# A design environment to articulate design intention of learning contents

Yusuke Hayashi* and Mitsuru Ikeda

School of Knowledge Science,
Japan Advanced Institute of Science and Technology,
1-1, Asahidai, Tatunokuchi, Nomi, Ishikawa, 9231292, Japan
E-mail: yusuke@jaist.ac.jp E-mail: ikeda@jaist.ac.jp
*Corresponding author

Riichiro Mizoguchi

The Institute of Scientific and Industrial Research,
Osaka University, 8-1, Mihogaoka, Ibaraki, Osaka, 5670047, Japan
E-mail: miz@ei.sanken.osaka-u.ac.jp

**Abstract:** Design of learning contents is more difficult to support than design of a physical substance because its object is abstract and it is difficult to establish a framework to express it appropriately. To address this issue, we converted an abstract concept about learning content design to an ontology. Then, on the basis of that ontology, we developed *i*Designer, a design support environment. *i*Designer realises the following two points:

- the designers are compelled to make the implicit results of their work explicit in order to deepen thinking about the design of an abstract matter
- the designers are provided with basic information to verify content validity by simulating the change of understanding of a learner in the learning contents model at a conceptual level, which is the intermediate result of design.

**Keywords:** learning contents; learning contents design environment; ontology-awareness.

**Reference** to this paper should be made as follows: Hayashi, Y., Ikeda, M. and Mizoguchi, R. (2004) 'A design environment to articulate design intention of learning contents', *Int. J. Cont. Engineering Education and Lifelong Learning*, Vol. 14, No. 3, pp.276–296.

**Biographical notes:** Yusuke Hayashi is a Research Associate at the School of Knowledge Science, Graduate School of Engineering Science, Japan Advanced Institute of Science and Technology (JAIST). He received his PhD degree from Osaka University, Osaka, Japan, in 2003. He has been engaged in research on Knowledge modelling, Ontological engineering, Intelligent learning support systems and Knowledge management. Dr. Hayashi is a member of the Japanese Society for Artificial Intelligence (JSAI), Information Processing Society of Japan (IPSJ) and Japan Society for Educational Technology (JSET), the International Society for AI in Education (IAIED), and Asia-Pacific Society for Computers in Education (APSCE).

Mitsuru Ikeda is a Professor at the School of Knowledge Science, Graduate School of Engineering Science, Japan Advanced Institute of Science and Technology (JAIST). He received his PhD degree from Osaka University in 1989. From 1989 to 1991 he was a Research Associate in Utsunomiya University, and from 1991 to 1997 – in ISIR, Osaka University. From 1997 to 2003 he was a Research Associate Professor in ISIR, Osaka University. His research interests include Knowledge-based systems, Ontological engineering, Learning support systems, and Knowledge management support systems. Dr Ikeda is a member of the Japanese Society for Artificial Intelligence (JSAI), the Institute of Electronics, Information and Communication Engineers, the Information Processing Society of Japan, the International Society for AI in Education (IAIED), AAAI, IEEE and Asia-Pacific Society for Computers in Education (APSCE). He received honorable mention for 10th Anniversary Paper Award from JSAI and Best paper Award of ICCE in 1996 and 1999, respectively.

Riichiro Mizoguchi is a Professor at the Department of Knowledge Systems, Institute of Scientific and Industrial Research (ISIR), Osaka University. He received his PhD degree from Osaka University in 1977. From 1978 to 1986 he was a Research Associate in ISIR, Osaka University. From 1986 to 1989 he was Associate Professor there. His research interests include Non-parametric data analyses, Knowledge-based systems, Ontological engineering and Intelligent learning support systems. Dr Mizoguchi is a member of the Japanese Society for Artificial Intelligence (JSAI), the Institute of Electronics, Information and Communication Engineers, the Information Processing Society of Japan, the International Society for AI in Education (IAIED), AAAI, IEEE and Asia-Pacific Society for Computers in Education (APSCE). He was President of the IAIED Society and APC of AACE. Currently, he is Editor-in-chief of JSAI. He received honorable mention for the Pattern Recognition Society Award, the Institute of Electronics, Information and Communication Engineers Award, 10th Anniversary Paper Award from JSAI and Best paper Award of ICCE99 in 1985, 1988, 1996 and 1999, respectively.

## 1 Introduction

A learning content designer designs a learning process carried out through using learning contents. First, the designer assumes learners' knowledge states and capacities. Next, considering a learning goal, he draws an ideal learning process for the learner to achieve it. Finally, he embodies that learning process as learning contents. We conceptualised this process as comprising the following four subtasks:

- *learning needs analysis* specifies what kind of learners should learn which subject
- *learning process design* constitutes a learning process that fulfills the learning needs
- *learning content production* arranges learning objects (learning materials) in a control structure (such as courseware), and embodies them as learning contents
- *learning object production* expresses learning matter using media.

Many commercial authoring tools, a salient example being Authorware [1], are designed around a media-editing function useful for production of learning materials and learning content. Typically, it makes use of digital media. *Learning needs analysis* and *learning process design*, however, are excluded from the current work on authoring tools development; and only auxiliary functions, such as a simple memorandum function, are provided. Murray [2] reports that decomposing and extracting knowledge for realising education appropriately is crucial for creating a system, which employs learning content intelligently. In this relation the importance of support for learning content design increases.

Among the principal reasons why there is no much work on tools to support *learning needs analysis* and *learning process design* are that the object of design is abstract and that the framework to express it appropriately is beyond imagination. That is, abstractness of the substance and difficulty in symbolisation of its expression prevent smooth support for designing the action of learning in contrast to designing physical substances. This paper discusses the role of ontology in the design of learning contents.

In the study reported here, the design and development of a learning content design environment were carried out by considering the *learning process design* and *learning contents production* as principal objects. Especially, we sought realisation of a tool equipped with the ability to support a user's intellectual work by understanding ontology for *learning process design* (ontology awareness). A study on learning needs analysis is in progress in a separate project [3]; both studies will be unified in the future. Moreover, for *learning object production*, learning objects created with conventional media editing tools can be imported to the design environment in standard formats such as HTML and PDF.

The following two types of support are considered as main objectives to realise an ontology-aware design environment:

- modelling a designer's intention: encouraging designers to model their intentions while designing; those intentions are recorded along with learning contents
- verification of the validity of a designer's intention: encouraging designers to confirm that the designers' intentions are appropriately reflected in learning contents; if needed, those contents are corrected.

In a previous study the authors have developed SmartTrainer/AT, an authoring tool for designing a training system for recovering from faults in electrical systems [4]. In that study, an ontology-aware design environment was developed for a specific object- a training system for recovering from electrical system faults. In this study, the development of *i*Designer, a learning content design support environment, proceeded as a rather general-purpose framework with knowledge obtained in the previous development as a starting point.

