An agent-oriented approach to holistic sustainability reporting

Sarah Hickmott
School of Computer Science and Information Technology, RMIT University, G.P.O. Box 2470, Melbourne, VIC 3001, Australia
Email: sarah.hickmott@gmail.com

Liam Magee
School of Global, Urban and Social Studies, RMIT University, G.P.O. Box 2470, Melbourne, VIC 3001, Australia
Email: liam.magee@rmit.edu.au

James Thom and Lin Padgham*
School of Computer Science and Information Technology, RMIT University, G.P.O. Box 2470, Melbourne, VIC 3001, Australia
Email: james.thom@rmit.edu.au
Email: lin.padgham@rmit.edu.au
*Corresponding author

Abstract: This paper presents a software application for sustainability reporting where a multi-agent system is an integral part of the overall architecture. We describe the social science philosophy and approach on which the application is based, and the ways in which an agent-based system is able to support these. In particular, we explore how the pro-active, goal oriented, context sensitive nature of the agent system is able to realise the principles of the underlying philosophy, which are to encourage holistic monitoring of sustainability and balance user relevance with standardisation. Further to the functional benefits, the paper looks at why agent oriented design is also valuable as a communication tool in the multi-disciplinary team, enabling the non-computer science members to be actively engaged in system design.

Keywords: agents; sustainability indicators; prototype; agent-oriented software engineering; holistic sustainability monitoring; holistic; proactivity; adaptivity; multi-disciplinary approach; design.

Biographical notes: Sarah Hickmott did her PhD in Engineering at the University of Adelaide and National ICT Australia. Following this, she worked as a Research Fellow in the Intelligent Systems Group at the RMIT University, Melbourne, Australia, looking at how to understand and influence social systems using agent oriented modelling and simulation.

Liam Magee is a Research Fellow in the Global Cities Research Institute at the RMIT University, Melbourne, Australia. He obtained his PhD in 2010 for work on the Semantic Web and its relationship to knowledge management and publishing. Since then, he has been active in research on urban development and sustainability, and developing software tools to support improved visualisation, analysis and decision-making in that area.

James Thom is an Associate Professor of Document Computing in the School of Computer Science and Information Technology at the RMIT University in Melbourne, Australia. He obtained his PhD in Computer Science from The University of Melbourne in 1993. Since then, he has been active in research on: databases; document management; information retrieval of text, image, video and XML; and ontologies for knowledge representation.

Lin Padgham is a Professor in Artificial Intelligence at the RMIT University, where she leads the Intelligent Agents Research Group. She obtained her PhD in 1989 from Linköping University, Sweden, for work on default inheritance hierarchies. Since 1995, she has worked in the field of intelligent multi agent systems, on a range of theoretical and practical issues. She co-developed the Prometheus methodology and Prometheus Design Tool (PDT) for designing agent systems, and has more recently been working in inter-disciplinary applications of agent technology, and agent-based modelling and simulation.

This paper is a revised and expanded version of a paper entitled ‘An adaptive system for proactively supporting sustainability goals’ presented at the 11th International Conference on Autonomous Agents and Multi Agent Systems, Valencia, Spain, 4–8 June 2012.

1 Introduction

This paper presents an application where belief desire intention (BDI) agent technology is used for an important component of the system. The application is a software support system for a new approach to sustainability reporting. The paper demonstrates, via a concrete application, how basic aspects of an agent oriented approach facilitate the desired functionality. This centres around balancing a responsive user-driven approach with a pro-active goal driven approach to use standardised or shared indicators where possible, and to assist users to cover specific aspects of sustainability in their planning. This balance between reactivity and pro-activity is at the core of agent systems and was of substantial value in this case. The paper also brings out a less discussed value of agent-based systems, as a representation that facilitates domain experts engaging with understanding system design that maps directly to implementation. A further value of the paper is the illustration of the agent system as part of a larger whole that is the application.
An agent-oriented approach to holistic sustainability reporting

It has recently been suggested that current reporting practices are failing to capture the full picture of whether an organisation’s practices are sustainable (Bell and Morse, 2008; Eccles and Krzus, 2010; Scerri and James, 2010). This is partly due to the way economic issues are often considered independently from environmental and social issues, and vice versa. Moreover, sustainability reporting is typically addressed using either external and standardised or locally-defined indicator sets. This forces a choice between choosing sustainability measures that allow for comparability with other organisations or measures that are most relevant to the organisation. Increasingly, though, there is a need to support organisations to use a combination of indicators that capture organisation-specific concerns and indicators that communicate and compare the organisation’s performance externally.

