# A Conceptual Information Model for a 'Green Economy Tourism System' (GETS)

*Research-in-Progress*

**G. Michael McGrath**
Victoria University, Melbourne, Australia
michael.mcgrath@vu.edu.au

**Paul A. Whitelaw**
Victoria University, Melbourne, Australia
paul.whitelaw@vu.edu.au

**Geoffrey H. Lipman**
Greenearth.travel, Brussels, Belgium
glipman@gmail.com

**Henk Meijerink**
Victoria University, Melbourne, Australia
hmeijeri@telstra.com

**Alexandra Law**
Victoria University, Melbourne, Australia
alexandra.law@live.vu.edu.au

**Terry DeLacy**
Victoria University, Melbourne, Australia
terry.delacy@vu.edu.au

## ABSTRACT

Many tourism destinations are currently pursuing green growth strategies but the development of appropriate policies is a complex task and, consequently, decision support technologies can be used to advantage here. The design and use of one such decision support system (DSS) is described in this paper. Key features of the system are that its design is underpinned both by a need to effectively manage the inherent complexity of the analysis domain and to allow iterative development with minimum impact on previous versions (i.e. to minimize ongoing maintenance costs). A key to realizing both these objectives is the use of a highly-abstracted conceptual information model and this is the major focus of this paper.

## KEYWORDS

Decision support; green-growth tourism; conceptual information model

## INTRODUCTION

The Intergovernmental Panel on Climate Change (IPCC) has declared that "*warming of the climate system is unequivocal*" (IPCC, 2007). As nations worldwide address the risks of climate change and aim to reduce their greenhouse gas (GHG) emissions, we are seeing the emergence of a new low-carbon, 'green' economy. The United Nations Environment Programme (UNEP) defines the green economy as "*an economy that results in improved human-wellbeing and reduced inequalities over the long term, while not exposing future generations to significant environmental risks and ecological scarcities"* (UNEP, 2010). This new green economy trend has wide ranging implications for the tourism sector. Destinations face complex new challenges, but also significant new opportunities, such as those presented by changing tourism demand, that need to be addressed to remain sustainable and competitive. In its Green Economy Initiative (GEI), UNEP identifies tourism as one of 11 priority sectors where investment in sustainable solutions can drive economic recovery and growth while simultaneously addressing social inequalities and environmental challenges (UNEP, 2010).

However, green economy planning in tourism is a complex process, characterized by high levels of uncertainty. For instance, tourism is not included as a sector in traditional emission inventories and as such, little information is avaiable on sources and magnitude of the sector's GHG emissions (Becken and Hay, 2007). Another example of uncertainty for the tourism sector is the emergence of 'green demand', which remains difficult to quantify. While it is anticipated that climate change and environmental perceptions will alter destination choice and consequently influence tourism demand (see e.g. Simpson at al, 2008), there is a lack of information available for destination policymakers and planners to understand the dynamics behind these changes. For targeted mitigation and adaptation strategies, the relationship and interdependencies between the green economy drivers must be understood. However, a planning framework for a green economy transition in tourism destinations does not currently exist. In this context, our 'Green Economy Tourism System' (GETS) is being developed to facilitate the capture, organization and access of required strategic planning data in a systematic and convenient way. In addition, an increasing array of 'add-on' decision support modules is being developed in order to allow destination planners and policy makers to investigate dynamic 'what if' scenarios around their destination and green economy developments.

Given that an iterative approach is required for GETS development and maintenance (because of the need to use the software sequentially in a range of destinations), the system design must allow for convenient modification with minimum impact on previous versions. Furthermore, since each successive application of the system should be able to utilize previously developed functionality and data, a system and information architecture that facilitates this sharing capability is essential. The keys to realizing these objectives within GETS are the use of data abstraction and generalization (Feldman and Miller, 1986) in specifying the conceptual information model and the ISO 3-schema architecture (van Griethuysen, 1982) as the basis for the overall system design. The focus in this paper is on the conceptual information model. The reader desiring detail on actual, field applications of GETS is referred to Law et al. (2012).

The paper is organized as follows: in the following section, background to the development and use of GETS and its system architecture is presented and this is followed by an overview of our research approach. The GETS conceptual information model is then introduced and the following section contains an example of an application developed around that model. The final section contains concluding remarks.

## *GETS:* BACKGROUND AND SYSTEM ARCHITECTURE

### Background: Green Growth Tourism and *GETS*

A detailed introduction to the GETS system and, in particular, the motivation and rationale that underpinned its development is presented in Law et al. (2012). A more technical treatment of the system is contained in McGrath, Law and De Lacy (2012).

*Green Growth* can be defined as coherent strategies to overcome economic free-fall, pervasive climate change, basic resource depletion, rapidly increasing populations and debilitating poverty. *Travelism* – the entire customer, company, and community value chain can play a much more significant role in the transformation to a fairer, happier society, based on renewable energy, web dynamics, social inclusion and biodiversity conservation. Such an approach will also lead to global temperature stabilization by 2050 through green transformation of production, consumption and investment.

In order to translate these ideas to a practical travel and tourism operational level, the Victoria University Centre for Tourism and Services Research has undertaken a number of destination focused evaluations in Africa, Asia, the Pacific and the Caribbean, over the past 5 years, culminating in 2012 in a major Green Growth and travelism-based study of Bali Indonesia. From the resultant "Green Growth 2050 Roadmap" (CTSR, 2012), we highlight the research action to model 'the Bali visitor economy'; the multi-stakeholder visioning and the strategic directions to deliver sustainable mobility, lifestyles and communities.

