

修士論文

e ラーニング XHTML エディタ eXe の SCORM テスト作成機能の拡張

Extension of the eLearning XHTML Editor eXe
for SCORM Test Function

社会文化科学研究科 教授システム学専攻 修士課程

077-G8115

福 島 誠 也

主指導教員：中野 裕司 教授

副指導教員：鈴木 克明 教授

副指導教員：松葉 龍一 准教授

2009 年 1 月

修士論文（又は特定課題研究）要旨 （日本語）

題 目

e ラーニング XHTML エディタ eXe の SCORM テスト作成機能の拡張

要 旨

e ラーニングの普及に伴い、その運用を担う学習管理システム（LMS: Learning Management System）も、コンテンツの作成・表示からテスト作成、採点、成績管理までを一元的に行えるようにシステム拡張・改善が行われてきた。しかし、現在の LMS におけるテスト機能は提示された選択肢の中から解答を選ぶだけのものや、語彙や数値を記入するだけの簡易なものが中心であり、科目によっては、理解度の把握、促進等の効果的な学習を実施するための適切な問題を作成しづらいのが現状である。特に、自然科学系の分野では、与えられた条件から数式を組み立て、解の導出を行うことを常に求められ、単に、計算結果の数値を覚えることは、その学習目的にはなっていない。そのような場合には、毎回、類題がランダムに出題されるような形式が好まれ、さらに、複数の解をもつ問題を出題した場合には、類題のランダム出題だけでなく、その全ての解を解答させ、それらの自動採点を行いたいと考えるのは当然である。

教育関係ソフトウェアの開発では、オープンソース形式による開発が世界的な主流になってきている現状をふまえ、本論文では、e ラーニング向けのオープンソース XHTML エディタとして知られている eXe のテスト作成モジュールを出題者の意図に沿うテスト問題の作成を可能にしかつ、作成した問題が特定の LMS に限定されずに利用できる標準規格 SCORM (Shareable Content Object Reference Model) に則った形式で出力するために行った eXe テストモジュールの拡張開発の成果を示す。合わせて、今後、eXe テストモジュール拡張開発への多くの開発者の参加を容易にするために、同モジュールの基本的な構造や開発手順等について述べる。

修士論文（又は特定課題研究）要旨 （英語）

題 目

Extension of the eLearning XHTML Editor "eXe" for SCORM Test Function

要 旨

The spread of e-Learning have developed unified performances of Learning Management Systems (LMS), such as displaying and making contents, making and marking online assessments, managing results. Unfortunately, assessment modules in the LMS may not be enough to support all subjects, because teachers are called on to set a test/quiz within the several type of questions like multiple choices and entering vocabularies and numbers, in an answer column. Therefore they cannot grasp how students understand each subjects properly to perform effective learning.

Now the open source approach for the development of an educational software becomes to be mainstream. Thus I have developed new assessment creating modules of the open source XHTML editor "eXe", which provide a new type of assessments and also the SCORM (Shareable Content Object Reference Model) assessments that have not set in the LMS until now.

In the present paper I show the new assessment creating modules of the XHTML editor "eXe" and its improved points and usability of systems, in addition, describe structures about the test modules of the "eXe" and how to creat new test modules.

目次

1.	はじめに	6
2.	e ラーニングにおけるテストツール	8
2.1.	学習におけるテストの効用	8
2.2.	テストツールの現状	8
2.3.	テストコンテンツの標準化	10
2.3.1.	SCORM	11
2.3.2.	QTI	12
3.	SCORM 形式テスト作成モジュールのためのエディタ開発	13
3.1.	eXe とは	13
3.2.	eXe のテスト作成モジュール構造	14
3.3.	新規開発拡張モジュールの構造	16
3.4.	開発概要	17
3.4.1.	作業手順（全体フロー）	17
3.4.2.	既存モジュールへの追記（手順 1. ー手順 3. の詳細）	18
3.4.3.	新規開発モジュールの追加（手順 4. ー手順 6. の詳細）	20
3.4.4.	日本語化	27
3.5.	SCORM 形式での出力結果	28
3.6.	開発モジュールの使用例と動作確認	32
3.7.	出力モジュールと LMS との連動	35
3.7.1.	Blackboard Learning System	36
3.7.2.	Moodle	36
4.	まとめと考察	39
	謝辞	41
	参考文献・参考サイト	42
	付録	44
	付録 1 新規開発モジュールによるテスト問題作成手順	44
	付録 2 既存ソースへの追記内容	47

付録 3	新規作成モジュール群【ファイル 1～ファイル 4 までの 4 ファイル】	69
付録 4	日本語化追記内容	89

1. はじめに

近年の高速インターネットの普及とコンピュータ技術の進展は、情報通信技術(ICT: Information Communication Technologies)を活用した教育、すなわち e ラーニングの発展を促し、学習者と教員間や学習者同士の距離と、学習時間帯等の制限を超えた学習の場の提供を可能にした。これにより、遠隔地の学習機関に所属した講師の講義を容易に受けられる時代が到来し、時間的、距離的に制約のある社会人等の、何らかの事情により通学できない学習者への学習の機会が広がってきている。教育への ICT の活用は、これまでの研究により示されてきた様々な教育・学習理論を実践し、学習成果に結び付けてきている[1]。ICT を活用した学習では、書籍同様のテキストや静止画の表示のみならず、動画による動きのある表現や、フルカラーでの表現、音声の再生、こちらからの入力に応じた画面表示など、書籍にはない学習方法が提供され[2]、さらに、ICT を利用しないこれまでの学習形態では不可能であった、学習到達度の最終的な把握という意味のテストだけでなく、学習者個々の弱点強化のための即座のフィードバックを目的としたテストの実施なども行うことができるようになった[3, 4]。

