INTELLIGENT VISUAL TRAVEL RECOMMENDER SYSTEMS MODEL FOR E-TOURISM WEBSITES

Nalin Sharda, Roopa Jakkilinki, Mladen Georgievski and Mohan Ponnadai
Technical Reports

The technical report series present data and its analysis, meta-studies and conceptual studies, and are considered to be of value to industry, government and researchers. Unlike the Sustainable Tourism Cooperative Research Centre’s Monograph series, these reports have not been subjected to an external peer review process. As such, the scientific accuracy and merit of the research reported here is the responsibility of the authors, who should be contacted for clarification of any content. Author contact details are at the back of this report.

We'd love to know what you think of our new research titles. If you have five minutes to spare, please visit our website to complete our online survey.

Sustainable Tourism CRC Tech Report Feedback

National Library of Australia Cataloguing-in-Publication entry

Intelligent visual travel recommender systems model for e-tourism websites / author: Nalin Sharda ... [et al.].

Publisher: Gold Coast, Qld. : Sustainable Tourism CRC, 2008.

ISBN: 9781920965709 (pbk and pdf)

Notes: Bibliography.


Other Authors/Contributors: Jakkilinki, Roopa. Georgievski, Mladen. Ponnada, Mohan.

338.47910285

Copyright © CRC for Sustainable Tourism Pty Ltd 2008

All rights reserved. Apart from fair dealing for the purposes of study, research, criticism or review as permitted under the Copyright Act, no part of this book may be reproduced by any process without written permission from the publisher. Any enquiries should be directed to:

General Manager, Communications and Industry Extension or Publishing Manager, info@crctourism.com.au

First published in Australia in 2008 by CRC for Sustainable Tourism Pty Ltd

Edited by Kelly Van Asperen

Printed in Australia (Gold Coast, Queensland)

Cover designed by Sin Design
CONTENTS

Abstract ........................................................................................................ v
Acknowledgements ...................................................................................... v
SUMMARY .................................................................................................... vi
Objectives of Study ...................................................................................... vi
Methodology .................................................................................................. vi
Key Findings ................................................................................................ vii
Future Action ............................................................................................... vii
CHAPTER 1 Introduction ................................................................................ 1
CHAPTER 2 AuSTO Ontology Creation Process .............................................. 2
Introduction to Ontology ................................................................................ 2
The AuSTO Ontology ...................................................................................... 2
Scope .............................................................................................................. 2
Users ............................................................................................................. 2
Structure ......................................................................................................... 2
AuSTO Development Methodology ............................................................... 2
Ontology levels ............................................................................................... 3
Foundational classes ................................................................................... 3
Intentional classes ....................................................................................... 4
Operational classes ..................................................................................... 4
Ontology roles ............................................................................................... 5
Ontology information flows ......................................................................... 5
The AuSTO Ontology Documentation ......................................................... 6
Traveller Preference ....................................................................................... 6
TravellerRequirement ..................................................................................... 7
TourismRelatedEvent .................................................................................... 7
TravellerPlan ................................................................................................ 7
Traveller ........................................................................................................ 7
Actor .............................................................................................................. 7
Activity .......................................................................................................... 7
ContactPoint ............................................................................................... 8
Destination .................................................................................................... 8
GeographicPlace .......................................................................................... 8
AuSTO Tour Planner System Design ............................................................ 8
Ontology Layer ............................................................................................. 9
Business Logic Layer ................................................................................... 9
Graphical User Interface Layer ................................................................. 10
CHAPTER 3 User Interface Design ................................................................. 11
Task Analysis and Task Modelling ............................................................... 11
Concur Task Tress Environment (CTTE) .................................................... 11
Task Model for Itinerary Planner ............................................................... 12
User Interface Design .................................................................................. 16
CHAPTER 4 Intelligence Layer ................................................................. 19
  What is the Semantic Web ................................................................. 19
  How can the Semantic Web help? ...................................................... 19
  Limitations of Semantic Web for Tourism Websites ......................... 19
  Implementation Overview .............................................................. 20
    Semantic Web Application Model for AuSTO .................................. 20
    Ajax (Asynchronous JavaScript and XML) ....................................... 21
    Web and XML Server .................................................................. 21
    Jena Middleware ......................................................................... 22
    Relational Database Server ......................................................... 22

CHAPTER 5 Demonstrating the Potential of Travel Recommender Systems ____ 23
  Recommender Systems .................................................................... 23
    Tourism Recommendation using Image Based Planning (TRIP) ........ 24
    Tour Recommendation using Image-Based Planning and SCORM (TRIPS): 24
    Visual tour using TRIPS .............................................................. 25
    TRIPS Architecture ..................................................................... 27

CHAPTER 6 Conclusion & Further Research ....................................... 28
  APPENDIX A: Peer Reviewed Publications (related to this project) ....... 29
  APPENDIX B: AuSTO Implementation Source Code Examples ............ 30
  REFERENCES ............................................................................... 31
  GLOSSARY .................................................................................. 32
  AUTHORS .................................................................................. 34
INTELLIGENT VISUAL TRAVEL RECOMMENDER SYSTEMS MODEL FOR E-TOURISM WEBSITES

ABSTRACT

Rapid advancements in Information Communication Technologies (ICT) have enabled the development of information rich e-Tourism websites. However, many of these websites are cumbersome to use and do not provide a satisfying browsing experience. Therefore, there is a need for developing better e-Tourism websites that are easy to use and provide more efficient access to the information and services that the user requires. This can be achieved with the help of Semantic Web technologies. The Semantic Web augments the current Web systems by using Ontology and software tools that can derive knowledge for the Ontology for developing intelligent applications.

This report presents the Australian Sustainable Tourism Ontology (AuSTO) developed in this project, based on the National Information Architecture for the Australian Tourism Industry (NIAATI) specifications. An open source, Semantic Web toolkit called Jena was used to develop the intelligence layer. A prototype user interface was developed using task modelling techniques, to abstract the essential tasks which a user would perform to access travel information and make reservations. This interface was then implemented using a design environment called CTTE (Concur Task Trees Environment). This methodology and collection of tools provides a framework for developing e-Tourism websites which provide easy and efficient access to travel information.

The other aspect of this report demonstrates the potential of Travel Recommender Systems and presents a model for developing Intelligent Visual Travel Recommender Systems (IV-TRS). This model uses the Sharable Content Object Reference Model (SCORM) and Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA) developed originally for the e-Learning application domain.

Acknowledgements

The Sustainable Tourism Cooperative Research Centre, established and supported under the Australian Government’s Cooperative Research Centres Program, funded this research.

The authors of this report would like to thank Mr. Henk Meijerink, and Mr Paul Moynihan for their contributions to project # 80051, and Mr Jayden Kimber and Mr S.M. Guru for their contributions to project # 80052.
INTELLIGENT VISUAL TRAVEL RECOMMENDER SYSTEMS MODEL FOR E-TOURISM WEBSITES

SUMMARY

Tourism is a dynamic industry. In recent years, with many countries turning to tourism to supplement their economies, there has been a massive expansion of tourism vendor offerings. As more travel arrangements are made online, pressure is put on e-Tourism website developers to provide efficient and easy to use interfaces and intelligent services.

A new model of the web called Semantic Web can be used to develop intelligent e-Tourism services. Semantic Web is a collection of models, technologies and tools that allow machines to create and process web content intelligently.

Travel Recommender System (TRS) is an application that makes use of expert systems to provide quality destination selection recommendations to the user; however current systems are cumbersome to use and do not provide a collated visualisation of the entire tour.

This report gives of the model of an Intelligent Visual Travel Recommender Systems (IV-TRS), which uses the Semantic Web technologies to help create efficient and easy to use e-Tourism websites for tour planning.

Objectives of Study

The main aim of this research was to investigate how Semantic Web technologies can be used to develop intelligent e-Tourism services. This aim was tackled by a three prong approach, as articulated in the following objectives:
1. Research the application of Semantic Web technologies for developing better e-Tourism systems.
2. Develop a framework for creating e-Tourism websites using Semantic Web tools.
3. Demonstrate the potential of Travel Recommender Systems and how an Intelligent Visual Travel Recommender System (IV-TRS) can be developed using Semantic Web tools.