Our research group of computer and social scientists has been developing a software system to facilitate sustainability reporting in accordance with a new reporting framework that has recently been proposed to address these issues (Scerri and James, 2010). The software needs to support organisations to combine their own locally defined indicators with global standards in a sustainability reporting template, and to look at their performance against a broad rather than narrow definition of sustainability – that is, holistically. It should to do this in a manner that both adapts to and guides the user. The application ought to be responsive to end user preferences as to how to perform tasks and define indicators, but also proactive with respect to the goals of the underlying philosophy. For example, the system should allow users to define their own indicators, guide them to identify a set of indicators that holistically monitors their performance, and suggest the use of standard indicators when compatible with their needs.

The system we are developing currently exists as a continually evolving prototype, being trialled with case study participants for ongoing feedback. The multi-agent component mediates between an interactive web interface and an extensible RDF-based data store for capturing information about sustainability projects, and the specific indicators used to measure them.

In this paper, we look at how fundamental properties of the BDI agent paradigm have readily met our above-mentioned needs: the proactive, goal-oriented features enable us to easily guide and support the user; and, the context-sensitive manner in which agents achieve their goals allows us to build a system that can readily adapt to specific user needs. Also, the goal-plan framework lends itself to the easy addition of automated reasoning support, and allows quick adaptation of the prototype system in response to case studies.

Furthermore, this paper looks at how the BDI framework, and the available design tools, have facilitated a highly interactive collaboration by providing an effective structure for communication between the computer and social scientists in our research team. We have found that the agent oriented entities of goals, events, plans and beliefs are sufficiently intuitive that it is possible for non-technical team members to understand and directly contribute to the agent design.

In Section 2, we first outline the philosophical motivation and approach behind the application system. In Section 3, we describe the system itself, including how the multi-agent design has been used to support the philosophy. In Section 4, we describe how the agent-oriented design methodology and tools assisted in communication within a cross-disciplinary team. We conclude by highlighting the benefits of multi-agent design we have experienced.
2 Sustainability reporting

2.1 Motivation

The concept of ‘sustainability’ refers to the capacity for something to survive, endure or maintain itself. In a human context, given our orientation towards economic and social growth, sustainability is frequently coupled with ‘development’. The most famous definition of ‘sustainable development’ was formulated by the World Commission on Economic and Development (1987):

“Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs.”

A quarter of a century since that definition was published, there has been considerable discussion of sustainability and sustainable development in the literature. In this section, we touch briefly on three aspects of this discussion. The first concerns the what kinds of needs have to be met, for both present and future generations. The second concerns who identifies those needs. The third concerns how those needs can be selected, measured and represented in a systematic way. Here we describe the main paradigm for addressing these concerns, and outline an approach which differs from this in important ways. That approach, which we elsewhere describe as the ‘Circles of Sustainability’ (Magee et al., 2013; Scerri and James, 2010), aims to be more holistic in how sustainability is conceptualised and measured. By holistic we mean

a encompassing a more comprehensive theory of sustainability

b being more inclusive of the relevant stakeholders in sustainable development

c supporting a broader range of indicators for measuring sustainability.

In the final part of this section, we describe a workflow that shows how this approach can be implemented in real-world scenarios. From this, we highlight features that a software system needs to exhibit to support the approach.

2.1.1 Models of sustainability

The current paradigm for sustainability assessment and reporting is the ‘Triple Bottom-Line’ (TBL) approach (Elkington, 1997). The core insight of TBL is that companies, governments and other organisations should not conceive of ‘development’ simply in economic terms. To develop sustainably requires also considering the environmental and social impacts of organisational activity. Accordingly, accounting, reporting and auditing procedures should measure against three ‘pillars’ or ‘bottom-lines’ of performance: economic, environmental and social.

In the wake of general concern about the impacts of businesses on the environment, the TBL has been highly influential in both the literature and in corporate and government practice (Eccles and Krzus, 2010). However, it has also been widely criticised:
The three pillars are under-specified, leaving considerable range for interpretation. A practical consequence is that TBL can become a ‘box-ticking’ exercise for organisations, since they can ignore ‘inconvenient’ aspects of environmental and social performance (Norman and MacDonald, 2004).

They are also often treated as distinct entities or categories, when in reality these entities often have complex interdependencies (for example, economic pressures also influence social outcomes) (Magee et al., 2013).

‘Social’ is used as a vague and miscellaneous category for non-economic and non-environmental activities. This ignores the fact that in non-corporate contexts, economic outcomes are often merely a means for achieving social ends (Scerri and James, 2010).

2.1.2 Stakeholders

Another distinction in the sustainability literature is between ‘top-down’ and ‘bottom-up’ indicator development. ‘Top-down’ refers to the use of standardised indicators, of which a number exist for measuring different dimensions of sustainability [for reviews, see Böhringer and Jochem (2007), Mayer (2008), Singh et al. (2009)]. ‘Bottom-up’ refers to the increasingly common practice of involving stakeholders, such as customers, suppliers, employees and members of the public, in determining critical sustainability issues and indicators (Bell and Morse, 2008).