The following three points are considered to be main objectives in the design and development of *i*Designer (compared to SmartTrainer/AT):

a *Generalisation of a learning contents design task ontology.* While SmartTrainer/AT was intended for construction of an authoring support tool for specific tasks based on detailed needs, this study develops an ontology built independently of an object on the general conceptual levels of education and training; a general-purpose design environment is developed on that foundation.

b *Coordinated support of a structuring process and a sequencing process.* SmartTrainer/AT realised support of a sequencing process that designs the control structure of the learning content. This study introduces a structuring process for building a learning item network that is a systematic model of learning matters, in addition to the sequencing process that is a process in which a designer arranges learning objects in advance of the systematic process. Then, a sequencing support function is developed that refers to the learning item network. This support function is the result of a structuring process.

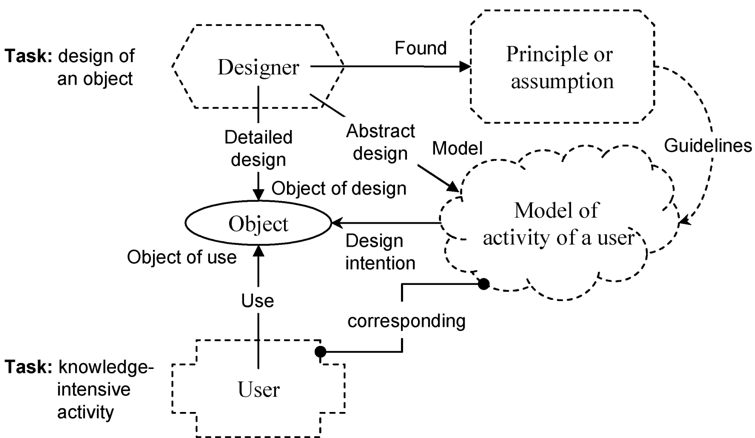
c *Introduction of conceptual level simulation.* In order to verify and visualise learning contents' validity, a conceptual level simulation is newly introduced as a function to simulate learning progress on the conceptual level.

Hereinafter, Section 2 discusses the general scheme of the learning content design task and describes the role of ontology in the scheme. Section 3 describes an ontology built in this study that aims at generalisation of the learning contents design task (see Objective (a)). Section 4 describes a learning content design support environment, *i*Designer, and its support functions implementing (b) and (c).

## 2 A model of learning contents design

Figure 1 shows a general diagram of the design of knowledge-intensive activities. This diagram illustrates two types of persons: a designer and a user. A user is the subject of knowledge-intensive activities, while the designer bears the role of providing the object that supports a user. Herein, the term 'object' means both the object supporting knowledge-intensive activities of a user and the object of design. The 'object' can be one of a range of concrete objects, such as an IT tool that a user uses, or abstract objects, such as the goal and background of activities or the role of a person in a group activity. A diagram on the left-hand side of Figure 1 illustrates how a user carries out knowledge-intensive activities using an object, which the designer has designed. The designer assumes a principle or assumption, which are requisite for a design. He designs an object, assuming the model of the knowledge-intensive activities of the user.

**Figure 1** Design of knowledge-intensive activity (the general form)



Two design processes represent a designer's activity in Figure 1:

a *abstract design:* modelling the user's activities while using the object, according to the assumption or principle of design.

b *detailed design:* specifying the object that supports the user's activities based on the model.

This study sets one of its objectives as supporting the rationality of the relation of the above-mentioned two design actions. Generally speaking, the importance of aspect (a) is recognised widely. For example, in the object-oriented design methodology in the software engineering field, the significance of modelling in the object world corresponding to (a) is recognised strongly as the basis for verification of a software specification. Furthermore, various modelling techniques for rationalising design are devised and have been accepted widely [5].

The specification of (a) is a more difficult issue when the support of a knowledge-intensive activity is an object of design. Such an object is abstract and difficult to model. It is rare that the assumption or the principle behind the model is specified, while it is often the case that an object is designed implicitly and provided to the user. As a result, the following cases may occur: a designer may not be able to define their intention in designing an object as a model because the premise is not clear; the embodiment is inadequate even if the designer himself can define a model; or the design intention is not conveyed to a user distinctly, so the intended function is not realised and the expected result is not obtained.

The following two design guidelines are set in this study to resolve such issues:

- a design intention should be defined so that a designer may investigate the design rationality
- a user should be able to act appropriately using the function of an object according to the designer's intention.

In this study, the role of an ontology [6,7] as a basis for showing the premise of a design is considered essential to embody these guidelines.

### *2.1 Modelling and design based on ontology*

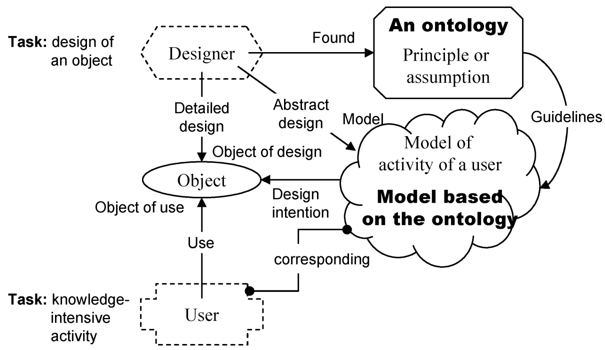
This section describes ontology-based modelling and design with special attention to two ontology roles:

- the revelation of implicit information
- the meta-model function.

The first role is related to specifying premises, such as assumptions and principles, as well as a model as a design intention (by describing the model considering the ontology as guidelines). In the second role the ontology provides design guidelines, i.e. is a rule which maintains the viewpoint, consistency and coherence in modelling the object. Figure 2 shows that 'principle or assumption' and 'model of knowledge-intensive activities' (from Figure 1) are represented as 'an ontology' as a rule and 'a model based on the ontology', respectively.

This fulfills the requirement of sharing the premise of the design and the design intention for an object, which were previously implicit, between designers or a designer and a user.

**Figure 2** Articulation of design intention as a model

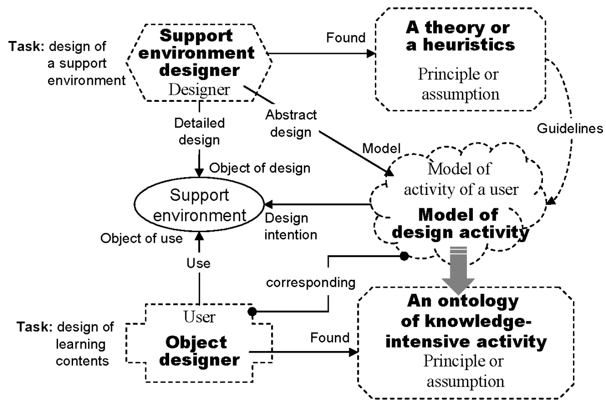



## 2.2 Design of support environment

This study aims at building a support environment for a designer of learning contents by considering an ontology as guidelines as outlined in the previous section. That is, we aim at both providing an ontology as design guidelines to a designer and providing a support environment for expressing a model based on it.