Methodology

The methodology used in this research included: theoretical study of new web technologies and standards, development of new e-Tourism models and their prototype implementations, followed by comparative analysis of various implementation options. This methodology was executed using the following five steps:
1. Investigated and evaluated Semantic Web technologies, standards and tools.
2. Applied these Semantic Web technologies, standards and tools to develop a framework for creating intelligent e-Tourism systems.
4. Analysed SCORM, CORDRA, and SMIL standards.
5. Developed the prototype of an Intelligent Visual Travel Recommender System (IV-TRS).
Key Findings

From the research carried out in this project we concluded that Semantic Web technologies can be used to develop intelligent e-Tourism service; however there will be some time before such systems reach maturity, and can be deployed for large scale e-Tourism systems. More specifically:

- It is possible to use Semantic Web technologies to develop better e-Tourism systems.
- Semantic Web application library called Jena is suitable for developing intelligent e-Tourism website.
- Current Travel Recommender Systems (TRSs) are useful; however they lack visualisation ability.
- Standards such as SCORM and CORDRA—originally developed for e-Learning systems—can be used to develop e-Tourism systems that provide greater content portability.
- The proposed Intelligent Visual Travel Recommender System (IV-TRS) can enhance the speed of information access and the quality of information presentation, by combining web-base content into a video-like presentation, and help travellers in tour planning.

Future Action

Future research will aim to conduct usability study on the effectiveness of continuous visualisation systems, and develop recommendations for the deployment of the proposed Intelligent Visual Travel Recommender System (IV-TRS). It will require the following steps to achieve this aim:

- Extend AuSTO ontology with more data and rules.
- Enhance the user interface and task modelling framework.
- Conduct usability tests on the user interfaces.
- Develop and deploy a demonstration e-Tourism website as an Intelligent Visual Travel Recommender System.
Chapter 1

INTRODUCTION

Tourism is one of the most successful and dynamic industries in the world. Most tourism companies have elaborate websites to present and sell their products. The tourism industry is an information intensive business, and the amount of information is increasing rapidly; however efficient access to this information is becoming a challenge. Due to the deluge of tourism vendor offerings and ever increasing numbers of travellers, the limitations associated with conventional e-Tourism websites are coming to the fore.

This project developed the Australian Sustainable Tourism Ontology (AuSTO), nevertheless it can be mapped to the requirements of other countries, as there are many concepts common to the tourism industry world-wide. To achieve this the developers for e-Tourism portals need better tools and methodology so they can swiftly create e-Tourism websites that are easy to use. This report proposes a solution using the Semantic Web. This is then liked to a visualisation system that presents users a video-like continuous presentation of the information.

The Semantic Web (Berners-Lee 2001) is a methodology and a set of tools that would enable computers to access the World Wide Web and use it intelligently. It does this by having websites created in a machine-readable form with a common language using a standard ontology and artificial intelligence technologies. The Semantic Web methodology requires a common language describing tourism entities and concepts; for this project, this was implemented as AuSTO. In addition to a common language, there should be a means to embed intelligence into a website. The Semantic Web uses inference engines and collections of logic rules to model intelligent behaviour. For this project, the reasoning engine used was called ‘Racer’. This reasoning engine uses the ontology, and the rules associate with the same.

To develop a user oriented system, a framework for modelling system behaviour from a user’s perspective is needed. For this project Concurrent Task Trees Environment (CTTE) was used. A framework for e-Tourism websites was developed using the AuSTO ontology, a Semantic Web toolkit called Jena, and the user interface modelling environment called CTTE. In addition, server programs were used to retrieve and respond to the user’s requests from web pages presented to the user. These websites can easily be given smart searching capability and the ability to anticipate user requirements. As the ontology contains knowledge of tourism concepts and rules of their interaction; the ontology can transport this intelligence, if transferred to another site.

Travel Recommender Systems are web-based systems that provide tourists with valuable recommendations based on their specifications. These recommendations are provided by an artificial intelligence based engine. Usually, when a person wants to go on a tour, there are many questions that need to be answered. The traveller is faced with the task of deciding on the most appropriate choices. The traveller will often start acquiring information from various sources such as websites, books, and suggestions from other people. The data collected in this manner functions as recommendations and help the traveller in making informed decisions about their desired tour. However the recommender systems being used today do not provide the facility to visualise the tour. This project tackles this problem by developing an Intelligent Visual Travel Recommender System (IV-TRS).

Chapter 2 presents the details of the AuSTO ontology; Chapter 3 presents the User Interface Design. Chapter 4 presents the Intelligence Layer framework. Chapter 5 discusses Travel Recommender Systems and aims to demonstrate their potential in e-Tourism and chapter 6 presents Conclusions and further research plans.
Chapter 2

AuSTO ONTOLOGY CREATION PROCESS

Introduction to Ontology
Ontology is a data model that represents a domain and is used to reason about the objects in that domain and the relations between them. Ontologies represent knowledge about the world or some part of it, they consist of:

- **classes**—collection of objects
- **attributes**—properties an object can have and share
- **relations**—the way the objects are related
- **individuals**—instances of a class

(Chandrasekaran, Josephson & Benjamins 1999).

An ontology can be a domain ontology or a theory ontology (Swartout, Patil, Knight & Russ 1997). A domain ontology models a specific domain; it represents the particular meanings of terms as they apply to that domain, e.g. tourism. A theory ontology provides a set of concepts for representing some aspect of the real world, such as time and space.

The AuSTO Ontology

**Scope**
The AuSTO ontology has been created for the tourism domain; it covers sub domains including vendor offerings, tourist requirements and tourism itinerary.

**Users**
The AuSTO ontology acts as a knowledge base for the application generator framework, the users of this ontology are tourism domain experts who maintain the ontology; the tourists do not interact with the ontology directly.

**Structure**
Creating an ontology involves delineating concepts into a class hierarchy. Three important approaches to develop class hierarchies are top-down, bottom-up and a combination approach (Uschold & Gruninger 1996). In the top-down approach the development process starts with the definition of the most general concepts in the domain, followed by specialised concepts. In the bottom-up approach the development process starts with the definition of the most specific classes, which form the leaves of the class hierarchy tree with subsequent grouping of these classes into more general concepts. The combined approach uses a combination of top-down and bottom-up processes. The approach followed for AuSTO is the bottom-up approach. This approach is usually driven by the need for having a workable vocabulary, which can be expanded as required. The AuSTO ontology is an (Web Ontology Language (OWL)—based ontology and it defines the various tourism terms and their relationships.

AuSTO Development Methodology
Ontology development must follow a systematic methodology. The methodology followed to develop the AuSTO ontology consists of the following steps (Jakkilinki, Sharda & Georgievski 2005):

- **Identify the purpose behind ontology development**: Understanding why the ontology is being developed is important for achieving the desired outcomes.
INTELLIGENT VISUAL TRAVEL RECommENDER SYSTEMS MODEL FOR E-TOURISM WEBSITES

- **Ontology capture mechanism**: Ontology capture comprises three stages:
  - determining the scope of the ontology
  - selecting a method to develop the ontology
  - defining the concepts in the ontology
- **Coding the ontology**: Coding refers to representing the ontology in some formal language. A suitable ontology editor has to be selected; for developing the AuSTO ontology the editor used is Protégé.
- **Refinement**: This consists of two phases, namely: Intra-coding refinement, and Extra-coding refinement.
- **Testing**: The testing process uncovers any defects in function logic and implementation.
- **Maintenance**: This involves all the activities undertaken to see that the ontology is functioning correctly. The three types of maintenance are corrective, adaptive and perfective.

Further details of the ontology development methodology are presented in Appendix A.

**Ontology Levels**

The classes in the AuSTO exist at three levels: foundational, intentional and operational, as shown in Figure 1.