This forces a choice between measuring what is most relevant to the organisation, and what allows for comparability with other organisations. While there are contexts in which one approach can and should be favoured over another, it is increasingly common to desire the ‘best of both worlds’ – indicators that map directly to local and specific concerns, but that can also be used to communicate and compare sustainability performance.

2.1.3 Heterogenous indicator sources

A more practical issue concerns how indicators are identified and included in a sustainability reporting framework when ‘bottom-up’ methods are used. Tools for selecting indicators are often very simple; in a community forum or corporate workshop environment, they can include a whiteboard, butcher paper or notepads and pens (Bell and Morse, 2008).

However, there are advantages to having flexible software systems to support this activity. This is especially the case in complex scenarios, when dealing with potentially large numbers of indicators, competing sustainability priorities and with different stakeholders. Software support for capturing indicators through community forums and workshops has largely been limited to ad hoc methods such as spreadsheets and documents. To date, there has not been a satisfactory approach to allow a range of users to browse, search and select indicators from heterogenous sources. This often results in existing indicators being ignored, or a single indicator system being selected to the exclusion of others.
2.1.4 Towards a holistic approach

An alternative approach, ‘Circles of Sustainability’, for sustainability reporting has been suggested (Scerri and James, 2010; Magee et al., 2013). This involves looking at economic, cultural, political and ecological issues inclusively, along with their interrelationships. It also specifies a number of sub-domains which specify more clearly what areas of sustainability need to be measured against. This comprehensiveness is the first sense in which we term the framework is ‘holistic’.

‘Circles of Sustainability’ is similar enough to the TBL that organisations do not need to discard existing indicator and data sets. However, it makes the following conceptual adjustments, which we summarise below:

1. the ‘social’ is separated into ‘political’ and ‘cultural’, to allow greater specificity and distinction, resulting in four domains (‘economy’, ‘ecology’, ‘politics’, ‘culture’)
2. organisational issues and indicators can belong to one or more of the four domains
3. relationships and interactions across domains are identified to examine the impacts of, for example, political issues on economic and ecological ones.

It advocates a blended approach to developing indicators for reporting on sustainable development. We recognise that it frequently makes sense to utilise existing sustainability indicator sets, both due to the expertise required to define them, and for the purposes of comparability. We also acknowledge the need to involve communities of stakeholders in determining what is most relevant to report against in a given context. Our approach therefore supports both ‘bottom-up’ and ‘top-down’ approaches – the second sense in which it aims to be holistic.

Finally, the approach emphasises indicator reuse, ideally through a software system that can simplify navigating and selecting indicators from heterogeneous sources. This is the third sense in which the approach is holistic.

2.2 Requirements and methodology

Our project team aims to develop a software system which supports the approach above. At a general level, the software should help stakeholders develop indicators that reflect a comprehensive and holistic sense of sustainability. At a more fine-grained level, the software needs to support the ‘four-domain’ model described previously. To allow for ‘top-down’ and ‘bottom-up’ indicator development, the software also needs to support both system- and user-defined indicators.

To develop a more precise set of domain requirements, we consulted with a number of organisations who regularly prepare sustainability reports. These included a large corporate organisation, a local government council and several non-government organisations. We consolidated their feedback with the generic requirements of the approach we have adopted towards sustainability reporting.

Below we present the list of requirements, along with a generic workflow for developing a sustainability report that we also established in discussion with the two partners. The main system requirements are summarised as follows:
An agent-oriented approach to holistic sustainability reporting

- Proactive guidance and support: Guide users through the process of defining a project, including proactively providing suggestions to assist in finding a holistic set of indicators, and identifying standard indicators, where these are relevant to the organisation.

- Flexible process: Be responsive to user needs by supporting them to build their project in whatever order they wish.

- Holistic perspective. Support the user to define a set of indicators that holistically monitors their sustainability issues, ensuring coverage of the four domains: economy, ecology, politics, and culture.

- Standard and locally-defined indicators: Provide access to existing indicator sets and support for querying these sets. Allow the specification of locally defined (local) indicators in many different forms, ranging from highly specified quantitative variables, to unstructured qualitative variables. Help the user to identify when a local indicator can be replaced with a standard indicator.

- Extensible automated reasoning: Provide automated reasoning to support, for example: ranking and relating project issues; suggesting possible indicators from a knowledge base; identifying conflicts between indicator targets.

- Manage fast changing requirements: Allow quick adaptation of the system as the methodology is refined in response to case studies.

The software also needed to implement the following workflow, to allow users to develop a sustainability report:

- Identify the main problem motivating the organisation to use this sustainability reporting process. For example, a city council may want to use the sustainability reporting process to look at the problem of ‘corruption’.

