Presented at WOFIEE'13, Athens, Greece, 16-17th July 2013

Towards Adjustable Autonomy in Adaptive Course Sequencing

Abdulkareem ALZAHRANI [a,1] , Vic CALLAGHAN [a] and Michael GARDNER [a]
[a] *Department of Computer Science, University of Essex, UK.*

Abstract. This paper presents adjustable autonomy and its application in adaptive course sequencing. The research aims to explore an adaptive course sequencing system (ACSS) which adapts the sequence of learning objects based on the student's profile and learning behaviour using soft computing techniques. The main contribution of this research is to make ACSS user driven by equipping it with adjustable autonomy mechanisms which allows student, tutor or some automated process to set the desired level of autonomous guidance. In addition, the system enables the teacher to reedit the sequence/guidance rules. Hence, we believe that a student will learn better when he/she learn using such a system, a hypothesis the longer term work of this student will seek to confirm. This paper gives an overview of the research area and presents an initial conceptual model which describes the ACSS's units and the relationship between these units.

Keywords. Adaptive course sequencing, intelligent tutoring system, adaptive educational hypermedia system

Introduction

Sequencing learning objects in intelligent tutoring system is one of a teacher's or instructor's responsibilities. Hence, he/she has to provide the domain model together with rules needed by the tutoring model to guide a student through a sequence of learning objects. However, this method is not sufficiently adaptive and personalized for individual students.

Therefore, we are proposing a new approach which we have named the Adaptive Course Sequencing System (ACSS). ACSS neither provides a fixed sequence of learning objects nor requires the teacher to dictate the rules for the guidance. Instead, it uses soft computing techniques to build an adaptive and dynamic sequence. This is done by offering number of learning objects and allowing the student to choose the learning objects he/she prefers to learn. ACSS then observes the student's learning behaviour and builds a profile for the student. The profile is then analyzed and informs a personalised tutoring agent so it can act as a sequence learning guide, suggesting the most appropriate learning path through a palette of learning objects.

By achieving that, we believe that students, particularly mature learners will learn better. However, as in real life, learners require differing levels of support a need we address by providing an adjustable autonomy tutor (i.e. an agent that can have the amount of help it provides adjustable). The amount of help a learner needs depends on

[1] Corresponding Author: Abdulkareem Alzahrani, School of Computer Science & Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK; E-mail: aaalza@essex.ac.uk

numerous factors such as their previous knowledge of the subject, the quality and teaching style of the current learning object and even how the student feels on that particular day. Therefore, equipping intelligent tutoring system with the adjustable autonomy mechanisms will make the system more adaptable and dynamic to the numerous variables that characterise the particular learning needs of a student. In addition, it will allow the learner or tutor to have the control and choose the preferred level of autonomy which in turns leads to learning better.

The main aim of this research is to devise an adaptive course sequencing system (ACSS) which learns students' needs and provides an adjustable level of support in guiding them through a set of learning object. This involves exploring the need for adjustable autonomy in ACSS and the ways of enabling adjustable autonomy in ACSS.

1. Related work

1.1. Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) are computer-based learning systems which attempt to adapt to the needs of learners [1]. It operates by facilitating ITS components (domain, student, tutoring, and user interface models) to interact so as to achieve various leaning and knowledge acquisitions[2]. In addition, ITS is the only part of the general IT and educational research which seeks, its scientific goals, to make computationally precise and explicit, forms of educational, psychological and social knowledge, which are more often left implicit" [1]. The following sections describe the ITS components (domain, student, tutoring, and user interface models) in more detail:

1.1.1. Domain model

This represents information about specific problems. It includes entities, definitions, processes, skills, and relations. In some problems it defines how experts should perform in a given domain, such as how to administer medications for disease [3], generate algebraic equations [4], or multiply numbers [5], to give but sum examples. There are three popular approaches for representing and reasoning domain knowledge which are; rule-based models, constraint-based models, and expert models.