We show how, in this process, the GETS system gives communities a much more comprehensive and better decision making framework and provides an incentive for evaluating/implementing the green growth and travelism options. It presents an approach for a decision support system (DSS) to assist destinations address challenges and opportunities in periods of rapid change with a core requirement of low carbon transformation. A key design factor is the capacity to support decision making for tourism destinations of varying sizes (small locations within a country, or transnational regions) and varying economic structures (regions exclusively reliant on tourism or where tourism is a major or even a minor industry) and varying composition (be it comprised of large international western style hotels and resorts, or small individual business with a high level of eco-tourism attractions and activities).

### System Architecture

A high-level view of the GETS architecture is illustrated in Figure 1. It is consistent with ISO '3-Schema Architecture' principles (van Griethuysen, 1982). The *Conceptual View* is a highly abstracted model of the total system, completely free of any implementation-level detail. The *Internal View* deals primarily with technical aspects of the various applications (relating to efficiency etc.) and is beyond the scope of this paper. *Application View 1, ----, Application View n* are external-level schemas developed for individual applications, implemented within specific software shells *(Software Shell 1, ----, Software Shell n).* Examples of these (used in applications implemented to date) are *Excel*™, *Access*™, a rule-based expert systems shell called *Flex*™ and the system dynamics simulator, *PowerSim*™. An example of how GETS concepts may be gradually broken down into more-detailed, lower-level concepts follows.

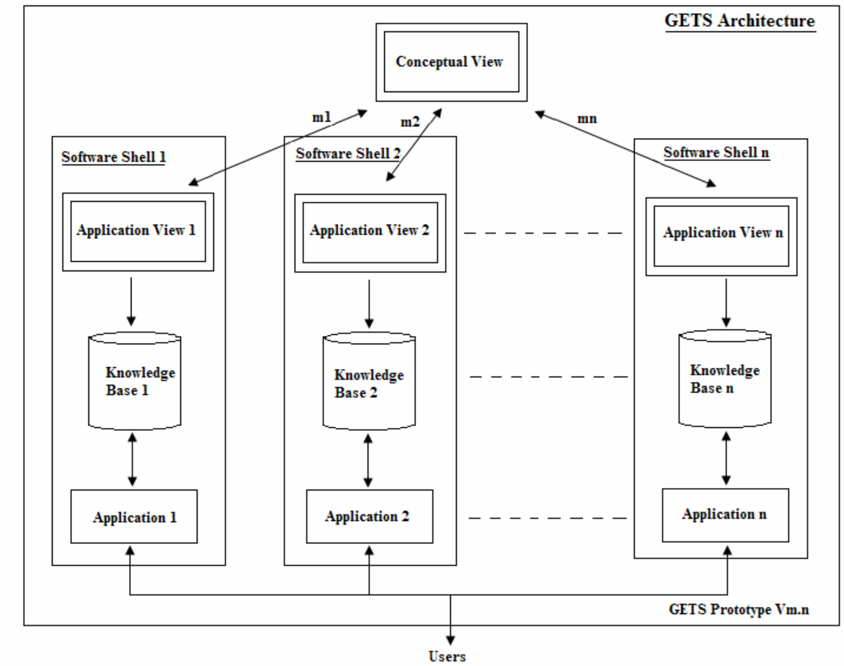


**Figure 1: GETS architecture – high-level view.**

The GETS domain contains a large number of variables, covering the economic, environmental and social dimensions, with variables interacting with each other in a complex myriad of ways; i.e a classic case of a "wicked' or 'messy' problem (Vennix, 1996). *System dynamics (SD)* (Maani and Cavana, 2000) is very well-suited to this task and many (but not all) GETS models have been specified and implemented using SD techniques and software packages.

SD has its origins in the work of Forrester (1961) and, more recently, has enjoyed something of a resurgence – largely due to Peter Senge's (1990) very influential work on ' the learning organization' and the development and release of easy-to-use, powerful, SD-based software modeling and simulation tools (such as *iThink*™*, Vensim*™ and *Powersim*™*).* Recent examples of where SD has been used in tourism include the 'Tourism Futures Simulator' of Walker et al. (1999), the hotel value chain modeling work of Georgantzas (2003) and the tourism modeling work of McGrath and More (2005).

In their simplest form, SD models are represented as 'causal-loop diagrams' (CLDs). The reader looking for a more thorough introduction to CLDs is referred to Manni and Cavana (2000) but, essentially, only one modeling construct is employed; an arrow connecting two domain variables, indicating a causal connection between them. Arrows are generally annotated with either a '+' or '-'; a '+' symbol meaning that both variables move in the same direction (i.e. increase or decrease together) and a '-' symbol meaning that the variables move in opposite directions. We employ a third annotation symbol, the question mark, '?' – meaning that we are unsure of the exact nature of the causal connection or that the connection is too complex to represent with the two basic annotations.

The usual approach in developing a SD model though, is to: i) specify the problem domain as a CLD and, then, ii) implement it in the slightly more complex *stock-flow* syntax employed by the software packages referred to above. In this section we restrict ourselves to CLDs. A very simple example of a causal connection is that, as *economic activity* increases, so will *energy demand.* This is represented as illustrated in Figure 2.

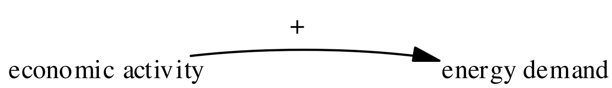


**Figure 2: Example of a simple very high-level (Level 0) CLD.**

In CLDs, decomposition can be based on either variables or the connections (relationships) between them. In this case, there is obviously much more to the activity-demand relationship than the very high-level representation contained in Figure 2, so we may decompose the connection further into the CLD presented in Figure 3.