- Assess the general problem against each of the four sustainability domains (economy, ecology, politics, culture). For example, this could show that ‘corruption’ is an issue that principally relates to the domains of politics and economics (but is also impacted by culture and ecology).

- Identify at least one critical issue for each of these domains. For example, ‘public distrust with officials’ may be a critical issue that belongs in the political domain.

- Associate at least one indicator to monitor and measure the achievement of this objective (e.g., ‘public satisfaction with government’). Users can define their own indicators, or choose indicators from existing sustainability indicator systems. Where possible, the system should prompt users about relevant indicators already included in the database.

- Set concrete targets for all indicators (e.g., ‘less than 40% public dissatisfaction in polling results’).

- Check the consistency of the report: have all domains been measured by at least one issue? Do all issues have at least one indicator to measure them against?

- Generate a sustainability report template that shows the domains, issues, indicators and domains in a logical and easy-to-read way.
The workflow is also not designed to be prescriptive, and steps can be reordered or removed to fit constraints of a particular reporting situation. The need for flexibility motivated our choice of a multi-agent system for the implementation, which we describe in the remainder of the paper. Once the initial version of the system was developed, it was then reviewed by potential end-users. We discuss their feedback and recommendations briefly in our conclusions.

3 System description

In this section, we first give a brief overview of the entire system, followed by a high level overview of the agent system component. We then explore in some detail how the agent architecture contributes to the key requirements to provide proactive support for holistic reporting and for combining standardised and local indicators, while also providing the flexibility that is central to the philosophy of the sustainability reporting approach adopted.

Figure 1  Overall architecture of system

We then discuss how the agent-based design methodology supported the communication between social and computer scientists, facilitating a much more detailed understanding of and contribution to the software than we believe would have been possible with more traditional approaches.
3.1 Architecture

The system architecture needs to support multiple users accessing multiple sustainability projects simultaneously via the web. Though an individual user may be associated with multiple projects, in a single browser session they can only be connected to one project. This led us to design the four main components of our architecture as shown in Figure 1. Users access the system via an HTML5-compatible browser. The web application running on an Apache Tomcat web server handles session management (including authentication), interaction with users via HTTP, sending and receiving messages to and from the agent system, and storing and retrieving information in the knowledge base. The Agent system has a separate instantiation that manages each project. The Knowledge base provides a persistent store for shared knowledge about sustainability indicator sets as well as project-specific knowledge.

The agent paradigm being used is that of belief, desire, intention (BDI) agents (Rao and Georgeff, 1991), a popular paradigm with a range of available development languages and platforms. Each agent in these systems is based essentially on a hierarchical set of predefined plans. These plans are triggered by internal subgoal events, by message events from another agent, or by information from some source external to the agent system (a percept). A plan can post subgoals, send messages, and in our case, send communications to other parts of the overall system. We have chosen GOal ORiented TEams (GORITE)\(^1\) (Rönnquist, 2008) as the implementation language, because we required open source freely available software. It is a Java-based platform that supports packaging and modularity – important for a large project. It also supports explicit teamwork and dynamic agent teams, although we have not utilised that in this application. For the agent system design we have used the Prometheus Design Tool (PDT) (Winikoff and Padgham, 2004; Thangarajah et al., 2005), and it is this which has been used for design discussions within the team, as well as providing the details for implementation. Design figures within this document are produced using PDT, though at times they are presented in a simplified form.

Figure 2 Overview of agent system for one project (see online version for colours)
Each aspect of the project, as defined by the methodology (Section 2.1.1), is managed by a different agent: a *Main Issue Agent* for assisting the users in identifying and recording information regarding their main issue; a *Critical Issues Agent* which assists in defining a balanced set of specific critical issues related to the main issue, and covering the four key domains; an *Indicators Agent* which assists users in selecting and recording indicators related to their critical issues; a *Monitoring Agent* for setting and monitoring targets for the indicators; and a *Status Agent* which tracks the overall status of the project with respect to the various phases. A *Reporting Agent* is also planned for providing regular reports and possibly also proactive alerting on the basis of data feeds.

An overview of the agent system for a single project is shown in Figure 2. Each of these agents have goals and context-related plans for both responding to user requests relevant to their project area and proactively guiding the user to meet the project specification requirements in a manner that supports the underlying philosophy. These agents operate independently and asynchronously, allowing and supporting the user to build a project in the manner which best suits them.

The user, via the web application, is able to query, and receive a response from the agent system via the *AgentInterface*. The agent system is also able to proactively make suggestions in an asynchronous manner, via the *MessageBus*. The agents can communicate both directly with the knowledge base as well as follow messages between the web application and the knowledge base. This is via Empire, a Java Persistence API (JPA) service that provides an object-oriented interface to the RDF-based knowledge base.