![Figure 1: Classes in AuSTO](image)

Foundational level classes (Herre & Loebe, 2005) are the most generic classes in AuSTO. Some of the classes are constructed into the generator as code classes. All of the classes are likely to be read from the ontology at run-time, generating the required tourism applications. Most of the foundational classes were derived from generic entities in the National Information Architecture for the Australian Tourism Industry.

Intentional level classes are motivated by AuSTO’s application domains, most are specialisations (sub-classes) of foundational classes. They are responsible for functionality and control in tourism applications. Some intentional classes are constructed into tourism applications as code classes by the generator. Others are read from the ontology at run-time as and when they are required to control some aspect of the e-tourism application.

Operational level classes are the most detailed and most numerous in AuSTO. These are mostly specialisations (sub-classes) of intentional classes. They constitute operational knowledge that is captured and exploited by tourism applications. A tourism application derives its value proposition from this operational knowledge.

**Foundational Classes**

Foundational classes include the most generic classes. AuSTO foundation classes comprise the following:
- **Involved party** functional classes include traveller, vendor/operator, etc.
INTELLIGENT VISUAL TRAVEL RECOMMENDER SYSTEMS MODEL FOR E-TOURISM WEBSITES

- **Requirement** functional classes include travel requirements
- **Offering** functional classes include travel products and services, etc.
- **Solution** functional classes include itineraries, etc.
- **Resource** functional classes include reserved rental items, etc.
- **Specification** functional classes include both offering specifications and requirement specifications
- **Preference** functional classes include travellers preferences such as date, time, location etc.

Each of these classes embodies a vast range of tourism information. For example, **Requirement** represents diverse travel requirements such as accommodation, entertainment, transport, etc., and **Offering** represents the wide range of travel products and services that vendors make available to the traveller, often as part of a packaged solution.

**Intentional Classes**

Many of the classes at this level are specialisations of foundational classes. All intentional classes aim to meet functional needs of specific tourism applications. For example, a travel planner application will need the following intentional classes:

- vendor type
- vendor-neutral
- offering type
- party-neutral requirement type
- specification element
- travel solution type (e.g. ‘itinerary’), destination, etc.

**Operational Classes**

Operational classes include a small number of concrete classes and a large number of abstract classes. Concrete classes are required for at least the following operational entities:

- tourism vendor
- vendor offering
- traveller travel requirement.

Abstract classes will be dynamically created to represent very detailed operational specialisations of at least the following intentional classes:

- vendor-neutral offering type
- party-neutral requirement type
- specification element.

Abstract operational classes are likely to be numerous and dynamically rationalised. Being dynamically rationalised implies that these classes will be merged and divided dynamically to ensure that each has a distinct and rational definition in terms of necessary and sufficient conditions for membership of the class (albeit ‘hypothetical’ membership in the case of abstract classes in an abstracted ontology). For example, as users enter their travel requirements, the application responds by creating abstractions in the form of sub-classes of **party-neutral requirement type**. In the background, the system dynamically rationalises these sub-classes in the overall ontology.
Abstract operational classes are not likely to have meaningful names. This is due to their abstract and dynamic nature. Instead, wherever meaningful descriptions are required, they will be gleaned from user interactions and captured into a property of the respective class.

**Ontology Roles**

The abstracted ontology will play several roles throughout the system’s lifecycle (Spyns, Meersman, & Jarrar, 2002), as described in the following:

- Foundational classes will drive much of the functional specification of the application generator. Many of these classes are likely to be implemented directly into the generator as application code classes.
- Intentional classes are derived through application domain specific analysis, to extend the ontology, enabling it to drive the running generator to construct custom tourism applications. In fact, the generator will implement many of the intentional classes directly into the target applications as application code classes.
- Intentional classes in the ontology will drive the running tourism applications to render meaningful labels and captions in the user interface.
- Intentional classes in the ontology assist information analysts in analysing complex vendor data and to capture their results. The most significant vendor data includes product and service offerings and related specifications. The results captured will extend the ontology further with a very large number of operational classes representing all kinds of vendor-neutral offering types, and a lexicon of all the different specification elements needed to define them.
- Having been extended with numerous operational classes representing travel offerings, the ontology will then drive the running tourism applications to assist its primary users in capturing their travel requirements and preferences. This extends the ontology further with an even greater number of operational classes.
- The ontology also drives the e-tourism applications, for example, to assemble optimal solutions that meet consumer travel requirements. Solution components are inferred through automated reasoning over the ontology (McGuiness, n.d.). This reasoning is possible because requirements and offerings are defined in terms of specification elements drawn from the same lexicon.

**Ontology Information Flows**

This section describes how different sources of knowledge flow into the ontology and the role played by ontology elements in the e-tourism system’s lifecycle.

The AuSTO ontology is based on the National Information Architecture for the Australian Tourism Industry (NIAATI). This architecture comprises among other things conceptual data models, conceptual process models and cross-references between them. It provides a contextual framework for the development of e-tourism systems; the AuSTO ontology is a case in point.

The high level industry-wide analysis that went into producing the NIAATI identified a range of generic entities. These became the basis for the foundational classes in AuSTO, and will enable the ontology to play its first big role as functional specification for developing the e-tourism application generator. The generic nature of these foundational classes ensures the generator will have ample domain extensibility, enlarging the range of applications it will be able to generate.

More detailed analysis is aimed at producing the specifications for particular applications to be generated. This analysis produces intentional classes into the ontology, enabling the ontology’s second role as functional specification for the range of e-tourism applications that can be generated.

Information analysts need to undertake more detailed analyses of vendor offerings (i.e. travel products and services). This will extend the ontology with operational classes, particularly sub-classes of *specification element* and sub-classes of *vendor-neutral offering type*. The latter is defined in terms of the former and together these amount to a very large number of operational classes. It is very likely...
that much of this analysis and knowledge capture will be performed using one or more of the generated applications. This operational knowledge about offering types contributes to preparing the ontology for its most important role: as the basis for inferring optimal travel solutions.

Vendors, including inbound tour operators, tourism facility operators, etc. will use one or more of the ontology driven tourism applications to link their specific offerings (e.g. accommodation, rental items, etc.) to the most relevant sub-classes of vendor-neutral offering type. As they do this, the application will extend the ontology with operational individuals for tourism vendors and vendor offering.

Finally, travellers will use one or more of the ontology driven tourism applications to help them analyse their travel plans, record their travel requirements and obtain travel solutions (e.g. itineraries, packaged tours, etc.). As users record their travel requirements, the application extends the ontology with more operational classes. In particular, it adds sub-classes of traveller-neutral requirement types that haven’t been encountered before. Importantly, these requirement types are defined in terms of the same specification elements used to define sub-classes of vendor-neutral offering type. With these additions the ontology is finally ready for its ultimate role as the basis for inferring optimal travel solutions. An automated reasoner infers sets of offering that match sets of requirement by applying logic over the specification elements in their definitions. A reasoner is the software that applies axiomatic knowledge in the knowledge base to the task-specific data embodied for arriving at some conclusions (Inference Engine, n.d.). A custom built component may be required to logically arrange the offerings in the optimal set, thus completing the solution.