1.1.2. Student Model

This represents the level of a student's conceptual understanding of a domain and describes how to reason about the student's knowledge of the domain. In its specific meaning, it contains information regarding a typical student's domain skill (stereotypic) and the information about current students. Examples of the latter include possible misconceptions about a domain, time spent on particular problem, the hints requested during learning, and the preferred presentation and learning style. [6].

1.1.3. Tutoring Model

This model comprises the series of decisions that should be implemented to identify a representation plus a descriptive assessment of the learning of knowledge, skills, and competencies. [8]

Woolf (2010) is of the view that tutoring models represents teaching strategies. This includes methods for encoding reasoning about the feedback from initial processing. Such encodings can be derived from empirical observation of teachers informed by learning theories, or enabled by technology, thus being only weakly related to a human analogue (simulations, animated characters).

1.1.4. Communication model

This model represents methods adopted for communicating between the students and computers. Examples of devices and processes used in communication models are graphical I/O interfaces, animated agents, and other dialogue mechanisms. Typical communication includes graphical illustrations, managing communication, and discussing student reasoning [2].

1.2. Adjustable Autonomy

Adjustable autonomy emerges from the underlying concept of agent autonomy which centres on building sets of actions, the relationship between the actions, respective scopes of each action, and the associated logical constraints governing the actions [9].

As identified by Bradshaw et-al. [9], a major challenge in adjustable autonomy is the requirement that the degree of autonomy is continuously and transparently consistent with declared policies ideally imposed and removed appropriately as desired. Bradshaw et-al also introduced the concept of a "sweet spot". According to this concept, an autonomous system is governed in a way that balances the convenience to delegate work with the assurance that delegating such work to a trusted system results in a minimum risk of failure.

Recently, a project [10], [11] at the University of Essex designed and built a smart home adjustable autonomy system which sought to collaborate with users to manage common tasks in homes.

In this project, Ball and Callaghan [10], [14] categorize four level of adjustable autonomy as being, full, high, low, and NO autonomy. In 'full autonomy' the agent manages the environment without user assistance. To do this it monitors the user's habitual behaviour creating rules (learning) that it uses to pre-emotively set the environment to a state it believes will match the user's needs . High-Semi-autonomous offer a mix of user and agent control which is commonly termed mixed initiative interaction. It is similar to fully-autonomous operation except that it requires the user to confirm automatically generated rules. This way, users can accept, reject, or edit the rules generated by the agent. Low- semi-autonomous uses the same mixed initiative interaction but uses an approach whereby the user offered suggestions by agent as an aid to generating behavioural rules. Finally, 'no autonomy' is similar to low autonomy with the added difference that, the user generates the rules without any assistance from an agent. The Essex work in rooted in Chin's end user programming research [13] and Hagras' fuzzy logic agents [12].

Various survey results indicate the usefulness of adjustable autonomy management system for intelligent environments. Differences in styles, user preference, trust, and dynamic control of autonomy level constitute the major findings as reported by many researchers [14]. A typical finding in these surveys is that people prefer higher level of control over personal systems such as entertainment and lower level of control over systems such as heating and lighting that are not associated with a particular user experience and perception. We hypothesis that this desire for personal control will extend to personalised learning.

2. Fictional Discussion to Illustrate Rationale

"*Smith is a University lecturer and he teaches a Software Engineering Module. Smith has been asked to use an online educational system to be used by students as a support tool, accessing the system in their own time. However, Smith said he had a bad experience with an online educational system. He said it had taken a long time to create the learning design scheme, particularly the guidance rules that are used for sequencing learning objects, which added greatly to the load on him. Therefore, he asked if there is an alternative way to determine and program the sequence of the learning objects. In addition, Smith said that he asked some students about their preferences regarding the online educational system. He found that some students asked more freedom in terms of choosing which learning objects would be studied. Those students argued that some learning objects in the system might be known before and if they are forced to study them again this would negatively affected their motivation towards the module. In other words, those students said they wanted the sequence of learning objects to be more flexible to allow them study only the learning objects they needed. Other students added that if the system could recommend the most appropriate learning path they should follow, that would improve their learning experience*."