しかし、これらテストを用いた学習効果の向上手法も、理解度がきちんと測定できる出題がおこなえて初めて可能となる。特に自然科学系の分野は、暗記をすれば済むような学習は少なく、与えられた問題文中の条件を読み取り、用いるべき数式を考え、計算を行い、解答を導き出すことが重要視される。このような学習目標の場合には、単純な選択式の問題や、解答の文字列が一致するかを確認するような穴埋め問題だけでは学習内容に対応しきれない。

本論文では、特に自然科学系分野において、より効果的にオンラインテストを利用するために、オープンソースの XHTML エディタである eXe をベースにテスト作成モジュールを拡張することで、既存の LMS やテストオーサリングツールでは作成が困難であった形式の問題を作成できるようにすることと、作成した問題を SCORM(Shareable Content Object Reference Model)形式にて出力可能にするための開発結果を示すとともに、その拡張開発の手法や、今回作成した新しいテストモジュールの活用場面、今後の拡張開発の視点について示している。

eXe と同様に SCORM 出力が可能なテスト問題作成ツールで有名なものに HotPotatoes があり、数値問題や計算問題こそ作成できないものの、多くの問題形式に対応し、かつ様々なフォーマットでのテスト問題作成と出力を可能にしている[5]。

しかし、HotPotatoes はオープンソースではないため、新規の問題形式に対応させようと考えた場合、それは開発元に依頼するしか方法がなく、拡張というスタイルでの開発を行う際のベースシステムにはなり得ない。一方 eXe は、SCORM 出力に対応した希少なオープンソースのツールであり、本研究において SCORM 対応のテスト問題作成モジュールを開発すると同時にその開発手順を明らかにすることは、多くのユーザと開発者にとって大きなメリット与えることと考える。

第 2 章では、現状の LMS やテスト作成専用ツールのテスト機能を調査した結果を整理し、あわせて、テストコンテンツと標準規格 SCORM の関係について述べた。第 3 章では、拡張に用いた eXe のシステム構造と、テスト作成モジュールを開発する際の基本的なモジュール構造を明らかにし、さらに、今回開発した数値解を複数解答できる問題を例に、その作成手順と実際の動作確認の結果について示した。最終章には、まとめと考察として本研究で得られた知見の総括と今後の開発に向けての課題点を記してある。付録として、今回拡張開発したテスト作成機能における実際の問題作成から SCORM 形式での出力までの手順ならびに、ソース追記内容と新規作成のソースを記載した。

2. e ラーニングにおけるテストツール

2.1. 学習におけるテストの効用

e ラーニングに限らず学習において、テストの果たす役割は大きい。「テストは教える側と学ぶ側の両方が少しずつ向上していくための道具である」というように[6]、テストは学習の成果を測定するだけでなく、教材そのものや教授方法の改善点を見つけたすことにも活用できる。例えば、

- 事前テストの実施において学習内容のおおよその把握が可能、
 - 学習内容の理解度を測定、確認することが可能、
 - 学習に対して理解が不足している点を把握でき、弱点の克服に繋げることが可能、
 - 事後テストの実施を事前に示すことで、学習内容理解に向けた意欲の向上、
- などが考えられる。

しかしながら、学習内容によって出題方法を適切に選ばねばテストによる効果を享受することができない。

- 偶然でも正解が生じ易い選択形式の問題や正誤形式の問題ばかりでは、その理解度が正確には把握できない、
- 学習目標に即した内容のテストとなっていない場合、テストに正解することが目的となってしまう、学習内容の理解に意識が向かない、
- 計算式などを活用して解答を導く学習内容では、類題を多数準備するなどして全く同じ問題を避ける工夫をしなければ、解答の暗記で答えられてしまう、

等がその主な観点である。このように、テストはその在り方によって学習全体にプラスの結果をもたらすことができる半面、マイナスの影響を与えてしまう可能性があることから、理解度を正しく反映できるテストを実施し活用することが重要となる[7]。

2.2. テストツールの現状

テストから得られる効用を最大限にするためには、学習内容に対する理解度を正しく把握できる問題を作成することが求められる。ここでは e ラーニングにおいて使用される様々なテストツールが、どのような問題作成方法に対応しているのかについての調査結果を示す。

e ラーニングにおいてテスト問題を作成する場合には、LMS 自身が備えている問題作

成機能を使う場合と、フリーウェアや市販ソフトといった外部の問題作成専用ツールで問題を作成し、その問題を LMS にインポートする場合とが考えられる[8]。表 1 に、代表的な LMS と問題作成専用ツールにおいて作成可能な問題形式の一覧を掲載する。