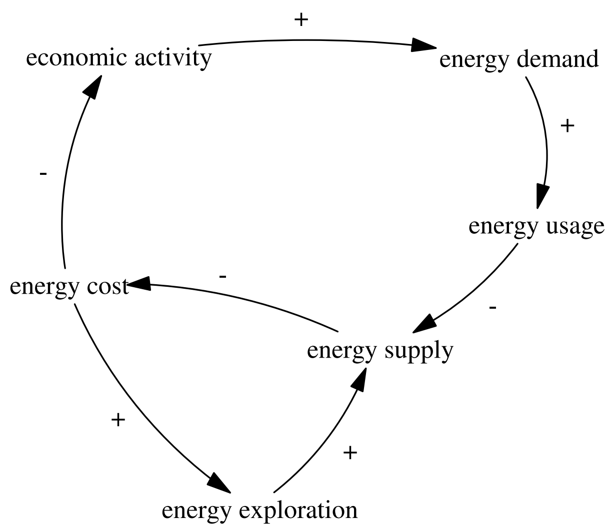


**Figure 3: Example of a 2nd level (Level 1) CLD.**

Here, high *energy demand* leads to greater *energy usage* (and vice versa) and greater usage, in turn, leads to a diminished *energy supply*. If *energy supply* is low though, this will probably lead to an increase in *energy cost* and this may have a consequent negative impact on *economic activity*. This leads us back to our starting point and completes the link to the top-level connection we have decomposed. Note though that there are additional constructs at the bottom of Figure 3: specifically, if the cost of energy is high, companies will be more inclined to involve themselves in increased *energy exploration* and, in turn, one would hope (and probably expect) that this will ultimately increase the *energy supply*.

The CLD in Figure 3 was the result of breaking down a connection into greater detail. Aspects of this model may be decomposed even further. For example, within a green economy context, in terms of energy production a distinction can and needs to be made between traditional *carbon-intensive energy (CIE)* (oil, coal and natural gas) and *renewable energy (RE)* (hydro, wind, biomass, waste etc.) sources. Figure 4 below demonstrates these relationships.

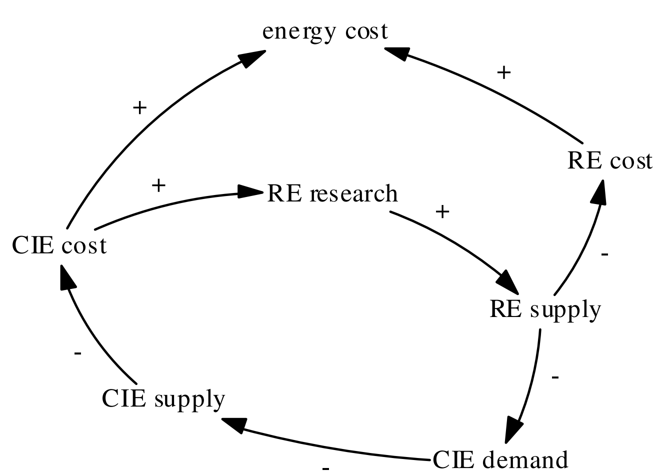


**Figure 4: A 3rd level (Level 2) CLD instance.**

Obviously, the total *energy cost* is dependent on the respective CIE and RE costs. If the *CIE cost* is high though, it is likely that more will be invested in *RE research* and this, in turn, should increase the *RE supply*. If the *RE supply* is high the *CIE demand* may drop, thus placing less pressure on *CIE supply* and, finally, without this supply pressure, the *CIE cost* should be less. This, of course, is the rough basis for the various carbon pricing and taxing schemes being introduced (or considered) in many countries throughout the world.

The models presented above illustrate one of the benefits of SD modeling as claimed by its proponents: specifically, the approach can counter our tendency to over-simplify complex problems and issues into simple cause-effect relationships we can readily understand within the limits of our cognitive powers (Vennix, 1996). Of course, this is true of many conceptual modeling approaches and each of these has their own strengths and weaknesses. SD, however, is particularly well-suited to domains where feedback loops and time are significant and both of these feature prominently in tourism models (see e.g. Ritchie and Crouch, 2003: 60-78).

A further strength of SD models is that, in basic CLD form, they are comprised of combinations of only one, simple construct (a causal connection between two variables), meaning that key stakeholders and end-users may readily contribute to modeling sessions. As noted earlier, CLD models are generally implemented in the stock-flow form favored by the more popular SD software packages. This increases complexity but it also enables the specification of critical concepts such as delays, queues, events and major environmental perturbations (e.g. the impacts of SARS or the periodic, dramatic increase in global oil prices).

## RESEARCH APPROACH: OVERVIEW

The following is a brief summary of the research approach, which is based on the idea that development of an Information System (IS) may, in certain circumstances, be considered a legitimate research strategy in its own right. Hasan (2003: 4) claims that IS development, in many cases, should be considered a valid research activity (and method) because, not only is knowledge created about the development process itself, but also because "a deeper understanding emerges about the organizational problem that the system is designed to solve". Markus et al. (2002) put forward a similar case in arguing that IS development is a particular instance of an *emergent knowledge process (EKP)* and that this constitutes original research where requirements elicitation, design and implementation are original and generate new knowledge on *how to proactively manage data and information in complex situations.* Hasan (2003: 6) further contends that this often involves a staged approach, where "systems evolve through a series of prototypes" with results of each stage informing requirements for the next and subsequent iterations.