Together the agents are able to support the various aspects of the application, each monitoring proactively to provide assistance as well as being available when requested explicitly by the user or another agent. The agents also communicate important status information (such as whether a phase is completed, and what suggestions have been sent) to the *Status Agent*, which maintains it in a shared store available to all the agents.

We now proceed to explain in more detail how the agent-based architecture specifically supports some key aspects of the application requirements.

### 3.2 Proactive behaviour

The main principles of the ‘Circles of Sustainability’ are to promote both a more holistic approach to sustainability planning and monitoring across the four domains, and to support both locally defined and standardised indicators. We found that supporting these principles in our application was relatively straightforward with an agent-based approach, due to the goal-oriented, proactive behaviour of agents.

Figure 3 shows a design diagram that indicates some of the goals and plans within the *Indicators Agent* that in addition to responding to a user request for indicator suggestions, also allows the agent to take the initiative and proactively generate an internal goal *IncreaseStdRatio* to attempt to increase the ratio of standardised indicators to locally defined indicators. On the left hand side of the diagram we see a (partial) specification of how the system will respond to a user request *ReqSuggestion* that contains either an existing indicator, or one or more keywords, and asks for indicator suggestions. The plan *PReqSuggestion* that responds to this request then has several subgoals which are achieved in sequence. The subgoal *DetPreferredSet* has several different plans for determining which indicator set is preferred for this user/project. The subgoal *ExtractKeywords* has a plan which extracts keywords from the existing indicator
An agent-oriented approach to holistic sustainability reporting

description if needed. The subgoal GetMatchingIndicator has a plan which queries the indicator database (possibly repeatedly) for indicators from the preferred set which match the keywords. Depending on what has been found, and the desired number of suggestions, the plan PReqSuggestion may also instantiate the subgoal GetRelatedIndicator to trigger a process of finding synonyms for the keywords and endeavouring to find related indicators from the preferred indicator set. The subgoal SuggestIndicators then triggers a process to package up the results and pass them to the web application.

**Figure 3** Proactive approach to promoting standard indicators (see online version for colours)

Although this process goes some way to support the philosophical intention to use standardised indicators where appropriate, it is totally dependent on the user taking the initiative to make a request. We prefer the application to be more proactive in achieving its purpose. So, in the upper right of the figure, one can see the goal IncreaseStdRatio which is automatically generated in cases where the ratio of local indicators to indicators from a standardised set drops below a certain level. This goal then has two possible plans. SuggestBrowseStd simply alerts the user to the low ratio and asks if they would like to browse a standard indicator set. This can be presented in a structured manner, based on project issues, currently used indicators, etc. The other plan GetReplacementOptions uses some of the same subgoals and plans that exist for satisfying the user request, and collects specific possibilities for replacing existing local indicators with standardised ones.
In this scenario the system much more actively seeks to support the goal of the philosophy to use some number of standardised indicators, where appropriate. These proactive, system-initiated goals are easy to include within an agent oriented architecture, due to the inherently autonomous, goal oriented, proactivity of agents. The system also includes similar proactive behaviour around monitoring for holistic coverage of issues, and making suggestions if some of the four domains are being neglected.

3.3 Flexible process

The methodology presented in Section 2.1.1 suggests a linear process for defining a sustainability project, starting with identifying the main issue, through to defining the comprising economic, ecological, political, and cultural issues and objectives, and ending with naming indicators and targets for monitoring these issues. Many users may not come with a main issue in mind however, or may not wish to define one initially. Those previously doing TBL reporting, for example, may already know some of the indicators they want to measure, but have never considered the underlying critical issues. Forcing these users to begin by identifying a main issue may encourage ‘box-ticking’ responses, thus failing the philosophy behind this reporting approach. Our aim is for the software to support users to build their project in the manner that suits them. The agent-oriented approach supports this in two particular ways.

Firstly, a user can define the different aspects of their project in any order, and agent responses and proactive suggestions will be generated in the manner described in Section 3.2. Consider, for example, the user-system interaction shown in Table 1. User activity is shown on the left side, with user requests shown in italics. On the right side are agent responses to these requests, also shown in italics, and proactive suggestions. In this example the user begins a project specification by first defining two local indicators, X and Y. While the user is focused on defining indicators, they are being proactively supported by not only the Indicators Agent, but also the Critical Issues and Status Agents. The Indicators Agent actively provides suggestions for strongly related standard indicators, and responds to user requests to look for (less strongly related) standard indicators. Meanwhile, the Status Agent brings the user’s attention to the fact these indicators are linked to Critical Issues and suggests two different ways the user could begin to consider this aspect of their project. Then, when the user subsequently decides to replace a local indicator with a standard one the Critical Issues Agent offers the possibility to look at issues and domains that have been associated with this indicator.