表 1：代表的な LMS と問題作成専用ツールにおける作成可能な問題形式一覧

	LMS (LMS内で使用するための作問機能)			LMS+問題作成ツール (SCORM出力)	問題作成ツール (SCORM出力)		
	Moodle	Web Class	Blackboard Learning System	Attain3 +インタラクション メーカーPRO	Hot Potatoes	QuizCreator	Adobe Captivate3
計算問題 (数値のランダム 設定可能)	△※		△※ 計算形式				
作文問題	○	○ 記述式	○ 短文形式			○ 小論文	
組み合わせ問題 (問いと答えの 1対1対応)	○	○ マッチング	○ 整合形式		○ Jmatch	○	○
穴埋め問題 (Cloze問題)	○		○ 穴埋め形式	○ ドラッグ&ドロップ	○ Jcloze		
多肢選択問題	○	○ 単数選択式・ 複数選択式	○ 組み合わせ形式	○ 択一・複数選択	○ JQuiz	○ 単一選択・ 複数選択	○
記述問題	○	○ 単語/数値入力	○ 記述形式	○ 記入問題 (テキスト入力問題)		○ 空所補充	○ 空所補充
数値問題 (数値の正解範囲を 設定可能)	△※		△※ 計算形式				
ランダム記述 組み合わせ問題	○						
○/×問題	○		○ 真偽形式	○ ○×問題(正誤二択)		○ 正誤	○
順序付け		○	○ 並べ替え形式		○ Jmix	○ 正序	
表形式		○					
クリックマップ						○	○ 画像のホットスポット
クロスワード					○ Jcross		
投票モジュール	○	○ レベル選択式					
課題モジュール	○	○ レポート提出					

※Moodle並びにBlackboard Learning Systemの計算問題、数値問題では複数解答欄は不可

表 1 において自然科学系の分野に関連する出題方法に着目すると、数値問題（数値の正解範囲を設定可能）や計算問題（問題に用いる数値のランダム設定）が可能なものは Moodle[9]と Blackboard Learning System[10]に限られている。さらに、いずれの LMS においても表 1 で△と表現したように、一つの問題に対して解答欄を一つしか設けることができない。現状では、数値問題や計算問題において複数解答欄を設けようと考えた場合、Moodle における Cloze 形式と呼ばれる穴埋め型の問題形式を用いれば、複数の穴埋め問題を利用し数値の解答を準備することは可能であるが、Cloze 形式は

問題テキストの中に正答情報等の制御文字列を埋め込む作業が発生し、問題を作成する際には、直観的なマウス操作による GUI(Graphical User Interface)ではなく、キーボード入力による CUI(Character-based User Interface)を行う必要があり、誰にとっても簡易な作業とは言えないのが現状である。

加えて、作成した問題を他の LMS でも利用可能な SCORM モジュールとして出力できるかどうかについての対応状況を調査したところ、表 2 に示すような結果を得た。

表 2 : LMS での作成テスト出力機能

	Moodle	Web Class	Blackboard Learning System
GIFTフォーマット	○	×	×
Moodle独自 XMLフォーマット	○	×	×
IMS QTI 2.0	○	×	×
XHTMLフォーマット	○	×	×
Web Class独自 CSVフォーマット	×	○	×
Blackboard独自フォーマット	×	×	○
SCORM	×	×	×

表 2 より明らかなように、LMS 標準のテスト機能においてはエクスポート機能がよく実装されているのは Moodle であり、Web Class[11]ならびに Blackboard Learning System においては商用 LMS ということもあり、他の LMS への移行にもつながるエクスポート機能は独自フォーマットへの出力に限られていることが分かった。

今回行った調査により、自然科学系に対応し得る数値問題や計算問題といった種類の出題は、テスト問題作成専用ツールでの実装が少なく、また一部 LMS に標準の出題機能では不十分な面があり、現状では非常に限定的な問題種類でしか出題できないことが明らかになった。

2.3. テストコンテンツの標準化

e ラーニングを実施、運用していく上で、常に意識しておく必要のある点に、異なる LMS 間での学習コンテンツの流通性を高めると同時に、コンテンツを組み合わせる

ことによって新たなコンテンツ制作の負担を少なくすることがあげられる。これを可能にする有効な方法が標準規格 SCORM を活用した導入手法である。SCORM の利用により LMS 間を超えた相互のコンテンツ流通だけでなく、LMS のバージョンアップにおいてもコンテンツに改修を加えることなく利用することができる。

2.3.1. SCORM

SCORM 規格とは、e ラーニングのプラットフォームとコンテンツを結ぶ標準規格として、アメリカ国防総省の ADL (Advanced Distributed Learning Initiative) [12] により制定されている e ラーニングのプラットフォームとコンテンツを結ぶ標準規格である。オンラインテストを考えた場合、SCORM 規格では、プラットフォームである LMS とコンテンツとの線引きとして、テストコンテンツの採点までをコンテンツ側に含めていることがその特徴である。つまり、SCORM では出題から採点までをコンテンツパッケージとして出力できるため、新たな問題種別を新規に作成したい場合や、採点方法に工夫を凝らした方法を設定したい際には SCORM による開発が適していると言える。ただし、SCORM 形式でのオンラインテストを利用する際には、次の点に留意しておく必要がある：

- LMS と SCORM コンテンツとのやり取りは規格上決まっており、それ故に学習者の入力した解答など、格納できる情報が制限される、
- 採点部分までが SCORM 側に組み込まれているため、学習者が問題を解かずに SCORM コンテンツを調べ、正しい答えを解答する可能性があるため、実際の運用では該当のファイルを隠蔽するなどの防衛策を取ったほうがよい、
- LMS 側が SCORM コンテンツから入手した得点等の情報の閲覧はできても、LMS 内の成績処理の一部として連動して扱う LMS が殆どない。

なお、SCORM の最新バージョンは SCORM2004 の 4th Edition(2008 年 12 月のアナウンスで、ベータ版のサンプルランタイム配布開始)であり、SCORM2004[13]は前バージョンの SCORM1.2[14]と比較し、学習コンテンツの学習順序について制御を行えるようになっている。この制御機能はシーケンシングと呼ばれ、この機能を用いることでコンテンツに目を通したかどうかや確認テストの点数などを条件にして、学習順序をコントロール可能にしている。