Thus, to summarize: the development of our DSS is a legitimate research activity in its own right, which draws on the more established, traditional research approaches of the design sciences and especially case study/action research. Each new application of the DSS (e.g. to a new destination) produces a new version of our prototype and extends our knowledge of the green tourism economy research domain. This is akin to employing a multi-case (study) research strategy - with each new case refining and extending results of previous iterations - and finally, many research findings and outputs are actually inherent in the various conceptual models (and implementations of these) that constitute the DSS.

## THE *GETS* CONCEPTUAL INFORMATION MODEL: A UNIFYING FRAMEWORK

Curtis et al. (1992) have argued that different modeling objectives, user diversity, conflicting requirements, the need to share models (between components) and the need for both large and small-grained levels of abstraction all demand decision support models permitting multi-paradigm representations. A modeling, 3-level framework that meets these requirements is presented in Figure 5. In developing this framework we have drawn on the work of IS0 Technical Committee 97 in defining the foundations of the 3-schema database management system architecture (van Griethuysen, 1982). Levels 0 and 1 (the UoD and conceptual views) are introduced in this section, with Level 2 (external views) discussed in the following section (by way of an example).

The *UoD* refers to *that collection of objects, from a real or postulated world that is* being described - in our case, the world of interest is centered on green growth tourism. The UoD representation at Level 0 was derived from the model described in detail by Law et al. (2012) and developed as part of a green growth tourism strategy, conducted with the Egyptian Government, at Sharm El Sheik. This study highlighted the following four key elements as being integral to a successful green economy transformation: i) GHG emissions reduction; ii) growing destination market demand; iii) enhancing the destination environment and ecosystems; and iv) sustainability of the destination's economy and socio-cultural traditions. The model, as represented at Level 0 in Figure 5, highlights the domain complexity: all elements (and relationships between them) are characterized by the fact that they are very tightly integrated, reflecting that targeted strategies rely on a holistic and systemic view.

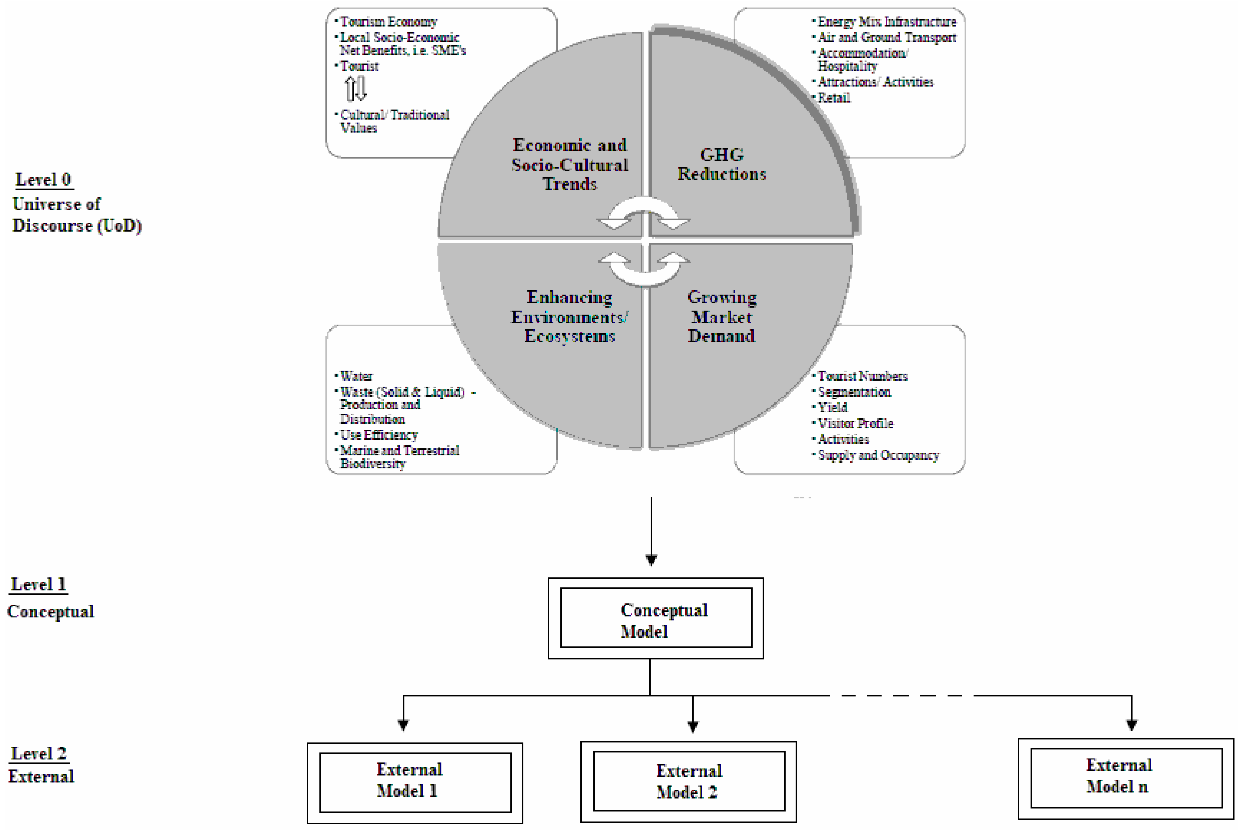


**Figure 5: GETS modeling framework.**

The *conceptual model* defines *the objects of the UoD, including rules governing allowable classifications, states, transitions and constraints* (van Griethuysen, 1982). Thus, the GETS conceptual model is a high-level conceptual representation of real world entities and their inter-relationships within the domain problem space. It is represented in entity-relationship form (Chen, 1976). It is spread over many subject area diagrams, where each diagram is, in effect, a view of part of the underlying overall model. Diagrams are constructed to partition off specific shared subject matters; data subject areas (DSAs) that are shared across a number of functions. An example of a GETS DSA is presented in Figure 6. It builds upon modeling work described in McGrath (2007) and provides a very abstracted view of relevant concepts at the *Destination* level. *GHG (emissions)* is shown at the bottom-right of Figure 6 and this is a super-entity which, itself, explodes into a further DSA. The majority of the variables represented in Figure 4 map back to this DSA.