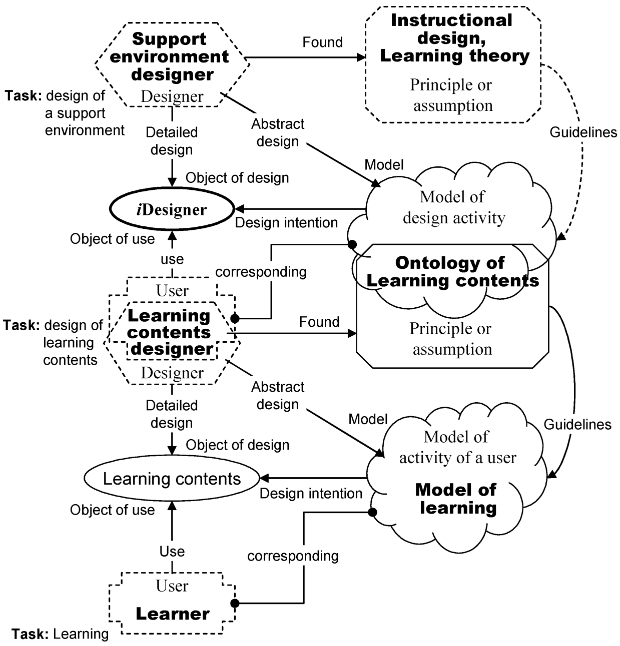
The design of a support environment can also be conceptualised according to the general diagram shown in Figure 1. That conceptualisation is shown in Figure 3. The object to design here is a support environment; the designer of this support environment is shown in the upper left part, and a designer of learning contents, who is the user of the designed support environment, is shown in the lower left in this figure. The position of the authors of this paper corresponds to the support environment designer. A support environment designer models the design activity that a user carries out with a support environment as shown in Figure 1. Sharing the premise of the support environment, specified as an ontology, by the environment and its users enables understanding of the context of the design activity of a user within the limits of the ontology, and offering more suitable support. The authors refer to the property and function of a system to offer more intelligent support by sharing a context of the activity with users based on an ontology as 'ontology awareness', and to such a system as 'an ontology aware system' [8].

**Figure 3** Design of a support environment



The design and development of *i*Designer, a learning content design environment, were carried out on the basis of the diagram given in Figure 3. Figure 4 presents both the design of learning contents and of a design support environment by unifying the user of the design environment and the designer of learning contents into one agent. As mentioned above, the left-hand side row shows the relationship between a designer, a design object, and a user. The support environment designer at the top corresponds to the authors of this paper; a support environment designer designs an environment to support a learning content designer. The learning content designer structures learning contents to support a learner by using *i*Designer. The authors designed and developed *i*Designer after considering the relationship between the modelling and design objects of both the environment designer and the learning contents designer. The ontology of learning contents built in this study has the role of indicating the design intention of *i*Designer and the role of clarifying the premise at designing learning contents. This is intended to raise the consistency of conceptual recognition surrounding learning activities of a support environment designer and a learning contents designer. This study concentrates on the *object* of design activity that will be described in the following section. It lacks a model of *process* of the design activity to create learning contents. Aroyo et al.'s work on the course authoring ontology [9] approaches an ontology and model to articulate the *process* of the design activity.

**Figure 4** Correspondence of learning contents design and the design of its support environment




## 3 Learning contents model

When we design and develop a tool that supports a user's knowledge-intensive activity, it is essential to clarify a conceptual system about the object of the activity. In our case, a conceptual system about learning content is needed for designing and developing a learning content design support environment. Inevitably, the conceptual system that the tool postulates is approximate. However, it is desirable to raise the approximation precision and to realise a user-friendly information processing function. This study adopts an ontology-engineering approach, as described in a in Section 1 [10,11].

### *3.1 Design task ontology of learning contents*

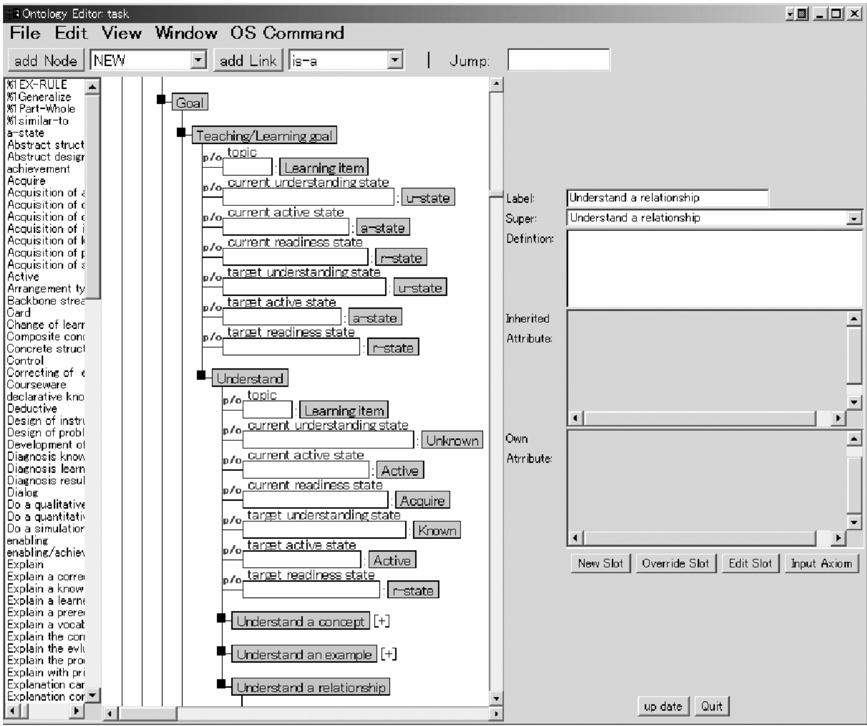
*i*Designer is a learning content design environment that is independent of specific learning matters. The ontology used as the basis of its data model is a task ontology, that is, a task conceptualisation of a learning content design comprehended on a level that is independent of a domain model. Hereinafter, we briefly describe the superordinate concept of ontology mounted in *i*Designer as a data model.

