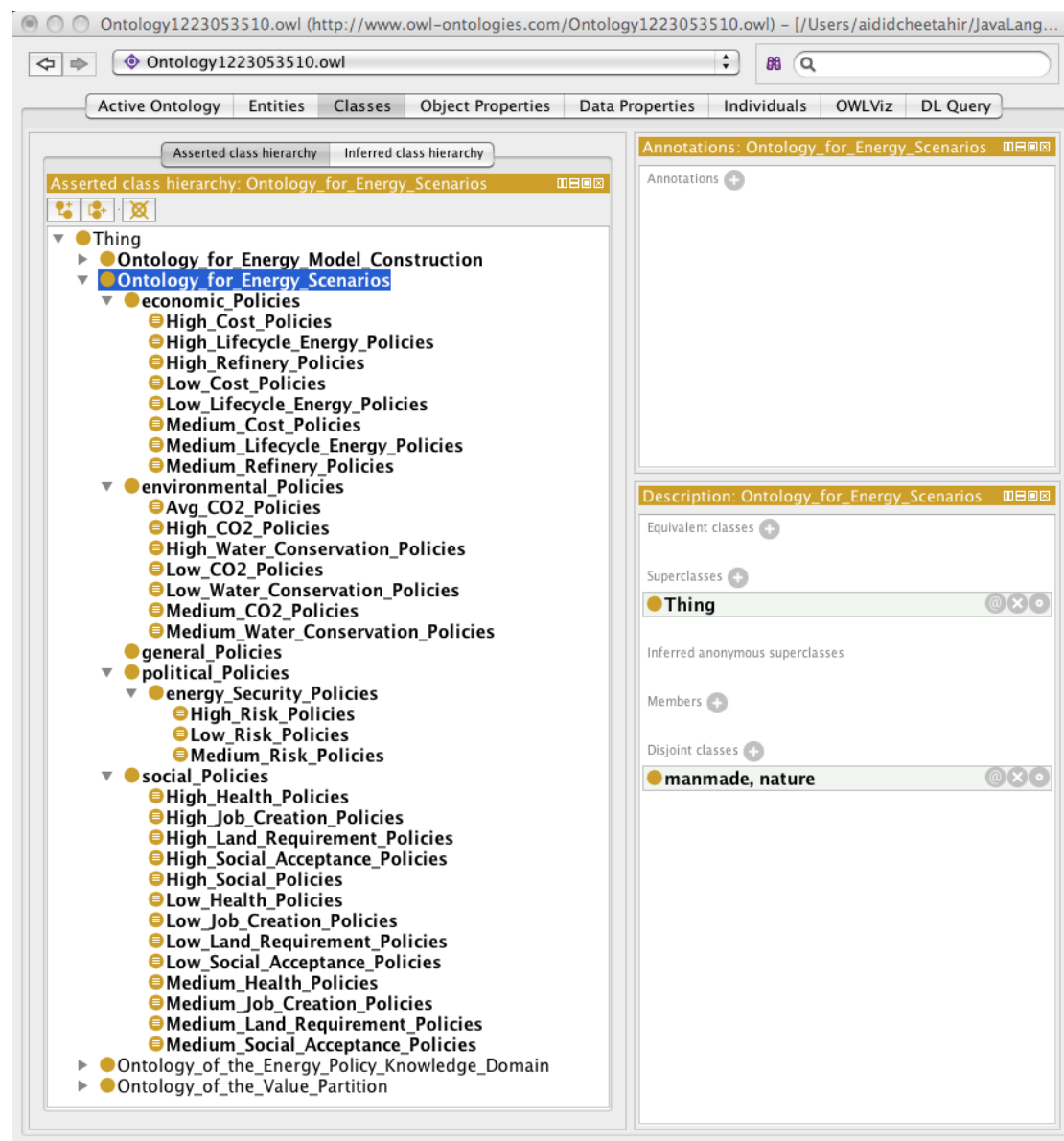


Figure 6.11. Protégé screen shot of the taxonomy of the *Energy Scenarios Ontology*.

From the literature, nine scenario drivers in relation to energy have been identified to represent the aspirations of society. Each policy goal has three selected policy targets for a total of 27 scenario driver options. The nine policy goals are:

- The minimisation of capital cost which reflects the trade-off between monetary cost and effective energy utilisation. Cheaper energy solutions are often less efficient than their more expensive counterparts. Hence, the cheapest financial solution might not be the best solution from the standpoint of energy utilisation.
- The maximisation of life cycle energy payback which reflects the trade-off between effective energy utilisation and monetary cost. More efficient energy solution are often more expensive.
- The future reduction of CO₂ emissions which reflects the commitment of the electricity generation industry to the reduction of CO₂ emissions in the future.
- The maximisation of water conservation which reflects the increased concern on the water consumption for electricity generation.
- The minimisation of public health risk which reflects the external cost to society due to emissions of NO_x, and CO₂ from electricity generation.
- New jobs creation, which reflects the potential creation of new jobs from the adoption of new electricity generation technologies and facilities
- Land utilisation which reflects the limitation of usable land for commercial, residential, industrial and electricity generation sites.
- The importance of social acceptability which reflects society's concern for the environmental impact caused by electricity generation facilities such as hydroelectric and nuclear power and society's objection to the aesthetic degradation caused by certain facilities such as wind power.

- The level of energy supply security risk which reflects the infrastructural and geopolitical risks of an energy source (e.g. the presence of oil platforms in politically unstable regions or the transactional complexities associated with the construction of nuclear power plants).

6.2.4 Energy Model Construction Ontology

The function of this ontology is to store the axioms necessary to enable logic inference of the variables and constants for (i) the construction of the objective function and constraints of the linear programming model, and (ii) the construction of the constraints for the system of inequalities.

Three classes with their associated axioms have been created for the generation of the objective functions corresponding to capital cost, total cost and life cycle energy payback (Figure 6.12). The objective function for cost, which is a more simple version of those employed in existing energy models [78,162], is the minimisation of capital cost of the energy infrastructure (i.e. power plants). The objective function for life cycle energy payback, which is a dimensionless number, is the maximisation of the energy efficiency of the energy supply and conversion process.

Several classes with axioms have been created for both supply and demand constraints. Supply based constraints limit the total electricity generation while demand based constraints dictate minimum electricity generation. The supply constraints represented are the maximum allowable CO₂ emissions from electricity generation and the maximum allowable electricity supply from intermittent generation sources (i.e. wind and solar). The total allowable intermittent generation is a supply constraint designed to guarantee the stability of the entire electricity network [163]. The demand constraints are the minimum average day and night electricity

generation, the minimum heat and power production and the minimum annual reserve electricity installation facilities. The minimum night electricity generation differs from that of the minimum day generation in that solar generation is excluded from the former. In addition, the required minimum night electricity demand is lower than the day demand due to reduced economic activities at night [164]. The reserve electricity installation facilities (primarily fossil fuel fired generation plants) represent the generation capacity in excess of demand which contributes to the electricity reserve margin.

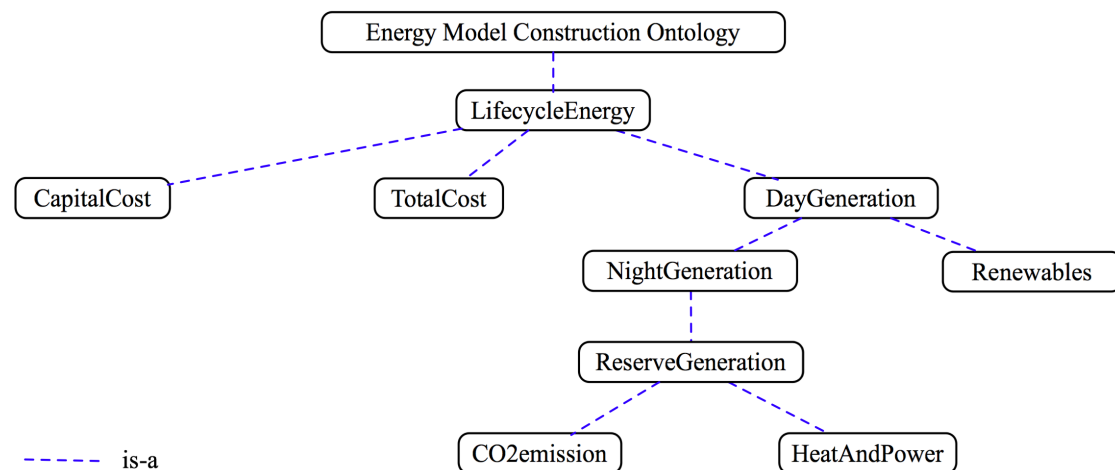


Figure 6.12. Taxonomy of the *Energy Model Construction Ontology*.

6.2.5 Engineering Models

The philosophy behind the use of mathematical-based engineering models is predicated on the fact that ontologies on their own are insufficient to provide a complete representation of the knowledge domain of energy policy. To represent the details of an engineering process such as the Brayton cycle, mathematical models are required. The models form part of the representation but are programmed in Java and linked to classes in the *Energy Policy Ontology*.

From a user standpoint, one advantage of the engineering models is the ability to quickly and consistently modify the plant representations in the ontology. By changing the operating parameters of the respective models, a new plant simulation can be made and the results of the simulation can be stored into the ontology automatically. Changes to the operating parameters for all the models can be effected from the *GUI* as in Figure 6.13. In addition while the *GUI* allows only a limited number of changes to the operating parameters of the plant, in principle, the amount and range of allowable changes can be expanded with additional programming.

Eight electricity generation plant models have been built to support the ontology in the *Prototype Energy Modelling System*. These are:

- *Coal Power Plant* which simulates a coal fired auxiliary boiler in a simple cycle. The model accepts two inputs: coal consumption in tons per day (2000 - 160000) and the temperature and pressure of the steam cycle (40 bars or 100 bars at 673 K or 993 K). The ontological, model and mathematical details are provided in Table 6.8
- *Gas Power Plant* which simulates a natural gas fired turbine in either a simple cycle, combined cycle or heat and power cycle. The combined and heat and power cycles have waste heat boilers that operate at 40 bars and 673 K. The

model accepts two inputs: natural gas consumption in tons per day and the method of combustion calculation. The ontological, model and mathematical details are provided in Table 6.9

- *Oil Power Plant* which simulates an oil fired gas turbine in either a combined cycle or heat and power cycle. The combined and heat and power cycles have waste heat boilers that operate at 40 bars and 673 K. The model accepts two inputs: oil consumption in tons per day and the method of combustion calculation. The ontological, model and mathematical details are provided in Table 6.10
- *Wind Power Plant* which simulates an onshore wind turbine. The model accepts three inputs: turbine blade diameter, the number of turbine units and geographical location. The wind speed is automatically determined from the geographical location. The ontological, model and mathematical details are provided in Table 6.11
- *Water Power Plant* which simulates a hydroelectric installation. The model accepts a single input: the geographical location of the plant. The height of the dam and river flow rates are automatically determined based on their geographical location. The ontological, model and mathematical details are provided in Table 6.12
- *Solar Power Plant* which simulates two solar technologies: solar photovoltaic and solar concentrator. The solar concentrator has a waste heat boiler operating at 40 bars 673K. The model accepts two inputs: the size of the photovoltaic or mirrors and the geographical location of the facility. The solar insolation values are automatically determined from the geographical location. The ontological, model and mathematical details are provided in Table 6.13

- *Biomass Power Plant* which simulates a biomass fired auxiliary boiler in a simple cycle. The model accepts two inputs: the feed type (generic biomass or palm oil empty fruit bunches) and the size of the biomass plantation. The auxiliary boiler operates at 40 bars 673 K. The ontological, model and mathematical details are provided in Table 6.14
- *Nuclear Power Plant* which simulates a nuclear reaction heated waste heat boiler in a combined cycle. The boiler operates at 100 bars 993 K. The ontological, model and mathematical details are provided in Table 6.15.

The GUI for electricity generation plant simulation features a top navigation bar with tabs: Profiles, Installation, Query, Equation Query, Charts, and Control. The main interface is organized into sections for different power generation technologies, each with a 'Calculate' button and a 'Select Country' button.

Panel Control: Calculate, Select Country, Store, Save

Coal and Coal CCS Electricity Generation:

- ☐ Coal Steam Boiler
- ☐ Coal Steam Boiler w Steam Turbine CCS
- Coal Consumption (2000–16000) tpd: 8000
- Plant Steam Cycle: ☐ 40 bars, 673 K ☐ 100 bars, 993 K

Wind Electricity Generation:

- ☐ Offshore Wind Turbine
- ☐ Onshore Wind Turbine
- Wind Turbine Diameter (60–120) m and Units (100–200): [Slider]
- ☐ UK Wind Velocity
- ☐ Malaysia Wind Velocity

Gas and Oil Electricity Generation:

- ☐ Gas Turbine
- ☐ Gas Turbine Combined Cycle
- ☐ Gas Turbine Heat and Power
- ☐ Oil Turbine Heat and Power
- ☐ Oil Turbine Combined Cycle
- NGas Consumption (100–200) mmscf: 200
- Oil Consumption (2000–16000) tpd: 8000
- Combustion Calculation Method: ☐ Stoic Combustion ☐ Enthalpy Diff

Biomass Electricity Generation:

- ☐ Biomass Steam Boiler (Generic)
- ☐ Biomass Steam Boiler (Palm Oil EFB)
- Biomass Plantation (5000 – 30000) hectares: 20000

Nuclear Electricity Generation:

- ☐ Nuclear Steam Boiler CC

Solar Electricity Generation:

- ☐ Solar Collector
- ☐ Solar Concentrator
- Solar PV or Mirror Area (50 – 500) hectares: 300
- ☐ Malaysia Solar Insolation
- ☐ UK Solar Insolation

Water Electricity Generation:

- ☐ Hydroelectric Turbine
- ☐ River Discharge (example of dams)
- ☐ 83 km3 Nile, Aswan
- ☐ Lochaber, Scotland
- ☐ 89 km3 Columbia, Coulee
- ☐ Temenggor, Malaysia

Figure 6.13. GUI for electricity generation plant.

As with the ontology, each electricity generation plant model has been constructed through the combination of its constituting equipment. These main equipment, which appear as individual classes in the ontological taxonomy also appear as individual Java Classes (programs) and Java Methods (procedures). This has been done to maintain consistency between the ontological representations and the engineering models.

Examples of main equipment that have been modelled include gas turbines, auxiliary and waste heat boilers, wind turbines, hydroelectric turbines, steam turbines, water pumps, generators, and solar photovoltaic panels. Examples of constituting equipment that have been modelled include compressor, combustor, gas generator and power turbine. Examples of mathematical formula include combustion, latent heat of vaporisation and net work. Examples of engineering processes that have been modelled include the Brayton and Rankine cycle.

To enhance the accuracy of the engineering models, data from the literature have been incorporated. Such data include average ambient air temperature [149], NO_x emissions per unit of energy generated, quality of fuel used, calorific fuel values, average wind speed, average solar insolation and total hydroelectric potential [28].

Tables 6.8-6.15 illustrate the ontology and engineering model arrangements for each electricity generation plant. These tables contain two main parts: the *Energy Policy Ontology* and the Mathematical-based Engineering Models

- The *Energy Policy Ontology* (columns 1 thru 4) consists of a set of objects (subclasses). At the top level, the main object is a power plant (column 1). This power plant is constructed from its constituting main equipment (column 2). In turn, these main equipment are made up of a set of main components (column 3) which uses specific engineering processes (column 4).

- The Mathematical-based Engineering Models (columns 5 thru 9) consist of a main program for each power plant (column 5). This program is formed by combining other programs which simulate the main equipment (column 6) of the plant. In turn these main programs that made up of smaller programs which simulate the main components (column 7) which contain the required procedures (column 8) to execute the calculations using the mathematical formulas for the respective processes (column 9). The details of the formulas are listed in the Appendix section of this thesis.

The main intention of Tables 6.8-6.15 is to illustrate the link between the ontology and the engineering models. The modelling system maintains a consistent taxonomy. Each class or subclass in the ontology has a corresponding class or procedure in Java. In addition, Tables 6.8-6.15 illustrate the reusability of both the ontologies and the engineering models. For example, while the system models three different kinds of gas power plants (Table 6.9) - simple cycle, combined cycle and heat and power - the ontologies and engineering models to represent the plants are similar from the main equipment and main program levels downwards. Simply by combining different main equipments (column 2) and main programs (column 5), different types of gas power plants can be represented and simulated.

Table 6.8. Relationship between the ontology and the engineering model for *Coal Power Plant*.

Energy Policy Ontology			Mathematical-based Engineering Model (Coal Power Plant)				
Subclass of plant_Generation	Subclass of manmade_Objects		Subclass of natural_Laws	Java Class (Program)	Java Class		Formulas (see Appendix)
	Subclass of other_Objects				Class (Program)	Method (Procedure)	
coal_Power_Plant	coal_Steam_Boiler	combustor	stoichiometric_Combustion	steam_Boiler_AuxiliaryBoiler.java	combustorCoalBoiler.java	stoichiometric_Combustion	A.1
		steam_Generator	IHOV_steam_Generator		steam_Generation.java	latentHeatOfVaporisation	A.2
	steam_Turbine	steam_Rotor	net_Work_Out	steam_Turbine.java	steam_Rotor.java	net_Work_Out	A.4
	generator	electrical_Generator	electricity_Generation	coal_Power_Plant.java	electricity_Generation.java	generator	G.1
	pump	pump_Rotor	net_Work_Out		pump_Rotor.java	net_Work_Out	A.6
	cold_Exchanger	-	zerothLaw		cold_Exchanger.java	zerothLaw	A.5
	water_Treatment	-	filter_osmosis_desalination		water_Treatment.java	waterBoilerInTreatment	A.8.1 A.8.2

This table illustrates the relationship between the Ontology and the Mathematical Model

Each object (Subclass) in the Ontology has a corresponding program (Java Class) or subroutine within the program (Java Method) in the Mathematical Model

The mathematical equation is stored as a subroutine (Method) in the program (Java Class).

The mathematical equations are listed in the Appendix

Table 6.9. Relationship between the ontology and the engineering model for *Gas Power Plant*.

Energy Policy Ontology			Mathematical-based Engineering Model (<i>Gas Power Plant</i>)				
Subclass of plant_Generation	Subclass of manmade_Objects		Subclass of natural_Laws	Java Class (Program)	Java Class		Formulas (see Appendix)
	Subclass of other_Objects	Subclass of engineering_Process	Class (Program)		Method (Procedure)		
gas_Power_Plant	gas_Turbine	compressor	air suction	gasTurbine. .java	compressor. .java	airSuction	B.1
		combustor	stoichiometric Combustion		combustorGasOil .java	stoichiometric Combustion	B.2
		power turbine	net_Work_Out		powerTurbine .java	net_Work_Out	B.3
	steam_Boiler	steam_Generator	IHOV_steam_Generator	steam_Boiler.WHBBBoiler .java	steam_Generator .java	latentHeatOfVaporisation	A.3
	steam_Turbine	steam_Rotor	net_Work_Out	steam_Turbine .java	steam_Rotor .java	net_Work_Out	A.4
	generator	electrical_Generator	electricity_Generation	gas_Power_Plant .java	electricity_Generation .java	generator	AG.1 BG.1
	pump	pump_Rotor	net_Work_Out		pump_Rotor .java	net_Work_Out	A.6
	heat_Exchanger	-	zerothLaw		heat_Exchanger .java	zerothLaw	A.7
	cold_Exchanger	-	zerothLaw		cold_Exchanger .java	zerothLaw	A.5
	water_Treatment	-	filter_osmosis_ desalination	-	water_Treatment .java	waterBoilerInTreatment	A.8.1 A.8.2
See Table 6.8 for description of table							
There are three configurations under gas_Power_Plant (i.e. Simple Cycle, Combined Cycle, Heat and Power). The objects which make up the different configurations are list below: - Simple Cycle is made up of the objects gas_Turbine and generator - Combined Cycle is made up of the objects gas_Turbine, steam_Boiler, steam_Turbine, generator, pump, cold_Exchanger, water_Treatment - Heat and Power is made up of the objects gas_Turbine, steam_Boiler, generator, heat_Exchanger, pump, cold_Exchanger, water_Treatment							

Table 6.10. Relationship between the ontology and the engineering model for *Oil Power Plant*.