2.3.2. QTI

eラーニングコンテンツにおいてテストを扱う際に、SCORMと並んで名が挙がるものの一つにIMS GLC(IMS Global Learning Consortium)の定めるQTI(Question & Test Interoperability)規格[15, 16]がある。QTI規格の最新バージョンは2.1ドラフトであったが、この規格は仕様が固まらないまま削除されており、2.0が最新の状態である

QTI 規格は問題種別や問題文、解答などを XML 規格に格納しパッケージングする。すなわち問題バンクの形態をとった規格であり、実際の利用にあたっては、その XML データをもとに LMS 画面上に受験画面を表示したり採点をさせたりする部分を、LMS 側で実装しなければならない。現在の QTI 規格に存在しないような問題種別については、当然既存の LMS に QTI 規格として実装されていないため、そういった出題機能や採点機能を開発対象としたい場合にはその用途としては適さないことが分かり、したがって、本研究において目指した開発は、QTI は不適切であることが分かった。

3. SCORM 形式テスト作成モジュールのためのエディタ開発

3.1. eXe とは

XHTML エディタ eXe は、CORE Education というニュージーランドの非営利教育研究開発組織にて現在サポートされているオープンソースのアプリケーションである[17]¹。eXe を利用することによって、教師はインターネットのウェブサイトの記述に用いる HTML 言語や、構造化されたデータを記述する XML 言語といった知識がなくても、テストを含むインタラクティブな e ラーニングコンテンツを容易に作成することが可能になる。また、制作したコンテンツ（テストコンテンツも含む）は、独自の形式のみならず SCORM1.2 や IMS Content Packaging[18]等での出力を可能としている。

eXe は標準で、以下の図 1 に列挙するような機能（制作するコンテンツのパーツとして、ページ内に配置できる機能）を持っており、これら機能の中でテスト問題を作成するモジュールをまとめると表 3 のようになる。

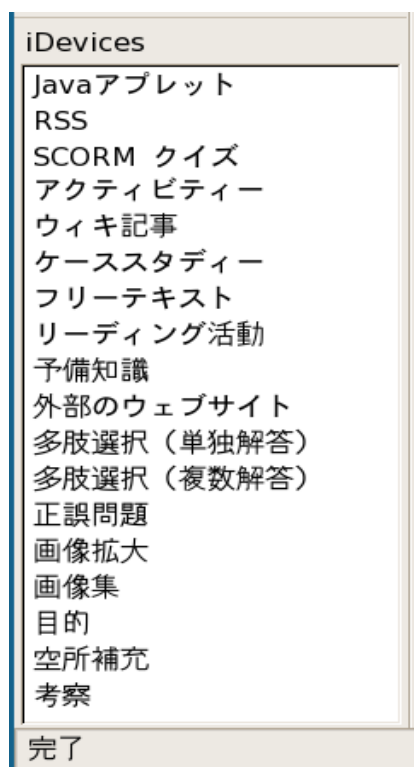


図 1 : eXe の標準搭載の機能

¹開発は、ニュージーランドのオークランド大学であったが、最近では eXe Project として進められている。2009 年 1 月現在の最新バージョンは 1.04。

表 3 : eXe 標準のテスト問題作成モジュール

モジュール名	概要	SCORM 出力
SCORM クイズ	多肢選択式の択一問題	可
多肢選択（単独解答）	多肢選択式の択一問題	不可
多肢選択（複数解答）	多肢選択式の複数選択問題	不可
正誤問題	True / False の二者択一問題	不可
空所補充	空所の文字を記入させる問題	不可

表 3 の示すように、eXe の持つ SCORM 出力に対応したテスト問題作成モジュールは「SCORM クイズ」と呼ばれるモジュールだけであり、しかも、この問題形式は多肢選択式の択一問題のみである。この問題形式のみでは非常にシンプルすぎて多くの学習分野に対して理解度の正確な測定に対応可能であるとは言えない。

eXe の最大の利点は SCORM 出力に対応した希少なオープンソースのツールであることにある。すべてのソースコードが公開されているので、開発力さえあれば、既存のモジュールを拡張し、様々な形式の SCORM 対応のテスト問題作成モジュールの開発を行うことができる。

3.2. eXe のテスト作成モジュール構造

eXe は e ラーニング XHTML エディタという名前が示すように、e ラーニングで様々なコンテンツを活用できるよう、多様な XHTML フォーマットでの出力を可能としたコンテンツ作成ツールである。今回はその中でもテスト問題作成モジュールに焦点を当て、新規にモジュールを開発した。開発にあたっては、まず eXe のソースコードを読むことから開始した。eXe のソースコードは、Python[19, 20]というフリーなオブジェクト指向プログラミング言語で書かれており、Web アプリケーションの開発言語としては一般的なものの一つである。

eXe は膨大なファイル群で構成されており、その全てに目を通すことは不可能であるため、当初はディレクトリ構造に注意して目を通し、以下のような構成をとっていることが分かった。