- *Learning item* represents a unit of learning, such as a knowledge item, skill, or competence.
- *Learning item link* represents a relationship, such as the prerequisite relationship between learning items and the generalising relationship.
- *Learning item network* represents learning matters with a network structure in which nodes of learning items are connected with learning item links.
- *Learning support system goal* is a concept showing the intention of an action of the system; it is classified into a teaching/learning goal and a classification goal.
- *Teaching/learning goal* represents a change of the learner's understanding through knowledge acquisition, skill acquisition, and competence development; the change in this case is what a designer expects from the teaching action of the system.
- *Classification goal* represents how the system categorises a learner's understanding.
- *Goal hierarchy* represents the hierarchical structure of educational intention implied in certain concrete learning contents. The structure details learning content goals gradually from a higher level to a lower level with *learning support system goals* as nodes.
- *Teaching action* represents system's behavior to a learner.
- *Learning content control structure* represents a structure that arranges learning objects, such as courseware.
- *Learning object* represents an electronic object shown to a learner.
- *Learner model* is a model showing a learner's understanding state.

A learning item works as an interface between a domain concept and a task concept. It bears the unique role of modelling learning matters. In *i*Designer, a learning item is embodied as data to store a pointer to a concept in a domain model.

Concepts and their relationships in this ontology are defined with the Ontology Editor (OE) [12] (see Figure 5). Sections 3.2 and 3.3 below describe abstract models of learning matters and a learning process based on this ontology, respectively.

**Figure 5** Ontology editor (OE)




## 3.2 Model of learning matters

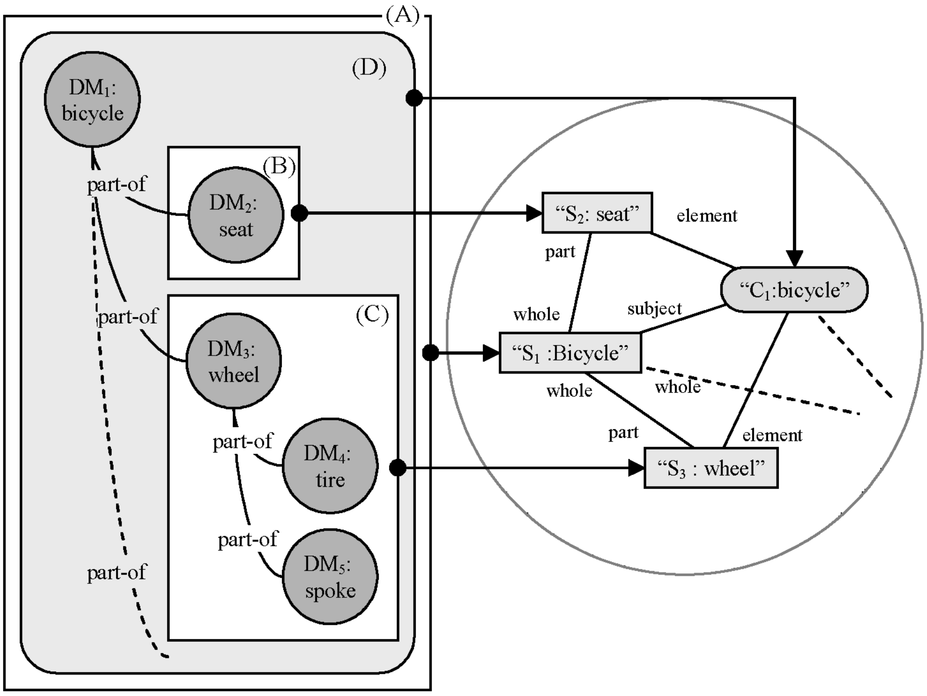
The model of learning matters reorganises a domain model in terms of learning. It is expressed as a learning item network in which a learning item, defined with consideration to granularity of knowledge and skills in terms of learning, comprises a node, and a relationship between learning items, such as a prerequisite relationship, is a learning item link. Table 1 summarises the construct of a learning item network.

**Table 1** Learning item and links

| | |
|---|---|
| *Learning item* | |
| Single learning item | Base unit of learning |
| Composite learning item | Aggregation of the related learning items in a context |
| *Learning item link* | |
| Generalised link | Relation between a generalised learning item and a specified one |
| Part-whole link | Relation between a whole learning item and a partial one |
| Prerequisite link | Order relation that should be followed in a sequencing |
| EGRUL link | Relation between an example and a rule |
| Similar-to link | Relation between similar learning items |
| Aggregation link | Relation between a composite learning item and the elements |

Figure 6 shows a domain model and the corresponding learning item network. The domain model illustrates the conceptual structure of a domain: e.g., ‘$DM_1$: bicycle’ has ‘$DM_2$: seat’ and ‘$DM_3$: wheel’ as parts. This is organised into a learning item network in terms of learning, e.g., the three domain concepts contained in area C are modelled as learning item $S_3$. $S_3$ is a ‘single’ learning item indicating the base unit of learning; it shows that the three domain concepts are recognised as one learning unit. This implies that domain concepts, such as a tire or a spoke, are not recognised as learning units in this model scheme. When a certain learning matter depends on other learning matters, a concept for expressing the learning of that dependence is referred to as a ‘composite’ learning item. For example, composite learning item $C_1$ in Figure 6 expresses a learning matter, which integrates learning items of the parts of a bicycle ($S_2$, $S_3$), for comprehension of a bicycle ($S_1$) based on their understanding. In this model scheme, two views are presumed for the learning process about a bicycle: the first, in which a learner progresses to the whole $S_1$ after learning parts in $S_2$ and $S_3$ through composite learning item $C_1$; and the second, in which a learner learns the concept contained in Figure 6(A) as single learning item $S_1$.

**Figure 6** Learning item network




A link between learning items expresses interdependence between those learning items, such as a prerequisite relationship in learning sequence. This link is used as reference information for learning process design.

## 3.3 Model of the learning process

A designer assumes a learning process satisfying a learning goal. Along with that learning process, a learning content control structure is assumed as an arrangement of learning objects. Learning content is a package of a learning content control structure and learning objects. This relationship is shown in Figure 7, which presents a learning content design model. The design model is classified into three levels: the conceptual level, the object level, and the delivery level. The goal hierarchy and a teaching/learning action structure are designed at the conceptual level. A learning content control structure is created at the object level. While the object level is substantiated as the final result of a design, the conceptual level is the intermediate result and an abstract of designer thought. Finally, the delivery level is defined as the learning content that operates in an execution environment. The model structure, mainly around the conceptual level, is described below.