Energy Policy Ontology			Mathematical-based Engineering Models (<i>Oil Power Plant</i>)				
Subclass of plant_Generation	Subclass of manmade_Objects		Subclass of natural_Laws	Java Class	Java Class		Formulas (see Appendix)
	Subclass of other_Objects	Subclass of engineering_Process			Class (Program)	Method (Procedure)	
oil_Power_Plant	gas_Turbine	compressor	air suction	gasTurbine.java	compressor.java	airSuction	B.1
		combustor	stoichiometric_Combustion		combustorGasOil.java	stoichiometric_Combustion	B.2
		power turbine	net_Work_Out		powerTurbine.java	net_Work_Out	B.3
	steam_Boiler	steam_Generator	IHOV_steam_Generator	steam_Boiler.WHBBoiler.java	steam_Generator.java	latentHeatOfVaporisation	A.3
	steam_Turbine	steam_Rotor	net_Work_Out	steam_Turbine.java	steam_Rotor.java	net_Work_Out	A.4
	generator	electrical_Generator	electricity_Generation	oil_Power_Plant.java	electricity_Generation.java	generator	AG.1 BG.1
	pump	pump_Rotor	net_Work_Out		pump_Rotor.java	net_Work_Out	A.6
	heat_Exchanger	-	zerothLaw		heat_Exchanger.java	zerothLaw	A.7
	cold_Exchanger	-	zerothLaw		cold_Exchanger.java	zerothLaw	A.5
	water_Treatment	-	filter_osmosis_desalination		water_Treatment.java	waterBoilerInTreatment	A.8.1 A.8.2

See Table 6.8 for description of table

There are two configurations under oil_Power_Plant (i.e. Simple Cycle, Combined Cycle, Heat and Power). The objects which make up the different configurations are list below:

- Combined Cycle is made up of the objects gas_Turbine, steam_Boiler, steam_Turbine, generator, pump, water_Treatment, cold_Exchanger
- Heat and Power is made up of the objects gas_Turbine, steam_Boiler, generator, heat_Exchanger, pump, water_Treatment, cold_Exchanger

Table 6.11. Relationship between the ontology and the engineering model for *Wind Power Plant*.

Energy Policy Ontology				Mathematical-based Engineering Models (<i>Wind Power Plant</i>)				
Subclass of manmade_Objects		Subclass of natural_Laws		Java Class	Java Class		Formulas (see Appendix)	
Subclass of plant_Generation	Subclass of other_Objects	Subclass of engineering_Process			Class (Program)	Method (Procedure)		
wind_Power_Plant	wind_Turbine	wind_TurbineRotor	net_Work_Out	wind_Power_Plant.java	wind_TurbineNacelle	wind_Rotor.java	net_Work_Out	C.1
		wind_TurbineNacelle	electricity_Generation			electricity_Generation.java	generator	CG.1
See Table 6.8 for description of table								

Table 6.12. Relationship between the ontology and the engineering model for *Hydroelectric Power Plant*.

Energy Policy Ontology				Mathematical-based Engineering Models (<i>Hydroelectric Power Plant</i>)				
Subclass of manmade_Objects		Subclass of natural_Laws		Java Class	Java Class		Formulas (see Appendix)	
Subclass of plant_Generation	Subclass of other_Objects	Subclass of engineering_Process			Class (Program)	Method (Procedure)		
water_Power_Plant	water_Turbine	water_TurbineRotor	net_Work_Out	hydroelectric_Power_Plant.java	-	water_TurbineNacelle.java	net_Work_Out	E.1
		electrical_Generator	electricity_Generation		-	electricity_Generation.java	generator	EG.1
See Table 6.8 for description of table								

Table 6.13. Relationship between the ontology and the engineering model for *Solar Power Plant*.

Energy Policy Ontology				Mathematical-based Engineering Models (<i>Solar Power Plant</i>)			
Subclass of plant_Generation	Subclass of manmade_Objects		Subclass of natural_Laws	Java Class	Java Class		Formulas (see Appendix)
	Subclass of other_Objects		Subclass of engineering_Process		Class (Program)	Method (Procedure)	
solar_Power_Plant	solar_Array	pV_Cell	net_Work_Out	solarArray .java	pVCell .java	photovoltaic	D.1
	solar_Concentrator	dish_Mirrors	electricity_Generation	solarConcentrator .java	pVMirror .java	photovoltaic	D.2
		steam_Generator	IHOV_steam_Generator	steam_Boiler.WHBBoiler .java	steam_Generation .java	latentHeatOfVaporisation	A.3
	steam_Turbine	steam_Rotor	net_Work_Out	steam_Turbine .java	steam_Rotor .java	net_Work_Out	A.4
	generator	electrical_Generator	electricity_Generation	electrical_Generator .java	generator .java	generator	AG.1
	pump	pump_Rotor	net_Work_Out	pump .java	pump_Rotor .java	net_Work_Out	A.6
	cold_Exchanger	-	zerothLaw	-	cold_Exchanger .java	zerothLaw	A.5
	water_Treatment	-	filter_osmosis_desalination	-	water_Treatment .java	waterBoilerInTreatment	A.8.1 A.8.2

See Table 6.8 for description of table

Table 6.14. Relationship between the ontology and the engineering model for *Biomass Power Plant*.

Energy Policy Ontology				Mathematical-based Engineering Model (<i>Biomass Power Plant</i>)			
Subclass of plant_Generation	Subclass of manmade_Objects		Subclass of natural_Laws	Java Class (Program)	Java Class		Formulas (see Appendix)
	Subclass of other_Objects		Subclass of engineering_Process		Class (Program)	Method (Procedure)	
biomass_Power_Plant	biomass_Plantation *	-	solar_Energy_Equation (photosynthesis)	-	photosynthesis.java	biomass	F.1
		-		-	photosynthesis.java	biomassEFB	F.2
	biomass_Steam_Boiler	combustor	stoichiometric_Combustion	steam_Boiler.AuxiliaryBoiler.java	combustorCoalBoiler.java	stoichiometric_Combustion	A.1
		steam_Generator	IHOV_steam_Generator		steam_Generation.java	latentHeatOfVaporisation	A.2
	steam_Turbine	steam_Rotor	net_Work_Out	steam_Turbine.java	steam_Rotor.java	net_Work_Out	A.4
	generator	electrical_Generator	electricity_Generation	electrical_Generator.java	generator.java	generator	AG.1
	pump	pump_Rotor	net_Work_Out	pump.java	pump_Rotor.java	net_Work_Out	A.6
	cold_Exchanger	-	zerothLaw	-	cold_Exchanger.java	zerothLaw	A.5
	water_Treatment	-	filter_osmosis_desalination	-	water_Treatment.java	waterBoilerInTreatment	A.8.1 A.8.2
	See Table 6.8 for description of table						

* biomass_Plantation is a subclass of upstream_Facilities and not other_Objects

Table 6.15. Relationship between the ontology and the engineering model for *Nuclear Power Plant*.

Energy Policy Ontology				Mathematical-based Engineering Model (<i>Nuclear Power Plant</i>)			
Subclass of plant_Generation	Subclass of manmade_Objects		Subclass of natural_Laws	Java Class (Program)	Java Class		Formulas (see Appendix)
	Subclass of other_Objects		Subclass of engineering_Process		Class (Program)	Method (Procedure)	
nuclear_Power_Plant	nuclear_Reactor *	-	nuclear_Reaction	-	-	-	-
		steam_Generator	IHOV_steam_Generator	steam_BoilerWHBBoiler.java	steam_Generation.java	latentHeatOfVaporisation	A.3
	steam_Turbine	steam_Rotor	net_Work_Out	steam_Turbine.java	steam_Rotor.java	net_Work_Out	A.4
	generator	electrical_Generator	electricity_Generation	electrical_Generator.java	generator.java	generator	AG.1
	pump	pump_Rotor	net_Work_Out	pump.java	pump_Rotor.java	net_Work_Out	A.6
	cold_Exchanger	-	zerothLaw	-	cold_Exchanger.java	zerothLaw	A.5
	water_Treatment	-	filter_osmosis_desalination	-	water_Treatment.java	waterBoilerInTreatment	A.8.1 A.8.2
See Table 6.8 for description of table							
* mathematical model for nuclear reaction not programmed. User must instead specify the expected heat production rate (Q_{reaction}) for the waste heat boiler.							

6.3 Application of the Semantic Representation of the Energy Policy Knowledge Domain

The application of the semantic representation of the energy policy knowledge domain is done through logic inference. This inference uses a logic reasoner, FACT++ and the mathematical software package Mathematica; both of which have been integrated into the *Prototype Energy Modelling System* through the use of Java.

As described in Section 5.2.2, the *Knowledge Inference and Expression Builder Module* employs a reasoner which (i) uses axioms stored in the *Energy Model Construction Ontology* to infer the required policy variables (i.e. electricity generation plants) for the construction of the mathematical expressions (i.e. objective function and constraints) and (ii) uses axioms stored in the *Energy Scenarios Ontology* to infer the required policy variables for the construction of the *Technology Preference Expression*.

In addition, the *Knowledge Inference and Expression Builder Module* loads the associated constants (e.g. plant generation installed capacity, CO₂ emissions, steam production capacity) and values for the constraints from the *Energy Policy Ontology*. Where necessary, the values that are not stored in the ontology are calculated, for example the total electricity demand which is based on both the population and the state of industrialisation in the specific simulation year. Mathematical expressions are formulated and then sent to the *Linear Programming Module*. The mathematical technique to solve the mathematical expression has been previously discussed in Section 5.2.3. The solution is then plotted.

This subsection demonstrates the application of the semantic representation within the *Prototype Energy Modelling System*. First, the mathematical expressions used for optimisation are presented. Second, the formulation of the Technology Preference Expressions is demonstrated. Third, examples of the formulation of the

objective function and constraints are provided. Fourth, a set of optimised electricity generation mixes for a given set of scenario drivers is shown.

6.3.1 Mathematical Expressions for the Sets of Linear Programming and System of Inequalities

As described in Section 4.4.4.IV, two sets of expressions are formulated: a set of linear programming expressions and a set of system of inequalities.

I. Set of Linear Programming Expressions

The set of linear programming expressions consists of:

- An objective function comprising of either the capital cost or the total cost for an electricity generation plant or a multiobjective function (explained in Section 5.2.3.II) comprising of either the capital cost or the total cost for an electricity generation plant and the life cycle energy payback of the electricity generation process. The capital cost objective function is shown in Expression 6.4.1, the total cost objective function is shown in Expression 6.4.2 and the life cycle energy payback objective function is shown in Expression 6.4.3. The variables teG and teP are the amount of electricity generation capacity and the amount of electricity production for each of the technologies in the generation mix that fulfils all the constraints in the set of linear programming expressions.

$$\text{minimise} \left(\sum_{i=1}^n c_i * teG_i \right)$$

where (6.4.1)

c = capital cost per MW of generation capacity (GBP£/MW)

i = electricity generation technology (e.g. coal, wind, solar)

teG = total electricity generation capacity in linear programming solution (MW)

$$\text{minimise} \left(\sum_{i=1}^n c_i * teG_i + om_i * teP_i \right)$$

where (6.4.2)

om = operation cost per MW of production (GBP£/MW)

teP = total electricity production (MW)

$$\text{maximise} \left(\sum_{i=1}^n lcep_i * teP_i \right)$$

where (6.4.3)

$lcep$ = life cycle energy payback per MW of production (1/MW)

- A CO₂ emissions constraint which limits electricity generation from fossil fuel installations as shown in Expression 6.4.4.

$$\sum_{j=1}^n teP_j * CP_j \leq CpC * pop$$

where

teP = total electricity production (MW) (6.4.4)

j = electricity generation technology (e.g. coal)

CP = CO₂ emission per MW of electricity production (tons/MW)

CpC = CO₂ emissions per capita allowance (tons/capita)

pop = population size (capita)

- A minimum electricity daily demand constraint based on electricity consumption by end users (consumers) as shown in Expression 6.4.5. The

total electricity demand per capita (teDp) is based on the societal state selected in the GUI (see Figure 6.2). The teDp values are taken from the societal state values published by Kraussman [165].

$$\sum_{i=1}^n teP_i \geq teDp * pop \quad (6.4.5)$$

where

teDp = total electricity demand per capita (MW/capita)

II. Set of System of Inequalities

The resulting capital cost, total cost and life cycle energy payback values from the linear programming solver are used in the set of inequalities which consists of:

- A modified objective function for either capital cost or total cost for electricity generation or dual modified objective functions (explained in Section 5.2.3.II) for either capital cost or total cost for electricity generation installations and a life cycle energy payback of electricity generation process. The modified objective function for capital cost is shown in Expression 6.4.6, for total cost is shown in Expression 6.4.7, and for life cycle energy payback is shown in Expression 6.4.8. The variables teGG and tePP are the amount of electricity generation capacity and the amount of electricity production for each of the technologies in the generation mix that fulfils all the constraints in the set of system of inequalities.

$$\text{minimise} \left(\sum_{i=1}^n c_i * teGG_i - \sum_{i=1}^n c_i * teG_i \right) \quad (6.4.6)$$

where

$teGG$ = total electricity generation capacity in system of inequalities (MW)

$$\text{minimise} \left(\sum_{i=1}^n c_i * teGG_i + om_i * tePP_i - \sum_{i=1}^n c_i * teG_i + om_i * teP_i \right) \quad (6.4.7)$$

where

$tePP$ = total electricity production in system of inequalities (MW)

$$\text{minimise} \left(\sum_{i=1}^n lcep_i * tePP_i - \sum_{i=1}^n lcep_i * teP_i \right) \quad (6.4.8)$$

- The *Technology Preference Expression* (see Section 6.3.2.I for example)
- A CO₂ emissions constraint which limits electricity generation from fossil fuel installations (Expression 6.4.9).

$$\sum_{j=1}^n tePP_j * CP_j \leq CpC * pop \quad (6.4.9)$$

- A life cycle energy payback constraint (if a single objective function is selected) as shown in Expression 6.4.10. The total life cycle energy payback for the set of inequalities must exceed the total life cycle energy payback for the set of linear programming because the latter's value is based on the cheapest (and possibly least efficient) electricity generation configuration.

$$\sum_{i=1}^n lcep_i * tePP_i \geq \sum_{i=1}^n lcep_i * teP_i \quad (6.4.10)$$

- A minimum daily electricity demand constraint based on electricity consumption by consumers as shown in Expression 6.4.11. The total electricity demand per capita (teDp) is based on the societal state selected in the GUI (see Figure 6.2). The teDp values are taken from the societal state values published by Kraussman [165].

$$\sum_{i=1}^n tePP_i \geq teDp * pop \quad (6.4.11)$$

- A minimum nightly electricity demand constraint based on electricity consumption by consumers as shown in Expression 6.4.12. The night time modifier (nMod) is specified in the GUI (see Figure 6.3).

$$\sum_{i=1}^n tePP_i \geq teDp * pop * nMod \quad (6.4.12)$$

nMod = night time electricity demand modifier

- A minimum heat (steam) demand constraint based on electricity consumption by consumers as shown in Expression 6.4.13. The heat demand modifier (hpMod) is specified in the GUI (see Figure 6.3).

$$\sum_{k=1}^n teHPP_k \geq teDp * pop * hpMod$$

where (6.4.13)

k = electricity generation technology (e.g. gas heat and power)

$teHPP$ = total heat production (MW)

$hpMod$ = heat demand modifier

- A minimum reserve electricity generation demand constraint (if selected)

as shown in Expression 6.4.14. The reserve power modifier ($resMod$) is specified in the GUI (see Figure 6.3).

$$\sum_{m=1}^n teRPP_m \geq teDp * pop * resMod$$

where (6.4.14)

m = electricity generation technology (i.e. gas, coal, oil, nuclear)

$teRPP$ = total reserve electricity production (MW)

$resMod$ = reserve power modifier

- A maximum intermittent (renewable energy) electricity supply constraint

as shown in Expression 6.4.15. The renewable energy modifier ($rewMod$) is specified in the GUI (see Figure 6.3).

$$\sum_{p=1}^n teREPP_p \leq teDp * pop * rewMod$$

where (6.4.15)

p = electricity generation technology (i.e. wind, solar)

$teREPP$ = total intermittent electricity production (MW)

$rewMod$ = renewable energy modifier

6.3.2 Formulation of the Mathematical Expressions

To facilitate a better interpretation of the mathematical expressions discussed in this section, the variables and constants used are tabulated in Table 6.16.

Table 6.16. Variables and constants used in Table 6.18 and Figures 6.14, 6.15 and 6.16.

Electricity Generation Technology	Variable Name	Electricity Production per plant (MW)	Capital Cost (GBP / MW)	Electricity Capacity per plant (MW)	CO₂ Emissions per plant (tons/hour)
Wind Turbine	w	35	675	117	0
Coal Boiler	cl	961	780	1068	824
Gas Turbine	g	727	333	1276	287
Gas Cogen	gh	880	400	1276	347
Gas Combined Cycle	gc	1306	400	1451	453
Oil Combined Cycle	oc	128	920	142	115
Oil Cogen	oh	86	781	125	88
Solar Array	sa	4	5750	38	0
Solar Concentrator	sc	6	5750	50	0
Hydroelectric	wt	97	1600	359	0
Biomass Boiler	bm	33	1575	39	0
Nuclear Boiler	n	423	2200	517	0

This section will demonstrate the automatic formulation of four mathematical expressions which are the *Technology Preference Expression*, objective function, supply constraint and demand constraint.

I. Technology Preference Expression

To demonstrate the formulation of the *Technology Preference Expression*, two different sets of scenario drivers have been selected. Table 6.17 lists the selected policy goals and targets.

Table 6.17. List of policy goals and the selected options for Scenarios 1 and 2.

Energy Policy Goals (Qualitative)		Scenario 1	Scenario 2
Political	Level of energy supply security risk	Low	Medium
Society	Minimisation of public health risk	Low Priority	Medium
	Maximisation of new jobs creation	Low Priority	Average
	Importance of social acceptability	Low	Low
Energy Policy Goals (Quantitative)		Scenario 1	Scenario 2
Economic	Minimisation of Capital Cost	High Priority	High Priority
	Maximisation of life cycle energy payback	Low Priority	High Priority
Environmental	Future reduction of CO ₂ emissions	Minor	Significant
	Land utilisation	Low	Medium
	Maximisation of water conservation	Minimum	Average

Table 6.18 lists the formulated *Technology Preference Expression* for the two scenarios using both the *Direct* and the *Random Probability* formulation methods. For the *Direct* formulation method, the order of preference for each generation technology (column 2) and the resulting expression are shown (column 3). For the *Random Probability* formulation method, only the probability of occurrence (in percentages) for each generation technology is shown (in brackets in column 2) as the formulated *Technology Preference Expression* changes randomly from iteration to iteration based on the percentages associated each generation technology.

Table 6.18. *Technology Preference Expressions* for Scenarios 1 and 2.

Scenario	Technology Preference Order	Technology Preference Expression
1	Coal (13%)	$0 < 423*n \leq 6*sc \leq 4*sa \leq 35*w \leq 97*wt < 128*oc+86*oh \leq 33*bm \leq 727*g+880*gh+1306*gc < 961*cl$
	Gas (12%), Biomass (12%), Oil (12%),	
	Water (10%), Wind (10%) Solar (10%), Nuclear (10%)	
2	Water (12%), Wind (12%) Solar (12%), Gas (12%), Biomass (12%)	$0 < 961*cl < 128*oc+86*oh < 423*n < 33*bm \leq 727*g+880*gh+1306*gc \leq 6*sc \leq 4*sa \leq 35*w \leq 97*wt$
	Nuclear (11%)	
	Oil (9%)	
	Coal (7%)	

The formulation of the *Technology Preference Expression* assumes that the preference of the user is proportional to the quantity of generated

electricity using a given technology (the constants for each variable is the electricity production per plant - see Table 6.16). Other basis of preference such as installed electricity generation capacity are also possible.

II. Objective Function

As explained in Section 5.2.2, axioms are used to infer the variables (policy variables as explained in Table 4.2) necessary to formulate the modified objective function for capital cost, total cost or life cycle energy payback. For capital cost the constants required are capital cost per MW and installed electricity generation capacity. Plant generation capacity is used for capital cost as such costs are incurred irrespective of actual generation. For total cost, a combination of electricity capacity for the capital component and electricity generation for the operating component is used. For life cycle energy payback, the constant required are life cycle energy payback per MW and electricity generated. Electricity generated is used as the benefit of life cycle efficiency is only realised during electricity production.

Figure 6.14 demonstrates the axiom that is used to formulate the modified objective function in the systems of inequalities (Expression 6.4.6). In the English equivalent, the axiom queries the capital cost for each electricity generation plant based on a calculation of its capital cost per MW and electricity generation capacity in MW. On the LHS, the data properties for the constants are *electricity_Capacity* and *capital_Cost* while the variable to be inferred are the electricity generation facilities. On the RHS, the value of the cost constraint was calculated from the value determined in the linear programming phase of the optimisation as described in Section 5.2.3.