Secondly, the use of context conditions in plans naturally allow goals to be achieved in different ways, depending on the particular situation and the needs of the user. For example, consider the goal DetPreferredSet shown in Figure 3. This goal determines which indicator set is preferred for this user/project, and has several different plans, which are applicable depending on the situation. The plan AskUserPrefSet simply asks the user what their preferences are with respect to a list of possible indicator sets (for example, the GRI set). DefaultPrefSet allows for the use of a default preferred indicator set, such as the GRI, if one has been specified. StoredPrefSet will use an indicator set that has been previously specified as the preferred indicator set for this project, perhaps by previously asking the user. InferPrefSet will infer the preferred set based on the indicators currently being used for the project if it is clear that the non-local indicators being used are predominantly from a particular standard set. In these ways the system is able to adapt to and engage with specific user needs.
Table 1  Agent responses and suggestions to user

<table>
<thead>
<tr>
<th>User activity</th>
<th>Agent responses and suggestions (sX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a new project and select the indicators</td>
<td>(s0) Browse indicator sets? From Indicator Agent</td>
</tr>
<tr>
<td>Define local indicators X, Y.</td>
<td>(s1) indicator X looks similar to the indicator GRI-123. Replace? From Indicator Agent (s2) Indicators X, Y are not linked to Critical Issues. Look at example critical issues? From Status Agent</td>
</tr>
<tr>
<td>Select suggestion (s1).</td>
<td>Request suggestion for standard indicator similar to Y.</td>
</tr>
<tr>
<td></td>
<td>Response: Select preferred indicator set: [Any, GRI, OECD, …]</td>
</tr>
<tr>
<td></td>
<td>from Indicator Agent</td>
</tr>
<tr>
<td></td>
<td>Set preference for OECD indicators.</td>
</tr>
<tr>
<td></td>
<td>Response: Indicator ‘Y looks similar to’ OECD-455 and OECD-466.</td>
</tr>
<tr>
<td></td>
<td>From Indicator Agent</td>
</tr>
<tr>
<td></td>
<td>Request suggestion for standard indicator similar to Y; no preferred indicator set.</td>
</tr>
<tr>
<td></td>
<td>Response: Indicator Y looks similar to: GCI-366, EVI-12, … [more]</td>
</tr>
<tr>
<td></td>
<td>from Indicator Agent</td>
</tr>
<tr>
<td></td>
<td>Replace indicator Y with indicator GCI-366.</td>
</tr>
<tr>
<td></td>
<td>(s3) View the Critical Issues commonly linked to indicator GCI-366? From Critical Issue Agent</td>
</tr>
<tr>
<td></td>
<td>Select suggestion (s3).</td>
</tr>
</tbody>
</table>

Figure 4  Screen shot of indicator suggestion (see online version for colours)
To see how the proactivity and flexibility described is facilitated in practice, refer to Figure 4. This shows a screen shot of the user interface prototype. On the left side is a list of project specification requirements, titled ‘Project Progress’. User input is obtained through the form in the middle. On the right side is a scrolling list of agent suggestions. In this particular instance the user has previously defined the main project issue and goal, and identified some critical issues and objectives. They have just begun defining a local indicator called ‘Number of species’; the agent system is proactively responding to this by suggesting standard indicators, which may be related (based on name similarity). Following any of these suggestions is at the user’s discretion. Observe that the user has control over the order in which they specify different aspects of their project; the agent system follows the user’s choices and proactively provides suggestions and guidance.

4 Evaluation

An additional advantage that we did not foresee with the agent paradigm, and especially the graphical presentations of detailed design using PDT, was the way in which this facilitated involvement of the social science team members in the software design. Consequently, we evaluate here two aspects: the first is the use of the agent paradigm for assisting design collaboration between technical and non-technical team members; the second is the result of some preliminary user studies.

4.1 Agents as a facilitator of collaboration

The BDI model concepts of beliefs, goals and plans (which have a clear mapping to implementation entities) were easily understood by non-programmers and helped both technical and non-technical members of our group communicate detailed information about how system behaviour could be delivered to end users.

Consider for example Figure 5(a) which is an initial version of a portion of the design shown earlier in Figure 3. Here we show the plan for offering some suggestions for indicators that are part of a standard indicator set, when the ratio of local to standardised indicators is undesirably high. In the initial design the agent programmer had planned to obtain all indicators that were related to the local indicator in focus, from any standard indicator set. When presenting this design, social science team members were able to indicate that they would prefer to present options from a particular standard, possibly chosen in a number of ways.

This then allowed for a revision and discussion of the design shown in Figure 5(b) where first the standard to be used is determined.