**Figure 7** Model structure of a learning process

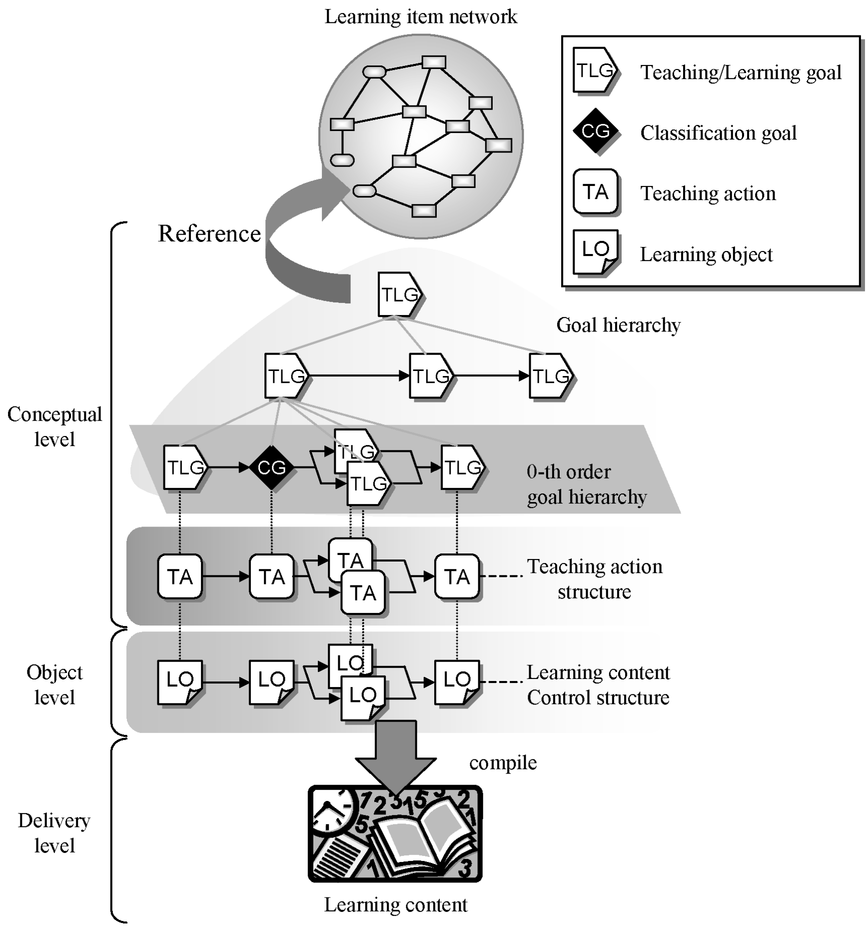


Figure 8 shows a part of the *is-a* hierarchy of the concept of a learning support system's goals. The nodes of the hierarchy are used for describing the educational intention of the system's actions. The goals are classified into two kinds: teaching/learning goals that anticipate a change of a learner's understanding condition, and classification goals that indicate the presumed understanding state of a learner. A teaching/learning goal is defined as an abstracted concept of 'an action that aims at the change of a learner's understanding about a learning item'. The knowledge state is defined in Table 2. A value is set for each of three types of assumed states: readiness, active state, and understanding. The description of a state change defines a change that each goal in Figure 8 directs. For example, 'to cause recognition' directs actions concerning readiness. On the other hand, a classification goal is defined as an abstracted concept of "the action that classifies a learner according to the knowledge state about a learning item". This corresponds to a branch on learning contents; the branch condition described here is a learner's state description that the designer assumes for the learning process of the branch destination. The authors referred to concepts, which appear in Gagne's five categories of learning outcomes [13] and Bloom's taxonomy of educational objectives [14], both widely known education theories, to set the description level of these goals.

**Figure 8** Concept hierarchy of goals (partial)

```
1.Learning support system goal
  2.-Teaching/learning goal
    3.-Recognize
      4.-Recognize an example
        5.-...
    3.-Recognize again
      4.-...
    3.-Understand
      4.-Understand an example
      4.-Understand a concept
      4.-Understand a relationship
        5.-...
    3.-Reject
      4.-...
    3.-Acquire
      4.-...
    3.-Master
      4.-...
    3.-Correct
      4.-...
  2.-Classification goal
    3.-Evaluate
      4.-...
    3.-Identify
      4.-...
```

"-" indicates "is-a" relation.

The number indicates the layer in the hierarchy.

**Table 2** Description of the knowledge state

| *Type* | *Explanation* | *Values* |
|---|---|---|
| Understanding state (u-state) | State of the learning item on long-term memory | Unknown\|Understood\|Acquired |
| Active state (a-state) | State of the learning item on short-term memory | Inactive\|Active |
| Readiness (r-state) | Learning activity for the learning item for which the learner is ready | To acquire\|To master\|To correct |

A goal hierarchy constituted to a certain concrete learning content expresses "what learning (effect) is expected for a learner in what state". A learning support system goal set at the top level is equivalent to learning needs, which the whole learning contents constituted at the bottom in Figure 3 should fill. The relation between an upper goal and a lower goal is a part-whole relationship. An upper goal is expanded to the sequence of multiple lower goals for achieving the upper goal. It is inferred that a reasonable sequence of goals gradually advances the state changes shown in Table 2. For example, a sequence to help a learner understand a rule by giving an example raises readiness to the rule by "letting the learner recognise the example of the rule for which the understanding state (*u-state*) is unknown". Consequently, the understanding state is transformed to 'understood' by 'letting the learner understand the rule'. The bottom of the goal hierarchy is called the hierarchy 0'th order goal.

A teaching action is a concept that connects learning objects to a learning support system goal: the learning object that a learner is shown to achieve the goal defines the roles that it plays in the teaching action (e.g. 'to explain' and 'to hint'). A learning content control structure is a model that arranges learning objects on control structures such as branch and iteration. The hierarchy 0'th order goal, a teaching action structure, and a learning content control structure, are structurally similar. The correspondence of these three defines the purpose and the type of action that a learning object embodies in the context of the overall learning content.

Thus, along with supporting design work, *i*Designer offers to designers a framework to express a design result maintaining continuity on two levels: the conceptual level and the object level.

## 4 *i*Designer: learning contents design support environment

In the development of *i*Designer, a fundamental data model and ontology awareness over the model are implemented based on the superordinate concept of ontology described in Section 3.1. For ontology awareness over a basic data model, the following specific functions are implemented with respect to goals b and c (see Section 1):

- providing a vocabulary to describe a conceptual level model
- providing a framework which expresses it visually
- carrying out a simulation with the conceptual level model
- presenting the behavior of learning contents visually and supporting its verification.

*i*Designer realises these as a capacity to interpret the content and process of a model of learning contents based on an ontology.

## *4.1 Design process*

The work, which a designer conducts on *i*Designer is divided roughly into the following three subtasks.

- *structuring*: to create a learning item network as production criteria of learning content
- *sequencing*: to arrange learning items in the learning item network into a learning content control structure
- *verification*: to verify the validity of learning content.

Structuring and sequencing correspond to the work of building a model of the learning matter mentioned in Section 3.2, and the work of building a model of the learning process mentioned in Section 3.3, respectively. Relationships between the two subtasks are similar to those between government guidelines for teaching and a textbook. That is, the government guidelines for teaching comprise the system of learning matter; a textbook represents the concrete arrangement of learning matter into a document structure. Various textbooks can be created that satisfy the restrictions provided by the government guidelines for teaching depending on author’s intentions in authoring the textbook, such as a learning goal and intended learners.