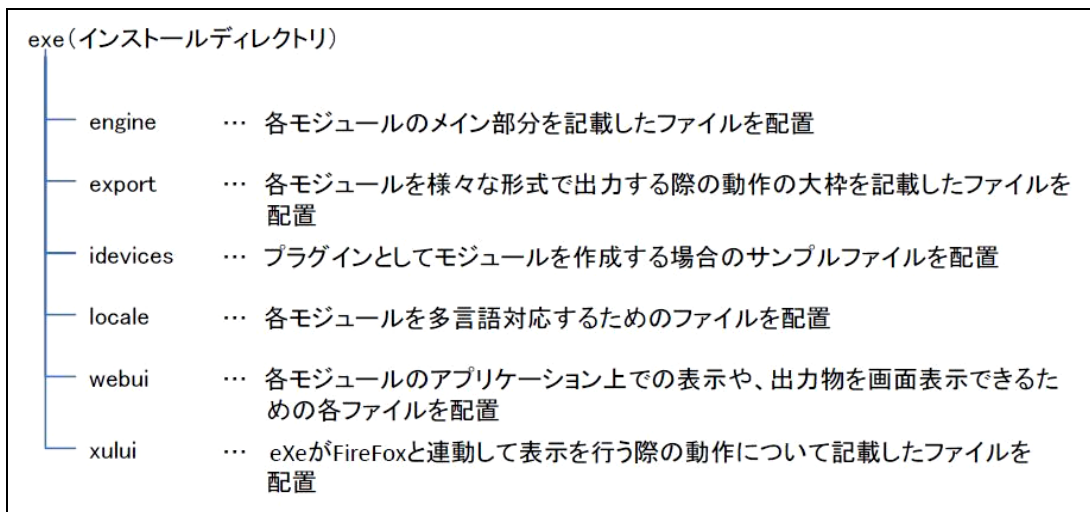


図 2 : eXe のディレクトリ概要

この中で実際のコンテンツ作成モジュール（eXe では iDevice と呼ばれる）に着目して、モジュールを構成する一連のファイル群がどのディレクトリのどのファイルなのかを調査した。以下では、前述の SCORM クイズを例に調査を行った結果を示す。

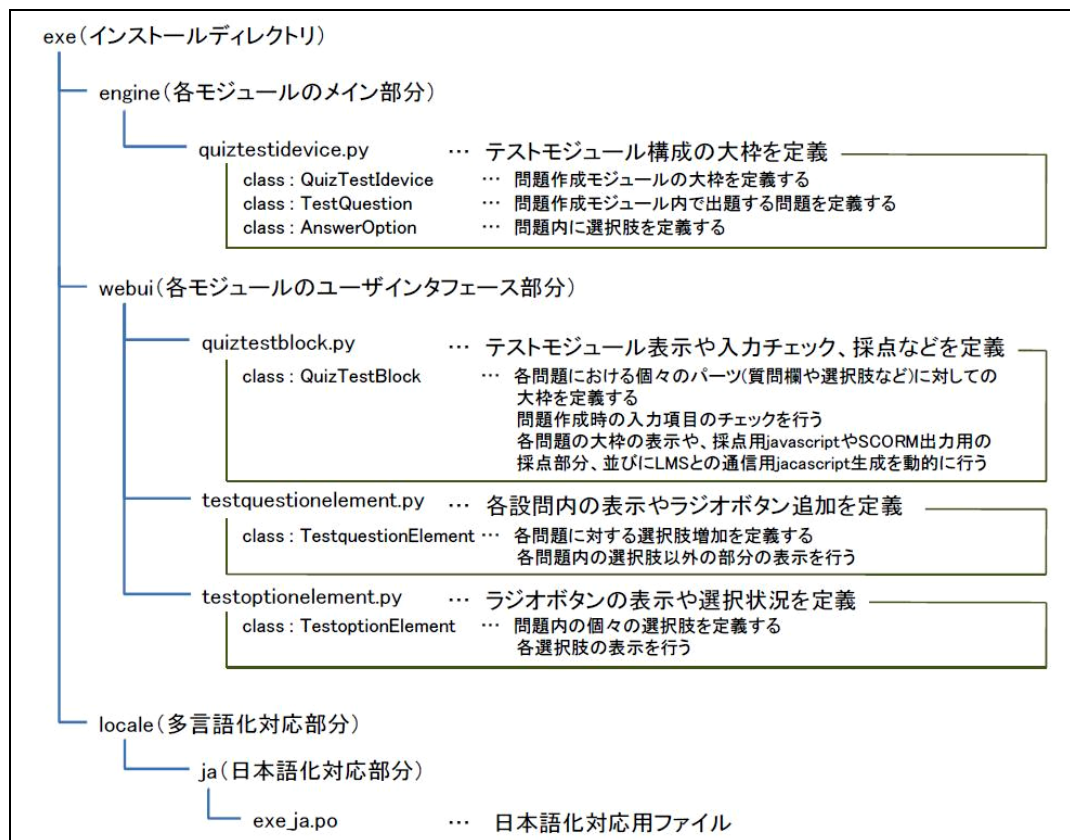


図 3 : QUIZ テストを例にとった構成ファイル群

細かい設定方法については 3.4. 開発概要にて詳しく述べるが、新規に作成すべきファイルは上記の図 3 に挙げられた 5 つのファイルである。図 3 の例では、ラジオボタンを用いた多肢選択式(択一解答)となっているため、選択肢を複数設けるためのファイル(testoptionelement.py)を含んでいる。例えば True/False の 2 択問題を SCORM 対応で開発するのであれば、True と False 以外の選択肢が増えることはないため、このファイル(testoptionelement.py)はなくすことも可能である。