The AuSTO Ontology Documentation

This section presents details of the following important classes in the AuSTO ontology:

- Traveller Preference
- TravellerRequirement
- TourismRelatedEvent
- TravellerPlan
- Traveller
- Actor
- Activity
- ContactPoint
- Destination
- GeographicPlace

Traveller Preference

The Traveller Preference class is used to store the likings of the traveller with regards to accommodation, transport etc. For example he/she might like three star accommodation and luxury travel. It has the following sub-classes:

- AccommodationPreference
- ActivityPreference
- BrandPreference
- ConferenceRoomPreference
- GuestRoomPreference
- TourismEventPreference
- TransportPreference
**TravellerRequirement**
The Traveller Requirement class is used to store the traveller’s requirements for a given trip. These requirements are used to search for suitable options so as to satisfy the user requirements. It comprises the following sub-classes:
- AccommodationRequirement
- ActivityRequirement
- EventRequirement
- TransportRequirement

**TourismRelatedEvent**
The TourismRelatedEvent class stores events of interest to travellers. It has the following sub-classes. As stated earlier, more sub-classes can be added to the ontology as and when required. These additions will take place in response to the information provided by the various operators.
- ConferenceEvent
  - HealthConference
  - ICTConference
  - TourismConference
- EntertainmentEvent
  - Ballet
  - PopConcert
- ExhibitionEvent
  - ArtExhibition
  - ComputerExhibition
  - TradeExhibition
- SportingEvent
  - AthleticEvent
  - SailingEvent

**TravellerPlan**
The Traveller Plan class represents the activity plan of a traveller, and is the basis of a more detailed travel itinerary. It has the following sub-classes:
- Itinerary
- PersonalPlan

**Traveller**
The Traveller class stores details of an individual who is travelling. It has the following sub-classes:
- IndividualTraveller
- TravelGroup

**Actor**
The Actor class represents an individual who maintains the ontology or uses the application. It has the following two sub-classes:
- OntologyAdministrator
- User

**Activity**
The Activity class represents a process or involvement which is of relevance for a journey or business trip. It has the following sub-classes:
- AdministrativeActivity
- EnterpriseActivity
- ImmunisationAppointment
- TourismActivity
INTELLIGENT VISUAL TRAVEL RECOMMENDER SYSTEMS MODEL FOR E-TOURISM WEBSITES

- Cinema
- FormalVisiting
- InformalVisiting
- Shopping

**ContactPoint**
The ContactPoint class represents a point of contact, typically with a person. It has the following sub-classes:
- Desk
- EmailAddress
- PhoneNo
- PhysicalAddress
- WebSiteAddress

**Destination**
The Destination class represents a place designated as the goal, typically a travel target, such as a hotel, park, attraction or organised activity. It has the following sub-classes:
- ElementaryDestination
- CompositeDestination

**GeographicPlace**
The Geographic Place class represents a physical place with a geographic address. It has the following sub-classes:
- Continent
- Country
- Province
- Region
- State

It is important to remember that this class hierarch can be extended vertically as well as horizontally. Including further sub-sub-classes will extend the hierarchy vertically. Adding more sub-classes at a given level will extend the hierarchy horizontally.

**AuSTO Tour Planner System Design**
The AuSTO Tour Planner System Design uses a layered architecture, as shown in Figure 2. It consists of three layers: the Ontology Layer (OL), the Business Logic Layer (BLL) and the Graphical User Interface Layer (GUIL). The OL includes the tourism ontology and any additional specifications required for system operation; the Business Logic Layer includes the two common components: Inference Engine (IE) and the Custom Logic (CL) and part of the tour planner applications. The GUIL layer includes the Graphical User Interface (GUI) components of the tour planner application.
The Ontology Layer consists of the ontology which embodies knowledge about the domain and some additional specifications. This layer is the core of all semantic web systems. Creating the ontology involves delineating travel concepts into a class hierarchy. The three important approaches to develop class hierarchies are top-down, bottom-up, and a combination approach (Uschold & Gruninger 1996).

- The top-down approach begins by defining the most general concepts in the domain, followed by expanding these into more specialised concepts.
- The bottom-up approach starts by defining the most specific classes—which form the leaves of the class hierarchy tree—with subsequent grouping of these classes into more general concepts.
- The combined approach uses a combination of top-down and bottom-up processes.

The AuSTO ontology has been developed using the bottom-up approach; this approach is usually driven by the need for having a workable vocabulary to bootstrap the ontology development process. AuSTO is implemented using Web Ontology Language (OWL). The Ontology Layer (OL) acts as a knowledge store for the AuSTO system as it contains data regarding the tourism domain and the Business Logic Layer acts upon this layer to generate itineraries.

The Business Logic Layer (BLL) is sandwiched between the GUIL and the OL, it takes the user queries from the GUIL and matches them with the domain knowledge residing in the Ontology Layer. The results are then displayed to the user via the GUIL. BLL comprises a reasoner and custom logic. Considering that the knowledge base is very large, inference mechanisms are required to search through the knowledge base and deduce results in an organised manner. A Reasoner software is used to apply axiomatic knowledge in the knowledge base to the task-specific data. The knowledge embodied in this data is used for arriving at some conclusions (Inference Engine n.d.). The Custom Logic is a mechanism which acts as a mediator between the reasoner, user interface and the ontology, and needs to be customised for each e-tourism application.
**Graphical User Interface Layer**

The Graphical User Interface Layer (GUIL) comprises intuitive interfaces that allow the user to interact with the AuSTO Tour Planner tool. These interfaces enable the users to specify travel requirements for a tour, such as accommodation, its star rating, transportation, and other tour booking preferences. Other interfaces included in the GUIL enable vendors to enter their offerings and services, such as hotels advertising their products. Users can view the itinerary produced by the system, based on the travel plans organised using the AuSTO tour planner. Further details of these interfaces are included in Chapter 3.
Chapter 3

USER INTERFACE DESIGN

User interface design is the design of the graphical elements on the computer screen, with which a user interacts to conduct application tasks. Good user interface design lowers training costs, lessens user stress, improves consistency in application usage, increases ability to recover from errors, and makes it easier to use the software, (Miller n.d.). Task analysis and modelling techniques are increasingly being used in designing user interfaces, they form an important part of user interface design process and help design more intuitive interfaces.

Task Analysis and Task Modelling

Task analysis is the process of analysing the way a task is performed. One of the most important applications of task analysis is designing user interfaces, where menus can be designed based on the task trees; such that top level menus can be labelled after the top level decomposition and the sub menus after the next level. After an informal task analysis, where the main tasks and their attributes have been identified, task modelling is used to understand the relationships among the various tasks in order to better address the design of interactive applications.

Task modelling is a technique used to model the behaviour of a system from users’ perspective, thus enabling the requirements and actions of the system to be defined as a set of tasks, and the behaviour of the system can be modelled as a scenario of tasks. This allows the designers to improve the human computer interaction aspects when designing a systems operation (Georgievski et al, 2003).

Task models can be classified under two categories:

- user task models
- system task models.

A user task model states the problems to be solved by a candidate system design, and thus consists of overlapping user scenarios. Actors involved in a user task model are generally humans; however, it may include external systems and the environment as some of the actors. A system task model forms the basis for specifying a solution in the form of system requirements. Actors involved in a system task model are generally subsystems, interfaces and humans (Georgievski et al, 2003).

Concur Task Tress Environment (CTTE)

The Concur Task Tree Environment (CTTE) is a Java Applet based tool developed by Human Computer Interaction Group—ISTI (Pisa). CTTE provides the ability to build task models from a visual perspective.

The main features of the CTTE tool are its focus on activities, hierarchical structure, graphical syntax, concurrent notation, task allocation, and objects (Giulio et al, 2002). Further details of the CTTE features can be found in Appendix A.

CTTE allows building of two types of task models:

- single user task models
- cooperative task models
Single user task models are used to represent systems where a single user controls the system. A cooperative task model is, in the main, similar to a single user task model; however, it includes some tasks executed by two or more users.

Other useful features of CTTE tool are:
- model comparison
- reachability analysis
- interactive task model simulator.

**Task Model for Itinerary Planner**

The task model for itinerary planner has been implemented using Concur Task Trees Environment (CTTE), this is a single user task model that represents the overall function of the user interface from a user perspective. We represent these tasks as tree diagrams in Figures 3 to 9. In the task tree diagram we define the execution sequence for each task using the temporal operators provided by CTTE. Details regarding the temporal operators can be found in Appendix A.

![Figure 3: Itinerary planner abstract model](image)

Figure 3 shows the *itinerary planner abstract task model*. It illustrates the tasks the user can perform on connecting to the itinerary planner website.
INTELLIGENT VISUAL TRAVEL RECOMMENDER SYSTEMS MODEL FOR E-TOURISM WEBSITES

Figure 4: Login activities

Figure 4 shows the task model that expands the *login activities* task. Login activities describe the tasks to be performed for the user to login, it allows for an existing user, as well as new user to login to the system.