*i*Designer provides an environment for each subtask, as described in Sections 4.2–4.4. It is assumed that the structuring environment described in Section 4.2 is provided only to designers who aim at systematisation of learning materials. For most general designers, a structuring environment is not required because the sequencing subtask and verification subtask are their main objectives. Therefore, they use mainly a sequencing environment and verification environment, discussed in Sections 4.3 and 4.4, respectively.

## *4.2 Structuring environment*

Figure 9 shows a learning item network editor for the structuring subtask. A learning item network is expressed with two kinds of nodes and six kinds of links corresponding to the classification of learning items and learning item links shown in Table 1. Each node in the network in Figures 9(A) and 9(B) expresses one learning item; lines between nodes express learning item links. Figure 9(A) shows the whole learning item network. When a node is chosen, it will display nearby learning items in the shape of a tree with the node as the root, as in Figure 9(B). Information added to the domain concept in terms of the learning content design task is displayed in Figure 9(C). Attribute data such as a name, a type, and a level of intended learner are described as shown in Table 3. Integration of this information and the information, which the domain concept itself possesses serves as the content of the learning item. Table 3 illustrates a single learning item set to instance 64relay#1 of a relay device in a certain electric power system; ‘Summary of 64relay’ as a name and ‘novice’ as the level of intended learner are provided.

**Figure 9** Structuring environment

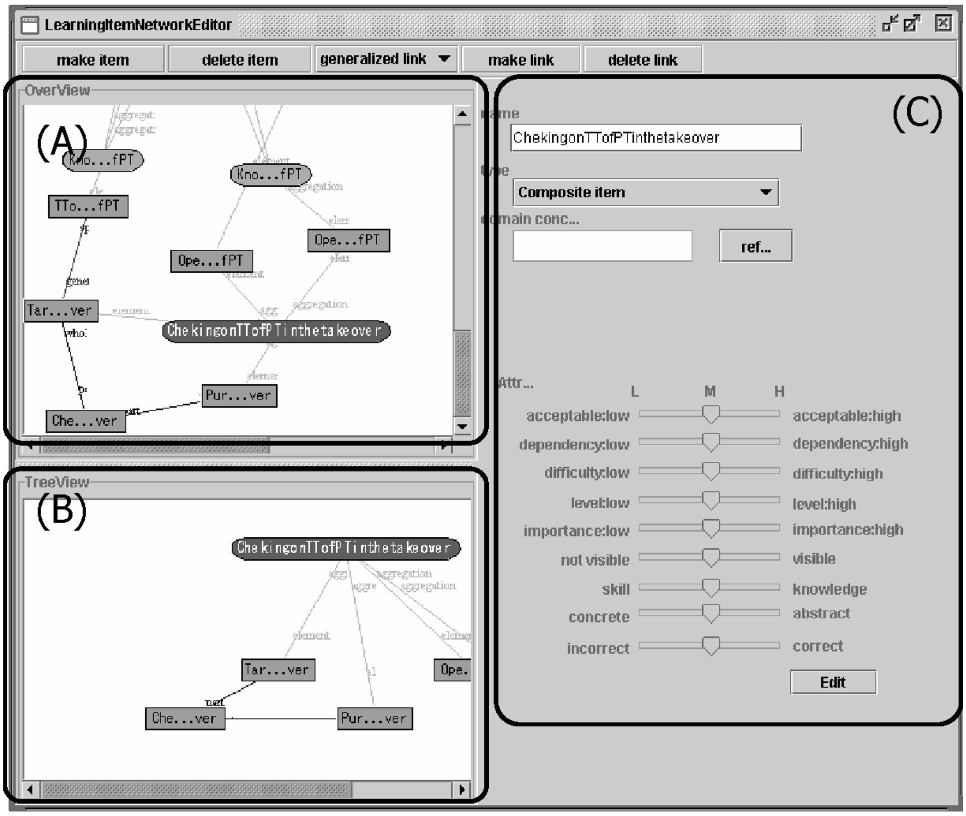


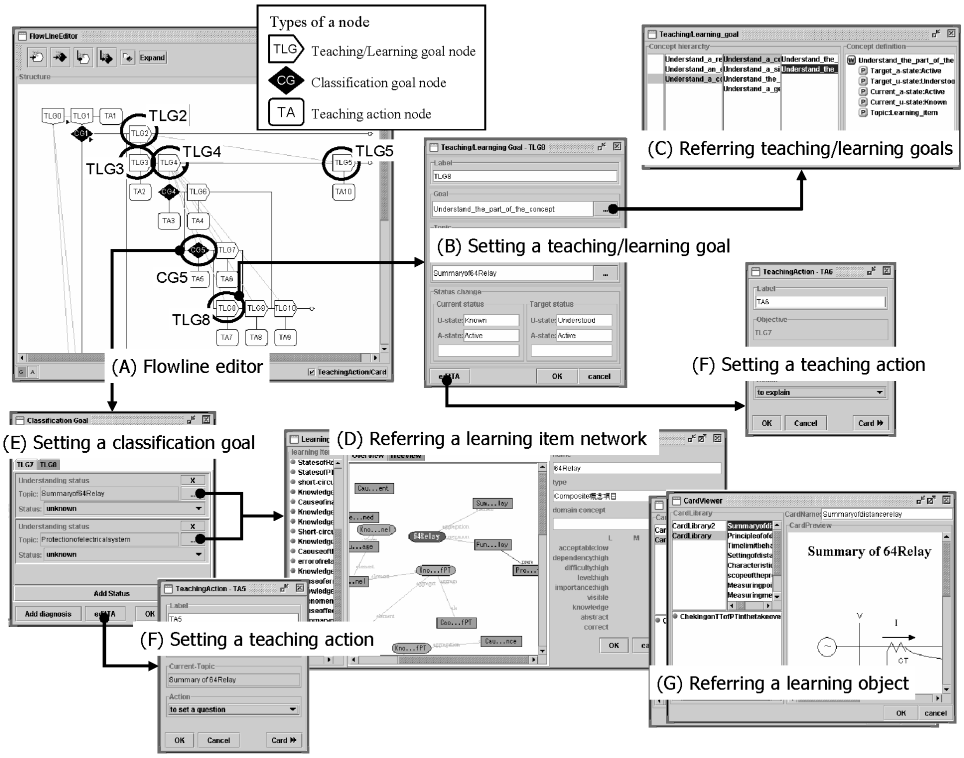

**Table 3** Data structure of a learning item

| *Name* | *Explanation* | *Range* | *Example* |
|---|---|---|---|
| Id | An identifier of an instance | Identifier | li-a |
| Name | Name of the learning item | String | 'Summary of 64relay' |
| Type | Type of learning item | Single\|Composite | Single |
| Domain | Corresponding domain instances to the learning item | Instance of domain concept | 64relay#1 |
| Acceptable | Recognisability, visibility, etc. | H\|M\|L | M |
| Difficulty | Difficulty for learners | H\|M\|L | L |
| Intended learners | Learners' level fit for the learning of the learning item | Novice\|Intermediate\|Advanced | Novice |
| Participant | Elements of the learning item, if type is 'Composite' | Learning items | |