これらのそれぞれのファイル(Python 言語においてモジュールと呼ばれる)の内容については、それぞれのモジュールに定義されたクラス(データとその操作をまとめたオブジェクト)[21]の果たす役割として、図 3 に併記した。

ここまでテスト問題作成モジュールの中心部分ばかりを細かく見てきたが、SCORM での出力を行う場合には、LMS との通信を行うための Javascript の function 群を収めた APIWrapper.js ファイルをはじめ、多くのファイル群を一つの zip ファイルとしてまとめる必要がある(exe によって書き出される一連のファイル群の詳細については、3.5. SCORM 形式での出力結果を参照のこととする)。これらのファイル群を一つにまとめる仕組みは、図 2 に示した exe(インストールディレクトリ)/export/というディレクトリ内の専用の Python モジュールが担当している。このファイル群出力モジュールは今回の開発による新規拡張モジュールと問題なく連動できている。そのためテスト問題作成モジュールを開発する際には、拡張モジュールが独自に必要とするエディタ画面表示ならびに入力制御部分や javascript 生成部分、採点部分などを作成すれば良いことが分かった。

つまり、今回の新規開発では、図 3 に表した 5 つ前後のファイル群を新規にコーディングすることによって実現できることが分かった。

3.3. 新規開発拡張モジュールの構造

今回開発した拡張モジュールの構造は、以下の図 4 のような構造ならびにファイル名、クラス名とした。基本的には図 3 で示した既存の SCORM クイズの構造に準じて作成した。大きな相違点はラジオボタンによる選択肢部分を、数値入力欄 2 つ(正答値と、正答とみなす許容範囲を表す 2 つの数値)へと変更し、入力内容に対しての制御(数値以外の文字列を受け付けないようにし制限し、正答値欄の必須項目化と許容範

囲値の省略許可) を実施した部分である。

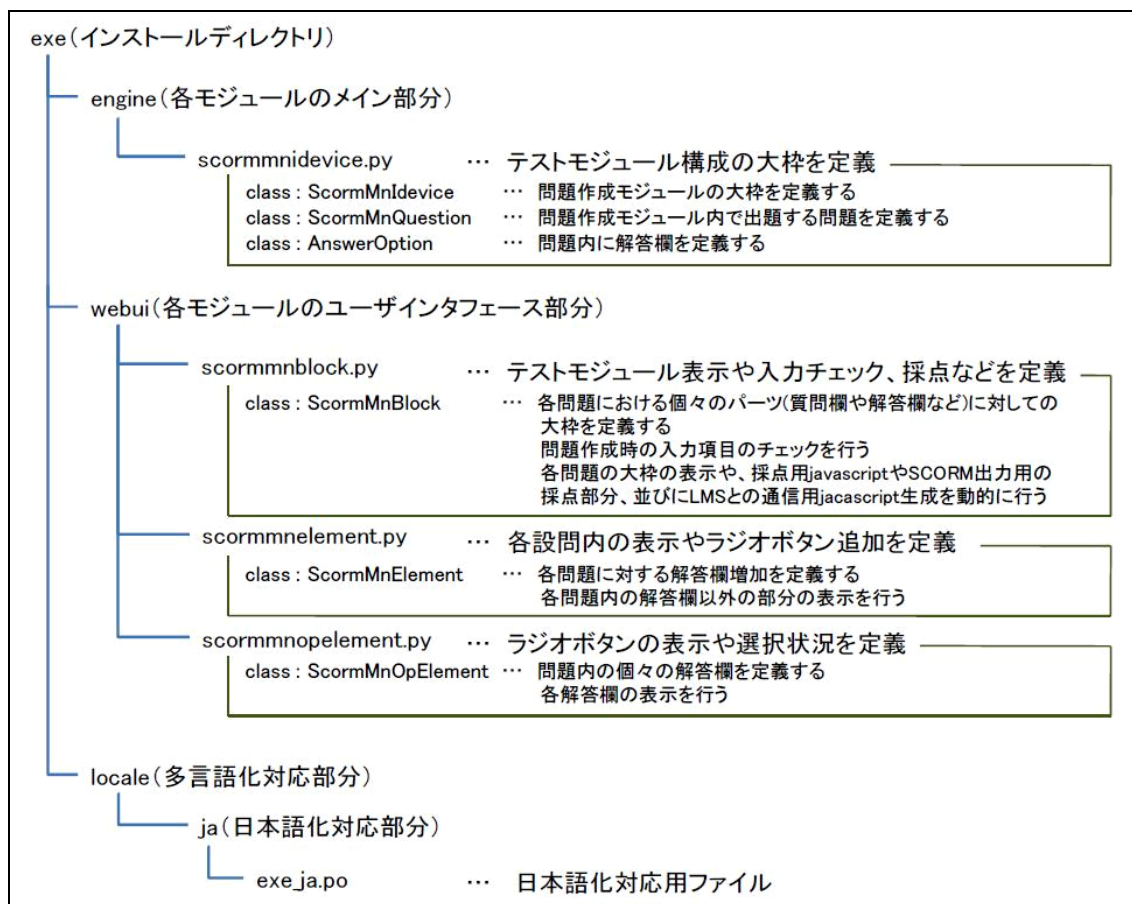


図 4：新規開発の問題作成拡張モジュール群の構造概略

3.4. 開発概要

ここでは最新版の eXe Version1.04(2008/05/09)に対して、Fedora8[22]上で実施した開発作業概要を記す。今回の開発は SCORM 出力に対応したテスト問題作成コンテンツのみのモジュール開発であるため、対応 LMS がより多く流通している SCORM1.2 に準じた開発をおこなった。これは、テストコンテンツについては SCORM1.2 と SCORM2004 の間に大きな相違点はないため、汎用性を重視した選択を行ったことによるものである。