Figure 5: Data maintenance activities

Figure 5 shows the *data maintenance activities* task tree; it provides a choice between four abstract tasks, namely:
- preference maintenance
- requirement maintenance
- itinerary maintenance
- offering maintenance.
Figure 6: Preference maintenance activities

Figure 6 shows the preference maintenance activities task tree; it consists of the tasks involved in maintaining the preferences of the traveller. Here preferences refer to what of the travellers likes, with regard to various facilities such as accommodation, transport, etc.

Figure 7: Requirement maintenance activities

Figure 7 shows the requirement maintenance activities task tree; it describes the tasks involved in maintaining the traveller requirements. Requirements refer to the traveller’s demands for the tour; for example, the traveller may want five-star accommodation and business class flight.
Figure 8: Itinerary maintenance activities

Figure 9: Offering Maintenance Activities

Figure 9 shows the offering maintenance activities task tree; it describes the tasks involved in maintaining the offerings being provided by various travel vendors. Tourism vendors can advertise their services such as room availability or tickets availability through this option.
User Interface Design

This section presents the user interface design for the itinerary planner; the final graphical user interface will be developed based on the prototype user interface presented here. The user interface for the itinerary planner has been developed based on the task model described in the previous section. This application is explained in further detail with the help of screen shots in the following.

Figure 10: New user screen

Figure 10 shows the new user screen, this allows the creation of a new user, as it is mandatory to have an account in order to use the application.

Figure 11: Login screen

Figure 11 shows the user login screen where an existing user can login to use the application. Only after the user logs in, the application provides access to such performs actions such as storing preferences or specifying requirements etc.
Figure 12: Requirements screen

Figure 12 shows the requirements screen where a tourist can enter the requirements for a trip. Once the requirements have been entered, the tourist clicks the Add Trip button. A trip can have many legs, therefore, the user can add a new leg with the help of the Add Leg button.

Figure 13: Itinerary screen

Figure 13 shows the itinerary screen, the tourists’ requirements are matched with the vendor offerings and an itinerary is produced. The tourist can accept offerings in the itinerary with the help of the checkboxes, and make booking by clicking on the Make Bookings button, and confirming bookings with the Confirm All Bookings button.
Figure 14: Preferences screen

Figure 14 shows the preferences screen, it allows the user to store the preferences about accommodation, transport facilities and other such services. Additionally, preferences for different kinds of trips can be stored, such as family trip, business trip etc.

Figure 15: Offerings maintenance screen

Figure 15 shows the offerings maintenance screen. Tourism vendors such as hotels and transport providers can advertise their offerings via this screen. For example hotels can advertise their room availability and transport providers can advertise their vehicle availability etc.
Chapter 4

INTELLIGENCE LAYER

What is the Semantic Web
The Semantic Web (Berners-Lee 2001) is a methodology and a set of tools that would enable computers to use the World Wide Web as easily as humans. The Semantic Web would enable computers to search intelligently and swiftly for items of interest, and operate websites to access services. For the past twelve years many research bodies have contributed to the ongoing development of the Semantic Web.

The Semantic Web is now changing how information is accessed and processed on the Internet. The Semantic Web needs to include extra information, called metadata, about web-based resources; then a computer would know what that information means in the given context, rather than just know how to present it on the computer screen. Without any metadata, computers view words just as a list of characters to be displayed in the specified format, and pictures as a sequence of bits that are rendered on the computer screen. A human user must be involved in interpreting the words and the pictures.

The Semantic Web uses various tools to mark-up information, to categorise it, and then access it with rule-based reasoning, thus building the ability to interpret textual information and with some less veracity, pictorial information.

How can the Semantic Web Help?
The Semantic Web technologies facilitate the building of intelligent and flexible web sites by using standards and proprietary-free tools. Some of these tools facilitate marking-up of information so that website resources can be given contextual meaning. For example, on one website the word ‘price’ may refer to the price of an item being sold. On another website the word ‘cost’ may be used to refer to the same thing i.e. the price of the item being sold. To website developers and eCommerce customers these mean the same thing. But if each of these websites has a special file with marked up resources, connecting these two terms, then it would be easy for the computer to know that “price” and “cost” are tags that refer to the same concept.

Software tools are now available which allow knowledge to be categorised and stored hierarchically in a specialised database called Ontology. Ontology is a hierarchically organised dictionary/thesaurus, coded in Extensible Markup Language (XML) syntax.

Knowledge can be stored as a combination of Ontology and logic-based rules that operate upon relevant data. The Semantic Web supports reasoning programs, which extract and then apply rules to the data stored in the Ontology. This process can derive new information, even if it is not explicitly stated, but implied; this new inferred knowledge can then be added to the data stored in the database.

For example, say there is a list of tourism resorts in an e-Tourism database, and each resort includes a list of activities available to their guests. A particular tourist wants to go to a resort that has activities that appeal to the sense of adventure. Each resort’s list of activities may not explicitly state that the activities would appeal to tourists with a sense of adventure. But if there are rules in the e-Tourism ontology that say if a tourist resort has ‘kayaking’ as an activity then it can be categorised as an adventure resort. The reasoning program would then include this resort when it searches for adventure tourism activities.

Limitations of Semantic Web for Tourism Websites
There are currently some limitations when applying Semantic Web techniques in developing e-Tourism
websites. These are not limitations of the Semantic Web technologies themselves, but are due to the environment in which e-Tourism Websites operate. One limitation is the fact that accommodation data is not always up-to-date unless (Hepp 2006). This requires manual intervention to confirm reservations.

On-demand availability accommodation information is therefore a business issue that needs to be tackled. While a huge amount of information is available on the Internet; most businesses don’t expose their inventories to public scrutiny and therefore to automated systems. The lack of availability of certain business data may be solved with some new security technology. This technology could act as a middleman, so that certain sensitive data would be accessible to automated systems. This issue needs to be solved; if large-scale automated tourism bookings are to become commonplace. This is a key factor that will limit the uptake of Semantic Web technologies for e-Tourism.

Another limitation is the lack of RDF (Resource Description Framework) mark-up on the various web sites (McCool 2006). An RDF description of a website is a file that the various Semantic Web tools expect to be available, for each website. There has been some research conducted (Abrahams 2005) on tools for automatically marking-up of websites; as this technology becomes readily available it will reduce the manual input required to mark-up current e-Tourism websites.

Implementation Overview

The e-Tourism system can be partitioned into two main components: a Browser Client that provide the user interface, and a Server-side System that uses Semantic Web software components, as shown in figure 16. The user interface consists of forms presented for user input, as shown in chapter 3.

Figure 16: Semantic Web application model for AuSTO

Semantic Web Application Model for AuSTO

The AuSTO system is developed using the Semantic Web application model shown in Figure 16. This system model employs the commonly used client-server paradigm. The user interfaces that reside on the client-side are developed as Rich Interface Applications, using the Ajax Engine, a web application development environment (Garrett 2005).
The Server-side System comprises the following three major components of the Semantic Web:

- **Apache Web-Server**—The Apache Web-Server provides web services to host the website. A web server constantly listens for http requests, such as `http://www.SomeTourismSite.com.au/8080/`, and responds with requested service components. This web server uses a custom Java Servlet\(^1\) to receive the options entered by the user. This Servlet also returns and displays the query results to the user.

- **Jena Middleware**—The Jena Middleware is responsible for managing the reasoning system, the queries from the custom Servlet, and communicating with the Relational Database Server.

- **Relational Database**—The Relational Database constitutes the knowledge base that includes the tourism information, based on the ontology. The application hosting the database services is called MySQL; retrieving information from this database is performed using Simple Protocol and Request Query Language (SPARQL) queries.