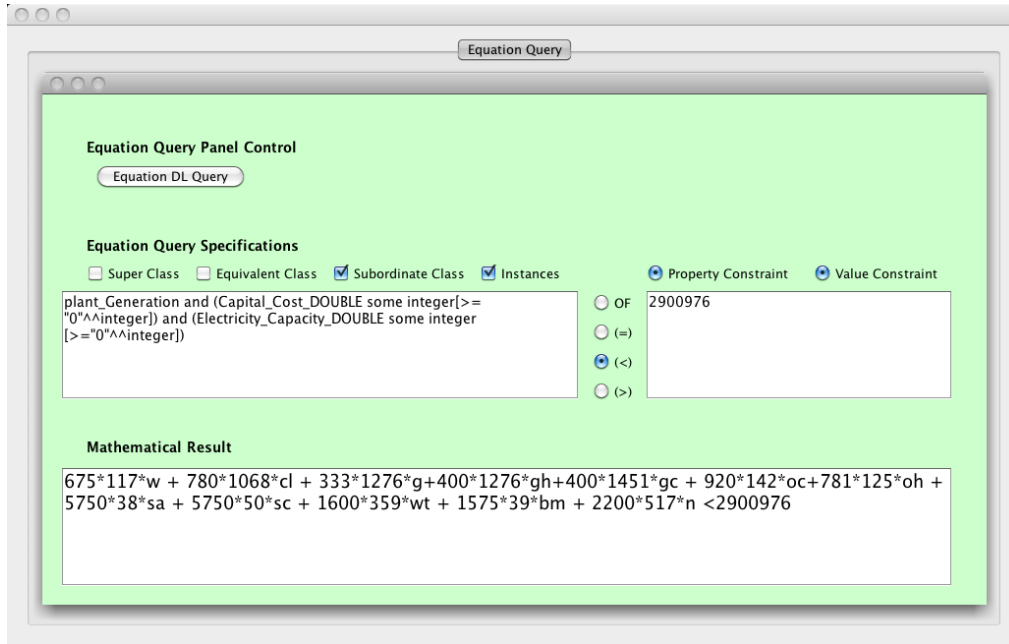


Figure 6.14. Capital cost objective function.

III. Supply Constraint

An example of a supply constraint which limits the maximum electricity generation from specific sources is that of maximum allowable CO₂ emissions. Figure 6.15 demonstrates the axiom used for the formulation of the LHS of this supply constraint. In the English equivalent, the axiom queries the amount of CO₂ emitted from electricity generation technologies which generate CO₂ from the use of the stoichiometric combustion processes where fuel is consumed. On the LHS, the required data properties for the constant is *CO2_Emissions* (which are stored in the ontology on a per plant basis in MT/hour) while the variables are the inferred electricity generation plants which employ the stoichiometric combustion process. On the RHS, the maximum CO₂ emissions constraint (in MT/hour) is calculated based on the population multiplied by the CO₂ per capita.

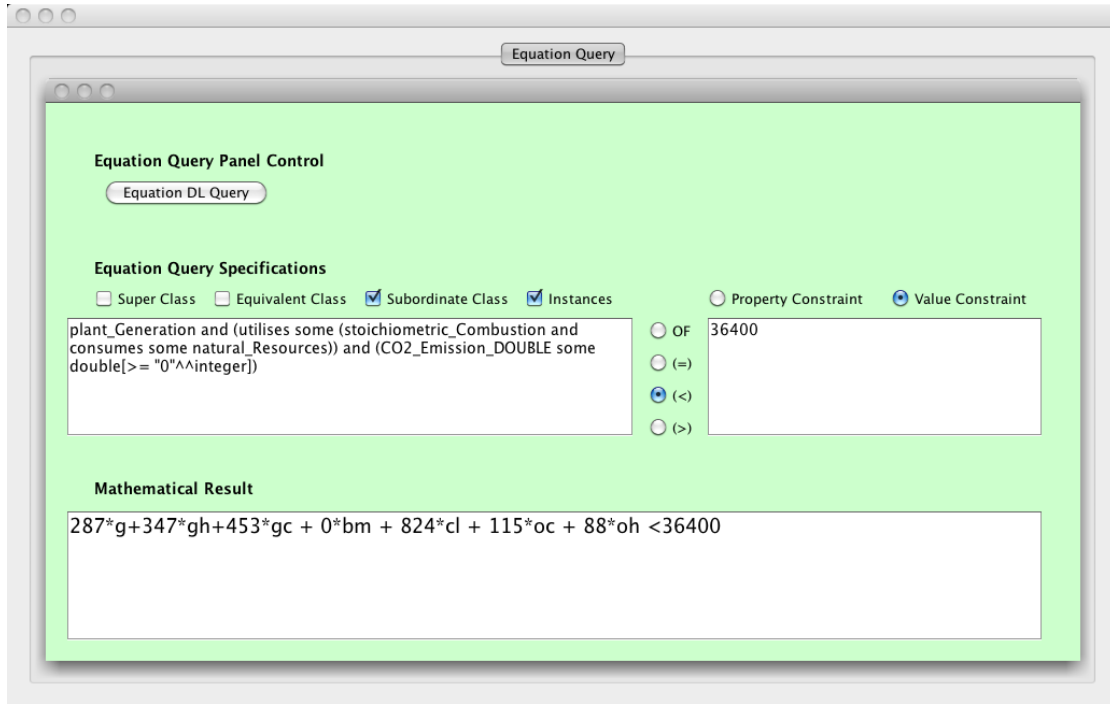


Figure 6.15. Maximum allowable CO2 emissions supply constraint.

IV. Demand Constraint

An example of a demand constraint which mandates the minimum electricity generation from specific sources is that of minimum electricity demand. Figure 6.16 demonstrates the axiom used for the formulation of the LHS of this demand constraint. In the English equivalent, the axiom queries the amount of electricity that can be generated from electricity generation technologies based on its electricity production which is a function of its installed electricity capacity and plant capacity. On the LHS, the required data property for the constants is *Electricity_Production* (which is stored in the ontology on a per plant basis in MW) while the variables are the inferred electricity generation facilities. On the RHS, the required data property is *Total_New_Electricity_Demand* which is calculated based on the population

multiplied by the societal state as defined by the user. The data for societal state originates from Kraussman [165].

The screenshot shows a software window titled "Equation Query". Inside, there is a sub-panel titled "Equation Query Panel Control" with a button labeled "Equation DL Query". Below this is the "Equation Query Specifications" section, which includes several checkboxes: "Super Class", "Equivalent Class", "Subordinate Class" (checked), and "Instances" (checked). To the right of these are radio buttons for "Property Constraint" (selected) and "Value Constraint". The main query area contains the text: "plant_Generation and (Electricity_Production_DOUBLE some integer{>= '0'^^integer})". To the right of the query is a list of radio buttons for comparison operators: "OF", "(=)", "<", and ">" (selected). The rightmost field contains the text "Total_New_Electricity_Generation_INT". At the bottom, the "Mathematical Result" section displays the equation: $35*w + 961*cl + 727*g + 880*gh + 1306*gc + 128*oc + 86*oh + 4*sa + 6*sc + 97*wt + 33*bm + 423*n > 20020.0$.

Figure 6.16. Minimum electricity generation demand constraint.

6.3.3 Optimisation of Electricity Generation Mix Based on Logic

Inference of a Set of Energy Policy Goals and Targets

This subsection demonstrates the ability of the program to optimise an electricity generation mix based on a set of scenario drivers. The system has been configured with a simulation of the United Kingdom with a population of 60 million with the energy policy goals and targets of both Scenario 1 and 2 (Table 6.17). The data is taken from the UK MARKAL database. The GDP linked energy consumption and population growth rates have been set at 1% for each 10-year period and the simulation has been set to run for 20 years. A plant decommissioning rate of 33% per 10 year period has been specified for a plant lifespan of 30 years with the exception of hydroelectric generation which has been set for continuous lifespan throughout the simulation period..

Four sets of results are presented:

- A Linear programming only optimisation which consist of expressions (6.4.1, 6.4.4 and 6.4.5) for Scenarios 1 and 2.
- A system of inequalities optimisation which consist of expressions (6.4.6, 6.4.9, 6.4.10, 6.4.11, 6.4.12, 6.4.13, 6.4.14 and 6.4.15) for Scenarios 1 and 2 using the *Direct* formulation method for the *Technology Preference Expression*.
- A multiobjective (two objectives) optimisation which consist of expressions (6.4.6, 6.4.8, 6.4.9, 6.4.11, 6.4.12, 6.4.13, 6.4.14 and 6.4.15) for Scenario 2 using the *Direct* formulation method for the *Technology Preference Expression*.
- A system of inequalities optimisation which consist of expressions (6.4.6, 6.4.9, 6.4.10, 6.4.11, 6.4.12, 6.4.13, 6.4.14 and 6.4.15) for Scenario 1 using the

Random Probability formulation method for the *Technology Preference Expression*.

I. Linear Programming Optimisation for Scenario 1 and 2

Figures 6.17 and 6.18 display the linear programming optimisation results for cost with only a selected set of constraints. The *Technology Preference Expression* is not considered in the optimisation. This optimisation represents ideal conditions which are not achievable in real life. The optimisation will always favour the electricity generation plant with the lowest cost. For the UK MARKAL dataset on which this simulation is based on, gas based electricity generation is the cheapest.

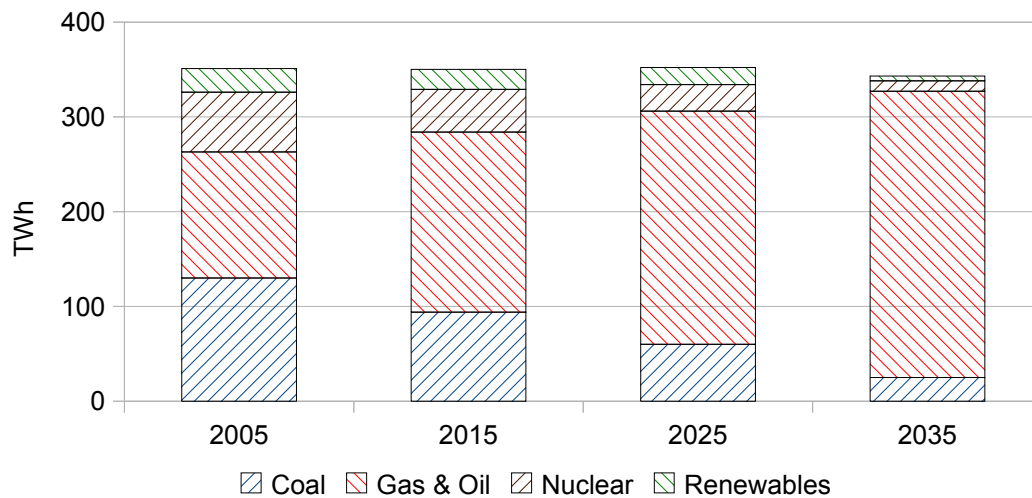


Figure 6.17. Linear programming optimised generation mixes in TWh for Scenario 1.

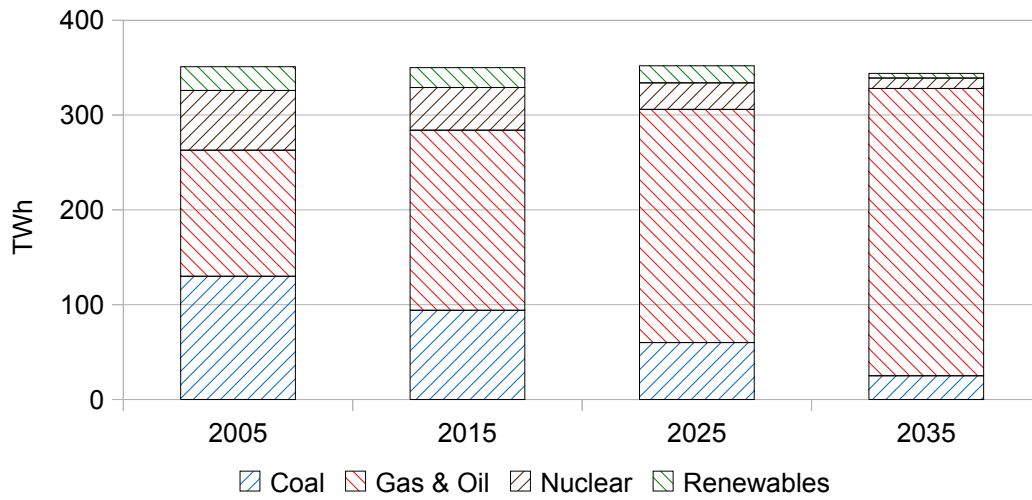


Figure 6.18. Linear programming optimised generation mixes in TWh for Scenario 2.

II. Optimisation for Scenario 1 and 2 using the Direct Formulation

Method for the Technology Preference Expression

Figures 6.19 and 6.20 displays the results from the cost optimisation using the complete set of constraints including the *Technology Preference Expression*. The 'optimised cost' value from the linear programming solution (Expression 6.4.1) is used as the initialisation point for Expression 6.4.6 of the set of inequalities.

In the case of Scenario 1, the effect of the *Technology Preference Expression* on the optimisation mix is the dominance of coal as a fuel source. Coal use increases because of the selected energy policy goals and targets. The final generation mix is different from the linear programming solution (Figure 6.17) where gas is the dominant source.

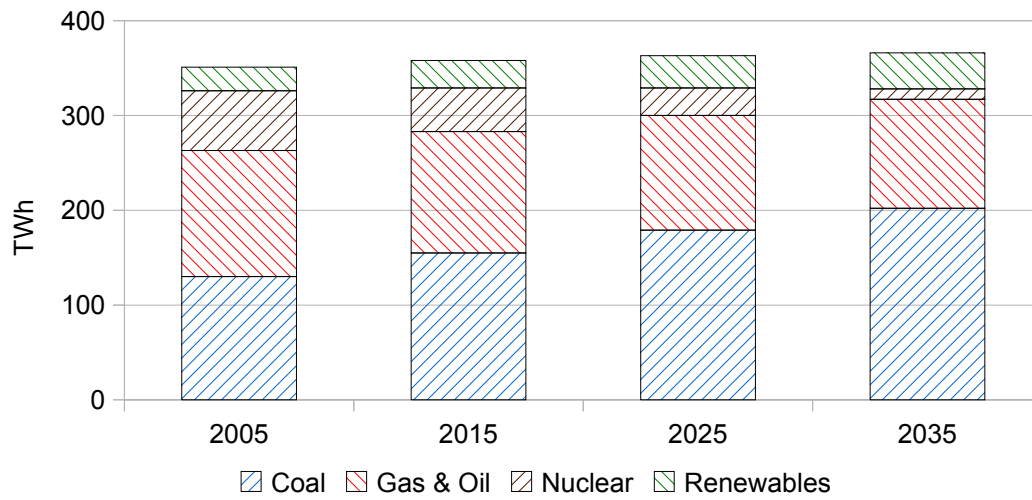


Figure 6.19. Optimised generation mixes in TWh for Scenario 1.

In the case of Scenario 2, the effect of the *Technology Preference Expression* on the optimisation mix results in a very different mix with renewables being the dominant source. The increase in renewable use is due to the effects of the selected energy policy goals and targets. This is different from the linear programming solution (Figure 6.18) where a gas based solution dominates.

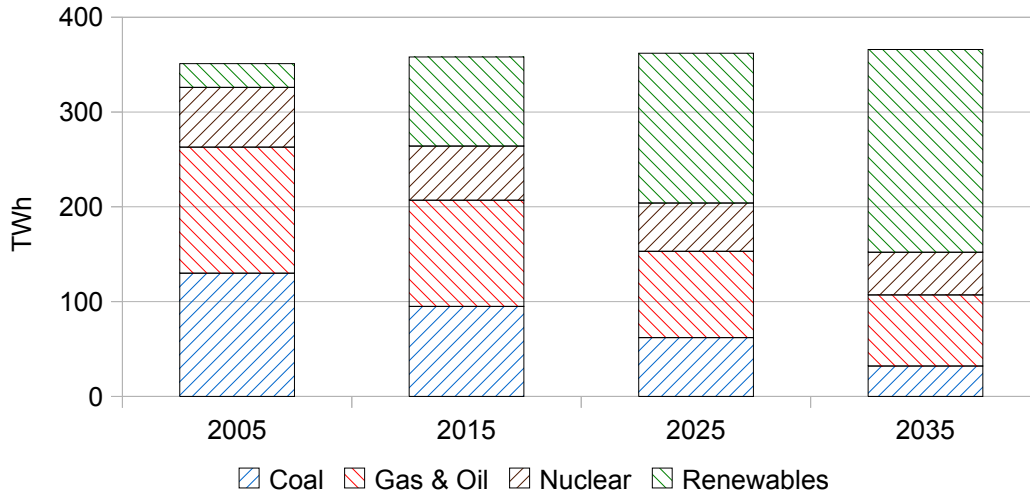


Figure 6.20. Optimised generation mixes in TWh for Scenario 2.

III. Multiobjective Optimisation for Scenario 2 using the *Direct Formulation Method for the Technology Preference Expression*.

Figure 6.21 displays the results from the optimisation of both cost and life cycle energy payback using the complete set of constraints using the *Direct* formulation method for the *Technology Preference Expression*. As explained in Section 5.2.3, the 'optimised' values for both cost and lifecycle energy payback from the linear programming solution (expression 6.4.1 and 6.4.3) are used as initialisation points for Expressions 6.4.6 and 6.4.8 of the set of inequalities. This solution represents the midpoint value between the minimisation of capital (or total) cost and the maximisation of life cycle energy payback. Compared to the cost optimised configuration in Figure 6.20, renewables, which have a higher life cycle energy payback, provide a greater contribution to the electricity generation mix. This difference in Figure 6.21

reflects the trade-off between capital cost and life cycle energy payback which differs from the cost only optimal point in Figure 6.20.

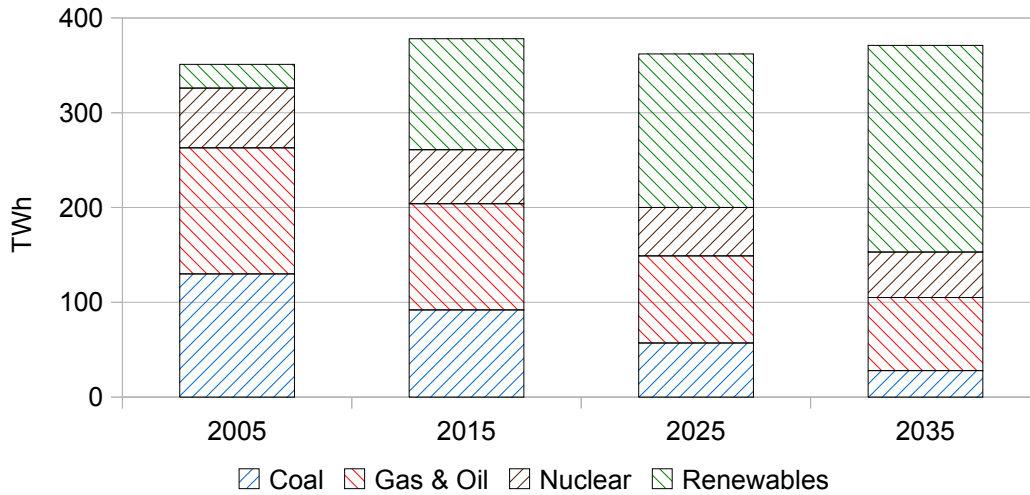


Figure 6.21. Two objective (multiobjective) optimised generation mixes in TWh for Scenario 2.

IV. Optimisation for Scenario 2 using the *Random Probability Formulation for the Technology Preference Expression*

Figures 6.22 and 6.23 display the results from cost optimisation using the complete set of constraints using the *Random Probability* formulation method for the *Technology Preference Expression*. The 'optimised cost' value from the linear programming solution (Expression 6.4.1) is used as an initialisation point for Expression 6.4.6 of the set of inequalities. Since the Technology Preference Expression changes from iteration to iteration, the results from Figure 6.22 and 6.23 reflects different generation mixes based on the same set of energy policy goals and targets.

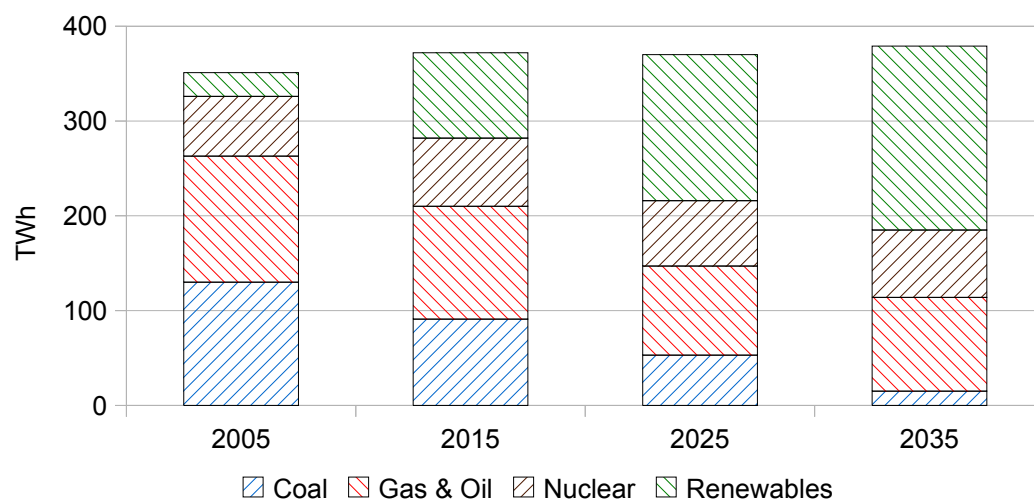


Figure 6.22. Optimised generation mixes in TWh for Scenario 2 using Random Probability formulation method for *Technology Preference Equation*.