From that story, we can advise Smith and his students to use the Adaptive Course Sequencing System (ACSS). ACSS is a web-based system which is capable of profiling its users and suggesting appropriate learning path based on their profile. The teacher is not required to provide ACSS with the sequence of the learning objects and their conditions, which are used for the guidance. Alternatively, ACSS uses the soft computing techniques to predict the most appropriate learning path for every student based on his/her profile.

ACSS has node-based interface which allows students to choose the learning objects they would like to learn. In addition, every node in this interface represents one learning object and contains introduction, explanations and multiple-answer questions (i.e. formative assessment). The student can read a node's introduction by hovering over the node with a mouse. Every node has one of three colours (blue: the student hasn't opened this node yet, green: the student has achieved the relevant learning objective and Red: means the student could not answer the questions in this node).

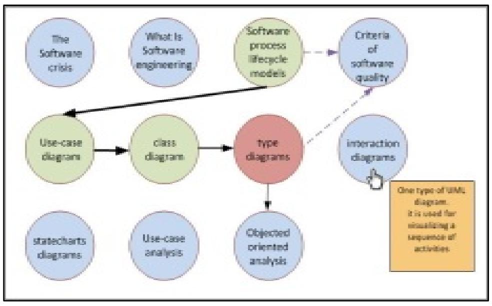

Figure 1. The node-based interface in ACSS

Every interaction between student and ACSS is recorded in student profile as well as some information about the student (e.g. his/her gender, age and level of knowledge …etc.). Once enough information is gathered about a student, ACSS, using soft computing techniques, analyses student profiles and observes the patterns of the sequences that students have made in order to generate guidance rules. Subsequently, ACSS uses these rules to build a tutor model.

The tutor model contains sets of rules which are generated and refined over the time by reasoning on student profiles. Each rule, when created, is assigned a value between 0 and 1 as a level of confidence. The value of the confidence level increases over the time based on who the repetition of the actions that make the rule. Rules that have high confidence level are stored as 'Active Rules' whereas other rules are stored as 'Potential Rules'. Both sets are part of the tutor model.

On the other hand, ACSS is trained by teaching a number of students *(software engineering)* and each lesson will have a number of learning objects with no sequencing between them. After the ACSS is trained sufficiently, it has got affective student profiles and tutor model. Hence, ACSS can present appropriate learning paths for students' based on their profile and experiences of other similar students. In addition, the most appropriate learning path for the student will be presented in bold solid black colour line and the alternative path will be presented in dashed grey colour.

2.1. The need of adjustable autonomy in ACSS

2.1.1. An Illustrative Scenario

"Sarah, Jane and David are learning using ACSS and they meet their teacher (Smith) to discuss the benefit of this system. Sarah said, she does not like the node-based interface (module map) in ACSS. She said that she is hesitant and the presentation of the nodes and the learning paths made her confused which, in-turn, affected her learning progress. She selected the option whereby ACSS would choose the most appropriate learning path for her and force her to follow it without giving her the ability to skip any learning object. In contrast, Jane said that the presentation of the learning paths ACSS performs (as shown in ***Figure 1****) is absolutely convenient for her and she loves it. David said he likes the freedom he is given by ACSS for jumping between learning objects but he does not want any kind of recommendation or guidance from ACSS. In other words, he asked to keep the presentation of the learning objects' nodes as it is shown in* ***Figure 1*** *but without it making suggestions as to his learning path.*

Smith (the teacher) asked if there is a way allowing him to access the tutor model to dictate and edit the rules.”