3.4.1. 作業手順（全体フロー）

詳細な作業内容の解説に入る前に、ここでは作業全体の手順を簡潔にまとめ提示する。手順は下記の通りである。

- 手順 1. %(install dir)%/exe/engine/idevicestore.py への追記。
- 手順 2. %(install dir)%/exe/engine/package.py への追記。
- 手順 3. %(install dir)%/exe/webui/builtinblocks.py への追記。
- 手順 4. %(install dir)%/exe/engine/ディレクトリに新規開発モジュールを追加。
(モジュール名は一般的に〇〇idevice.py という命名がされる)
- 手順 5. %(install dir)%/exe/webui/ディレクトリに新規開発モジュールを追加。
(モジュール名は一般的に〇〇block.py という命名がされる)
- 手順 6. %(install dir)%/exe/webui/ディレクトリに手順 5. で追加したモジュールから呼び出されるモジュールを追加。
- 手順 7. %(install dir)%/exe/locale/ja/exe_ja.po へこれまで追加した新規開発モジュールの日本語化部分を追記。

手順 1. ～手順 3. と、手順 7. については既存ファイルへの追記であるため、作業量は非常に少ない。逆に手順 4. ～手順 6. では必要なだけの新規開発ファイルを作成する作業となるため、作成するモジュールの規模によって作業量は大きく変動する。

3.4.2. 既存モジュールへの追記(手順 1. - 手順 3. の詳細)

eXe の様々なコンテンツモジュールの一つに、今回開発のモジュールを読み込むよう登録作業を実施する。具体的には、下記の手順 1. ～手順 3. の通り追記を実施する。

手順 1. %(install dir)%/exe/engine/idevicestore.py への追記。

- ・作業内容: __loadExtended 関数において作成したモジュール scormmnidevice.py モジュールを from ならびに import する部分を追記する。
IdeviceStore クラス自身が持つ extended リストに、新規開発した ScormMnIdevice クラスを追加する部分を追記する。

```
def __loadExtended(self):
    from exe.engine.freetextidevice      import FreeTextIdevice
    ~~~ 途中略 ~~~
    from exe.engine.quiztestidevice      import QuizTestIdevice
    from exe.engine.scormmnidevice       import ScormMnIdevice
    self.extended.append(FreeTextIdevice())
    ~~~ 途中略 ~~~
```

追記

```

self.extended.append(QuizTestIdevice())
self.extended.append(ScormMnldevice())

# generate new ids for these iDevices, to avoid any clashes

    ~~~ 以下略   ~~~

```

追記

図 5 : %(install dir)%/exe/engine/idevicestore.py への追記内容

ここで、追記箇所は太字下線により強調して表示した。以降も同様に表記する。

手順 2. %(install dir)%/exe/engine/package.py への追記。

- ・ 作業内容 : loadNodesIdevices 関数において、新規開発の scormmnidevice に対応する部分を追記する。

```

def loadNodesIdevices(node, s):
    soup = BeautifulSoup(s)

    ~~~ 途中略   ~~~

    elif i.attrMap['class']=="TrueFalseIdevice":
        idevice = burstIdevice('True-False Question', i, node)
    elif i.attrMap['class']=="ScormMnldevice":
        idevice = burstIdevice('SCORM MultiNumerical Question', i,
node)
    else:

        ~~~ 以下略   ~~~

```

追記

図 6 : %(install dir)%/exe/engine/package.py への追記内容

手順 3. %(install dir)%/exe/webui/builtinblocks.py への追記。

- ・ 作業内容 : builtinblocks.py において、新規開発の scormmnidevice の表示や入出力制御を行う scormmnblock.py モジュールの import 部分を追記する。

```

from exe.webui.freetextblock      import FreeTextBlock

    ~~~ 途中略   ~~~

from exe.webui.quiztestblock      import QuizTestBlock
from exe.webui.scormmnblock      import ScormMnBlock

```

追記

図 7 : %(install dir)%/exe/webui/builtinblocks.py への追記内容

3. 4. 3. 新規開発モジュールの追加 (手順 4. - 手順 6. の詳細)

今回の開発において新規に開発した拡張モジュール群は、3. 3. 新規開発拡張モジュールの構造にて図 4 で示した 4 つのファイルとなる。これらのファイルを作成し下記、手順 4. ～手順 6. のようにして、適切な場所へと配置する。

手順 4. %(install dir)%/exe/engine/ディレクトリに新規開発モジュールを追加。

開発を行った全リストについては付録にて掲載することとし、ここでは今回の開発にあたって特徴的な部分を抜粋にて解説する。

以下に、scormmnidevice.py ファイル(抜粋)を記す。

```
# =====
class AnswerOption(Persistable):

    def __init__(self, question, idevice, answer=""):
        """
        Initialize
        """
        self.question = question
        self.idevice = idevice
        self.answerTextArea = TextAreaField(x_(u'Instructions For Answer( and
Correct Answer)'), self.question._optionInstruc, answer)
        self.answerTextArea.idevice = idevice
        self.correctmain = ""
        self.correcttole = ""
        self.ans = ""

    def getResourcesField(self, this_resource):

        ~~~ 途中略 ~~~

# =====
class ScormMnIdevice(Idevice):
    def __init__(self):
        Idevice.__init__(self,
            x_(u"SCORM MultiNumerical Question"),
            x_(u"Seiya Fukushima"),
            x_(u""Unlike the MCQ the SCORM quiz is used to test
the learners knowledge on a topic without providing the learner with feedback to
the correct answer. The quiz will often be given once the learner has had time to
learn and practice using the information or skill. """), u"", "question")
        self.isQuiz = True
        self.emphasis = Idevice.SomeEmphasis
        self.score = -1
        self.isAnswered = True
        self.passRate = "50"

        ~~~ 以下略 ~~~
```