The Browser Client comprises of the Ajax Engine and the user interface. The Browser Client enables users to interact with the system using intuitive web interfaces that are developed as Rich Interface Applications using Ajax. These interfaces enable users to enter queries related to their tourism interest; these queries are then converted into JavaScript and sent to the Ajax Engine. The Ajax Engine sends a HTTP request to the Apache Web Server. The Web Server passes this request to the Jena Middleware. Jena Middleware passes and executes a SPARQL query, to retrieve the required information from the database. The results are then sent back to the Browser Client in XML form. The information received by the Browser Client is interpreted by the Ajax Engine and presented to the user in a readable format. These components, as used in the Semantic Web Application Model, are further described in the following.

**Ajax (Asynchronous JavaScript and XML)**

The Ajax Engine comprises a combination of technologies that allow the development of interactive web applications; also known as Rich Interface Applications (RIA). One of the shortcomings of the current (standard) web applications is that each time the user places a request, a HTTP request is sent to the server, and when the server responds the entire page is reloaded. Ajax overcomes this by adding an Ajax Engine between the user interface and the server.

The Ajax Engine sends to the server only the essential parts of the data rather than the entire page; this eliminates the necessity of reloading the entire page each time the server responds to a request. Ajax enhances interactivity, usability, and performance of web interfaces; therefore users can operate with the web application as though it was installed locally—on their own computer. This leads to a richer user experience, and thus the name: Rich Internet Applications. Ajax includes a number of different technologies, including:

- XHTML and CSS for marking up information
- JavaScript to display information dynamically
- XMLHttpRequest object to exchange data asynchronously with the web server (Garrett, 2005).

**Web and XML Server**

The Web and XML server is used to receive, interpret and generate static and dynamic XML and HTTP documents. The XML documents generated for the AuSTO system serve the purpose of data interchange between the user interface applications and the server-side applications that form the AuSTO semantic web. Queries entered by the client are retrieved by the Web Server and then forwarded to the Jena Middleware, which in turn extracts the knowledge from the Relational Database. The implementation for the AuSTO system includes using the Apache Web Server, which hosts the services for exchange of XML and HTML documents. Further details for the Web and XML Server are given in section A2 of Appendix A.

\(^1\) Servlets are Java server applications that respond to requests from web browsers.
Jena Middleware

Middleware is the software that connects the reasoner, the ontology and the user interface. Custom made middleware can be created for an application but it is much easier to use existing middleware software such as Jena (Jena: A semantic Web Framework for Java n.d.).

All operations in Jena are completed by manipulating the Jena model. To manipulate an ontology, first the ontology needs to be loaded into the Jena model. Once loaded, it becomes possible to extract knowledge from this ontology and update the knowledge base as and when possible.

Jena has four subsystems:
- query engine
- database interface
- reasoning engine
- ontology management system.

Jena’s architecture allows external reasoners to be plugged in into the Jena models. The user queries are sent via a user interface to the Jena model, Java code converts these queries into SPARQL format, which can be read by Jena; and the result is sent back to the user.

Figure 17 gives an example of a SPARQL query used in the AuSTO ontology. This query retrieves Business Name and URL of all apartment-holiday units in St Kilda with five-star rating that have swimming pool and air-conditioning facilities. In the operational system, information retrieved by this query is converted into XML format and then forwarded to the browser client, which then displays the information in readable form to the user via the user interface.

```
PREFIX Q: http://www.owl-ontologies.com/Accommodation.owl#
SELECT? BusinessName? URL
WHERE {?Accommodation  Q:hasCategory  Q:Apartment_HolidayUnit .
    ?Accommodation  Q:hasDestination  Q:StKilda .
    ?Accommodation  Q:hasFacility  Q:Airconditioning .
    ?Accommodation  :hasBusinessName ?BusinessName .
    ?Accommodation  :hasURL ?URL}
```

Figure 17: SPARQL query example

Relational Database Server

The AuSTO ontology is stored in a Relational Database Server, which becomes the knowledge base in the tourism domain. A relational database is a database that conforms to the relational model, i.e. a data model based on predicate logic and set theory (Bickel n.d.). The software used to create a relational database is called the Relational Database Management System (RDBMS). The development of the AuSTO system includes the use of MySQL—an open source RDBMS that relies on processing data in the database. MySQL is commonly used in web applications. MySQL provides support to Jena libraries, thus allowing the Jena Middleware to manipulate the ontology stored in the Relational Database Server. The Apache Server is used to host the Web Services and the XML server, as well as the MySQL server.
Chapter 5

DEMONSTRATING THE POTENTIAL OF TRAVEL RECOMMENDER SYSTEMS

Recommender Systems

As more people buy tourism products and services over the Internet, the concept of Recommender Systems has become increasingly popular; Recommender Systems are being applied in different areas and are becoming more prevalent with the development in e-Commerce. These technologies are being improved constantly to keep up with the growing demand for shoppers who purchase products and services over the Internet. Most businesses are adopting Internet as a medium to showcase their products and services to increase sales. Adding a recommendation service to help the user make a good decision is an added advantage.

Recommender Systems can be classified based on the following four recommendation methodologies.

- **Content-based Recommendation Systems** consider the users’ desires, requirements and constraints. After receiving the required data, the system matches the user profile to the product description using information retrieval techniques. This system accepts characteristics and product ratings from users, based on which it tries to understand the user preferences.

- **Collaborative-filtering Recommendation Systems** are the most widely used recommender systems where feedback, reviews, and ratings given by other users are relied upon to recommend an item (Hill, Stead, Rosenstein & Furnas 1995). This system suits best for items that have a large volume of ratings associated with them.

- **Knowledge-based Recommendation Systems** make recommendations based on the knowledge of products on offer and the knowledge about the user. These systems require extensive knowledge about different purchasing habits of users, and some knowledge of the item on offer. Queries are generally used to help obtain this knowledge about the users and their buying preferences.

- **Hybrid Systems** combine different recommendation methodologies. The combination of one or more recommendation methodologies helps in overcoming the limitation of individual methodologies. In most cases collaborative-filtering is combined with another recommendation methodology.

Travel Recommender Systems

The Internet has been very helpful in advancing the tourism industry. Services are being offered on the Internet that allows users to plan their entire trip. Travel Recommender Systems (TRSs) have been developed to recommend different travel destinations to the user. Two of the most popular recommender systems currently available are the Triplehop’s TripMatcher™ and VacationCoach Me-Print™.

Users can plan their entire trip in a more interactive and interesting environment using the facilities built into a TRS. The aim of a TRS is also to reduce the time spent on the entire planning process. The TRS will usually display a range of products based on user query.
Each query and recommendation made can also be used to help the recommender system to learn and identify a better set of products for future recommendations.

Even though current TRSs aim to assist the user in making informed decisions about the trip they have certain limitations. For example, current Travel Recommender Systems fail to perform certain important functions such as generating a complete travel itinerary and are unable to provide a clear picture of what the user is going to experience on the trip. In this project we aimed to develop an Intelligent Visual Travel Recommender System (IV-TRS) to overcome the above mentioned shortcomings of current TRSs.

Tourism Recommendation using Image Based Planning (TRIP)
Travel Recommender Systems available today have certain limitations; they help with destination selection but fail to help with many other important functions related to tour planning. A potential traveller often spends most of the time in researching about the various places of interest and different activities that they can enjoy during the trip. Destination selection is a very complex issue and various factors affect the process of making a decision. In order to make this process simpler, travel websites must try and provide visual information rather than just textual information. The Tourism Recommendation using Image-based Planning (TRIP) proposed by Keen & Rawlings was developed to make this possible.

TRIP Overview
A potential traveller has many questions about the trip being planned, such as ‘How will I organise the trip?’, and ‘How will I be able to get the best out of my investment?’ To answer such questions, one needs some quantitative information that relates to emotional satisfaction, which plays a major role in the decision making process. Gathering such emotional information is not easy. TRIP eliminates these hindrances by efficiently presenting all this information directly to the user in visual magazine style.