Another example is a more complex use case for supporting ranking and selection of critical issues that need to be measured in the system, based on evaluation against multiple criteria. Social science members of the group first tried a specific multi-criteria decision making approach, called the analytic hierarchy process (AHP) (Saaty, 1980), with end users using an Excel spreadsheet. Based on feedback, they worked with the computer science team to break the process down into a series of goals and subgoals that could be implemented in the agent system. We found analysing and describing AHP behaviour in terms of goals and subgoals relatively straightforward. For example, a top
level goal of AHP is to produce a principled ranked list, based on importance weightings of multiple criteria. A subgoal in this process is to obtain pairwise preferences for each pair of issues, on each criteria. Once the whole process was broken down into these kind of goals and subgoals, it was then possible to develop and confirm with the social science team the plans to achieve each subgoal, and to determine when one would use one approach over another (giving the context condition of the plans). This was an efficient and effective approach to co-design, which social science team members found intuitive and natural.

Figure 5  Suggesting standardised indicators, (a) initial design (b) revised design (see online version for colours)

4.2 End user evaluations

We presented the initial version of the system to a group of 4 social scientists who were familiar with the ‘Circles of Sustainability’ approach, to assist us with reviewing the system. We first demonstrated several scenarios in which users could enter issues and select appropriate indicators from different sets included in the system. As issues were entered, reviewers noticed that the agent system would suggest appropriate indicators from the database. They then could click on the final sustainability report template, which showed issues and indicators logically grouped together.
Reviewers were then encouraged to explore the system, and to provide verbal feedback at the time of the presentation, as well as respond to an online questionnaire. In verbal conversation, they generally acknowledged the usefulness of the system, and appreciated the guidance it provided in suggesting indicators. The reviewers also offered several suggestions, which we aim to incorporate into future versions:

1. supporting real-time discussion on issues between different stakeholders
2. allowing for indicator measurements to be tracked and compared over time.
3. improved export functions for project and reporting data
4. further support and assistance for complex parts of the application, such as prioritising issues.

In their own time, the reviewers then completed an online questionnaire, responding to the following open-ended questions:

1. Was the presentation of the software clear? How could it be made clearer?
2. Does the software seem to fit with what has been described in the associated methodology?
3. Does the software seem useable? How could it be more user-friendly? Does it need to be adapted or extended to accommodate particular user needs?
4. The software ultimately aims to help an organisation select indicators for measuring a series of critical issues that a stakeholder group has identified. Do you think it does that job well? How could it do better?
5. Do you anticipate problems harmonising the set of indicators recommended by the software with indicators and data points your organisation already collects?
6. Can you imagine using the software in everyday sustainability assessment and reporting practices?
7. Please let us know any other feedback you have about the system.

Questions 1, 2 and 3 related, respectively, to the clarity, relevance and usability of the software. Questions 4 and 5 relate to the specific tasks of selecting and harmonising indicators. Question 6 relates to perceptions of whether the software could be adapted to existing organisational practices. Question 7 solicited any further remarks.

In response to questions 1 and 2, all reviewers felt the presentation of the software was clear, and fitted well with the associated methodology. In the words of one participant:

“Yes it does, I find the methodology quite complex to understand and the software is a practical vehicle to conceptualise the theory.”

In relation to usability (question 3), responses were mixed. All respondents agreed the software was relatively intuitive, but thought there was potential to simplify the navigation and steps. As one respondent put it:

“The software is certainly useable, but it needs a stronger visual map of the phases of assessment work that need to be done in the process (and have been undertaken).”
Again respondents also agreed the software would be helpful to organisations in selecting and harmonising indicators (questions 4 and 5). One respondent stated:

“I think the software does that reasonably well. It is tackling a very complex area with indicators, and facilitating region/issue specific development of customised indicators is very valuable.”

Another suggested the software could help coordinate between standardised indicators and existing sustainability datasets:

“If there is a mis-match between the indicators recommended and the data already collected, then the software can help provide a basis for discussion and contemplation.”

In terms of the general question of whether the system could be used in everyday sustainability assessment and reporting practices (question 6), all respondents were enthusiastic about its potential. One stated the software “offers time benefits and replaces double handling of data collected”, while another suggested:

“With the capacity for a reporting printout, this system could have profound consequences for managing major and complex projects where the manager does not have access to proprietary and expensive tools.”

When asked to provide general feedback, one reviewer provided the following overall assessment of the system:

“Undoubtedly, this tool has significant application for those involved in city governance and its step-by-step functionality helps to remove much of the “mysticism” surrounding sustainability.”

The responses overall suggest the agent-based system met the goals of implementing the Circles of Sustainability methodology, and would have broad applicability for organisations reporting on sustainability. We plan to incorporate suggestions, mostly in the area of user interface design, into future iterations of the software.