The conceptual model is a *common denominator* schema, as defined by Curtis et al. (1992): i.e. it is a schema that specifies only the essentials of the vital or core domain elements, leaving aside any external representation considerations. The entity-relationship approach has been employed not because it is unarguably superior to competing formalisms but because: i) it has long been the most popular and best known data modeling approach used within the information systems arena; ii) there is a well-defined abstraction process for entity-relationship models that employs much the same "super" entity types used within most data and process modeling; and iii) the entire rationale for the development of the entity-relationship modeling approach was to provide a unified view of data.

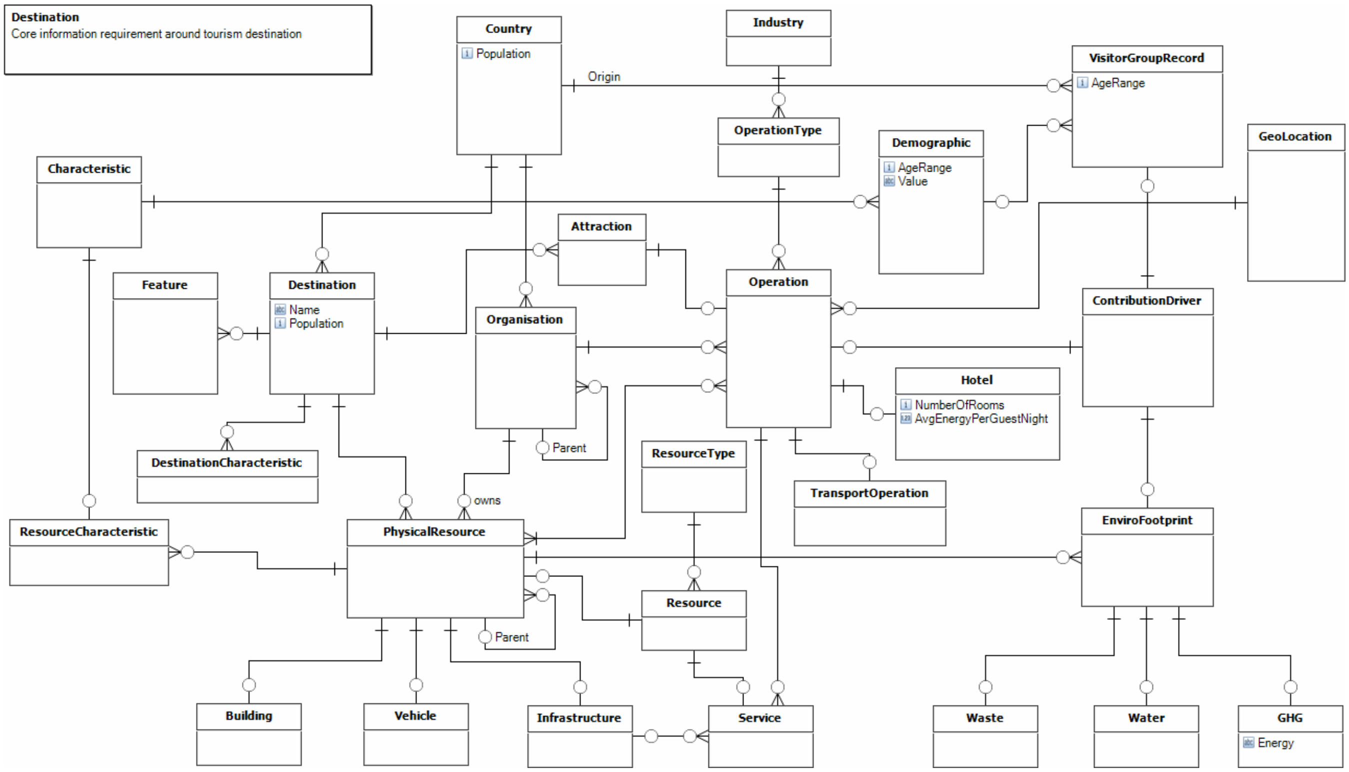


**Figure 6: *Destination* DSA.**

Mappings between DSAs and between external application models and the conceptual model are facilitated by a modelling approach based fundamentally on the abstraction/generalization principles detailed by Feldman and Miller (1986). Extensive use is made of what are, essentially, class hierarchies and generic archetypes such as that presented in Figure 7.

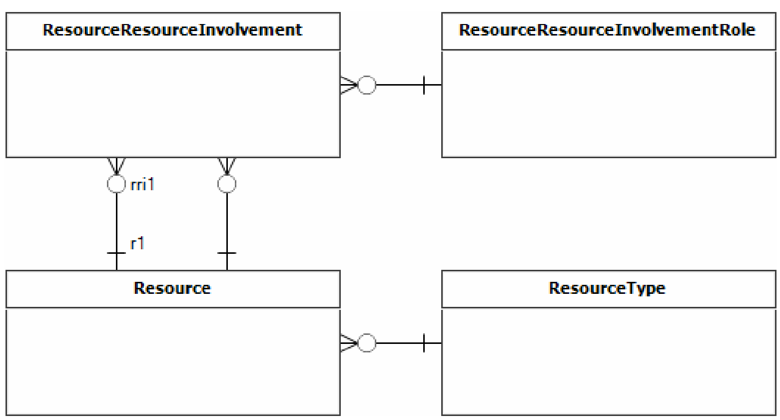


**Figure 7: Modeling archetype: *resource-resource involvement (rri).***

Implemented as a relational database application, the intersecting entity, *rri (resource-resource involvement),* would translate to something like the *Access*™ table presented in Figure 8. This table details some of the important subtype relationships that need to be captured. In this case, *energy* is specified as a *resource* subtype, *energy* may be decomposed further into *airTransportEnergy, accmdnEnergy, attractionEnergy* and *activityEnergy* and these, in turn, may be broken down even further (as illustrated in Figure 8).