From this story, it can be said that the best way to make ACSS user-driven is by implementing the mechanisms of Adjustable Autonomy. By doing this, every student can choose from various levels of autonomy as described in this table:

Table 1. The levels of adjustable autonomy in ACSS

Autonomy Level	Description
Full autonomy (3)	The tutor model agent takes guidance responsibility and guides the student from learning object to another without using the node-based interface.
Partial autonomy (2)	Students can choose their own sequence from the module map. In addition, ACSS presents the appropriate learning paths and recommends one of them.
No autonomy (1)	Students can choose their own sequence from the module map. The ACSS will not recommend the best learning path.
Instructor Programmed	The teacher can access the tutor model and dictate new rules or edit the existent rules.

After adjusting the autonomy of the tutor agent in ACSS, Sarah is then forced by ACSS to follow the most appropriate learning path for her. ACSS guides her from learning object to another depend on her profile. In addition, Sarah does not need to deal with the node-based interface.

For Jane, who prefers the partial autonomy level, ACSS will continue acting as before and she will not recognize any change.

For David who prefers the no autonomy level, the ACSS does not present the recommended learning paths. He is free to choose the learning objects he would like to learn from the module map (as before).

Smith (the teacher), can then access the tutor model and write or edit the rules. In addition, he can reset the level of confidence for any rules.

3. ACSS Conceptual Model

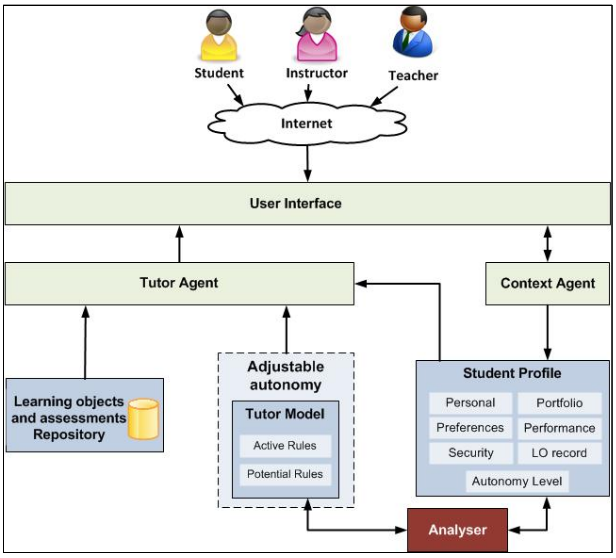

Figure 2. ACSS Conceptual Model

Using the earlier scenario, we have proposed this conceptual model as a way of delivering the required functionality. From the student's perspective, this model is able to give students' the freedom to choose a path through learning objects as well as recommend appropriate learning paths. In addition, it gives the student the ability to choose the guidance method (e.g. enforce, recommend or no guidance). On the other hand, the teacher/instructor can set the guidance rules simply and the learning objects as well.

The model is divided into the following main parts: communication, user interface, tutor-agent, context-agent, Analyzer, LO and assessments repository, student profile, the tutor model and adjustable autonomy mechanism which are briefly described in the following sections.

3.1. The communication

ACSS is a web-based application. Therefore, the learner uses the internet network to access ACSS.

3.2. User Interface

ACSS has a node-based interface. That means every learning object will be presented as a node, as previously described in the scenario. This interface will be presented to a student who chooses one of these levels of autonomy namely: No autonomy (1) or partial autonomy (2) or full-autonomy. However, the difference between these levels in terms of the interface is that the module map will be presented for student who chooses the partial autonomy level. In addition, a student who chooses the full autonomy level will simply be presented with learning objects to complete (not the node-based interface or the module map). Technically, the development of the node based and module map interface is being developed using Python and JavaScript (D3.js).

The teachers/instructors have a different user interface which allows them to access the tutor model and set the learning sequence rules manually.