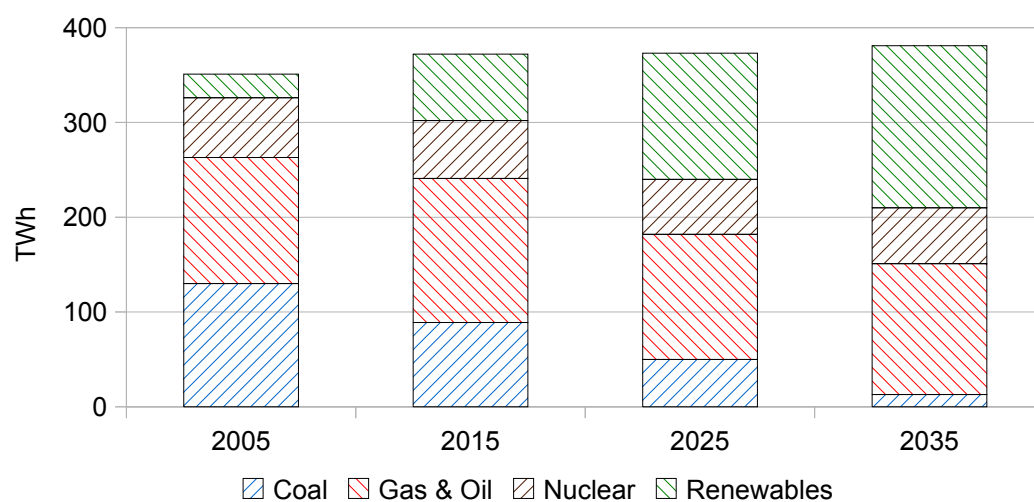


Figure 6.23. Optimised generation mixes in TWh for Scenario 2 using Random Probability formulation method for *Technology Preference Equation*.

6.4 Conclusions

This chapter has described the details of the *Prototype Energy Modelling System* which has been developed using the framework proposed in Chapter 4. The details of the main components of the *Prototype Energy Modelling System* have been discussed while the knowledge relevant to the energy policy domain and how it has been represented in the ontology has been presented.

This chapter has illustrated how through the use of logic inference, the *Prototype Energy Modelling System* can formulate expressions to be used as an input to the optimisation of an electricity generation mix. A significant aspect of this formulation is the ability of the modelling system to represent the objectives, goals and targets of the policy maker using mathematical expressions. This chapter has demonstrated that for a set of scenario drivers, an electricity generation mix that fulfils the constraints can be obtained.

In the next chapter, a case study is conducted using the *Prototype Energy Modelling System*.

Chapter 7 Case study

This chapter discusses a case study for an electricity generation mix in Malaysia using the *Prototype Energy Modelling System*. The purpose of the case study is to demonstrate the modelling system's capability to generate an *Energy Model* which can propose an electricity generation configuration that is inferred from the energy policy goals and targets specified by the user. The chapter consists of a brief introduction to the Malaysian energy policy (Section 7.1), an explanation of the data configuration for the case study (Section 7.2), a description of the three scenarios used (Section 7.3), a presentation of the results (Section 7.4) and a conclusion (Section 7.5).

7.1 Introduction to the Malaysian Energy Policy

Malaysia is a country located in South East Asia with a population of 27 million. A former British colony, the Federation of Malaysia consists of Peninsular Malaysia on the west and Sabah and Sarawak on the east. Although Malaysia recognises 1957 as its year of independence, the formation of Malaysia only occurred in 1963 when Sabah and Sarawak received their independence from Britain. As a country, Malaysia is blessed with an abundance of natural resources. Malaysia is a net exporter of Liquefied Natural Gas (LNG), crude oil, palm oil and rubber.

The Malaysian energy policy began as a reaction to the energy crises of the 1970s. These energy crises saw a shift of power and influence from the International Oil Companies (IOC) to the governments of the producer countries. In many cases, natural resources were established as the patrimony of the producer country; hence for example, oil production was the prerogative of governments rather than that of oil companies. A wave of nationalisation of oil assets occurred and a new concept of oil

production called the Production Sharing Contract (PSC) was established [166].

In Malaysia, the discovery of significant oil and gas resources in Peninsular Malaysia, coupled with a desire for the control of oil concessions in Sabah and Sarawak, resulted in the enactment of the Petroleum Development Act (PDA) in 1974 [167]. The PDA established Petroliaam Nasional Berhad (PETRONAS) as the national oil company and vested it with the responsibility for exploration, development, refining, processing, manufacturing, marketing and distribution of petroleum products. Soon thereafter, the National Petroleum Policy [168] was formulated in 1975 to guide and regulate the fast growing petroleum industry. Subsequently in 1979, an overall National Energy Policy [169] was developed with broad guidelines on long-term energy objectives and strategies. Its main objectives are to ensure adequate, secure and cost effective energy supplies, to promote the efficient use of energy and to minimise the negative impacts of energy utilisation.

The need to safeguard the depletion rate of oil and gas reserves resulted in the establishment of the National Depletion Policy in 1980 [169]. This policy mandates a maximum depletion rate of 3% which currently results in a reserve life of 16 years for oil and 70 years for gas. In addition, to improve energy security and to reduce the over dependence on oil, the Four Fuel Diversification Policy [170] was adopted in 1981 with the aim of increasing the use of gas, coal and hydroelectric sources for electricity generation. A further revision was made in 2001 with the Fifth Fuel Policy [171] which seeks to develop renewable energy, in particular biomass, biogas, municipal waste, solar and hydroelectric sources for the electricity generation industry.

The push for the further development of green technology, energy efficiency and energy security saw the introduction of new master plans in 2009 and 2010 respectively [171]. Preliminary plans include proposals for up to 5% of the total electricity generation to be from renewable sources by 2015 [172].

7.2 Data Configuration of the Prototype Energy Modelling System

To simulate the conditions in Malaysia, localised information and data have been stored in the ontology. Geographical data for solar radiation [173,174], wind velocity [175,176], maximum hydroelectric potential [177] and biomass type and availability [178] are required to recalibrate the electricity generation plants to reflect the conditions in Malaysia. The engineering models for solar array, solar concentrator, biomass, wind and hydroelectric sources were recalculated using the localised data and are summarised in Table 7.1 below.

Table 7.1. Geographical parameters, values, and applicable electricity generation plants for Malaysia.

Geographical Parameters	Value	Applicable Electricity Generation Plant
Solar Global Radiation (Solar Insolation)	4.83 kWh/m ²	Solar Array
Solar Direct Radiation	2.42 kWh/m ²	Solar Concentrator
Wind Velocity	8 m/s	Wind Turbine
Biomass Type	Palm Oil Empty Fruit Bunches	Biomass Fired Boiler with Rankine Cycle
Biomass Gross Heating Value (GHV)	18838 kJ/kg	
Maximum Hydroelectric Potential	123 TWh/year	Hydroelectric

Socio-economic data such as population, energy consumption per capita, electricity generation mix, percentage of electricity in energy consumption per capita, installed electricity generation mix and electricity generation demand which were required to reflect the conditions in Malaysia for the start of the scenario in 2005 were

obtained from reports by the Malaysian Economic Planning Unit [179,180]. Table 7.2 summarises the values used in the case study.

Table 7.2. Socio-economic parameters for Malaysia at start of case study in Year 2005.

Parameter	Value in Year 2005	Notes
Initial Population	26000000	-
Energy Consumption Per Capita	62.2GJ per capita	Final commercial energy demand basis
% of Electricity in Energy Consumption Per Capita	19.90%	Final commercial energy demand basis
Electricity Generation Mix	22% - Coal 70% - Gas 6% Hydroelectric 2% - Oil	-
Electricity Generation Demand (MW)	9360MW	From [180]
<i>Electricity Generation Peak Demand (MW)</i>	13779MW	From [180]

Population growth rates (Table 7.3) and electricity per capita growth rates (Table 7.4) for the period from 2005 - 2035 were estimated with reference to historical figures. Population growth rates on an average annual (compounded) growth rate (AAGR) has declined from 2.5% between 1991 and 2005 [181] to 2.3% between 1991 and 2010 [178]. For the three scenarios in this case study, this decline is maintained as it is viewed to be consistent with the general situation where the rate of population growth declines as a nation progresses from developing to developed status. Given these rates, the estimated population of Malaysia for the three scenarios is approximately 33 million by the year 2020 which is close to the government

estimation of 34 million [173].

Table 7.3. Historical and forecast population growth rates for Malaysia.

Parameter	1991 - 2005	1991 - 2010	2005 - 2015	2015 - 2025	2025 - 2035
Population Growth (AAGR) %	2.5	2.3	1.84	0.96	0.48

Electricity per capita growth rates on an average annual (compounded) growth rate (AAGR) have also decreased. While the period from 1995 to 2005 [180] saw a significant growth in electricity use, it is expected that the growth rate will decline as Malaysia transitions from a top tier developing nation to a developed nation. It is expected that by 2020, Malaysia should attain the status of a newly developed nation. The current government forecast of electricity generation of 18000MW (peak demand) in 2015 which is approximately 13000MW annual average demand based on a 40% difference with peak demand is close to the scenario values of 13400MW for annual average demand.

Table 7.4. Historical and forecast electricity growth rates for Malaysia.

Parameter	1995 - 2005	2005 - 2015	2015 - 2025	2025 - 2035
Electricity Consumption Per Capita Growth (AAGR) %	5.67	2.09	1.05	0.49

7.3 Scenarios

Three scenarios have been considered for this case study. The first is Scenario 1 BAU (Business as Usual) which represents the business as usual condition. Scenario 2 EcoFriend places a priority on efficiency, conservation and acceptability. Scenario 3 Nuclear explores an initiative to shift towards nuclear power as a major source of electricity generation. The selected policy goals in the three scenarios are intended to cover the range of issues that a future Malaysian energy policy must address.

Scenario 1 BAU reflects the BAU case as outlined in the 10th Malaysian Plan [172]. The plan calls for continued fossil fuel use with an introduction of renewable energy within the generation mix. Current plans target a renewable generation percentage of 5% by 2015. To better reflect the background of this plan, not all nine energy policy goals have been selected. The energy policy goals for future CO₂ emissions reductions, land utilisation, water conservation and social acceptability have been deactivated. As an Annex III country, Malaysia has no obligation to reduce CO₂ emissions under the Kyoto protocols. Given its tropical climate and 60% forest coverage, water and land resources remain in abundance in Malaysia. There is little societal opposition toward electricity generation installations.

Scenario 2 EcoFriend explores a priority on efficiency, conservation and acceptability. This mimics the general evolution of electricity generation in developed countries. It presupposes that as Malaysia evolves into an industrialised nation, her government and citizens will also adopt the attitudes and views of developed countries. In particular, this means an increased societal concern regarding CO₂ emissions and opposition towards both nuclear and coal energy sources. Greater emphasis on energy efficiencies and water utilisation is also expected in addition to greater conservation of tropical rain forests. As part of the drive towards a high income nation, increased emphasis is given towards the creation of jobs related to new

electricity generation technologies such as wind turbines, photovoltaic cells and third generation biomass processing.

Scenario 3 Nuclear represents a shift in approach towards electricity generation. It places nuclear power as a major source of electricity generation in Malaysia. As with Scenario 1 BAU, not all of the nine policies are used. The energy policy goals for future CO₂ emissions reduction, water conservation and new jobs creation have been deactivated. Even though the pursuit of nuclear power will result in the reduction of emissions, Malaysia has no obligation to do so under the Kyoto protocols. Water remains an abundant resource while the goal for the creation of new jobs within a nuclear technology development and research programme is not a feasible option for Malaysia.

The energy policy goals and selected level of application for the three scenarios are listed in Table 7.5.

Table 7.5. Energy policy goals and the selected options for Scenarios 1, 2 and 3.

Energy Policy Goals (Qualitative)		Scenario 1 BAU	Scenario 2 EcoFriend	Scenario 3 Nuclear
Political	Level of energy supply security	Medium	Medium	Medium
Social	Minimisation of public health risk	Low Priority	High Priority	Low Priority
	Maximisation of new jobs creation	Low Priority	High Priority	-
	Importance of social acceptability	-	High	Low
Energy Policy Goals (Quantitative)		Scenario 1 BAU	Scenario 2 EcoFriend	Scenario 3 Nuclear
Economic	Minimisation of Capital Cost	High	Low	Medium
	Maximisation of life cycle energy payback	High	High	Medium
Environmental	Future reduction of CO ₂ emissions	-	Significant	-
	Land utilisation	-	High	Low
	Maximisation of water conservation	-	Maximum	-

7.4 Results

The stipulated energy policy goals for Scenario 1 BAU result in the continued dominance of fossil fuel based electricity generation. Since the early 1980s, gas turbine electricity generation plants have been an extremely attractive option in Peninsular Malaysia because of its abundant natural gas supply and its artificially low subsidised price. However, the rapid decline in supplies since 2005 coupled with the gradual removal of subsidies has resulted in the pursuit of alternative sources such as coal and hydroelectric. Sarawak in particular, will see a massive expansion in the use of hydroelectric energy generation with the completion of the Bakun Dam. Thus, under Scenario 1 BAU, the utilisation of coal to alleviate the increasing constraints on natural gas continues. There is also an expansion of renewable energy use and an introduction to nuclear power generation. Figure 7.1 illustrates the electricity generation mix for Scenario 1 BAU.

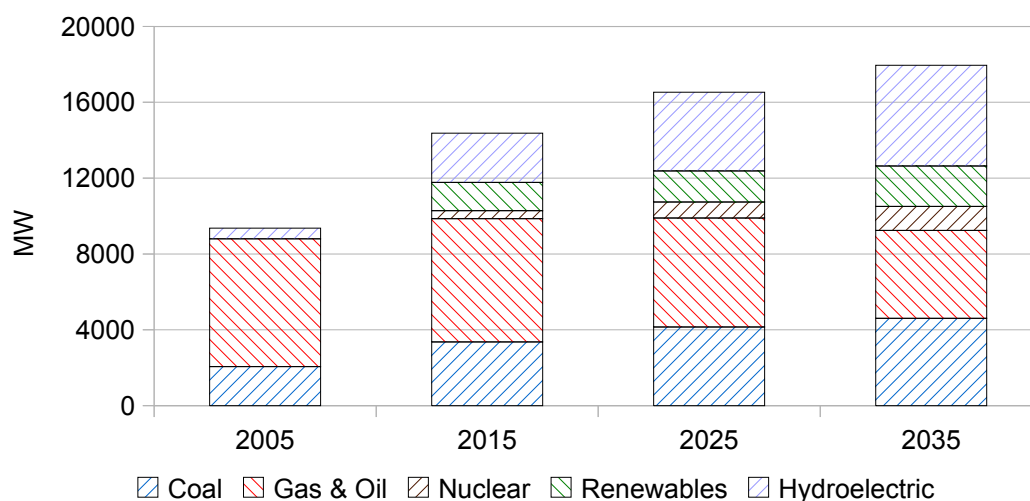


Figure 7.1: Proposed final electricity generation mix in MW for Scenario 1 BAU

Under Scenario 2 EcoFriend significant changes in the electricity generation mix occur. A commitment to the future reduction of CO₂ emissions, an emphasis on the creation of jobs in the new technology industry and a concern for the social acceptability of electricity generation plants are policy goals which will encourage the adoption of renewable energy sources when coupled with the right incentives. Figure 7.2 illustrates the mix for Scenario 2 EcoFriend.

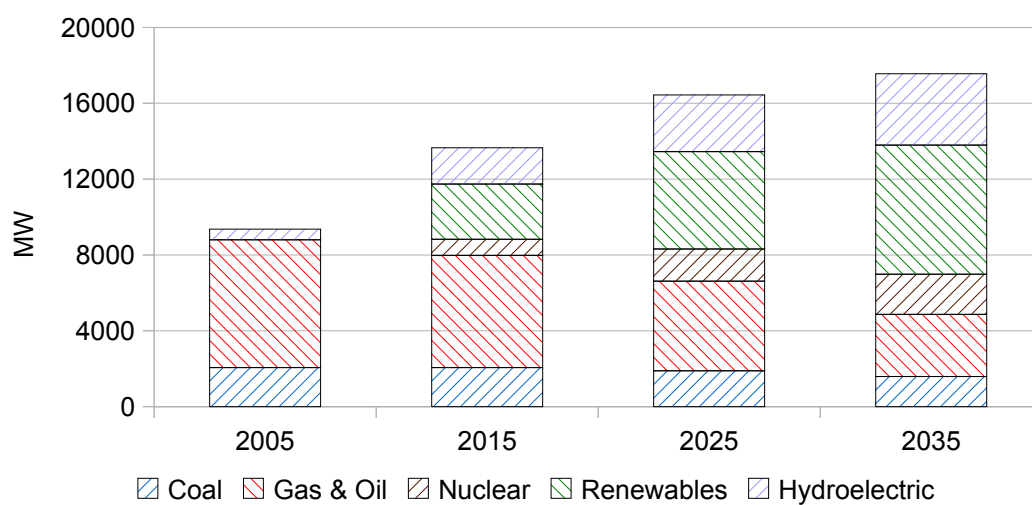


Figure 7.2: Proposed final electricity generation mix in MW for Scenario 2 EcoFriend

As Scenario 2 EcoFriend, illustrates, under ecologically friendly conditions the contribution of natural gas and coal declines and is replaced by renewable energy. This is not unexpected. As a tropical country, Malaysia enjoys a good amount of consistent sunlight throughout the year. This solar insolation represents an untapped energy resource. In addition, given Malaysia's current status as the second largest palm oil producer with other significant agricultural production, there is the potential for the use of biomass as a fuel source for electricity generation. The contribution towards renewable generation in this scenario is approximately 55% solar energy with

the rest coming from biomass and wind.

Scenario 3 Nuclear envisions an emphasis on nuclear power as a major contributor to electricity generation. This mix requires the construction of nuclear generation capacity of a significant size, which is consistent with the aspirations of Tenaga Nasional that has previously indicated an intention in the future to construct four reactors of approximately 1000 MW each in the future. In addition, it envisions the use of biomass as a source of renewable energy. One attraction of this scenario is the balanced supply of energy sources which is consistent with the fuel diversification policy. Figure 7.3 illustrates the mix for Scenario 3 Nuclear.

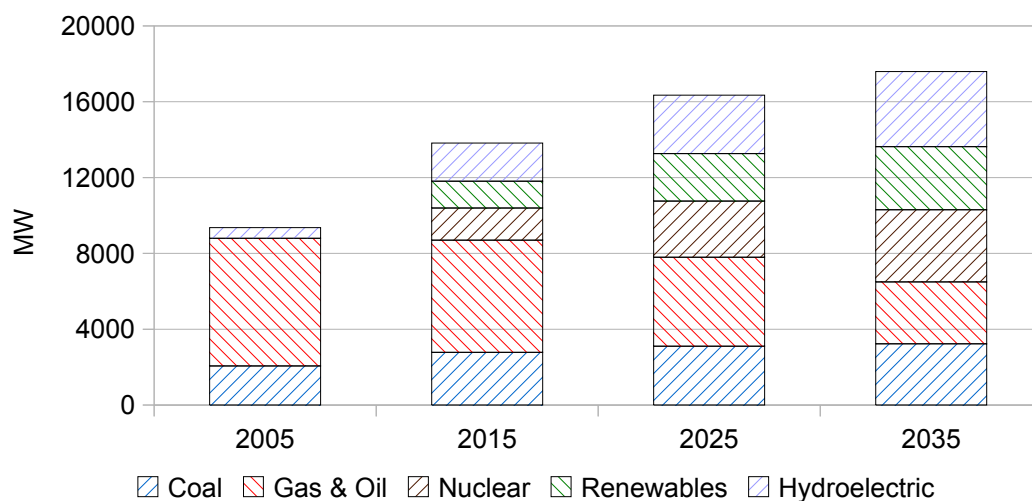


Figure 7.3: Proposed final electricity generation mix in MW for Scenario 3 Nuclear

The very different electricity generation mixes for all three scenarios illustrate the impact of the different policy goals. The overall effect of the energy policy goals in Scenario 1 BAU encourages the continued preference for fossil fuel based technologies. Fossil fuel based electricity generation technologies tend to have lower capital costs. These low costs are the benefits gained from the use of established

engineering equipment and processes; however such technologies also tend to have limited potential for the creation of new jobs and have a trade-off with regards to their impact on the environment and health.