rri : Table

| rriId | resType1 | resType2 | rriRole |
|---|---|---|---|
| 22 | resource | energy | subtype |
| 23 | energy | airTransportEnergy | subtype |
| 24 | energy | accmdnEnergy | subtype |
| 25 | energy | attractionEnergy | subtype |
| 26 | energy | activityEnergy | subtype |
| 27 | accmdnEnergy | hotelEnergy | subtype |
| 28 | accmdnEnergy | motelEnergy | subtype |
| 29 | accmdnEnergy | bedAndBEnergy | subtype |
| 30 | hotelEnergy | airConEnergy | subtype |
| 31 | hotelEnergy | lightingEnergy | subtype |
| 32 | hotelEnergy | appliancesEnergy | subtype |
| 33 | hotelEnergy | transportEnergy | subtype |
| (AutoNumber) | | | |

**Figure 8: *Access*™ implementation of *rri* class hierarchy (partial).**

Representing the conceptual model in an abstracted form produces a number of benefits, including: i) where appropriate, common functionality may be coded around the abstracted view, leading to a reduction in system development effort; ii) integration of applications, developed around external views, is facilitated because core data types are all mapped back to the common conceptual view (model); iii) better integration means that functionality may be more conveniently shared between applications (which also means less coding effort); and iv) ongoing system maintenance is reduced (again resulting in a reduction of total development effort).

The easier maintenance benefit is extremely important and, consequently, deserves additional attention. A common requirement in decision support applications of this type is to derive all subtypes (at whatever level removed) of a given super-type. Employing (quasi) *Prolog* (Bratko, 1986) as the programming language, this functionality may be implemented as the following recursive procedure:

*RT_x isaSubtypeOf RT_y* ***if***
*rri(RRIId, RT_y, RT_y, subtype).*

*RT_x isaSubtypeOf RT_y* ***if***
*rri(RRIId, RT_y, RT_z, subtype)* ***and***
*RT_x isaSubtypeOf RT_z.*

Assume now that, in place of the decomposition illustrated in Figure 8, we wish to break energy resources down into carbon-intensive and renewable varieties (and at the next level into coal, gas, oil, hydro, biofuel etc.). With a more concrete conceptual model, this would require much code revision. With the abstract view however, this is not required and, as an example, the hierarchy retrieval procedure above still works perfectly well without any revision. All that is required is replacement of the Figure 8 table entries with a new set.

Finally, a conceptual information model is an ontology covering a problem space. In that sense it is a design, an intention, a schema. It may be concrete or physical in relation to any element existing or required in the problem space. However, it should always be conceptual or abstract in relation to any element in the solution. In addition, a conceptual model is suggestive of solution design only to the very limited extent that an element in the requirements specification might serendipitously map directly to an element in the solution. However, a complete and well-crafted information model will definitely inform a resilient solution architecture and adaptable solution design elements. To be clear, flexible generic design is often informed by the generic elements in a well-crafted information model. We now turn our attention to this issue: specifically GETS applications derived from the conceptual information model.

## EXTERNAL APPLICATION: EXAMPLE

Individual GETS applications (see Figure 1) are developed around external-level models. An *external model* or user *view* is *a mapping from all or part of a conceptual model to a language or representational form of the user's choosing* (van Griethuysen, 1982). In addition, it must be possible to map in the reverse direction: i.e. from external to conceptual model. In the previous section, part of a conceptual model implementation using *Access*™ and the very high-level programming language *Prolog* was presented. GETS contains external-level applications using these software products but additional packages (and associated modeling and coding techniques) are also employed. In particular (and as noted earlier), a number of key system functions are implemented as external components using SD and, specifically, the SD product *Powersim*™. We shall now illustrate the mapping process and provide an example of an external model/application through the leveled set of CLDs presented earlier in Figures 2-4 (specifying the economic-activity – energy-usage causal relationship).

Referring to Figure 4 again, this CLD contains the circular set of causal relationships: *REResearch* → *RESupply* → *CIEDemand* → *CIESupply* → *CIECost* → *REResearch.* At the conceptual level, these are also represented as *rri* relationships, as illustrated in Figure 9. In contrast to Figure 8 though, the involvement role here is *causal* (as opposed to *subtype*).

rri : Table

| | rriId | resType1 | resType2 | rriRole |
|---|---|---|---|---|
| ▶ | 34 | REResearch | RESupply | causal |
| | 35 | RESupply | CIEDemand | causal |
| | 36 | CIEDemand | CIESupply | causal |
| | 37 | CIESupply | CIECost | causal |
| | 38 | CIECost | REResearch | causal |
| * | (AutoNumber) | | | |

**Figure 9: Additional *rri* relationships (see also, Figure 8).**

Part of the SD external application's model dealing with this particular set of relationships is presented in Figure 10. Specifically, this is a *Powersim*™ model and a major reason for converting the CLD to this particular form is to take advantage of *Powersim's* powerful simulation and scenario generation/evaluation capabilities.