Tour Recommendation using Image-Based Planning and SCORM (TRIPS):
Currently, a traveller needs to gather information from many different websites to get a feel of what will be experienced on the trip. TRIPS (Tour Recommendation using Image-Based Planning using SCORM) aims to remove this hindrance of gathering information with the help of the Sharable Content Object Reference Model (SCORM). The TRIPS system facilitates the application of SCORM standard to tourism information stored on various web servers. The proposed system retrieves the required multimedia content posted using SCORM standard from different websites and prepares a presentation that can then be viewed by the user.

SCORM
The SCORM standard was proposed by The Department of Defence (DoD) and Advanced Distributed Learning (ADL) laboratories in 1997 to standardise the format of eLearning contents on the Web. This model aims to improve performance and reduce costs for e-Learning. However our aim is to use this standard to post tourism information on the web, so that the TRIPS system can easily collate the same.

CORDRA
Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA) was created to expand the SCORM functionality. Its purpose is to reference and locate SCORM repositories on the Internet; CORDRA does this by searching the Internet for resources, once their location is resolved, CORDRA delivers these contents to a Learning Management System. Figure 18 displays a conceptual model of how CORDRA works.
INTELLIGENT VISUAL TRAVEL RECOMMENDER SYSTEMS MODEL FOR E-TOURISM WEBSITES

Figure 18: Finding and delivering content as SCOs

Figure 18 demonstrates how CORDRA locates, collates and delivers the content stored as SCOs, based on a user query.

**Visual tour using TRIPS**

Sharable Content Objects (SCO's) contain information parcels such as still and moving images, and textual description of destinations. When a destination selection has been made by the holiday shopper, there are different ways in which he/she can spend his time on the vacation.

The user can enter these options via the web portal, specifying preferences—such as hotel class and constraints—such as the length of the trip. This information about the destination, tourist's interests and preferences are contained by the TRIPS system. TRIPS uses a management system similar to Learning Management Systems, to collects different SCO's that contain the required information. The Learning Management System then creates an activity tree of the user’s itinerary and all the collected SCOs are placed in an XML file called IMSmanifest (imsmanifest.xml). This IMSmanifest file is then delivered to the web portal.

A visual presentation can be created from the IMSmanifest file, which is contained inside a Package Interchange File (PIF). This PIF file includes a structured hierarchy of all the different activities and resources available for the user at the destination. The contents stored as multimedia files in the PIF file can be then delivered to the user’s display in the desired sequence. One option considered in this project was to convert the multimedia contents of the PIF file in to a Flash movie with the help of a tool called ActiveSWF (Activeswf, 2005).

Figure 19 shows the steps required to convert the IMSmanifest.xml file to a Flash movie using the ActiveSWF tool. A special purpose SWF translator was written to convert the IMSmanifest file into the SWFmanifest format. The ActiveSWF tool then created the Flash movie from the SWFmanifest file.
The ActiveSWF is proprietary tool, and requires payment of royalties. While the conversion of the IMSmanifest.xml file into a Flash movie was explored using a trial version of the ActiveSWF tool, it was concluded that this approach is not the most desirable option for implementation, due to proprietary nature of the software. Also, the Flash movie requires a Flash plug-in before it can be viewed. Whereas, other options can produce the visual presentation in a format the will play in any web browser.

The other approach investigated for converting the IMSmanifest.xml file into a movie clip was the Synchronized Multimedia Integration Language (SMIL) standard.

Figure 20 shows the process used in converting the PIF file (which contains the IMSmanifest file) into SMIL standard based vitalisations. In may ways this process works similar to the process used for creating the Flash file from the PIF file.

Required multimedia content is collated from different websites and packed into a PIF file using the XML standard. A SMIL translator program was created for this project that converted the PIF file into a SMIL file. As SMIL is an international standard, SMIL files can play in most modern web browsers, without downloading a plug-in; where as a Flash plug-in is required to play any Flash file.

The other advantage of the SMIL standard is that it allows the same presentation to be played out in different formats. Thus, the SMIL file produced by the SMIL Translator can be repurposed for specific user devise.

A SMIL customisation program interacts with the user to get parameters such as the length of payout and resolution. Prior to displaying the SMIL file to the user, this program checks for various system and device parameters such as the connection speed, bandwidth, screen resolution, and the type of device. The final play out SMIL file is then customised to suit these parameters, thereby creating a better presentation.
**TRIPS Architecture**

The TRIP architecture can be enhanced by using the SCORM standard; creating the TRIP using SCORM (TRIPS) model shown in figure 21. The advantage of using the SCORM standard for storing web content using the SCORM standard is that is can then be searched more effectively using SCORM’s companion standard CORDRA as explained earlier.

The TRIPS architecture will allow information to be store as reusable packages. These packages can then be accessed by local SCORM repositories or through the CORDRA enhancement. This enables information to be easily accessible, maintainable, and adaptable.

![TRIPS Architecture Diagram](image)

**Figure 21: Tour Recommendation using Image-based Planning using SCORM (TRIPS)**

In the TRIPS architecture the Dynamic Tourism Information Repository is the central storage device whose primary function is to store all the Sharable Content Objects (SCOs) that will be used for tour visualisation. Information within this repository can be accessed via a console or a portal. The user enters a query using the V-TRS portal. An LMS is used to interpret this query. The LMS enters a request to the Dynamic Tourism Information Repository to access information required to respond to the user query. The CORDRA system can then search for the relevant SCO’s on the Web or on specialised tourism information repositories such as the ATDW (Australian Tourism Data Warehouse). The tourist will interact with the V-TRS portal, while a system Administrator can access the Repository via a Repository Access portal. Content producers can mount content on to the web as SCOs or enter it into specialised repository like ATDW.
Chapter 6

CONCLUSION & FURTHER RESEARCH

There is a need for developing better e-Tourism websites which are easy to use and provide more efficient access to the information and services that users require. This report presented the outcomes of two related projects that investigated the benefits of the Semantic Web technologies and their application in developing Intelligent Visual Travel Recommender System.

It was found that the Semantic Web tools and technologies would be useful in developing the next generation of e-Tourism websites. Some of the benefits would be more flexible searching of appropriate websites, less precise input from users in requesting what they want and a standard way of packaging intelligence for a suite of e-Tourism applications.

Some limitations in applying the Semantic Web are due to the nature of the Internet as it stands today. For example how much automation is it safe to apply on the Internet, given that security remains a problem? This report presented an overview of the Semantic Web and described in detail a framework for developing e-Tourism applications based on the technologies of the Semantic Web.

Recommender systems are built to simplify the task of destination selection. They help the user to make a better choice from the options available. Recommender systems must be designed so that they can handle the Human-Computer Interaction (HCI) well and make the task of tour planning less cumbersome.

This report has also presented an overview of Travel Recommender Systems (TRSs) their classification and application in the present incarnation of the Internet. We provide recommendations as to what future recommender systems will be able to do and what other work needs to be done to enhance develop an Intelligent Visual Travel Recommender System (IV-TRS). The IV-TRS will enhance current recommender systems into more visual and easy to use systems that work faster, more efficiently and support more sophisticated functions. These enhancements will increase the confidence levels of the user in their tour planning, leading to enhanced tourism business.

Current breed of TRSs can provide recommendations for destination selection; however they fail to provide a good visualisation of the entire tour. In this project we have shown how continuous visualisation systems can be developed using current content packaging standards such as SCORM & CORDRA and how this content can be combined as a Flash or SMIL presentation.

Future research will aim to conduct usability study on the effectiveness of continuous visualisation systems, and develop recommendations for the deployment of the proposed Intelligent Visual Travel Recommender System (IV-TRS), using the following steps:

- Extend AuSTO ontology with more data and rules.
- Extend user interface and task modelling framework.
- Conduct usability tests on the user interfaces.
- Develop a Demonstration e-Tourism website for providing Intelligent Visual Travel Recommendations.
APPENDIX A: Peer Reviewed Publications (relate to this project)

List of research papers and book chapters published based on the research conducted in these projects.