3.3. Tutor Agent

This agent manages the entire process. When a student starts using the ACSS, the tutor agent will request student profile for this student. If there is no information about the student's knowledge level, the tutor agent will generate a formative assessment (taken from the assessment repository) and ask the student to undertake it, storing the result in the student profile. However, if the student has already done the formative assessment, the tutor-agent will request some relevant information from student profile (i.e. student's preferences and performance) which is uses to request the related rules from the tutor model. In addition, the tutor-agent takes into consideration the chosen autonomy level and, based on that, the tutor-agent will chose the guidance method as described in section **Error! Reference source not found.**.

When the student finishes studying a learning object, the tutor-agent will request the relevant assessment from the repository to test the student and then store the marks in the student profile.

3.4. Context Agent

This agent is responsible for observing student learning behaviour and store the information in the student profile.

3.5. Analyzer

This is the main component that provides the ACSS with intelligent behaviour based on the use of fuzzy logic to generate rules which are subsequently analysed by a reasoning engine operating on the student profiles.

3.6. Learning Objects and Assessments Repository

This component is responsible for storing learning objects with their information. This information includes the title, description, main content (i.e. explanations) and multiple-answer questions (i.e. formative assessment). In addition, the teacher can access the repository to add, delete or edit any learning object.

3.7. Student Profile

This component uses a database to store the student's personal information (e.g. name, age and email…etc.) and the student's learning information (e.g. completed learning tasks and time on tasks….etc.). In addition, this component follows, partially, the enhancement of IEEE-PAPI specification that was made by Wei and Yan [15]. Thus, this component has these categories:

- Personal information: <name, telephone, address, reference, e-mail, post address>
- Portfolio information: <knowledge level, degree, transcription, qualifications, certificates, licenses>.
- Security information: <user name, password>
- Preference information: <language, region, age level, input/output device preference, content preference, prefer time on each study>.
- Performance information: <learner ID, content ID, recoding-date-time (time begin, time end), complete percentage, score>.
- Learning objects record <learning object ID >.
- Autonomy level: <level of autonomy>.

3.8. Tutor Model

The tutor model contains rules to be used by the tutor-agent. In addition, the rules in tutor model are classified into two components; Active rules component and potential rules component.

The active rules component stores the rules that have high confidence level (initially more than 0.75 of 1.00). Whereas, the potential rules component stores the low level confidence rules.

These rules will be readable (thanks to fuzzy logic rule sets). Below is an initial view of the active rules component.

Table 2. Example of the active rule component

Rule	Confidence Level
If the previous domain knowledge is poor AND the learning object A mark is good AND the learning object C mark is bad THEN next learning object is B.	0.78

3.9. Adjustable Autonomy

This component offers three choices of autonomy for student and gives the teacher the ability to add, delete and edit the rules in the tutor model (as explained in section **Error! Reference source not found.**2.1). It can give suggestions for rules to aid the teacher.

4. Experimental work to date

In this section, we summarise the work to date.

4.1. Learning Design (LD)

Using LAMS, we have structured the evaluation scenario for the learning phase. Firstly, the student will be forced by the system to do a formative assessment. Then, the student will be able to choose one of the learning objects. Every learning object is designed to have an introduction, explanation and multiple questions to assess the student's performance in that learning object. By the end of this process, the student will have been completed an assessment and survey which, in addition to supporting the system operation, will provide an experimental record and data.

Figure 3. Learning design first lesson using LAMS

4.2. Student profile

Using postgreSQL we have designed and built a student profile scheme. As mentioned earlier in section 3.73.7 the student profile will follow IEEE PAPI. **Figure 4** shows the seven tables with the relations between them, that comprises this profile.