In comparison, the combination of energy policy goals in Scenario 2 EcoFriend encourages the adoption of renewable energy based technologies alongside fossil fuel generation sources. Renewable energy electricity generation technologies tend to have more stringent environmental performance when compared to fossil fuel technologies but at the expense of a greater cost.

Lastly, Scenario 3 Nuclear provides another possible mix for electricity generation through the extensive use of nuclear and biomass sources. It represents, to an extent, a compromise between environment and economic costs. The use of nuclear energy allows for the reduction of greenhouse gases and harmful particulates emissions without the high capital costs and generation intermittency risks often associated with wind and solar generation plants.

The different generation mixes of the three scenarios result in different resource requirements, benefits and costs. Table 7.6 lists the life cycle energy payback, CO₂ emissions, land utilisation, water withdrawal and water consumption on a final electricity generation basis.

Table 7.6. Comparison of selected parameters between initial condition in 2005, Scenario 1 BAU in 2035, Scenario 2 EcoFriend in 2035 and Scenario 3 Nuclear in 2035.

Parameters	Initial Condition 2005	Scenario 1 BAU 2035	Scenario 2 EcoFriend 2035	Scenario 3 Nuclear 2035
Final Electricity Generation (MW)	9,360	17,952	17,560	17,588
Life Cycle Energy Payback	200,538	1,594,092	1,195,275	1,218,810
CO ₂ Emissions (Mtons / Year)	39	49	32	35
Land Requirement (km ²)	443	4,218	3,132	3,007
Water Withdrawal (tons / hr)	21,473	190,818	137,142	146,297
Water Consumption (tons / hr)	322	774	924	1,352

The figures in Table 7.6 illustrate the impact caused by the expansion in electricity generation between 2005 and 2035. CO₂ emission will increase under Scenario 1 BAU by approximately 30% even with an increased use of hydroelectric sources and the initiatives to implement renewable energy. Land usage by electricity generation plants will increase by almost ten fold with the greatest amount of land required for hydroelectric and solar generation plants.

Water withdrawal and consumption rates will increase between seven and ten times with the greatest increase in water withdrawal occurring in Scenario 1 BAU due to the significant expansion in hydroelectric sources while the greatest increase in water consumption occurs in Scenario 3 Nuclear due to the significant expansion of

nuclear generation.

For the purposes of illustration, the annualised capital and total cost profiles for all three scenarios have been manually calculated. Total cost was obtained from the sum of capital, fixed operation and maintenance costs, variable operation and maintenance costs and fuel costs. With the exception of hydroelectric energy, these costs are annualised over the construction and lifetime period of the plant. Coal, gas, oil, solar, wind and biomass plants have a lifetime of 30 years with a five year construction period. Nuclear plants have a lifespan of 30 years with a 10 year construction period. Hydroelectric plants have a lifespan that spans the entire simulation period; however the annualised cost life span period is fixed at 30 years with a 10 year construction period. The expected operational lifespan and plant capacity factors have also been used in the calculation of both the capital cost and total costs. The interest rate for the entire annualised period has been held at 5% and the costs have been tabulated in GBP£ in real terms for the year 2005 in order to eliminate the influence of currency exchange and inflation rates.

An important fact to remember with the calculated costs presented here is that the costs reflect only that of the proposed new electricity installations within the 2005 – 2035 time period. The actual cost of electricity generation for the whole system at the start of 2005 will be considerably more due to the presence of legacy facilities for which we have not calculated the costs. Similarly, the actual cost of electricity generation for the whole system at the end of 2070 will also be considerably more due to the presence of new facilities which we have not accounted for. Thus both the beginning and end of the cost curve is not representative of total costs at those time periods.

On the basis of annualised capital cost, Scenario 2 EcoFriend is the most expensive of the three scenarios while Scenario 1 BAU and Scenario 3 Nuclear have

similar costs. The four-fold increase in capital cost for Scenario 2 EcoFriend is due to the widespread use of solar electricity generation. Figure 7.4 illustrates the capital cost for the scenarios.

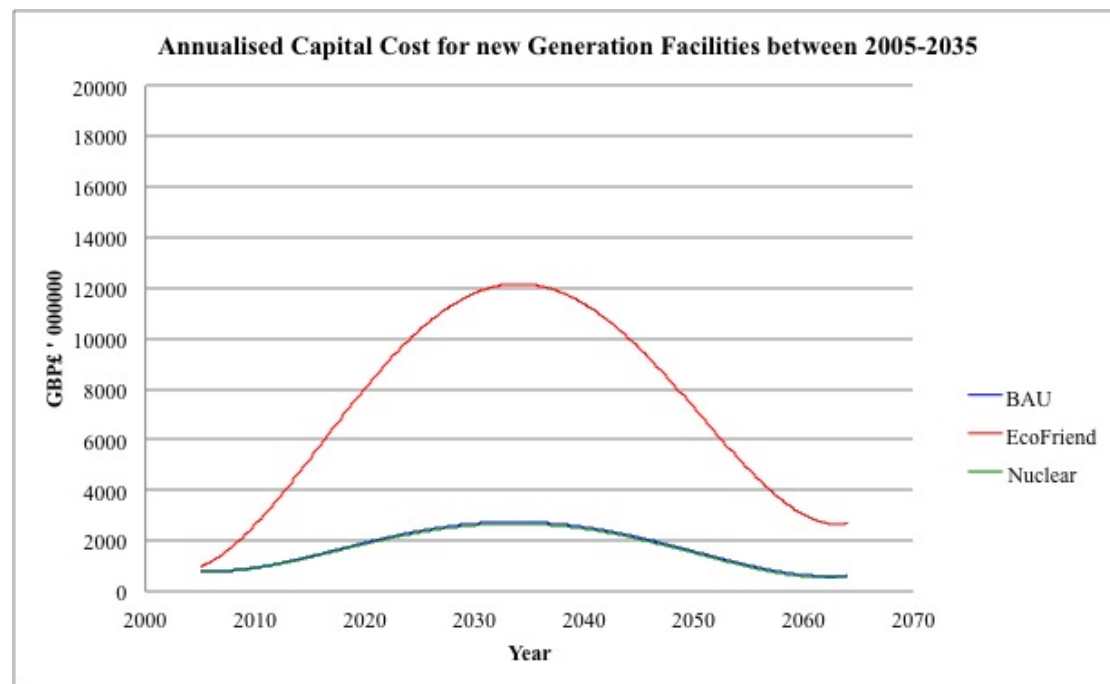


Figure 7.4. Annualised capital cost for Scenarios 1, 2 and 3.

A comparison of capital cost between the three scenarios does not, however, provide a complete representation of the actual financial costs. This is because fuel costs comprise approximately 70% - 80% of the cost of a fossil fuel electricity generation plant and approximately 50% of a nuclear electricity generation plant [182]. On the other hand, renewable energy plants such as those using solar, hydroelectric and wind sources have negligible fuel costs but significant capital costs. Therefore, a comparison of total cost over the lifetime of the facility is required to provide a complete picture.

On the basis of annualised total cost, Scenario 2 EcoFriend remains the most expensive option. However, the difference between the scenarios reduces to a two-

fold increase in total costs. Figure 7.5 illustrates the total cost for the scenarios.

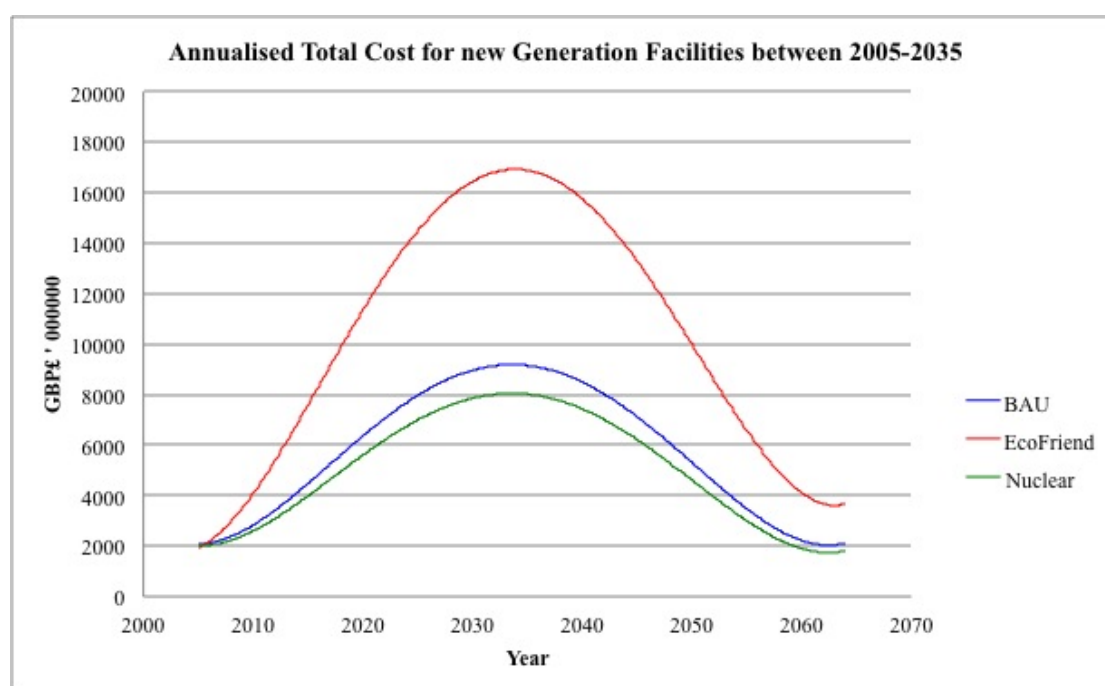


Figure 7.5. Annualised total cost for Scenarios 1, 2 and 3.

On a total cost basis, the per ton of CO₂ abatement costs for the scenarios range between GBP£62.5 – GBP£400. This large variation in range values is comparable to that found in other published work [183].

A comparison between Scenario 1 BAU and the official government plans provides a useful insight into the validity of the optimisation. Even though a direct apples to apples comparison is neither possible given that the forecast charts and detailed figures of the government's plans are not available in the public domain nor fair given that the government's policy goals and targets have been selected based on our best interpretation of public domain information, nevertheless some observations can be made as listed below:

1. The government anticipates coal and natural gas to remain the dominant energy

sources [184]. While a percentage has not been specified, it is generally expected to be at least 50% of electricity generation. Scenario 1 BAU also maintains a more than 50% electricity generation from coal and gas.

2. The government forecasts a decline in indigenous natural gas production due to maturing gas fields. Indeed, the national electricity company Tenaga Nasional (TNB) already suffers from the consistent gas curtailment and new facilities have been constructed to enable the importation of Liquefied Natural Gas (LNG) to fulfil future gas production shortfalls. Given no further increase until 2025 of the current 1000mmsfcd allocated to the power sector, the generated electricity of approximately 6500MW is close to the Scenario 1 BAU value of not less than 6000MW from 2005 to 2025.
3. The government has set a target of 5% renewable electricity generation by 2015 [171,185]. This translates to a generation mix of approximately 985MW [171] in 2015. In addition, the government is amenable to the expansion of solar generation to approximately 1300MW by 2020 [186]. These plans are comparable to the 1500MW - 1700MW forecasted in Scenario 1 BAU for the 2015 to 2025 time period.
4. TNB has committed to the construction of a nuclear generation plant within the next 20 years. Current plans call for a total generation capacity of 2000MW with the first of two units in operation by 2021 [186]. Scenario 1 BAU predicts a generation of 400MW in 2015 and 800MW in 2025 respectively.
5. The government plans to increase the total hydroelectric generation capacity by to approximately 5200MW by 2015 [172]. Given the historical generation to capacity historical ratio of between 0.3 to 0.4 [187,188], total hydroelectric generation is comparable to the Scenario 1 BAU value of 2600MW in 2015.

The similarities between Scenario 1 BAU and the official government plans

are encouraging given the fact that (i) different methodologies and approaches have been used and (ii) the policy goals and targets in Scenario 1 BAU is based on our estimation of what the official government policy goals and targets are from information extracted in the 10th Malaysian Plan [172].

7.5 Conclusions

The similarity between Scenario 1 BAU and the official government plans is a good result which lends credence to the capability of the *Energy Model* created by the *Prototype Energy Modelling System* to optimise electricity generation mixes that are acceptable and believable.

It is encouraging to note that both Scenario 1 BAU and the official government plans entail the introduction of renewables with a target of 5% by 2015. This laudable goal represents a good starting point for source diversification of the electricity generation mix. Given the life span of electricity generation plants, an introduction of renewable energy at the start of an increased electricity demand cycle will avoid the technological lock-in to fossil fuels that can be a hindrance if renewable energy is to be adopted in the later phases of cycle.

It is also encouraging to find that the CO₂ emissions from Malaysia can be reduced even with increased electricity demand through the adoption of the electricity generation mixes resulting from Scenario 2 EcoFriend and Scenario 3 Nuclear. It might therefore be advantageous to pursue for a more aggressive growth for renewable generation than the current target of 5% by 2015.

The results from this case study reflect the large influence that diverse policy goals have on a proposed electricity generation configuration. This influence is modelled by means of logic inference in the ontology and demonstrates the ability of the system to consider qualitative factors (e.g. social acceptance) in addition to

quantitative inputs (e.g. capital costs). It is however acknowledged that the models used in the proposed system are rudimentary and the results are, at best, rough approximations.

In conclusion, it is important for policy makers to consider issues other than cost in the development of energy policies. As Malaysia progresses to become a developed country, unanticipated changes in societal attitude towards electricity generation facilities could occur. Increased concerns on the 'external cost of energy' that the Malaysian society pays in the form of health and environmental effects may require changes to the future energy plans of Malaysia. Similarly, as part of Malaysia's plan to become a high income nation, the pursuit of energy life cycle efficiencies and the growth of a manufacturing sector based on the production of new electricity generation equipment may be necessary. To support this, changes in the energy plan to support technologies such as algae based biofuels, wave and tidal technology, wind turbines and solar towers could be necessary. This *Prototype Energy Modelling System*, which allows for the exploration of such scenarios, could be an invaluable tool for policy planners in their development of an energy policy for Malaysia that is both appropriate for the current environment and flexible enough to accommodate future developments.

Chapter 8 Conclusions and Future Work

This section discusses the conclusions, future work and contributions of this research.

8.1 Conclusions

This research has been predicated on the hypothesis that it is possible to develop a modelling system which supports policy makers by automatically converting their policy goals and targets into an optimisation model consisting of a set of equivalent mathematical expressions. This modelling system, referred to in this thesis as the *Optimisation Modelling System*, addresses the disconnect between the policy maker and the optimisation model itself. To address this disconnect, a modeller is currently employed whose function is to translate the policy goals and targets from the policy maker into inputs compatible with the optimisation model. This solution however reduces the overall efficiency of the policy evaluation and analysis process and more importantly, gives rise to misinterpretations due to miscommunication between the modeller and the policy maker.

To support the hypothesis, this research project has sought to extend the boundaries of the formulation of optimisation models used for policy evaluation and analysis with particular emphasis given to energy models. This has been achieved by semi-automating part of the function and the associated tasks of the modeller.

To test the hypothesis, a framework for the development of an *Optimisation Modelling System* has been developed (Chapters 4 and 5). This modelling system creates a *Model* by automatically formulating the necessary mathematical expressions for optimisation based on both quantitative and qualitative inputs from the user. The modelling system achieves this performance through the combined use of a semantic

representation, knowledge inference, mathematical techniques and procedural programming.

To validate the framework (Chapter 6), a *Prototype Energy Modelling System* has been developed which is made up of five modules:

- a *Semantic Representation Module* which uses ontologies and engineering models to represent the knowledge related to energy policy,
- a *Knowledge Inference and Expression Builder Module* which uses logic inference to determine the variables for the automatic formulation of mathematical expressions,
- a *Linear Programming Module* which solves a set of mathematical expressions,
- a *Graphical User Interface*, and
- a *Support Module* which contains additional Java code for overall program integration.

To demonstrate the capability of the *Prototype Energy Modelling System*, a case study on the future electricity generation mixes of Malaysia was conducted (Chapter 7).

The *Optimisation Modelling System* offers unique advantages and capabilities. However, as with all systems, it also has disadvantages and limitations. Thus room for further improvements remain. These issues are discussed in the following subsections.

8.1.1 Advantages and Capabilities

The proposed framework breaks new ground with regards to the use of ontologies. As far as can be ascertained, this is the first use of ontologies as a method to formulate mathematical expressions through inference in the knowledge domain.

This allows for a reduced dependence on manual programming of mathematical expressions where a single human mistake can cause inaccurate results.

From an application standpoint, the *Prototype Energy Modelling System* and the *Energy Model* it formulates offer the advantage of a single evaluation platform, explicit model documentation, rapid model reconfiguration, ease of use, and an opportunity to explore new results.

- Single evaluation platform: The *Prototype Energy Modelling System* accepts both qualitative and quantitative inputs in the areas of politics, economics, society and the environment. This differs from current energy models which tend to concentrate on issues of economics and the environment. Furthermore, the optimised results from the system reflect consideration of both qualitative and quantitative aspects.
- Explicit documentation: The semantic representation (ontologies) within the modelling system provides explicit documentation of both the energy policy knowledge domain and the mathematical expressions used in the *Energy Model*. The basis and assumptions behind an optimisation are stored in the ontology and can therefore be easily checked.
- Rapid model reconfiguration: The use of axioms to formulate mathematical expressions allows for the quick and easy inclusion of additional mathematical equations. To add new equations to the *Energy Model*, the user essentially has only to create new axioms in the ontology. Current energy models require the services of a dedicated computer programmer to achieve the same results.
- Ease of use: The emphasis on policy goals and targets as the main inputs for scenario analysis makes the modelling system easier to use when compared to current energy models. Policy makers are able to utilise the modelling system

with minimal assistance from a modeller. In addition, the emphasis on policy goals and targets caters to the needs of the policy maker whose interest lies not with the programming of the model but with its application as a tool for the analysis of policy effects.

- Opportunity to explore new results: The modelling system enables the linear programming based *Energy Model* to accept both qualitative and quantitative inputs. This feature creates new results of which other models are not able to produce.

8.1.2 Disadvantages and Limitations

All modelling systems suffer from the modelling bias, a bias that arises from the view of the world that the programmer holds. In addition, the use of ontologies as the modelling technique raises the issue of the ontology bias, a bias that arises from the belief that the world can be represented as classes, instances, properties and axioms. Furthermore, the codification of the knowledge domain introduces the expert bias, a bias which arises from the opinions of the domain expert on what constitutes the information and data of the knowledge domain.

While the framework allows the *Optimisation Modelling System* to accept qualitative inputs, the analysis of the inputs using qualitative data remains limited. The ability of the *Optimisation Modelling System* to process qualitative data is limited by the richness of the ontology which in turn is limited by the available knowledge for codification. Under the framework, the use of the data is confined to the interpretation of the policy goals and targets. In addition, the type of policies that can be accepted for analysis is subject to the availability of accurate data related to the policy in the published literature.

From an application standpoint, the *Prototype Energy Modelling System* and

the *Energy Model* suffers from the vulnerability of an infeasible solution space of which only limited problem finding and solving functions have been incorporated. In addition, solutions from the *Energy Model* is subject to dispute given that certain non universally accepted presumptions have been incorporated.

- Infeasible solution space vulnerability: It is sometimes not possible for the *Energy Model* to determine an optimal solution because a feasible solution space does not exist. There are two reasons as to why this can occur. First, the constraints specified for a particular case study may have ranges which are too narrow. Second, the policy goals and targets selected by the user results in contradictions of which no solution is possible.
- Problem finding function: The modelling system does not have a mechanism to determine the reasons which can cause an infeasible solution. Given that the intended user of the modelling system are policy makers who may lack programming skills, the lack of an error finding mechanism severely limits its ease of use when such problems occur.
- Limited problem solving function: The modelling system incorporates features to preclude the production of unrealistic optimisation results. These include, for example, the automatic relaxation of constraints where allowable. In addition, the *Prototype Energy Modelling System* limits both the type of policy goals and the available policy targets. These features are workable but unsatisfactory.
- Modelling presumptions: Some of the presumptions in the *Energy Model* are not universally agreed upon, by the domain experts. For example, the presumption that the minimisation of cost (capital or total) shall be part of the objective function in the *Energy Model* is not universally accepted as a good

idea when in the context of energy savings and carbon reduction.

8.2 Future Work

The proposed framework would benefit from additional research in the areas of semantic representation and linear programming. The four areas of improvement are the expansion of the knowledge codification paths, the improvement in the integration between the ontology and the engineering models, the enhancement of the representation quality of the ontology and the adoption of a better multiobjective function optimisation method.