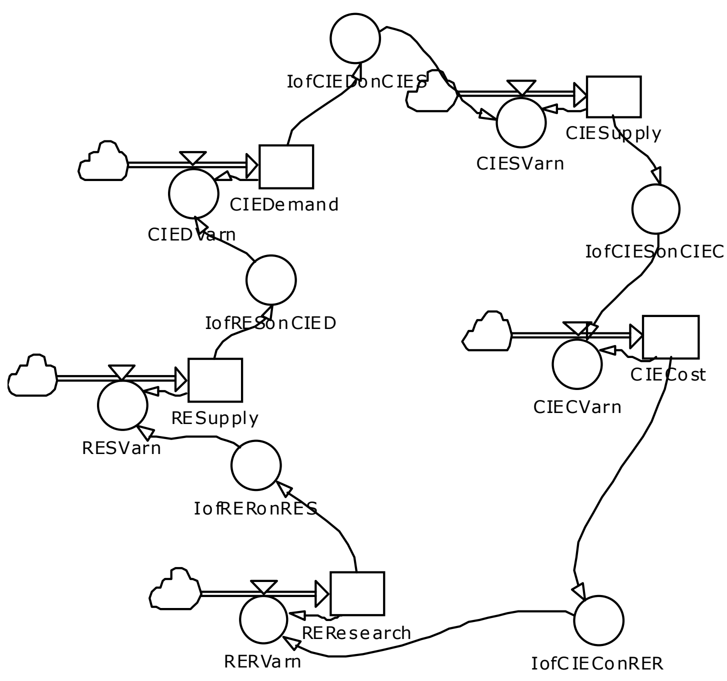


**Figure 10: Stock-flow representation of Figure 4 (partial).**

The basic building blocks of SD (stock-flow) models are *stocks* (represented as rectangles), *flows* (represented as arrows with circular flow regulators attached), *converters* (represented as circles) and *constants* (represented as diamonds). In our model, examples of stocks are *RECost* and *CIEDemand.* There is a level associated with each stock, which can be an actual value or a value bounded by some artificial scale. Stock levels vary with flows, which may be *inflows, outflows* or *bidirectional.* For example, *CIEDVarn* (CIE demand variation) is a bidirectional flow such that:

$CIEDemand_t = f(CIEDemand_{t-1}, CIEDVarn_t)$.

That is, in our model, the CIE demand level at time, *t,* is a function *(f)* of the CIE demand level at time, *t-1,* and its variation at time, *t.* These equations are the foundation of *Powersim's* formidable simulation capabilities. The third of our basic constructs, converters, serve a utilitarian role: they hold values for constants, calculate mathematical relationships and serve as repositories for graphical functions. In general, they convert inputs into outputs (hence, the name, 'converter').

The reader may have noted a similarity between each of the causal connections in Figure 10: in fact, they are identical in their basic structure, with each pair of stocks connected by a converter named, *IofXonY (impact of X on Y),* representing a mathematical relationship between *X* and the variation to *Y (YVarn).* This type of structure is extremely common in SD models and the general form of a relationship of this type between two variables (represented as stocks), $S_1 \rightarrow S_2$, is:

$$S_{2,t} = S_{2,t-1} + g_2(S_{1,t}) \quad - \quad (1)$$

where $g_2$ is the input function, *Iof* $S_1onS_2$. An example (derived from the United Nations 'Green Economy Report', (UNEP, 2010), illustrating the relationship between RE supply and CIE demand variation, is presented in Figure 11.

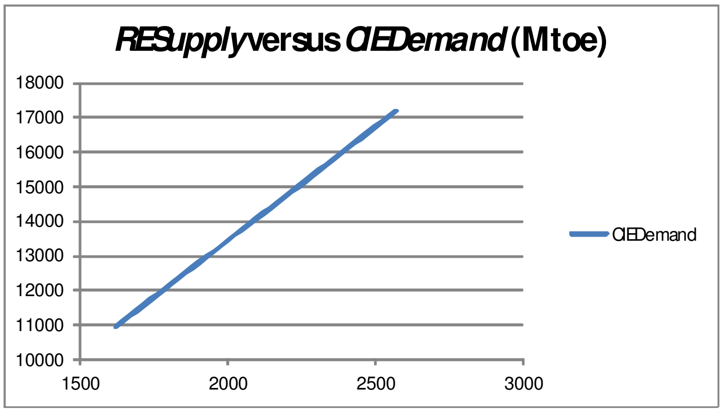


**Figure 11: RE supply on CIE demand relationship (Source, UNEP, 2010).**

More generally, a sequence of *n* stocks, $S_1$,---, $S_n$ ,causally connected by this type of impact relationship, has the general form:

$$S_{n,t} = S_{n,t-1} + g_n(S_{n-1,t-1} + \text{----} + g_2(S_{1,t-1})))\text{----}) \quad - \quad (2)$$

This is a classic recursive definition and within the conceptual model, using the very high-level, logic-based programming language *(Prolog)* we employ to manipulate basic relational data (implemented in *Access*™*)* at this level, it could be implemented as:

```
RT_x hasImpactOn RT_y if
        rri(RRIId, RT_x, RT_y, causal) and
        FnY(RT_X, RT_Y).

RT_x hasImpactOn RT_y if
        rri(RRIId, RT_x, RT_z, causal) and
        RT_z hasImpactOn RT_y.
```

where *FnY* represents a call to the actual procedure used to compute the impact function, $g_y$.