APPENDIX B: AuSTO Implementation Source Code Examples

This appendix shows example of the source code used in the system. Figure C1 is a screenshot of two restriction blocks in the AuSTO ontology; it shows how OWL syntax based ontology stores logic rules.

Figure C1: Example of restriction in AuSTO Ontology

Figure C2 shows a SPARQL query. It requests Business Name and website address of all apartment-holiday units in Lorne with four-star rating and having swimming pool and air-conditioning.

Figure C2: Example of SPARQL query
REFERENCES


## GLOSSARY

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADL</td>
<td>Advanced Distributed Learning</td>
</tr>
<tr>
<td>Ajax</td>
<td>Asynchronous JavaScript and XML</td>
</tr>
<tr>
<td>ATDW</td>
<td>Australian Tourism Data Warehouse</td>
</tr>
<tr>
<td>AuSTO</td>
<td>Australian Sustainable Tourism Ontology</td>
</tr>
<tr>
<td>BLL</td>
<td>Business Logic Layer</td>
</tr>
<tr>
<td>CL</td>
<td>Custom Logic</td>
</tr>
<tr>
<td>CORDRA</td>
<td>Content Object Repository Discovery and Registration/Resolution Architecture</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>CTTE</td>
<td>Concurrent Task Trees Environment</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defence</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>GUIL</td>
<td>Graphical User Interface Layer</td>
</tr>
<tr>
<td>HCI</td>
<td>Human-Computer Interaction</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>ICT</td>
<td>Information Communication Technologies</td>
</tr>
<tr>
<td>IE</td>
<td>Inference Engine</td>
</tr>
<tr>
<td>IV-TRS</td>
<td>Intelligent Visual Travel Recommender Systems</td>
</tr>
<tr>
<td>Jena</td>
<td>Java framework for building Semantic Web applications</td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management System</td>
</tr>
<tr>
<td>NIAATI</td>
<td>National Information Architecture for the Australian Tourism Industry</td>
</tr>
<tr>
<td>OL</td>
<td>Ontology Layer</td>
</tr>
<tr>
<td>OWL</td>
<td>Ontology Web Language an XML type language for describing ontologies</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>PIF</td>
<td>Package Interchange File</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>RDF</td>
<td>An XML type language for describing Web site resources unambiguously</td>
</tr>
<tr>
<td>RIA</td>
<td>Rich Interface Applications</td>
</tr>
<tr>
<td>SCO</td>
<td>Sharable Content Object</td>
</tr>
<tr>
<td>SCORM</td>
<td>Sharable Content Object Reference Model</td>
</tr>
<tr>
<td>Semantic Web</td>
<td>A methodology and tools to process of Web-based resources by machines</td>
</tr>
<tr>
<td>Servlet</td>
<td>A server computer based Java program that responds to web based http requests</td>
</tr>
<tr>
<td>SMIL</td>
<td>Synchronised Multimedia Integration Language</td>
</tr>
<tr>
<td>SPARQL</td>
<td>A W3C standard SQL-like query language for retrieving data from ontologies</td>
</tr>
<tr>
<td>SPARQL</td>
<td>Simple Protocol and Request Query Language</td>
</tr>
<tr>
<td>ST-CRC</td>
<td>Sustainable Tourism Cooperative Research Centre</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Tomcat</td>
<td>A server computer based Java program that manages Servlets.</td>
</tr>
<tr>
<td>TRIP</td>
<td>Tourism Recommendation using Image Based Planning</td>
</tr>
<tr>
<td>TRIPS</td>
<td>Tour Recommendation using Image-Based Planning and SCORM</td>
</tr>
<tr>
<td>TRS</td>
<td>Travel Recommender Systems</td>
</tr>
<tr>
<td>V-TRS</td>
<td>Visual TRS</td>
</tr>
<tr>
<td>XHTML</td>
<td>Extensible HyperText Markup Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
AUTHORS

**Associate Professor Nalin Sharda**
Nalin Sharda gained B. Tech. and PhD. degrees from the Indian Institute of Technology, Delhi. Presently he teaches and leads research in multimedia and Internet communications at the School of Computer Science and Mathematics, Victoria University, Australia. Nalin’s publications include the Multimedia Information Networking textbook and around 100 papers and handbook chapters. Nalin has invented Movement Oriented Design (MOD) paradigm for the creation of effective multimedia content based experience and applied it to e-Learning and e-Tourism. Nalin has led various projects for the Australian Sustainable Tourism Cooperative Research Centre, to develop e-Tourism systems using Semantic Web technologies and innovative visualisation methodologies. Nalin has been invited to present lectures and seminars in the Distinguished Lecturer series of the European Union’s Prolearn program. He has presented over fifty seminars, lectures, and key note addresses in Austria, Australia, Finland, Germany, Hong Kong, India, Malaysia, Pakistan, Japan, Singapore, Sweden, Switzerland, UAE, and USA.
Email: Nalin.Sharda@vu.edu.au

**Dr. Mladen Georgievski**
Mladen Georgievski completed his PhD. from Victoria University in 2007; prior to that he completed a Bachelor of Science in Computer Science (honours) in 2001 and Bachelor of Science in Computer Technology in 1999. His research interests are in the field of networked multimedia systems, usability studies, and quality of service management for multimedia communications. Mladen has worked as a tutor at Victoria University and as a research assistant for e-Tourism projects funded by the Sustainable Tourism Cooperative Research Centre. He is a member of the Institution of Electrical and Electronics Engineering (IEEE) and the Australian Computer Society (ACS).
Email: mladeng@csm.vu.edu.au

**Roopa Jakkilinki**
Roopa Jakkilinki is a PhD. research scholar at Victoria University, Melbourne, Australia. She gained a Master of Science in Computer Science in 2004 and a Graduate Diploma in Computer Science in 2002, both from Victoria University; prior to that she completed a Postgraduate Diploma in Information Technology from MAHE, India, in 1999 and a Bachelor of Arts in Computer Technology from Andhra University, India. Roopa has been employed as a tutor at Victoria University and as a research assistant for e-Tourism projects funded by the Australian Sustainable Tourism Cooperative Research Centre. Her research interests include Semantic Web systems for e-Tourism.
Email: roopa@csm.vu.edu.au

**Mohan Ponnada**
Mohan Ponnada is completing his Bachelor in Computer Science at Victoria University. He has worked as a research assistant for e-Tourism projects funded by the Sustainable Tourism Cooperative Research Centre and as a member of the system support team for Australian Unity. His research interests include visualisation systems for e-Tourism.
Email: mohan.ponnada@students.vu.edu.au
EC3, a wholly-owned commercialisation company, takes the outcomes from the relevant STCRC research; develops them for market; and delivers them to industry as products and services. EC3 delivers significant benefits to the STCRC through the provision of a wide range of business services both nationally and internationally.
The Sustainable Tourism Cooperative Research Centre (STCRC) is established under the Australian Government’s Cooperative Research Centres Program. STCRC is the world’s leading scientific institution delivering research to support the sustainability of travel and tourism – one of the world’s largest and fastest growing industries.

The program emphasises collaboration between businesses and researchers to maximise the benefits of research through utilisation, commercialisation and technology transfer. An education component focuses on producing graduates with skills relevant to industry needs.

**STCRC’s objectives are to enhance:**
- the contribution of long-term scientific and technological research and innovation to Australia’s sustainable economic and social development;
- the transfer of research outputs into outcomes of economic, environmental or social benefit to Australia;
- the value of graduate researchers to Australia;
- collaboration among researchers, between researchers and industry or other users; and efficiency in the use of intellectual and other research outcomes.

**Introduction**
The STCRC has grown to be the largest, dedicated tourism research organisation in the world, with $187 million invested in tourism research programs, commercialisation and education since 1997.

The STCRC was established in July 2003 under the Commonwealth Government’s CRC program and is an extension of the previous Tourism CRC, which operated from 1997 to 2003.

**Role and responsibilities**
The Commonwealth CRC program aims to turn research outcomes into successful new products, services and technologies. This enables Australian industries to be more efficient, productive and competitive.