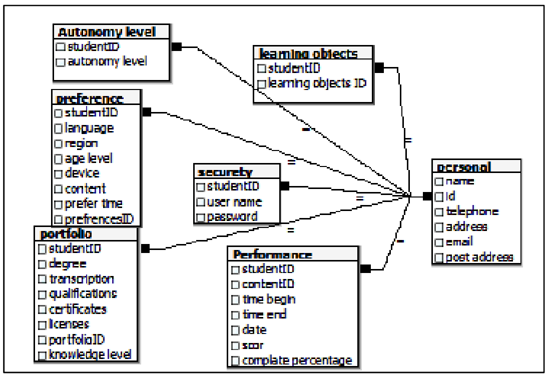

Figure 4. The initialized student profile

5. Future work

We are now entering the experimental phases of the research. The first phase will investigate the students' learning needs and preferences regarding adaptive course sequencing in general and adjustable autonomy in particular. Therefore, a questionnaire will be distributed and analyzed to partially contribute in achieving these research objectives; 1) *understanding the student's learning needs and preferences, 2) studying the need for adjustable autonomy in adaptive course sequencing*.

The second phase will involve exploring the chosen topic (*Software Engineering*) might be implemented as a practical test of ACSS. This wills involve choosing number of lessons, constructing the related learning objects and constructing the formative and summative assessments from those lessons.

In the third phase, a web-based adaptive course sequencing will be built. This will involve mapping the subject space to learning objects, initializing student profile, developing the observation mechanisms to observe student doing learning, building tutor model and implementing the soft computing techniques needed for reasoning student profile to enrich the tutor model. By doing this phase, the question *"How do we encode the learning experiences in fuzzy rules?"* will be answered.

The fourth phase, will refine and implement the conceptual architecture model for the adjustable autonomy as part of a broader intelligent educational environment (AAIEE). In addition, this phase will involve defining how adjustable autonomy can be designed and what the steps are needed for this design in educational context. In completing this phase we aim to answer the question "*How best can adjustable autonomy be enabled in adaptive course sequencing system)*. Moreover, we will also answer the question *(Does adjustable autonomy intelligent tutoring system brings significant benefits to student learning?).*

The fifth phase is the training phase which involves teaching a number of student's lessons *(in software engineering)* with each lesson having a number of learning objects. By doing this phase, we will answer the question "how much training is required".

The sixth phase is the evaluation phase which involves teaching students the same lessons and assessing their behaviour and progress. This phase consists: 1) Equipping the system with adjustable autonomy mechanisms; 2) Evaluating the guidance in the system by giving the students summative assessments after they finished every lesson (and comparing their marks with students studying similar lessons but in fixed sequencing method); 3) Using interviews to provide a qualitative assessment of student satisfaction towards the adjustable autonomy together with a more quantitative comparison of students' marks from every level of autonomy. Another area of evaluation is will be to analyze the usage of ACSS based on the data gathered from the behaviour of the adjustable autonomy. By doing these phases, we will be able to answer the question "*What are the benefits of adjustable autonomy and adaptive course sequencing to students"*. Moreover, we hope to answer many more research questions as part of this work.

6. Conclusion

This paper describes early stage research that is setting out to investigate how adjustable autonomy might be applied to sequencing learning objects to empower students to personalise better their learning experiences. Our research objects concern both the technical possibilities as well as the learning benefits, so we will be evaluating this work using real students and lessons. The work itself is an early stage and we believe that the main contributions of this paper are a through literature the rational for applying adjust able autonomy to education and, most importantly, a conceptual model (the ACSS) for how this might be implemented. The proposed model implements the adjustable autonomous agent as part of an intelligent pedagogical tutoring system. The level of guidance can be adjusted autonomously to cater for the need of the pupil. The adjustment can be carried out by the teacher, the pupil, or the machine based preference of educational stakeholder's policies (E.g. curricula) and the pedagogical needs of the target pupil. This type of intelligent environment is important as it closely reflects the actual teaching environment where the teacher alters the level of help or guidance a pupil receives based on their capabilities and immediate needs. Adding a more naturalistic view to alter the autonomy provided by such system will, we hope, have a positive impact on learning systems. We suggest that this adds a more dynamic adaptability to supporting the role of teachers' and needs and progressions of learners using online systems.