Relatively simple recursive procedures such as the above are used to facilitate the conceptual – external view mappings discussed earlier. Moreover, recursion has long been recognized as a highly-effective means of managing complexity in IS design and development. Recursion allows a solution for a problem to be derived through solutions to many smaller instances of the same problem. Furthermore, every recursive function can be transformed into an iterative procedure by replacing recursive calls with iterative control constructs (Kowalski, 1979: 107-129). Insofar as GETS is concerned, this means (for example) that the external SD model presented in Figure 10 (and the corresponding conceptual specification detailed above) may conveniently be re-specified iteratively. This allows redevelopment of the function as an alternative external application, using a more conventional, procedural software development environment (e.g. for execution-time efficiency reasons or to take advantage of specific features available in an alternative software development shell).

## CONCLUSION

In this paper we sought to demonstrate how IS and SD techniques can be employed to help tourism planners mitigate against and adapt to the impacts of climate change in tourist destinations. The variety of tourism destinations, in terms of; size, scope, location, engagement with the local community, reliance on the natural environment, energy reliance, and underlying economic structure (as well as governance and political processes) means that such modeling needs to be highly flexible. High level systems modeling, supported by well specified concepts and functions and applied by well developed SD models can help the tourism industry better prepare for the impending period of intense change as the global economy seeks to decarbonize itself. Importantly, GETS gives communities a much more comprehensive and better decision making framework and provides an incentive for evaluating/implementing their green growth and travelism options.

## REFERENCES


1. Becken, S. and Hay, J. E. (2007) Tourism and Climate Change: Risks and Opportunities, Channel View Publications, Clevedon, Buffalo.
2. Bratko, I. (1986) Prolog Programming for Artificial Intelligence, Addison-Wesley, Wokingham, England.
3. Chen, P.P.S. (1976) The Entity-Relationship Model: Towards a Unified View of Data, *ACM Transactions on Database Systems,* 1, 1, 9-36.
4. CTSR (2012) Green Growth 2050 Roadmap for Bali Sustainable Tourism Development, Centre for Tourism and Services Research, Victoria University, Melbourne, Australia.
5. Curtis, B., Kellner, M.I. and Over, J. (1992) Process Modelling, *Communications of the ACM,* 35, 9, 75-90.
6. Feldman, P. and Miller, D. (1986) Entity Model Clustering: Structuring a data Model by Abstraction, *The Computer Journal,* 29, 4, 348-360.
7. Forrester, J.W. (1961) Industrial Dynamics, MIT Press, Cambridge, MA.
8. Georgantzas, N.C. (2003) Tourism Dynamics: Cyprus' Hotel Value Chain and Profitability, *System Dynamics Review,* 19, 3, 175-212.
9. Hasan, H. (2003) Information Systems Development as a Research Method, *Australasian Journal of Information Systems,*11, 1, 4-13.
10. IPCC (2007) *Climate change 2007: Impacts, adaptation and vulnerability. Contribution of working group II to the fourth assessment report of the Intergovernmental Panel on Climate Change (IPCC),* In M. L. Parry, O. F. Canziani, J. P. Palutikof, P. J. van der Linden & C. E. Hanson (Eds.), Cambridge, UK.
11. Kowalski, R.A. (1979) Logic for Problem Solving, North-Holland, New York.
12. Law, A., De Lacy, T., McGrath, G. M., Whitelaw, P. A., Lipman, G., & Buckley, G. (2012) Towards a Green Economy Decision Support System for Tourism Destinations, *Journal of Sustainable Tourism,* 20, 6, 823-843.
13. Maani, K.E. and Cavana, R.Y. (2000) Systems Thinking and Modelling: Understanding Change and Complexity, Prentice-Hall, Auckland, New Zealand.
14. Markus, L., Majchrzak and Gasser, L. (2002) A Design Theory for Systems that Support Emergent Knowledge Processes, *MIS Quarterly,* 26, 3, 179-212.
15. McGrath, G.M. and More, E. (2005) An Extended Tourism Information Architecture: Capturing and Modelling Change, in A.J. Frew (Ed.), *Proceedings of the 12th International Conference on Information Technology in Travel and Tourism,* 26-28 January, Innsbruck, Austria, 1-12.
16. McGrath, G.M. (2007) The Identification of ICT Gaps and Needs within the Australian Tourism Industry, *Journal of Information Technology in Hospitality,* 4, 4, 161-178.
17. McGrath, G.M., Law, A. and DeLacy, T. (2012) Decision Support for the Development of Green Growth Tourism Strategies: Managing Complexity, *Journal of Knowledge Management Practice,* 13, 4, December, 2012.
18. Ritchie, J.R.B. and Crouch, G.I. (2003) The Competitive Destination, CABI Publishing, Cambridge, MA.
19. Senge, P.M. (1990) The Fifth Discipline: The Art and Practice of the Learning Organization, Random House, Milsons Point, NSW.

20. Simpson, M.C., Gossling, S., Scott, S., Hall, C.M. and Gladin, E. (2008). *Climate Change Adaptation and Mitigation in the Tourism Sector: Frameworks, Tools and Practices,* United Nations Environment Program, Madrid.
21. UNEP (2010) Green Economy Report, Towards a Green Economy: Pathways to Sustainable Development and Poverty Eradication, downloaded 23/12/2011 from : http://www.unep.org/greeneconomy/v2/GreenEconomyReport/tabid/29846/Default.aspx.
22. van Griethuysen, J.J. (1982) Concepts and Terminology for the Conceptual Schema and the Information Base, *ISO Technical Report ISO/TC9/SC5/WG3,* ISO, Geneva, Switzerland.
23. Vennix, J.A.M. (1996) Group Model Building: Facilitating Team Learning Using System Dynamics, Wiley, Chichester, UK.
24. Walker, P.A., Greiner, R., McDonald, D. and Lyne, V. (1999) The Tourism Futures Simulator: A Systems Thinking Approach, *Environmental Modelling and Software,* 14, 59-67.