- Expansion of the knowledge codification paths: As discussed in Section 4.4.3, the semi-automatic codification path (path 4) has not been implemented due to the lack of resources and possible duplication of work by other researchers. It is recommended that a method to semi-automatically extract data from sources (e.g. from the literature or the web) for automatic incorporation into the ontology be developed using the methods discussed in literature [90,91,92].
- Improved integration between the ontology and the engineering models: As identified in Section 5.2.1.5 there is currently no established API to connect a class or instance in the ontology to a program in Java.
- Enhancement of representation quality: Given that the accurate translation of the policy goals into mathematical expressions is highly dependent on the comprehensiveness of the ontological representation - it would be useful if an established ontology such as OntoCAPE was to be used as the basis for knowledge representation. Such an approach however, will require changes to the ontology construction method to one based on the philosophy advocated by Swartout *et al.* (Section 2.2.4).
- Better multiobjective function optimisation: The technique used in the

prototypes, which is to gradually increment the values for each objective function until a Pareto optimal point is found, is an unsatisfactory method. More advanced optimisation techniques such as that published by Yan *et al.* could be used [189].

From an application standpoint, the *Prototype Energy Modelling System* has potential for additional improvements. The created *Energy Model* is currently limited to the optimisation of electricity generation within a narrow scope. By comparison, models such as MARKAL and MESSAGE optimise all facets of energy supply and demand. Sustainability of energy supply through portfolio diversification and the mitigation of energy demand through efficiency improvements are examples of issues that could be incorporated to expand the scope of optimisation. The framework for this expansion could be based on the works by Darton who proposes the use of scenarios and metrics to measure sustainability in energy futures [190] and Stirling who proposes the use of multicriteria diversity analysis to appraise energy portfolios [191]. Similarly, the level of detail of the energy supply chain, as represented by the class *Energy_Chain_Objects*, should be expanded to include new technologies such as carbon capture and storage. Additionally, extension to account for regional supply and demand would allow for simulation of interstate electricity import and export. Enhancements in all these areas would improve the optimisation results.

The scope of the *Prototype Energy Modelling System* could also be expanded. The ontological representation could be enhanced to include the economics knowledge domain while the modelling system could be enhanced to allow for the automatic creation of a macroeconomic model. This would make the modelling system more in line with current energy models such as MARKAL-MACRO and MESSAGE-MACRO.

Lastly, the applicability of the framework to optimisation models for sectors

other than energy should be explored. Indeed, we have already developed part of a *Prototype Water Modelling System* which optimises the water supply and treatment for agricultural, industrial and municipal uses for the Malaysian state of Penang. Unfortunately, due to the constraints of space, the details of this model and the case study have not been discussed in this thesis. Another potential area for exploration is the integration of both water and energy in a new *Prototype Energy-Water Modelling System* system. Such a system would address the interactions between water and energy with application in areas such as the large scale treatment of water using renewable energy and the use of water as a storage medium for energy.

8.3 Contributions

The main contributions from this research are:

- A framework for the development of an *Optimisation Modelling System* which automatically formulates an optimisation model. The modelling system uses a semantic representation and knowledge inference to automatically formulate mathematical expressions representative of the intentions of the user.
- A semantic representation of the energy policy knowledge domain using ontologies. This ontology further expands the library of ontologies available to the ontology community.
- A *Prototype Energy Modelling System* which automatically creates an *Energy Model* based on the qualitative and quantitative energy policy goals and targets selected by the user.
- An energy policy case study of Malaysia which optimises electricity generation mixes based on energy policy goals and targets.

8.4 Conference and Journal Papers

We have published the results from this research in two conference papers:

- Chee Tahir, A., Bañares-Alcántara, R., 2011. An Ontology-Based Semantic Representation of Energy Policy Goals in the Optimisation of Electricity Generation Mixes. *6th Dubrovnik Conference on Sustainable Development of Energy Water and Environment Systems*. Dubrovnik, Croatia 25-29 September 2011. (Awarded Best Conference Paper).
- Chee Tahir, A., Bañares-Alcántara, R., 2011. A Knowledge Representation Model for the Optimisation of Electricity Generation Mix. *International Conference on Applied Energy (ICAE2011)*. Perugia, Italy 16-18 May 2011. (Recommended for Journal Publication).

In addition, we have been invited to publish our results in the journals below:

- Chee Tahir, A., Bañares-Alcántara, R., 2011. An Ontology-Based Semantic Representation of Energy Policy Goals for the Optimisation of Electricity Generation Mixes. *Energy Special Issue SWEDES2011*. (Submitted).
- Chee Tahir, A., Bañares-Alcántara, R., 2011. A Knowledge Representation Model for the Optimisation of Electricity Generation Mixes. *Applied Energy Special Issue ICAE2011*. (2nd Review in Progress).

8.5 Final Remark

In conclusion, the framework and created prototypes discussed in this thesis have, in our opinion, supported the hypothesis of the research. Our new approach to the formulation of optimisation models used for policy goals evaluation and analysis caters more effectively to the needs of policy makers which should in turn allow for a more effective development of policies for the betterment of mankind.

Appendix A Rankine Steam Cycle

This appendix lists the equations associated with the Rankine cycle when used for electricity generation. The relations between the equipment, engineering process and formulas are listed in Table A.1. The specific configuration of the engineering model for each power plant is shown in Tables 6.8 - 6.15.

Table A.1: Equipment and formula for Rankine steam cycle

Equipment and Component		Engineering Process	Formula
Auxiliary Steam Boiler	Combustor	stoichiometric combustion	A.1
	Steam Generator	latent heat of vaporisation	A.2
Waste Heat Steam Boiler	Steam Generator	latent heat of vaporisation	A.3
Steam Turbine	Steam Rotor	net work out	A.4
Cold Exchanger		net work out *	A.5
Pump	Pump Rotor	net work out *	A.6
Heat Exchanger		net work out	A.7
Water Treatment			A.8.1, A.8.2
Generator	Electricity Generator	electricity generation	AG.1
* "net work out" when boundary condition is set to only encompass the equipment. When boundary condition is set to the entire Rankine cycle, it is "net work in" as shown in the formula notations			

I. Stoichiometric combustion (A.1)

For a steady state combustion process in the auxiliary steam boiler, the energy released is given by equation A.1. In the engineering models, three fuel compositions are available: methane for natural gas based power plants, octane for oil based power plants and a lignite sub-bituminous mixture for coal based power plants. Enthalpy values for the components have been taken from Wark and Richards [192]

$$\dot{Q}_{\text{combustion}} = \sum_{\text{products}} \dot{M}_{ic} h_{ic} - \sum_{\text{reactants}} \dot{M}_{ic} h_{ic}$$

where

$$\dot{Q}_{\text{combustion}} = \text{combustion heat production rate (kJ/s)} \quad (\text{A.1})$$

ic = components (e.g. air, oxygen, water, carbon dioxide)

$\dot{M}$ = mass flow rate of components (kg/s)

h = enthalpy at specified temperature and pressure (kJ/kg)

II. Latent heat of vaporisation (A.2)

The rate of steam production from the auxiliary steam boiler is given by equation A.2. The engineering models simulate two superheated steam states: 100 bars at 720°C and 40 bars at 400°C. The enthalpy values from the steam tables published by Wark and Richards [192] have been used.

$$\dot{m}_{\text{steamout}} = \frac{\dot{Q}_{\text{combustion}}}{h_{\text{steamout}} - h_{\text{boilerwaterin}}}$$

where

$$\dot{m}_{\text{steamout}} = \text{steam production rate (kg/s)} \quad (\text{A.2})$$

h_{steamout} = enthalpy of produced steam (kJ/kg)

$h_{\text{boilerwaterin}}$ = enthalpy of boiler feedwater (kJ/kg)

III. Latent heat of vaporisation (A.3)

The rate of steam production from the waste heat steam boiler is given by equation A.3. The engineering models simulate a single superheated steam state of 40 bars and 400°C. The waste heat into the boiler is simulated with an superheater inlet temperature of 700°C and an riser outlet temperature of 500°C.

$$\dot{m}_{steamout} = \frac{\dot{Q}_{wasteheat}}{\dot{h}_{steamout} - \dot{h}_{boilerwaterin}} \quad (A.3)$$

where

$$\dot{Q}_{wasteheat} = \text{waste heat production rate (kg/s)}$$

IV. Steam turbine net work out (A.4)

The steam turbine work out is given by the set of equations in A.4. The engineering model optimises for a turbine outlet steam pressure of 4 bars at a steam quality of more than 90%. The enthalpy values from the steam tables published by Wark and Richards [192] have been used.

$$\begin{aligned}\dot{W}_{ST,out} &= \dot{m}_{steamout} * (h_{ST,out} - h_{steamout}) \\ \dot{s}_{ST,out} &= (1-x_{ST,out})\dot{s}_{ST,liq} + x_{ST,out}\dot{s}_{ST,vap} \\ \dot{s}_{ST,out} &= \dot{s}_{steamout} \\ h_{ST,out} &= (1-x_{ST,out})h_{ST,liq} + x_{ST,out}h_{ST,vap}\end{aligned}$$

where

(A.4)

$\dot{W}_{ST,out}$ = net work out rate from steam turbine (kJ/s)
 $h_{ST,out}$ = enthalpy of steam at steam turbine exit (kJ/kg)
 $\dot{s}_{ST,out}$ = entropy of steam at steam turbine exit (kJ/kg.K)
 $x_{ST,out}$ = vapour fraction at steam turbine exit
 $\dot{s}_{ST,liq}$ = entropy of liquid phase at steam turbine exit (kJ/kg.K)
 $\dot{s}_{ST,vap}$ = entropy of vapour phase at steam turbine exit (kJ/kg.K)
 $h_{ST,liq}$ = enthalpy of liquid phase at steam turbine exit (kJ/kg)
 $h_{ST,vap}$ = enthalpy of vapour phase at steam turbine exit (kJ/kg)

V. Cold exchanger (Steam condenser) net work out (A.5)

The steam condenser net work in is given by the set of equations in A.5. The engineering model simulates a steam condensate outlet of approximately 110°C at 1.1 bars. Inlet pressures are simulated at either 4 bars (for steam turbine outlet) or 2 bars (for cold exchanger outlet). Cooling water is the cooling medium. The enthalpy values from the steam tables published by Wark and Richards [192] have been used.

$$\begin{aligned}\dot{W}_{COOL,in} &= \dot{m}_{boilerwaterin} * (h_{COOL,in} - h_{COOL,out}) \\ h_{COOL,in} &= h_{ST,out} \\ h_{COOL,in} &= h_{HEX,out} \\ \dot{m}_{boilerwaterin} &= \dot{m}_{steamout}\end{aligned}$$

where

(A.5)

$\dot{W}_{COOL,in}$ = net work in by steam condenser (kJ/s)
 $h_{COOL,in}$ = enthalpy of boiler feed water at steam condenser outlet (kJ/kg)
 $\dot{m}_{boilerwaterin}$ = boiler feed water flow at steam condenser outlet (kg/s)
 $h_{COOL,out}$ = enthalpy at steam condenser outlet (kJ/kg)

VI. Pump net work out (A.6)

The pump work in is given by equation A.6. The engineering model simulates a boiler feed water pump discharge pressure of either 40 or 100 bars and a discharge temperature of 110°C. The enthalpy values from the steam tables published by Wark and Richards [192] have been used.

$$\dot{W}_{PUMP, in} = \dot{m}_{boilerwaterin} * (h_{PUMP, out} - h_{PUMP, in})$$

where

$$\dot{W}_{PUMP, in} = \text{net work in by boiler feed water pump (kJ/s)} \quad (\text{A.6})$$

$$h_{PUMP, out} = \text{enthalpy of boiler feed water at pump discharge (kJ/kg)}$$

$$h_{PUMP, in} = \text{enthalpy of boiler feed water at pump suction (kJ/kg)}$$

VII. Heat exchanger net work out (A.7)

The heat exchanger work (heat) out is given by the set of equations in A.7. The engineering model optimises for a outlet steam pressure of 2 bars at a steam quality of more than 80%. The enthalpy values from the steam tables published by Wark and Richards [192] have been used.

$$\dot{Q}_{HEX, out} = \dot{m}_{steamout} * (h_{HEX, out} - h_{steamout})$$

$$s_{HEX, out} = (1 - x_{HEX, out}) s_{HEX, liq} + x_{HEX, out} s_{HEX, vap}$$

$$s_{HEX, out} = s_{steamout}$$

$$h_{HEX, out} = (1 - x_{HEX, out}) h_{liq} + x_{HEX, out} h_{vap}$$

where

(A.7)

$\dot{Q}_{HEX, out}$ = net heat production rate from heat exchanger (kJ/s)

$h_{HEX, out}$ = enthalpy of steam at heat exchanger exit (kJ/kg)

$s_{HEX, out}$ = entropy of steam at heat exchanger exit (kJ/kg.K)

$x_{HEX, out}$ = vapour fraction

$s_{HEX, liq}$ = entropy of liquid phase at heat exchanger exit (kJ/kg.K)

$s_{HEX, vap}$ = entropy of vapour phase at heat exchanger exit (kJ/kg.K)

$h_{HEX, liq}$ = enthalpy of liquid phase at heat exchanger exit (kJ/kg)

$h_{HEX, vap}$ = enthalpy of vapour phase at heat exchanger exit (kJ/kg)

VIII. Water treatment (A.8.1 and A.8.2)

Two sets of water treatment equations are modelled: plant start up and steady state operation. During plant start up, the water withdrawal rate is given by the set of equations in A.8.1. The engineering model simulates a three system treatment: sand and activated carbon filters, reverse osmosis membranes, and cation-anion demineraliser beds. The loss rates for each system is based on propriety data published in the process flow diagrams of an operational plant [193].

$$\begin{aligned}
 m_{BWTotal} &= m_{boilerwaterin} + m_{boilerwaterloss} \\
 m_{coolingwaterin} &= \frac{W_{COOLin}}{(h_{COOLout} - h_{COOLin})} \\
 m_{CWTotat} &= m_{coolingwaterin} + m_{CWloss} \\
 m_{BWTotal} + m_{CWTotat} &= m_{demintotal} - m_{deminloss} \\
 m_{demintotal} &= m_{ROtotal} - m_{ROloss} \\
 m_{ROtotal} &= m_{filtertotal} - m_{filterloss} \\
 m_{withdrawal} &= m_{filtertotal}
 \end{aligned}$$

where

$$\begin{aligned}
 m_{BWTotal} &= \text{total water flow rate in steam system (kg/s)} \\
 m_{BWloss} &= \text{boiler feed water loss rate (kg/s)} \\
 m_{coolingwaterin} &= \text{cooling water in flow rate (kg/s)} \\
 m_{CWTotat} &= \text{total cooling water flow rate (kg/s)} \\
 m_{CWloss} &= \text{cooling water loss rate (kg/s)} \\
 m_{demintotal} &= \text{total demineralizer flow rate (kg/s)} \\
 m_{deminloss} &= \text{demineralizer water loss rate (kg/s)} \\
 m_{ROtotal} &= \text{total reverse osmosis flow rate (kg/s)} \\
 m_{ROloss} &= \text{reverse osmosis water loss rate (kg/s)} \\
 m_{filtertotal} &= \text{total filtered water flow rate (kg/s)} \\
 m_{filterloss} &= \text{filtered water loss rate (kg/s)} \\
 m_{withdrawal} &= \text{water withdrawal rate (kg/s)}
 \end{aligned} \tag{A.8.1}$$

At steady state operation, the water consumption rate is given by the set of equations in A.8.2. The water consumption rate is taken to be approximately equal to the water loss rates for the steam and cooling water systems. The boiler feed water system loss of 3% accountants for intermittent and continuous blowdown of the boilers [194]. A cooling water loss of 5% is based on our experience.

$$\begin{aligned}
 m_{BWTopup} &= m_{boilerwaterloss} = 0.03 * m_{boilerwaterin} \\
 m_{CWTopup} &= m_{coolwaterloss} = 0.05 * m_{coolwaterin} \\
 m_{BWTopup} + m_{CWTopup} &= m_{demintotal} - m_{deminloss} \\
 m_{demintotal} &= m_{ROtotal} - m_{ROloss} \\
 m_{ROtotal} &= m_{filtertotal} - m_{filterloss} \\
 m_{consumption} &= m_{filtertotal}
 \end{aligned} \tag{A.8.2}$$

where

$m_{BWTopup}$ = total water top up in steam system (kg/s)

$m_{CWTopup}$ = total water top up in cooling water system (kg/s)

$m_{consumption}$ = water consumption rate of electricity generation plant (kg/s)

IX. Electricity generation (AG.1)

The electricity generated by the steam turbine coupled generator is given by equation AG.1.

$$P_{electrical, out} = \eta * W_{ST, out}$$

where

$P_{electrical, out}$ = electrical power generated (kW)

η = generator efficiency

(AG.1)

Appendix B Brayton Cycle

This appendix lists equations associated with the Brayton cycle when used for electricity generation. The relations between the equipment, engineering process and formulas are listed in Table B.1. The specific configuration of the engineering model for each power plant is shown in Tables 6.8 - 6.15.

Table B.1: Equipment and formula for Brayton cycle

Equipment and Component		Engineering Process	Equations
Gas Turbine	Compressor	air suction	B.1
	Combustor	stoichiometric combustion	B.2
	Power Turbine	net work out	B.3
Generator	Electricity Generator	electricity generation	BG.1

I. Air suction flow rate (B.1)

For a steady state condition in the gas turbine, the air flow rate is given by equation B.1. The standard enthalpy of combustion of methane and octane have been obtained from Wark and Richards [192].

$$\dot{m}_{air} = \frac{\Delta H_c^o}{h_{C,out} - h_{C,in}} * \frac{\dot{m}_{fuel}}{MM_{fuel,c}}$$

where

$$\dot{m}_{air} = \text{air flowrate (kg/s)} \quad (B.1)$$

$$\dot{m}_{fuel} = \text{fuel flowrate (kg/s)}$$

$$\Delta H_c^o = \text{enthalpy of combustion (kJ/mol)}$$

$$MM_{fuel,c} = \text{molar mass of fuel required to produce } \Delta H_c^o \text{ (kg/mol)}$$

$$h_{C,in} = \text{enthalpy of air at combustor inlet (kJ/kg)}$$

$$h_{C,out} = \text{enthalpy of flue gas at combustor outlet (kJ/kg)}$$

II. Stoichiometric combustion (B.2)

For a steady state combustion process in the gas turbine, the enthalpy of combustion for either methane or octane can be calculated based on the combustion of 1 mole of methane or octane respectively. The enthalpy of combustion equation is given by B.2. The mole of air required is calculated from stoichiometry. Enthalpy values for the components have been taken from Wark and Richards [192]. In the engineering models, two fuel compositions are available: methane for natural gas based power plants and octane for oil based power plants.

$$\Delta H_c^o = \sum_{products} N_i h_i - \sum_{reactants} N_i h_i$$

where

$$i = \text{components (e.g. air, oxygen, water, carbon dioxide)} \quad (B.2)$$

$$N = \text{molar mass (kg/mol)}$$

$$h = \text{enthalpy at specified temperature and pressure (kJ/kg)}$$

III. Power turbine net work out (B.3)

The power turbine work out is given by equation B.3. The enthalpy values from the steam tables published by Wark and Richards [192] have been used.

$$\dot{W}_{PT,out} = ((h_{C,out} - h_{PT,out}) - (h_{C,in} - h_{atm})) * \dot{m}_{air}$$

where

$$\dot{W}_{PT,out} = \text{net work out rate from power turbine (kW)}$$

$$\dot{m}_{air} = \text{air flow rate} \sim \text{flue gas flow rate (kg/s)} \quad (\text{B.3})$$

$$h_{C,out} = \text{enthalpy of flue gas at combustor outlet (kJ/kg)}$$

$$h_{PT,out} = \text{enthalpy of flue gas at power turbine outlet (kJ/kg)}$$

$$h_{C,in} = \text{enthalpy of air at combustor inlet (kJ/kg)}$$

$$h_{atm} = \text{enthalpy of air at compressor inlet (kJ/kg)}$$

IV. Electricity generation (BG.1)

The electricity generated by the power turbine coupled generator is given by equation BG.1.

$$P_{electrical,out} = \eta * \dot{W}_{PT,out}$$

where

$$P_{electrical,out} = \text{electrical power generated (kW)}$$

$$\eta = \text{generator efficiency}$$

(BG.1)

Appendix C Wind Turbine Model

This appendix lists the equations for wind turbine electricity generation.