References

[1] J. Self, "The defining characteristics of intelligent tutoring systems research: ITSs care, precisely," *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 10, pp. 350–364, 1998.

[2] B. . Woolf, *Building Intelligent Interactive Tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann, 2010.

[3] C. R. Eliot, K. A. Williams, and B. P. Woolf, "An intelligent learning environment for advanced cardiac life support.," in *Proceedings of the AMIA annual fall symposium*, 1996, p. 7.

[4] A. Corbett, "Cognitive tutor algebra I: Adaptive student modeling in widespread classroom use," in *Technology and Assessment: Thinking Ahead. Proceedings from a Workshop. National Academy Press, Washington*, 2002.

[5] I. Arroyo, J. E. Beck, B. P. Woolf, C. R. Beal, and K. Schultz, "Macroadapting Animalwatch to Gender and Cognitive Differences with Respect to Hint Interactivity and Symbolism," in in *Intelligent Tutoring Systems*, G. Gauthier, C. Frasson, and K. VanLehn, Eds. Springer Berlin Heidelberg, 2000, pp. 574–583.

[6] Y. Zhou and M. W. Evens, "A practical student model in an intelligent tutoring system," in *11th IEEE International Conference on Tools with Artificial Intelligence, 1999. Proceedings*, 1999, pp. 13–18.

[7] A. S. Drigas, K. Argyri, and J. Vrettaros, "Decade Review (1999-2009): Artificial Intelligence Techniques in Student Modeling," in in *Best Practices for the Knowledge Society. Knowledge, Learning, Development and Technology for All*, M. D. Lytras, P. O. de Pablos, E. Damiani, D. Avison, A. Naeve, and D. G. Horner, Eds. Springer Berlin Heidelberg, 2009, pp. 552–564.

[8] J. Bourdeau and M. Grandbastien, "Modeling Tutoring Knowledge," in in *Advances in Intelligent Tutoring Systems*, R. Nkambou, J. Bourdeau, and R. Mizoguchi, Eds. Springer Berlin Heidelberg, 2010, pp. 123–143.

[9] J. Bradshaw, M. Sierhuis, A. Acquisti, P. Feltovich, R. Hoffman, R. Jeffers, D. Prescott, N. Suri, A. Uszok, and R. V. Hoof, "Adjustable Autonomy and Human-Agent Teamwork in Practice: An Interim Report on Space Applications," in in *Agent Autonomy*, H. Hexmoor, C. Castelfranchi, and R. Falcone, Eds. Springer US, 2003, pp. 243–280.

[10] M. Ball and V. Callaghan, *Managing Control, Convenience and Autonomy*. Special Issue on Agent-Based Approaches to Ambient Intelligence, AISE book series, IOS Press, 2012.

[11] M. Ball, V. Callaghan, and M. Gardner, "An Adjustable-Autonomy Agent for Intelligent Environments," in *2010 Sixth International Conference on Intelligent Environments (IE)*, 2010, pp. 1 – 6.

[12] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman, "Creating an ambient-intelligence environment using embedded agents," *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 12 – 20, Dec. 2004.

[13] J. Chin, V. Callaghan, G. Clarke, "End-user Customisation of Intelligent Environments". In the handbook of Ambient Intelligence and Smart Environments, Editors: H. Nakashima, J. Augusto, H. Aghajan, Springer, 2010, Spring, pp 371-407, ISBN 978-0-387-93807-3.

[14] M. Ball and V. Callaghan, "Introducing Intelligent Environments, Agents and Autonomy to Users," in *2011 7th International Conference on Intelligent Environments (IE)*, 2011, pp. 382 –385.

[15] X. Wei and J. Yan, "Learner Profile Design for Personalized E-Learning Systems," in *International Conference on Computational Intelligence and Software Engineering, 2009. CiSE 2009*, 2009, pp. 1–4.