5 Related work

Numerous commercial vendors have developed systems for environmental sustainability reporting, such as VERISAE (VERISAE Inc., 2013), Cloudapps (Cloudapps, 2013), TBL2 UK (Centre for Sustainability Accounting, 2013), IHS (IHS, 2013) and SIMPLiFi (SIMPLiFi Solutions, 2013). Ecological Footprint calculators are the most widely used examples of such software, and can be used by individuals, enterprises, cities and countries to measure the embodied biocapacity to sustain production of goods and services (Ewing et al., 2010). While such tools are very useful in calculating specific measures, they are built using specific and inflexible assumptions, and are generally designed to measure only environmental and economic aspects of sustainability. Social indicators, if measured at all, cover internal organisational characteristics such as employee diversity, rather than broader impacts (Smith, 2010).

More closely related to our application is a widely-used online reporting tool, the MDG Dashboard (European Commission Joint Research Centre, 2010), that supports visualising different sustainability data sets taken from the UN MDG indicator database.
The MDG Dashboard allows other datasets to be used, but requires specialist technical knowledge to prepare them, and once created, does not allow for multi-user editing. Our application is similarly holistic in philosophy to this system, but is far more customisable in terms of the types and relationships between indicators. It also differs in allowing multiple indicator sets to be applied to a single project, and for these to be edited by end users.

Web-based multi-user collaborative environments, such as wikis, blogs and social networking sites, have become increasingly popular for documenting and reporting on projects. Such systems, even where they support structured data sets (such as semantic wikis), still required considerable customisation for the specialised case of sustainability reporting. Hence, while flexible, they do not offer the guidance necessary for developing complex indicator reporting structures.

Existing systems support an impressive array of reporting approaches. However none sufficiently address the challenges of ‘bottom-up’ sustainability reporting – supporting a high degree of flexibility without sacrificing context-specific guidance. While extensive qualitative and quantitative processes exist in the literature for ‘bottom-up’ reporting, to date these have not made their way into supporting software. By using agent systems to provide unobtrusive support to the process of developing sustainability indicators, our application facilitates flexible collaboration and structured guidance in a novel way.

Various research groups have also applied agent technology to sustainability issues. For example, Kwak et al. (2012) report an agent system which decreases building energy use by suggesting to users alternative, more energy efficient arrangements, particularly for situations involving multiple users. Like our system, this application benefits from the balance between reactivity and pro-activity which is a hallmark of agent systems. The agents are responsive to users, but also have their own goals (energy use reduction). The major focus of this work is however on the novel MDP processes which determine the multi-objective optimisation, given multiple users and the system goals. Mamidi et al. (2012) also use intelligent agents to obtain greater energy efficiency in a smart building, although here the focus is on agent-based sensing and control of the heating, ventilation and air conditioning (HVAC) system.

Pulter et al. (2011) investigate via simulation a potential agent-based infrastructure for intersection control using an architecture of intersection agents and driver assistant agents. This work does not however address specifically the interaction between the human users and the agent system.

Electricity grids and power distribution are an area where there has been substantial work within multi-agent systems. Chalkiadakis et al. (2011) describe one such example where a game theoretic approach to pricing mechanisms ensures that none of the autonomous agents have a financial incentive to break away from their local cooperative. This work however is primarily about mechanism design, rather than the agent architecture as such, and does not have any component of interaction with end users. Another example of agent-based smart grids is Vandael et al. (2011) where electric cars and the scheduling of their recharging is used to reduce load imbalances.

Also somewhat related to our research is the use of agents to assist users in finding information, for example for meta-search (Wei and Junjie, 2007) and for searching the semantic web (Lu and Rahman, 2007).
6 Conclusions

Our work demonstrates how an agent oriented approach to sustainability software tools can proactively facilitate a flexible process for managing sustainability goals, that guides an organisation to examine their sustainability goals holistically and supports the integration of global standards with the specific needs and measurement capabilities of the organisation.

More generally, what we have found is that the agent paradigm is extremely suitable for the kind of application which needs to include both proactive system generated behaviour, as well as responsiveness to different users and their way of operating. In addition, the BDI agent framework aided communication in a cross-disciplinary team, providing an important linkage between high-level philosophy and requirements and low-level technical implementation. Whilst it is well understood that agent systems are suitable for use in dynamic environments where the responsiveness is key to managing environmental change, the value of the approach for enabling systems which are responsive to users, while still driving particular system goals, has perhaps been less clearly articulated. This application provides a good example of this situation.

Acknowledgements

This research was supported under Australian Research Council’s Linkage Projects funding scheme, project number LP0990509.

References


Notes

1 http://www.intendico.com/gorite/.