Table C.1: Wind Turbine Model

Equipment and Component		Engineering Process	Equations
Wind Turbine	Rotor	net work out	C.1
	Electricity Generator	electricity generation	CG.1

I. Wind turbine net work out (C.1)

For a steady state condition, the work generated from a wind turbine is given by equation C.1. The Betz constant has been set to 0.59. Wind velocity values have been either loaded from a GIS database [149] or taken from the literature [175,176].

$$W_{WDT, out} = \beta * \frac{1}{2} * \rho_{air} V^3 * \pi * r^2$$

where

$W_{WDT, out}$ = wind turbine power output (W) (C.1)

β = Betz constant

ρ = air density (kg/m^3)

V = wind velocity (ms^{-1})

r = rotor radius (m)

II. Electricity generation (CG.1)

The electricity generated by the wind turbine is given by equation CG.1

$$P_{electrical, out} = \eta * W_{WDT, out}$$
(CG.1)

where

$P_{electrical, out}$ = electrical power generated (W)

η = generator efficiency

Appendix D Solar Photovoltaic and Concentrator Model

This appendix lists the equations for solar electricity generation.

I. Photovoltaic electricity production (D.1)

The amount of electricity generated by a photovoltaic array is given by expression D.1 [195]. Solar global radiation values have been either loaded from a GIS database [149] or taken from the literature [173,174]. An efficiency value of 10% is used.

$$P_{electrical, out} = \eta_{pv} * \tau_g * A_{pv}$$

where

$$P_{electrical, out} = \text{electrical power generated (kWh/day)} \quad (D.1)$$

$$\eta_{pv} = \text{photovoltaic generator efficiency}$$

$$\tau_g = \text{solar global radiation (kWh/m}^2 \cdot \text{day)}$$

$$A_{pv} = \text{photovoltaic surface area (m}^2 \text{)}$$

I. Solar concentrator heat production (D.2)

The amount of heat generated by a solar concentrator has been assumed to be similar to that of a solar thermal collector. Expression D.2 is based on EN12975 [196]. An efficiency value of 12% is used.

$$\dot{Q}_{col} = A_{col} * (\eta_{col} * \tau_g - a_1 * (T_m - T_a) - a_2 * (T_m - T_a)^2)$$

where

$$\dot{Q}_{col} = \text{thermal heat production rate (kWh/day)}$$

$$A_{col} = \text{collector surface area (m}^2 \text{)} \quad (D.2)$$

$$\eta_{col} = \text{collector efficiency}$$

$$a_1 = \text{1st order heat loss coefficient (kW/K)}$$

$$a_2 = \text{2nd order heat loss coefficient (kW/K)}$$

$$T_m = \text{collector mean temperature (K)}$$

$$T_a = \text{ambient air temperature (K)}$$

Appendix E Hydroelectric Model

This appendix lists the equations associated with hydroelectric power.

Table E.1: Hydroelectric Turbine Model

Equipment and Component		Engineering Process	Equations
Hydroelectric Turbine	Rotor	net work out	E.1
	Electricity Generator	electricity generation	EG.1

I. Hydroelectric turbine net work out (E.1)

For a steady state condition, the work generated from a hydroelectric turbine is given by E.1

$$\dot{W}_{WT,out} = \rho_{wt} * g * h * Q_{water}$$

where

$$\dot{W}_{WT,out} = \text{net work our rate from water turbine (J/s)} \quad (E.1)$$

$$\rho_{wt} = \text{water density (kg/m}^3\text{)}$$

$$g = \text{gravity (ms}^{-2}\text{)}$$

$$h = \text{height (m)}$$

$$Q_{water} = \text{water flowrate (m}^3\text{/s)}$$

II. Electricity generation (EG.1)

The generated electricity from the hydroelectric turbine is given by equation EG.1.

$$P_{electrical,out} = \eta * \dot{W}_{WT,out} \quad (EG.1)$$

where

$$P_{electrical,out} = \text{electrical power generated (W)}$$

$$\eta = \text{generator efficiency}$$

Appendix F Biomass Production Model

This appendix lists the equations for bioenergy production.

I. Biomass production and energy for generic plant (F.1)

The amount of bioenergy produced by a living plant is given by equation F.1 where α represents the maximum electron transport pathway process efficiency (33%), ω represents the maximum solar absorption efficiency (12%), β represents the percentage of solar radiation that lands on leaves (20%), γ represents the amount of energy overheads required to maintain the plant itself (60%) [197]. The constant 365 converts from days to year.

$$U_{bio} = (\tau * A_{bio} * \alpha * \omega * \beta * \gamma * 365)$$

where

τ = global solar radiation (kJ/m².day)

A_{bio} = biomass leaves or plant surface area (m²)

α = process efficiency

ω = absorption efficiency

β = solar radiation factor

γ = mass production efficiency

U_{bio} = amount of energy from biomass (kJ/year)

(F.1)

II. Biomass energy for Palm Oil (F.2)

The energy available in an empty fruit bunch is given by equation F.2.

$$U_{bio} = EFB_{CV} * Yield_{plant} * P_{bio}$$

where

EFB_{CV} = empty fruit bunches calorific value (kJ/kg)

$Yield_{plant}$ = biomass energy yield (kg/hectare.year)

P_{bio} = plantation area (hectare)

(F.2)

References

- [1] Schichl, H., 2004. Models and History of Modeling. In: J. Kalrath , ed. 2004. *Modeling Languages in Mathematical Optimization*. Dordrecht: Kluwer Academic Publishers. Ch. 2
- [2] Hafele, W., Rogner, H-H., 1984. A Technical Appraisal of the IIASA Energy Scenarios? A Rebuttal. *Policy Sciences*, 17(4), pp.341-365
- [3] Candy, L. 1998. Representations of strategic knowledge in design. *Knowledge-Based Systems*, 11(7-8), pp.379-390
- [4] Studer, R., Richard Benjamins, V., Fensel, D., 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*. 25(1-2), pp.161-197
- [5] Koskinen, K.U., Pihlanto, P., Vanharanta, H., 2003. Tacit knowledge acquisition and sharing in a project work context. *International Journal of Project Management*, 21(4), pp.281-290
- [6] Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hubner, S., 2001. Ontology-based integration of information -- a survey of existing approaches. In: A. Gomez Perez, M Gruninger, H. Stuckenschmidt, M. Uschold, ed., *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*. Seattle, United States of America 4-10 August 2001.
- [7] Mizoguchi, R., Kozaki, L., Sano, T., Kitamura, Y., 2000. Construction and deployment of a plant ontology. In: R. Dieng and O. Corby, ed., *Proceedings of the 12th European workshop on knowledge acquisition, modeling and management*. Juan-les-Pins, France 2-6 October 2000. London: Springer-Verlag
- [8] O'Leary, D. E., 1998. Using AI in knowledge management: knowledge bases and ontologies. *Intelligent Systems and their Applications*, 13(3), pp.34-39
- [9] Staab, S., Studer, R., Schnurr, H. P., Sure, Y., 2001. Knowledge Processes and Ontologies. *Intelligent Systems and their Applications*, 16(1), pp.26-34
- [10] Guarino, N., Giarretta, P., 1995. Ontologies and Knowledge Bases: Towards a terminological clarification. In: N.J.I. Mars, ed. 1995. *Towards Very Large Knowledge Bases*. Amsterdam: IOS Press, pp.25-32.
- [11] Noy, N., Sintek, M., Decker, S., Crubezy, M., Fergerson, R., Musen, M., 2001. Creating semantic Web contents with Protégé-2000, *IEEE Intelligent Systems*, 16(2), pp.60-71.
- [12] Neches, R., Fikes, R., Finin, T.m Gruber, T., Patil, R., Senator, T., Swartout, W.R., 1991. Enabling Technology for Knowledge Sharing. *AI Magazine*. 12(3), pp.36-56
- [13] Issa, R., Arciszewski, T., 2007. *Ontology: An Introduction, Teaching Modules in ASCE Global Center of Excellence in Computing*. [online] American Socieity of Civil Engineers. Available at: <http://www.asceglobalcenter.org/php-bin/modules.htm> [Accessed 6 October 2010]

- [14] Schreiber, G., Wielinga, B., Jansweijer, W., 1995. The KACTUS View on the 'O' Word. In: Workshop on Basic Ontological Issues in Knowledge Sharing, *International Joint Conference on Artificial Intelligence*. Montreal, Canada 20-25 August 1995.
- [15] Morbach, J., Wiesner, A., & Marquardt, W., 2009. OntoCAPE 2.0 - a (Re)Usable Ontology for Computer-Aided Process Engineering. *Computers and Chemical Engineering*, 33(10), pp.1546-1556.
- [16] Patel-Schneider, P.F., Horrocks, I., 2007. A comparison of two modelling paradigms in the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4), pp.240-250.
- [17] Yang, K.H., Olson, D., Kim, J., 2004. Comparison of first order predicate logic, fuzzy logic and non-monotonic logic as knowledge representation methodology. *Expert Systems with Applications*, 27(4), pp.501-519.
- [18] Lacy, L.W., 2005. *OWL: Representing Information Using the Web Ontology Language*. Victoria: Trafford Publishing.
- [19] Morbach, J., Yang, A., Marquardt, W., 2007. OntoCAPE-A large-scale ontology for chemical process engineering. *Engineering Applications of Artificial Intelligence*, 20(2), pp.147-161.
- [20] Venkatasubramanian, V., Zhao, C., Joglekar, G., Jain, A., Hailemaran, L., Suresh, P., Akkisetty, P., Morris, K., Reklaitis, G.V., 2006. Ontological informatics infrastructure for pharmaceutical product development and manufacturing. *Computers & Chemical Engineering*, 30(10-12), pp.1482-1496.
- [21] Berkan Sesen, M., Suresh, P., Bañares-Alcántara, R., Venkatasubramanian, V., 2010. An ontological framework for automated regulatory compliance in pharmaceutical manufacturing. *Computers & Chemical Engineering*, 34(7), pp.1155-1169.
- [22] Peachavanish, R., Karimi, H., Akinci, B., Boukamp, F. 2006. An ontological engineering approach for integrating CAD and GIS in support of infrastructure management, *Advanced Engineering Informatics*, 20(1), pp.71-88.
- [23] Kitamura, Y., Kashiwase, M., Fuse, M., Mizoguchi, R., 2004. Deployment of an ontological framework of functional design knowledge. *Advance Engineering Informatics*, 18(2), pp.115-127.
- [24] Kitamura, Y., Mizoguchi, R., 2003. Ontology-based description of functional design knowledge and its use in a functional way server. *Expert Systems with Applications*, 24(2), pp.153-166.
- [25] IBM alphaWorks, 2008. *Introduction to semantics technology*. [online] Available at: <http://www.alphaworks.ibm.com/contentnr/introsemantics> [Accessed 19 July 2008].
- [26] Shadbolt, N., Hall, W., Berners-Lee, T., 2006. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3), pp.96-101.
- [27] Marquardt, W., Morbach, J., Wesner, A., Yang, A., 2010. *OntoCAPE A Re-Usable Ontology for Chemical Process Engineering*. London: Springer

- [28] Braunschweig, B., Fraga, E., Guessoum, Z., Marquardt, W., Nadjemi, O., Paen, D., Pinol, D., Roux, R., Sma, S., Serra, M., Stalker, I., Yang, A., 2004. CAPE web services: the COGents way. *Computer Aided Process Engineering*, 18, pp.1021-1026.
- [29] Yang, A., Morbach, J., Marquardt, W., 2004. From conceptualization to model generation: the toles of ontologies in process modelling. In: C.A. Floudas, R. Agrawal, ed., *Proceedings of Sixth International Conference on Foundations of Computer-Aided Process Design*. Princeton, New Jersey 11-16 July 2004. Austin: CACHE Corp.
- [30] Kitamura, Y., Sano, T., Namba, K., Mizoguchi, R., 2002. A functional concept ontology and its application to automatic identification of functional structures. *Advanced Engineering Informatics*, 16(2), pp.145-163.
- [31] Keirstead, J., van Dam, K. H. 2010. A comparison of two ontologies for agent-based modelling of energy systems. In: *Proceedings of the First International Workshop on Agent Technologies for Energy Systems*. Toronto: Canada. 11 May 2010.
- [32] van Dam, K.H., Keirstead, J., 2010. Re-use of an ontology for modelling urban energy systems. In: *Proceedings of the Third International Conference on Infrastructure Systems and Services: Next Generation Infrastructure Systems for Eco-Cities*. Shenzhen: China. 11-13 November 2010.
- [33] Hung, C. Q., Batanov, D. N., Lefevre, T., 1998. KBS and macro-level systems: support of energy demand forecasting. *Computers in Industry*, 37(2), pp.87-95.
- [34] Batres, R., West, M., Leal, D., Price, D., Masaki, K., Shimada, Y., Fuchino, T., Naka, Y., 2007. An upper ontology based on ISO 15926. *Computers & Chemical Engineering*. 31(2-6), pp.519-534 .
- [35] Yao, J., Raghavan, V., Wu, Z., 2008 Web information fusion: A review of the state of the art. *Information Fusion*, 9(4), pp.446-449.
- [36] Buccella, A., Cechich, A., 2009. Ontology-Driven Geographic Information Integration: A Survey of Current Approaches. *Computer & Geosciences*, 35(4), pp.710-723.
- [37] Chandrasekaran, B., Josephson, J., Benjamins, V., 1999. What are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*, 14(1), pp.20-25.
- [38] Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A., 2002. Methodologies, tools and languages for building ontologies. Where is their meeting point?. *Data & Knowledge Engineering*, 46(1), pp.41-64.
- [39] Pulido, J. R. G., Ruiz, M. A. G., Herrera, R., Cabello, E., Legrand, S., Elliman, D., 2006. Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems*. 19(7), pp.489-497.
- [40] Su, X., Ilebrikke, L., 2002. A Comparative Study of Ontology Languages and Tools. *LNCS 2348 Springer-Verlag*, pp.761-765.
- [41] Patil, R.S., Fikes, R.E., Patel-Schneider, P.F., Mckay, D., Finin, T., Gruber, T., Neches, R., 1992. The DARPA Knowledge Sharing Effort: Progress Report.

In: *Proceedings of the Third International Conference on Principles Of Knowledge Representation and Reasoning*. Cambridge: Morgan Kaufmann.

- [42] Kifer, M., Lausen, G., Wu, J., 1995. Logical Foundation of Object-Oriented and Frame-Based Languages. *Journal of the Association for Computing Machinery*, 42(4), pp.741-843.
- [43] Lenat, D.B., Guha, R.V., 1990 *Building large knowledgebased systems : representation and inference in the Cyc project*. Wokingham: Addison-Wesley
- [44] W3C, 2008. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation 26 November 2008. [online] W3C. Available at: <http://www.w3.org/TR/xml/>
- [45] Coombs, J.H., Renear, A.H., DeRose, S.J., 1987. Markup systems and the future of scholarly text processing. *Communications of the ACM*, 30(11), pp.933-947.
- [46] W3C, 2004. *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation 10 February 2004. [online] W3C. Available at: <http://www.w3.org/TR/rdf-schema/>
- [47] Lassila, O., van Harmelen, F., Horrocks, I., Hendler, J., McGuinness, D.L., 2000. The semantic Web and its languages. *Intelligent Systems and their Applications*, 15(6), pp.67-73.
- [48] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., 2004. *OWL Web Ontology Language Reference*, W3C Recommendation 10 February 2004. [online] W3C. Available at: <http://www.w3.org/TR/owl-ref/>
- [49] Horrocks, I., Patel-Schneider, P.F., van Harmelen, F., 2003. From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1), pp.7-26.
- [50] Noy, N., Klein, M. 2004. Ontology Evolution: Not the Same as Schema Evolution. *Knowledge and Information Systems*, 6(4), pp.428-440.
- [51] Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A., 2004. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In: S. A. McIlraith, Plexousakis, D., Harmelen, F. ed., *Proceedings of the Third International Semantic Web Conference*. Hiroshima: Japan 7-11 November 2004.
- [52] Yang, S.Y., 2009. OntoPortal: An ontology-supported portal architecture with linguistically enhanced and focused crawler technologies. *Expert Systems with Applications*, 36(6), pp.10148-10157.
- [53] Kalyanpur, A., Parsia, B., Sirin, E., Cuenca Grau, B., Hendler, J., 2005. Swoop: A Web Ontology Editing Browser. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(2), pp.144-153.
- [54] Uschold, M., King, M., 1995. Towards a Methodology for Building Ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, *International Joint Conference on Artificial Intelligence*. Montreal, Canada 20-25 August 1995.

- [55] Gruninger, M., Fox, M.S., 1995. Methodology for the Design and Evaluation of Ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, *International Joint Conference on Artificial Intelligence*. Montreal, Canada 20-25 August 1995.
- [56] A. Bernaras, I. Laresgoiti, J. Corera, Building and reusing ontologies for electrical network applications, in: *Proc. European Conference on Artificial Intelligence (ECAI'96)*, Budapest: Hungary, 1996, pp. 298–302.
- [57] Swartout, B., Ramesh, P., Knight, K., Russ, T., 1997. Toward Distributed Use of Large-Scale Ontologies. *AAAI Symposium on Ontological Engineering*. Stanford, United States of America 24-27 March 1997.
- [58] Fernandez-Lopez, M, Gomerz-Perez, A., Sierra, J.P., Sierra, A.P., 1999. Building a chemical ontology using Methontology and the Ontology Design Environment. *Intelligent Systems and their Applications*, 14(1), pp.37-46.
- [59] Noy, N. F., McGuinness, D.L., 2001. *Ontology Development 101: A Guide to Creating Your First Ontology: Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*. Stanford: Stanford University Knowledge Systems Laboratory.
- [60] Fernandez Lopez, M., 1999. Overview Of Methodologies For Building Ontologies. . In: Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends, *International Joint Conference on Artificial Intelligence*. Stockholm, Sweden 31 July - 6 August 1999.
- [61] Fernandez Lopez, M., Gomez-Perez, A., 2003. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, 17(2), pp.129.156.
- [62] Gomez-Perez, A., Fernanzer-Lopez, M., Corcho, O., 2004. *Ontological Engineering*. London: Springer-Verlag
- [63] Sure, Y., Staab, S., Studer, R., 2003. On-To-Knowledge Methodology. in: S. Staab, R. Studer, ed., 2003. *Handbook on Ontologies*. Berlin: Springer-Verlag, pp.117-132.
- [64] Barstow, D., 1984. A Perspective on Automatic Programming. *AI Magazine*, 5(1), pp.5-27.
- [65] Mills, K.L., Gomass, H., 2002. Knowledge-based automation of a design method for concurrent systems. *IEEE Transactions on Software Engineering*, 28(3), pp.228-255.
- [66] Rich, C., Waters, R.C., 1988. Automatic programming: myths and prospects. *Computer*, 21(8), pp.40-51.
- [67] Barstow, D.R., 1985. Domain-Specific Automatic Programming. *IEEE Transactions on Software Engineering*, SE11(11), pp.1321-1336.
- [68] Vazquez-Roman, R., King, J.M.P., Bañares-Alcántara, R., 1996. KBMoSS: A process engineering modelling support system. *Computers & Chemical Engineering*, 20(1), pp.309-314.

- [69] Manna, Z., Waldinger, R.J., 1971. Towards Automatic Program Synthesis. *Communications of the ACM*, 14(3), pp.151-165.
- [70] Manna, Z., Waldinger, R., 1980. A Deductive Approach to Program Synthesis. *ACM Transactions on Programming Languages and Systems*, 2(1), pp.90-121.
- [71] Waldinger, R., Lee, R., 1969. PROW: a step toward automatic programming. *1st International Joint Conference on Artificial Intelligence*. Washington, United States May 1969. Waltham: William Kaufman.
- [72] Perkins, J.D., Sargent, W.H., Vazquez-Roman, R., Cho, J.H., 1996. Computer generation of process models. *Computers & Chemical Engineering*, 20(67), pp.635-639.
- [73] Uschold, M., 2008. Ontology-driven Information Systems: Past, Present and Future. *5th International Conference on Formal Ontology in Information Systems*, Saarbrücken, 31 Oct - 3 Nov 2008. Amsterdam: IOS Press.
- [74] Barstow, D.R., 1979. An experiment in knowledge-based automatic programming. *Artificial Intelligence*, 12(2), pp.73-119.
- [75] Borgida, A., Greenspan, S., Mylopoulos, J., 1985. Knowledge Representation as the Basis for Requirements Specifications. *Computer*, 18(4), pp.82-91.
- [76] Nakayama, Y., Mizutani, H., Sadashige, K., 1990. Model-based Automatic Programming for Plant Control. *Sixth Conference on Artificial Intelligence Applications*. Santa Barbara, United States of America 5-9 May 1990. Piscataway: IEEE Press.
- [77] Uschold, M., 1996. The use of the typed lambda calculus for guiding naive users in the representation and acquisition of part-whole knowledge. *Data & Knowledge Engineering*, 20(3), pp.385-404.
- [78] Messner S., Golodnikov A., Gritsevskii A., 1996. A stochastic version of the dynamic linear programming model MESSAGE III. *Energy*, 21(9) pp.775-784.
- [79] Klaassen G., Riahi K., 2007. Internalizing externalities of electricity generation: An analysis with MESSAGE-MACRO. *Energy Policy*, 35(2), pp.815-827.
- [80] Messner S., Schrattenholzer L., 2000. MESSAGE-MACRO: linking an energy supply model with a macroeconomic module and solving it iteratively. *Energy*, 25(3), pp.267-282.
- [81] Seebregts A.J., Goldstein G. and Smekens K., 2001. Energy/Environmental modelling using the MARKAL family of model. In: *Operations Research Proceedings 2001—Selected Papers of the International Conference on Operations Research (OR2001)*, Duisburg, Germany 3-5 September 2001.
- [82] Nakata T., 2004. Energy-economic models and the environment. *Progress in Energy and Combustion Science*, 30(4), pp.417-475.
- [83] Akimoto K., Tomoda T., Fujii Y. & Yamaji K., 2004. Assessment of global warming mitigation options with integrated assessment model DNE21. *Energy Economics*, 26(4), pp.635-653.