### *4.3 Sequencing environment*

Figure 10 shows a set of tool groups for a sequencing subtask. For this, the flow line editor in Figure 10(A) bears a main role. A flow line is a visualisation format of a learning process model. It has two levels of representation: the conceptual level and the object level. Figure 10(A) shows a conceptual level flow line, which expresses the goal hierarchy and a teaching action sequence with three types of nodes: a teaching/learning goal node, a classification goal node, and a teaching action node. It also uses two types of directed links: a control link and a goal specialisation link. On the other hand, an object level flow line is the visualisation of a learning contents control structure, expressed by learning object nodes and control links.

**Figure 10** Sequencing environment




A teaching/learning goal node describes what (learning item) and for what (goal type) the action is made, as shown in Figure 10(B). A designer is shown acceptable goal types in the teaching/learning goal reference window (Figure 10(C)). The list sequence indicates the concept hierarchy; it presents to the designer the list of labels of the teaching/learning goals gradually from the left-hand side (high level) to the right-hand side (low level) based on the *is-a* and *part-of* relationships between concepts. Moreover, acceptable learning items are shown in the form of a learning item network, as shown in Figure 10(D).

A classification goal represents classifying a learner's understanding to constitute an adaptive control structure. It describes information to judge what (learning item) is in what understanding state, such as 'unknown', 'understood', and 'acquired'. This node describes branches in a control structure corresponding to a learner's answer and the understanding state supposed at each of them. For example, classification goal node CG5 in Figure 10(A) is described as shown in Figure 10(E): a branch to two directions is depicted according to two understanding states – 'understood' and 'unknown'. The classification goal is finally embodied with a diagnostic action (such as a test), which includes a direct approach to a learner, or indirect approach of referring to a learner model.

In Figure 10(A), broken and solid lines denote goal specialisation links and control links, respectively. For example, teaching/learning goal TLG2 is specialised to three goals: TLG3, TLG4, and TLG5. Various patterns are assumed for detailing, such as a pattern to organise learning about a larger learning item into learning sequence of smaller learning items, and a pattern to specialise an abstract learning goal into multiple learning goals at a lower level. In this work, it is vital to comprehend the learning item relevant to specialisation and then to sequence it appropriately along with the designer's intention. The learning item link set in the structuring environment serves as its reference information. For example, to specialise an upper level goal for a single learning item $S_1$ shown in Figure 6, it is proposed to the designer that $S_1$ can be divided into learning items $C_1$, $S_2$, and $S_3$ by using the composite learning item $C_1$ and the aggregation relationships related to it. Furthermore, if prerequisite relationships among $C_1$, $S_2$, and $S_3$ are set, they will serve as basic information for sequencing lower level goals.

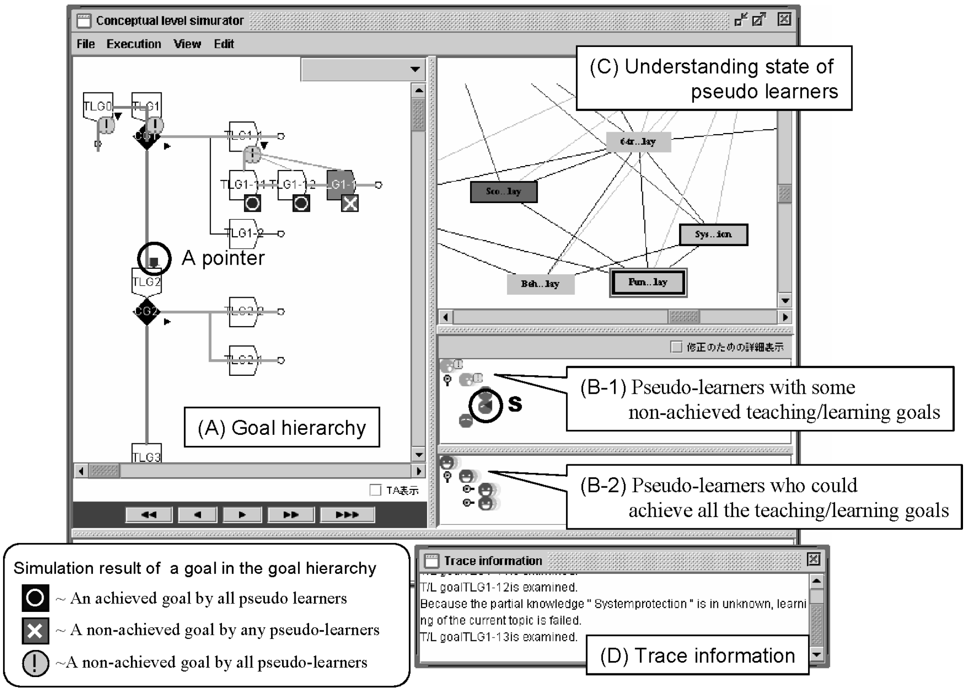
Learning contents are expressed with an object flow line. An object flow line is the structure of a concrete learning object (see Figure 10(G)). A learning object is set in correspondence to the description of a teaching action, as shown in Figure 10(F).

The design of this environment presumes that a designer first develops a goal hierarchy from an upper level to a lower level and then embodies it gradually to a learning content control structure. However, it should not necessarily be a multi-layered goal hierarchy; simple design of single layer goal hierarchy is also allowed.

### *4.4 Verification environment*

In the verification environment, information about behavior of learning contents at the conceptual level is shown visually. The learner assumed to learn the contents in Figure 11(A) (hereinafter referred to as a pseudo-learner) is iconified and shown in Figure 11(B); his understanding state is shown in Figure 11(C). A pseudo-learner is generated based on a conceptual level flow line. For example, the classification goal of two branches implies that learners of two states that differ in branch conditions are assumed; therefore two pseudo-learners are generated. In this study, the function to trace the learning process of a pseudo-learner according to the hierarchy 0'th order goal is referred to as a conceptual level simulation.

**Figure 11** Verification environment




Execution of a simulation classifies and displays pseudo-learners who were not able to achieve the learning goal (above) and pseudo-learners who were able to (below), as shown in Figure 11. The upper part becomes empty when the intention that the designer expressed in the learning content model is appropriate. This implies that it is very likely that a learner will obtain the expected result based on a theory or heuristics referred to when building an ontology. The basic work of a designer in the verification environment is to trace the learning process of the pseudo-learner with a non-achieved goal and to reveal the cause of its lack of achievement. The designer chooses a pseudo-learner in Figure 11(B) and controls the progress of learning by the control button in the lower part of Figure 11(A) to identify whether any learning problem exists. At that time, the learner type is identified based on information (such as change of understanding state shown in Figure 11(C)), which the system presents. For example, Figure 11 shows the state in which the cause of a non-achieved goal by the chosen pseudo-learner (indicated with icon S) is investigated. When the learning path of the pseudo-learner is highlighted in the conceptual level flow line and learning process is advanced to the position of the pointer on the path, the understanding state at that time is displayed on the learning item network (see Figure 11(C)). The trace information window (Figure 11(D)) shows that the reason why learning of the learning item 'Function of 64relay' failed is that the prerequisite learning item 'system protection' a, was in the 'unknown' state.

## 5 Conclusion

It is difficult to establish a framework to symbolise and express learning contents design because the design of the target phenomenon, i.e. learning, is so abstract. As an approach to this issue, in this study, the authors built an ontology of learning contents design and developed a design support environment, *i*Designer, based on it. The environment was tested by designing learning contents for high school physics and for the relay test operation in an electric power system, which is a company-training subject. In the case of physics learning contents, a relatively small-scale learning item network (about 50 learning items) concerning motion of material was created and sequenced. Flow lines with various forms of branches were drawn in this sequencing. The learning item network for the relay test operation in an electric power system has about 300 learning items, and corresponds to the actual condition of a company training field. A relatively small flow line was created in sequencing for one of the activities included in novice training, having ten accompanying questions and 33 associated branches. All kinds of assumed conditions, such as the lack of learning items and sequential flaws, were embedded intentionally in these flow lines to verify the operation of the conceptual level simulator. As a result, we confirmed that while *i*Designer presents the progress of assumed learning visually, it detects all embedded errors and points out their positions.

In this paper we described the conceptual level representation of the design result and the design support environment. In this environment, encouraging a designer to externalise the work result, which has heretofore been implicit, impels that designer to recognise the design process visually. It therefore enhances consideration during design. iDesigner interprets the design intention expressed by the conceptual level model. Then it presumes, visualises, and feeds-back to the designer the conceptual level behavior intended in the learning content. The conceptual level interaction between the designer and the tool is realised on the basis of an ontology. The authors consider that the conceptual level simulation is a good implementation example to demonstrate merits of ontology awareness of a tool.

On the other hand, it is needless to point out the demerit of forcing a designer's modelling according to ontology: poor agreement on the ontology causes poor intimacy of interaction between the tool and the user. However, it is not the issue of alternative, but the issue of trade-off in which merits should be increased while demerits should be suppressed as much as possible. It is crucial to mount ontology of the highest agreement in the tool to solve this. So far, we have advanced this study aiming at mounting the ontology awareness of a tool; however, we have made no direct discussion about the ontological quality and design tool convenience. Reproducibility of existing instructional theories and models (such as Gagne's five classifications of learning results [13] and Bloom's taxonomy of the educational objectives [14]) within the proposed learning contents design task ontology need to be verified. We will advance further discussion on this topic in the future.

A different ontology is needed for education of other modes. In such cases, an ontology can be constituted with the Ontology Editor and a corresponding data structure can be created by importing the XML form from the Ontology Editor into *i*Designer. Iconified objects on the editor screen corresponding to a superordinate concept of an ontology (for example, a learning item node) and conditions of examination for the conceptual level simulator are hard-coded. Therefore, recoding is required for the superordinate concept shown in Section 3.1.

As a future direction, the authors expect to conduct verification of the design of the object level model of other forms, specifically investigating the correspondence to SCORM [15], which is the standard specification of learning contents.

## References


**1** Authorware (1997) *Computer Software*, Macromedia.

**2** Murray, T. (1998) 'Authoring knowledge based tutors: 'tools for content, instructional strategy, student model, and interface design', *Journal of the Learning Sciences*, Vol. 7, No. 1, pp.5–64.

**3** Hirata, K., Ikeda, M. and Mizoguchi, R. (2001) 'Total resolution for human resource development based on competency ontology', *Proceedings of the International Conference on Computers in Education 2001 (ICCE 2001)*, Korea, pp.1149–1152.

**4** Jin, L., Chen, W., Hayashi, Y., Ikeda, M., Mizoguchi, R., Takaoka, Y. and Ohta, M. (1999) 'An ontology-aware authoring tool – functional structure and guidance generation', *Proceedings of 9th International Conference on Artificial Intelligence in Education (AI-ED'99)*, France, pp.85–92.

**5** Rumbaugh, J. *et al*. (1990) *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, N.J.

**6** Mizoguchi, R. (1998) 'A step towards ontological engineering', *Proceedings of the 12th National Conference on AI of JSAI*, pp.24–31. (English translation is available in http://www.ei.sanken.osaka-u.ac.jp/english/step-onteng.html)

**7** Fensel, D. (2001) *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, New York.

**8** Ikeda, M., Hayashi, Y., Jin, L., Chen, W., Bourdeau, J., Seta, K. and Mizoguchi, R. (1999) 'An ontology – more than a shared vocabulary', *Proceedings of 9th International Conference on Artificial Intelligence in Education (AI-ED'99) Workshop on Ontologies for Intelligent Educational Systems*, France, pp.1–10.

**9** Aroyo, L., Dicheva, D. and Cristea, A. (2002) 'Ontological support for web courseware authoring', *Proceedings of 6th International Conference on Intelligent Tutoring Systems (ITS 2002)*, pp.270–280.

**10** Mizoguchi, R. and Bourdeau, J. (2000) 'Using ontological engineering to overcome AI-ED problems', *International Journal of Artificial Intelligence in Education*, Vol. 11, No. 2, pp.107–121.

**11** Ikeda, M., Seta, K., and Mizoguchi, R. (1997) 'Task ontology makes it easier to use authoring tools', *Proceedings of 5th International Joint Conference on Artificial Intelligence (IJCAI 97)*, pp.342–347.

**12** Kouji, K., Kitamura, Y., Ikeda, M. and Mizoguchi, R. (2000) 'Development of an environment for building ontologies which is based on a fundamental consideration of 'Relationship' and 'Role', *Proceedings of the 2000 Pacific Rim Knowledge Acquisition Workshop (PKAW 2000)*, pp.205–221.

**13** Gagne, A.M. and Briggs, L.J. (1974) *Principles of Instructional Design*, Holt Rinehart and Winston, Inc., New York.

**14** Bloom, B.S., Hastings, J.T. and Maclaus, G.F. (1971) *Handbook on Formative and Summative Evaluation of Student Learning*, McGraw-Hill, New York.

**15** ADLNet (2001) *Sharable Content Object Reference Model: SCORM*, Ver. 1.1, http://www.adlnet.org/.