- [84] Hogan, W., 1975. Energy Policy Models For Project Independence. *Computers & Operations Research*, 2(3-4), pp.251-271.
- [85] Hafele, W., 1981. *Energy in a finite world*. Laxenburg: International Institute for Applied Systems Analysis, Energy Systems Group.
- [86] Keepin, B., Wynne, B., 1984. Technical analysis of IIASA energy scenarios. *Nature*, 312, pp.691-695.
- [87] Keepin, B., 1984. A technical appraisal of the IIASA energy scenarios. *Policy Sciences*, 17(3), pp.199-276.
- [88] Grubb, M., Edmonds, J., ten Brink, P., Morrison, M., 1993. The Costs of Limiting Fossil-Fuel CO₂ Emissions: A Survey and Analysis. *Annual Review of Energy and the Environment*, 18, pp.397-478.
- [89] Urban, F., Benders, R.M.J., Moll, H.C., 2007. Modelling energy systems for developing countries. *Energy policy*, 35(6), pp.3473-3482.
- [90] Van Beeck, N., 1999. *Classification of Energy Models*. Tilburg: Tilburg University.
- [91] Connolly, D., Lund, H., Mathiesen, B.V., Leahy, M., 2010. A review of computer tools for analysing the integration of renewable energy into various energy systems. *Applied Energy*, 87(4), pp.1059-1082.
- [92] Robinson, J.B., 1982. Energy backcasting A proposed method of policy analysis. *Energy Policy*, 10(4), pp.337-344.
- [93] Schwartz, P., 1996. *The Art of the Long View*. New York: Doubleday Dell Publishing Group.
- [94] Connell, L., Keane, M.T., 2004. What plausibly affects plausibility? Concept coherence and distributional word coherence as factors influencing plausibility judgments. *Memory & Cognition*. 32(2), pp.185-197.
- [95] Chumacero, R., Schmidt-Hebbel, K., 2005. *General Equilibrium Models for the Chilean Economy*. Santiago: Central Bank of Chile.
- [96] McFarland, J.R., Reilly, J.M., Herzog, H.J., 2004. Representing energy technologies in top-down economic models using bottom-up information. *Energy Economics*, 26(4), pp.685-707.
- [97] Kainuma, M., Matsuoka, Y., Morita, T., 2001. CO₂ emission forecast in Japan by AIM/end-use model. *Opsearch - Journal of the Operational Research Society of India*, 38(1), pp.109-125.
- [98] Krause, F., 1996. The costs of mitigating carbon emissions: A review of methods and finding from European studies. *Energy Policy*, 24(10-11), pp.899-915.
- [99] Manne, A.S., Wene, C.O., 1992. *MARKAL-MACRO: A Linked model for Energy-Economy Analysis*. Upton: Bookhaven National Laboratory.
- [100] Loulou, R., Goldstein, G., Noble, K., 2004. *Documentation of the MARKAL Family of Models*. Paris: ETSAP.

- [101] Beenstock, M., Daiziel, A., 1996. The demand for energy in the UK: A general equilibrium analysis. *Energy Economics*, 8(2), pp.90-98.
- [102] Bhattacharyya, D., 2005. *First Course in Linear Programming*. Kolkatta: Dr. Subhankar Dhur of Academic Publishers.
- [103] Zonooz, M.R.F., Nopiah, Z.M., Yusof, A.M., Sopian, K., 2009. A Review of MARKAL Energy Modelling. *European Journal of Scientific Research*, 26(3), pp.352-361.
- [104] Energieonderzoek Centrum Nederland (ECN), 1993. Appendix A 1.1 MARKAL Demystified. *National Energy Options For Reducing CO₂ Emissions, Volume 1: The International Connection*. Paris : International Energy Agency.
- [105] Kannan, R., Strachan, N., Pye, S., Anandarajah, G., Balta-Ozkan, N., 2007. *UK MARKAL Model Documentation*. [online] UKERC. Available at: <http://www.ukerc.ac.uk>
- [106] IPCC, 2000. *IPCC Special Reports on Climate Change: Emissions Scenarios*. [online] UNEP. Available at: http://www.grida.no/publications/other/ipcc_sr/
- [107] National Resources Canada, 2011. *RETScreen International*. [online] Available at: <http://www.etscreen.net>
- [108] Stadler, M., Kranzl, L., Huber, C., 2006. *The Invert Simulation Tool*. [online] Vienna University of Technology. Available at: http://www.invert.at/downloads/Manual_Report.pdf
- [109] Borges, A.M., 1986. Applied General Equilibrium Models : An assessment of their usefulness for policy analysis. *Economic Studies*, 7, pp.7-41.
- [110] Craig, P.P., Gadgil, A., Koomey, J.G., 2002. What can history teach us? A Retrospective Examination of Long-Term Energy Forecasts for the United States. *Annual Review of Energy and the Environment*, 27, pp.81-118.
- [111] Stevenson, A., Lindberg, C.A., ed., 2010. *New Oxford American Dictionary*. Oxford: Oxford University Press.
- [112] Mattsson, M., Bosch, J., 1997. Framework composition: problems, causes and solutions. In: R. Ege, M. Singh, B. Meyer, ed., *Proceedings of the International Conference on Technology of Object-Oriented Systems and Language*. Santa Barbara, United States of America 28 July - 1 August 1997. Los Alamitos: IEEE Computer Society.
- [113] van der Linden, N.H., Uytendinck, M.A., Vrolijk, C., Nilsson, L.J., Khan, J., Astrand, K., Ericsson, K., Wiser, R., 2005. *Review of International Experience With Renewable Energy Obligation Support Mechanisms*. Petten: Energieonderzoek Centrum Nederland (ECN).
- [114] Stroustrup, B., 1988. What is object-oriented programming?. *IEEE Software*, 5(3), pp.10-20.
- [115] Dobson, P. W., Starkey, K., Richard, J., 2004. *Strategic Management: Issues and Cases*. Oxford: Blackwell Publishing.

- [116] Aguilar, F.J., 1967. *Scanning the business environment*. New York: Macmillan.
- [117] Shilei, L., Yong, W. 2009. Target-oriented obstacle analysis by PESTEL modeling of energy efficiency retrofit for existing residential buildings in China's northern heating region. *Energy Policy*, 37(6), pp.2098-2101.
- [118] Chermack, T.J., Kasshanna, B.K., 2007. The Use and Misuse of SWOT Analysis and Implications for HRD Professionals. *Human Resource Development International*, 10(4), pp.383-399.
- [119] Guitouni, A., Martel, J.M., 1998. Tentative guidelines to help choosing an appropriate MCDA method. *European Journal of Operational Research*, 109(2), pp.501-521.
- [120] Haurant, P., Oberti, P., Muselli, M., 2011. Multicriteria selection aiding related to photovoltaic plants on farming fields on Corsica island: A real case study using the ELECTRE outranking framework. *Energy Policy*, 39(2), pp.676-688.
- [121] Bolden, R., Moscarola, J., 2000. Bridging the Quantitative-Qualitative Divide : The Lexical Approach to Textual Data Analysis. *Social Science Computer Review*, 18(4), pp. 450-460.
- [122] Drass, K., 1980. The Analysis of Qualitative Data : A Computer Program. *Journal of Contemporary Ethnography*, 9(3), pp. 332-353.
- [123] Maedche, A., Staab, S., 2000. Semi-Automatic Engineering Ontologies from Text. In: *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering*. Juan-les-Pins, France 2-6 October 2000.
- [124] Celjuska, D., Vargas-Vera, M., 2004. Semi-Automatic Population of Ontologies from Text. In: J. Paralic, G. Polzlbauer, A. Rauber, ed. *Proceedings of the Fifth Workshop on Data Analysis*. Tatranska Polianka, Slovak Republic 24-27 June 2004.
- [125] Balakrishna, M., Moldovan, D., Tatu, M., Olteanu, M. 2010,.Semi-Automatic Domain Ontology Creation from Text Resources. In: N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, D. Tapias, ed. *Proceedings of the Seventh International Conference on Language Resources and Evaluation*. Malta, 19 - 21 May 2010. Paris: European Language Resources Association
- [126] GAMS Software GmBH, 2011. *Welcome to the GAMS Home Page!*. [online] Available at: <http://www.gams.com>
- [127] AMPL Optimization LLC, 2011. *A Modeling Language for Mathematica Programming*. [online] Available at: <http://www.ampl.com>
- [128] Hoste, J., 2009. *Mathematica Demystified*. New York: McGraw-Hill.
- [129] Fateman, R.J., 1992. A review of mathematica. *Journal of Symbolic Computation*. 13(5), pp.545-579.
- [130] Zotos, K., 2007. Performance comparison of Maple and Mathematica. *Applied Mathematics and Computation*. 188(2), pp.1426-1429.

- [131] Cornil, J. M., Testud, P., 2000. *An introduction to Maple V*. Berlin:Springer-Verlag.
- [132] Attaway, S., 2009. *MATLAB: A Practical Introduction to Programming and Problem Solving*. Boston: Elsevier.
- [133] Tsarkov, D., Horrocks, I., 2006. FaCT++ Description Logic Reasoner: System Description, *LNCS 4130/2006 Springer*, pp.292-297.
- [134] Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., & Katz, Y., 2007. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*. 5(2), pp.51-53.
- [135] University of Oxford, 2011. *Hermit OWL Reasoner*. [online] Available at: <http://hermit-reasoner.com>
- [136] Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data Clustering: A Review. *ACM Computing Surveys*, 31(3), pp.264-323.
- [137] Object Refinery Limited, 2005. *Welcome To JfreeChart!* [online]. Available at: www.jfree.org/jfreechart
- [138] Weisser, H., 2007. The security of gas supply – a critical issues for Europe?, *Energy Policy*, 35(1), pp.1-5.
- [139] Vivoda, V., 2009. Diversification of oil import sources and energy security: A key strategy or an elusive objective?, *Energy Policy*, 37(11), pp.4615-4623.
- [140] Stringer, K., 2008. Energy Security: Applying a Portfolio Approach. *Baltic Security and Defence Review*, 10, pp.121-142.
- [141] Schneider, S. H., 1979. Comparative risk assessment of energy systems. *Energy*, 4(5), pp.919-931.
- [142] Gagnon, L., 2008. Civilisation and Energy Payback. *Energy Policy*, 36(9), pp.3317-3322.
- [143] Ardente, F., Beccali, M., Cellura, M., Brano, V.L., 2008. Energy performances and life cycle assessment of an Italian wind farm. *Renewable & Sustainable Energy Reviews*, 12(1), pp. 200-217.
- [144] Hondo, H., 2005. Life cycle GHG emission analysis of power generation systems: Japanese case. *Energy*, 39(11-12), pp.2042-2056.
- [145] Sovacool, B. K., 2008. Valuing the greenhouse gas emissions from nuclear power: A critical survey. *Energy Policy*, 38(8), pp.2950-2963.
- [146] Chatzimouratidis A.I., Pilavachi P.A., 2008. Multicriteria evaluation of power plants impact on the living standard using the analytic hierarchy process. *Energy Policy*, 36(3), pp.1074-1089.
- [147] Wei, M., Patadia, S., Kammen, D.M., 2010. Putting renewables and energy efficiency to work: How many jobs can the clean energy industry generation in the US?. *Energy Policy*, 38(2), pp.919-931.
- [148] Kenley, C.R., Klingler, R. D., Plowman, C. M., Soto, R., Turk, R. J., Baker, R.L., Close, S.A., McDonnell, V. L., Paul, S.W., Rabideau, L.R., Rao, S.S.,

- Reilly, B.P., 2009. Job creation due to nuclear power resurgence in the United States. *Energy Policy*, 37(11), pp.4894-4900.
- [149] United States Department of Energy, 2009. *Energy Efficiency and Renewable Energy, Weather Data Sources*. [online] Available at: http://apps1.eere.energy.gov/buildings/energyplus/weatherdata_sources.cfm
- [150] Stewart, C.T., 1992. *Air Pollution, Human Health, and Public Policy*. Toronto: Lexington Book.
- [151] Agterbosch, S., Meertens, R.M., Vermeulen, J.V., 2009. The relative importance of social and institutional conditions in the planning of wind power projects. *Renewable and Sustainable Energy Reviews*, 13(2), pp.393-405.
- [152] Yap, L.K., 2011. Letter to the PM on gas shortage. *The Star Newspaper*, 1 November.
- [153] Sukumaran, T., 2011. Task force to solve TNB's fuel shortage predicament. *The Edge Daily*, 12 October.
- [154] CEC, 2004. *North American Power Plant Air Emissions*. Montreal: Commission for Environmental Cooperation of North America.
- [155] Strupczewski, A., 2003. Accident risks in nuclear power plants. *Applied Energy*, 75(1-2), pp.79-86.
- [156] Sovacool, B.K., 2009. The intermittency of wind, solar, and renewable electricity generators: Technical barrier or rhetorical excuse?. *Utilities Policy*, 17(3-4), pp. 288-296.
- [157] Le Coq, C., Paltseva, E., 2009. Measuring the security of external energy supply in the European Union. *Energy Policy*, 37(11), pp.4474-4481.
- [158] Cabalu, H., 2010. Indicators of security of natural gas supply in Asia. *Energy Policy*, 38(1), pp.218-225.
- [159] Gupta, E., 2008. Oil vulnerability index of oil-importing countries. *Energy Policy*, 36(3), pp.1195-1211.
- [160] Institute For Economics & Peace, 2011. *Global Peace Index: 2011*. [online] Available at: <http://www.visionofhumanity.org/info-center/global-peace-index-2011/>
- [161] Romerio, F., 1998. The risks of the nuclear policies. *Energy Policy*, 26(3), pp.239-246.
- [162] Shimazaki, Y., Akisawa, A., Kashiwagi, T., 2000. A model analysis of clean development mechanisms to reduce both CO₂ and SO₂ emissions between Japan and China. *Applied Energy*, 66(4), pp.311-324.
- [163] EWEA, 2005. *Large Scale Integration of Wind Energy in the European Power Supply: analysis, issues and recommendations*. Brussels: European Wind Energy Association.
- [164] Ibl, N., Robertson, P.M., 1973. Optimisation of use of surplus night electricity in electrochemical processes. *Electrochimica Acta*, 18(11), pp.897-906.

- [165] Krausmann, F., Schandl, H., Rolf, P.S., 2008. Socio-ecological regime transitions in Austria and the United Kingdom. *Ecological Economics*, 65(1), pp.187-201.
- [166] Gale, B., 1981. Petronas: Malaysia's National Oil Corporation. *Asian Survey*, 21(11), pp.1129-1144.
- [167] *Petroleum Development Act 1974*. (c.3), Kuala Lumpur: Commissioner Of Law Revision Malaysia.
- [168] Kementerian Tenaga, Teknologi Hijau dan Air, 2010. *National Petroleum Policy*. [online] Available at: <http://www.kettha.gov.my/> [Accessed 7 Sept 2011].
- [169] Mohamed, A.R., Lee, K.T., 2006. Energy for sustainable development in Malaysia: Energy policy and alternative energy. *Energy Policy*, 34(15), pp.2388-2397.
- [170] Chua, S.C., Oh, T.H., 2010. Review on Malaysia's national energy developments: Key policies, agencies, programmes and international involvements. *Renewable and Sustainable Energy Reviews*, 14(9), pp.2916-2925.
- [171] Hashim, H., Ho, W.S., 2011. Renewable energy policies and initiatives for a sustainable energy future in Malaysia. *Renewable and Sustainable Energy Reviews*, 15(6), pp.4780-4787.
- [172] Economic Planning Unit, 2010. *10th Malaysian Plan*, Putrajaya: Prime Minister's Department.
- [173] Alamsyah, T.M.I., Sopian, K., Shahrir, A., 2004. Predicting average energy conversion of photovoltaic system in Malaysia using a simplified method. *Renewable Energy*, 29(3), pp.403-411.
- [174] Azhari, A.W., Sopian, K., Zaharim, A., Ghoul, M.A., 2008. A New Approach For Predicting Solar Radiation In Tropical Environment Using Satellite Images - Case Study Of Malaysia. *WSEAS Transactions on Environment and Development*, 4(4), pp.373-378.
- [175] Sopian, K., Othman, M. Y. Hj., Wirsat, A., 1995. The wind energy potential in Malaysia. *Renewable Energy*, 6(8), pp.1005-1016.
- [176] Exell, R.H.B., Fook, C.T., 1986. The wind energy potential of Malaysia. *Solar Energy*, 36(3), pp.281-289.
- [177] WEC, 2010. *Survey of Energy Resources 2010*. London: World Energy Council.
- [178] Ong, H.C., Mahili, T.M.I., Masjuki, H.H., 2011. A review on energy scenario and sustainable energy in Malaysia. *Renewable and Sustainable Energy Reviews*, 15(1), pp.639-647.
- [179] Economic Planning Unit, 2010. *The Malaysian Economy In Figures 2010*. Putrajaya: Prime Minister's Department.

- [180] Economic Planning Unit, 2006. *9th Malaysian Plan*. Putrajaya: Prime Minister's Department.
- [181] Economic Planning Unit, 2005. *The Malaysian Economy In Figures 2005*. Putrajaya: Prime Minister's Department.
- [182] Santisirisomboon, J., Limmeechokchai, B., Chungpaibulpatana, S., 2003. Least cost electricity generation options based on environmental impact abatement. *Environmental Science & Policy*, 6(6), pp.533-541.
- [183] Wang, Q., Cui, Q., Zhou, D., Wang, S., 2011. Marginal abatement costs of carbon dioxide in China: A nonparametric analysis. *Energy Procedia*, 5, pp.2316-2320.
- [184] NRE, 2011. *Malaysia Second National Communication to the UNFCCC*. Putrajaya: Ministry of Natural Resources and Environment Malaysia
- [185] Yee, L.H., 2011. Enter a new era of green energy. *The Star Newspaper*, 3 December.
- [186] Economic Transformation Programme, 2010. *A Roadmap for Malaysia*. Putrajaya: Prime Minister's Department.
- [187] TNB, 2004. *Annual Report 2004*. Kuala Lumpur: Tenaga Nasional Berhad
- [188] TNB, 2008. *Annual Report 2008*. Kuala Lumpur: Tenaga Nasional Berhad
- [189] Yan, H., Wei, Q., Wang, J., 2005. Constructing efficient solutions structure of multiobjective linear programming. *Journal of Mathematical Analysis and Applications*, 307(2), pp.504-523.
- [190] Darton, R.C., 2003. Scenarios and Metrics as Guides to a Sustainable Future: *The Case of Energy Supply. Process Safety and Environmental Protection*, 81(5), pp.295-302.
- [191] Stirling, A., 2010. Multicriteria diversity analysis: A novel heuristic framework for appraising energy portfolios. *Energy Policy*, 38(4), pp.1622-1634.
- [192] Wark, K., Richards, D., 1999. *Thermodynamics*. Boston: WCB McGraw-Hill.
- [193] PETRONAS GAS Bhd, 1996. *Water Treatment System 056/1009/056-03*. [Process Flow Diagram] Revision 3. Kuala Lumpur: PETRONAS GAS Bhd.
- [194] Kohan, A.L., 1997. *Boiler Operator's Guide*. Boston: McGraw-Hill.
- [195] Sediadi, E., 2004. Simple calculation of Photovoltaic/PV Solar Electricity Product in Buildings. In: *Proceedings of the 5th International Conference on Sustainable Environment and Architecture*. Skudai, Malaysia 10-12 December 2004. Kuala Lumpur: Universiti Teknologi Malaysia.
- [196] ESTIF, 2007. *Simple Calculation of energy delivery of (small) ST systems*. Brussels: European Solar Thermal Industry Federation.
- [197] Andrews, J., Jelley, N., 2007. *energy science: principles, technologies, and impacts*. Oxford : Oxford University Press.