AnswerOption (複数解答欄の class) を初期化する際に、正答と許容範囲、学習者解答を格納する変数を作成する

ScormMnIdevice class の初期化部分において、idevice(eXe におけるコンテンツ作成機能をこう呼ぶ) の名前や作成者情報を格納する

図 8 : %(install dir)%/exe/engine/ scormmnidevice.py (抜粋)

手順 5. %(install dir)%/exe/webui/ディレクトリに新規開発モジュールを追加。

開発を行った全リストについては付録にて掲載することとし、ここでは今回の開発にあたって特徴的な部分を抜粋にて解説する。

本ファイルでは大きく次のような事項を実現している。

- ・ 数値入力用の解答欄(変数名：正答欄=>**correctmain**、正答に対する許容誤差欄=>**correcttole**) に対して、正答欄の空白を禁止するよう制御し、許容誤差欄の空白はゼロと判断することとした。また、数値以外の文字を入力した場合は受け付けられないように作成した。
- ・ **web** 形式での出力の際に、LMS とは通信を行わず採点と結果表示のプレビューを行える Javascript が自動生成されるように作成した。
- ・ 上記の機能に加え、正答や学習者の解答、正解/不正解、得点などの結果を LMS と通信[23]を行い情報の格納(**doLMSSetValue** と命名された javascript で命令)を行うようなプログラムを自動生成するように作成した。なお LMS に格納する情報は以下の通りとした。

問題ID	cmi.interactions.X.id	⇒格納する値の例： keyXb0
問題タイプ	cmi.interactions.X.type	⇒格納する値の例： performance
正答	cmi.interactions.X.correct_responses.Y.pattern	⇒格納する値の例： 21 plus_minus0
学習者の答え	cmi.interactions.X.student_response	⇒格納する値の例： 21
正誤結果	cmi.interactions.X.result	⇒格納する値の例： correct
学習者得点	cmi.core.score.raw	⇒格納する値の例： 100
学習状況	cmi.core.lesson_status	⇒格納する値の例： passed
学習時間	cmi.core.session_time	⇒格納する値の例： 00:00:54.89

図 9：LMS に格納する情報の代表例

以下に、scormmnblock.py ファイル(抜粋)を記す。

```
# =====
class ScormMnBlock(Block):
    def process(self, request):

        ~~~ 途中略 ~~~

        if ("action" in request.args and request.args["action"][0] == "done"
            or not self.iddevice.edit):
            self.iddevice.isAnswered = True
            # remove the undo flag in order to reenale it next time:
            if hasattr(self.iddevice,'undo'):
                del self.iddevice.undo

            ### fuku input correctmain(correct main answer check)
            for question in self.iddevice.questions:
                for option in question.options:
                    if option.correctmain == "":
                        self.iddevice.edit = True
                    else:
                        try:
                            float(option.correctmain)
                        except ValueError:
                            self.iddevice.edit = True
                        except TypeError:
                            self.iddevice.edit = True

            ### fuku input correcttole(correct answer tolerance check)
            for question in self.iddevice.questions:
                for option in question.options:
                    if not option.correcttole == "":
                        try:
                            float(option.correcttole)
                        except ValueError:
                            self.iddevice.edit = True
                        except TypeError:
                            self.iddevice.edit = True

            if "submitScore" in request.args ¥

            ~~~ 途中略 ~~~

        def renderEdit(self, style):
            """
            Returns an XHTML string with the form element for editing this block
            """
            html = "<div class=¥"iDevice¥">¥n"

            ### fuku input correctmain(correct main answer check)
            for question in self.iddevice.questions:
                for option in question.options:
                    if option.correctmain == "":
                        html += common.editModeHeading(
                            _("Please input correct answer(main) with numerical
```

正答欄の正解未記入や数値以外の
入力に対して制御をおこなう部分

正答に対する許容範囲
欄に対し、数値以外の
入力が行われていない
か判定、制御をおこな
う部分

```

value.))
    else:
        try:
            float(option.correctmain)
        except ValueError:
            html += common.editModeHeading(
                _("Please input correct answer(main) with numerical
value.))
            except TypeError:
                html += common.editModeHeading(
                    _("Please input correct answer(main) with numerical
value.))

    ### fuku input correcttolerance(correct answer tolerance check)
    for question in self.iddevice.questions:
        for option in question.options:
            if not option.correcttolerance == "":
                try:
                    float(option.correcttolerance)
                except ValueError:
                    html += common.editModeHeading(
                        _("Please input correct answer(tolerance) with
numerical value.))
                    except TypeError:
                        html += common.editModeHeading(
                            _("Please input correct answer(tolerance) with
numerical value.))

    html += common.textInput("title"+self.id, self.iddevice.title)

    ~~~ 途中略 ~~~

    return html

def renderJavascriptForWeb(self):
    """
    Return an XHTML string for generating the javascript for web export
    """
    scriptStr = '<script type="text/javascript">¥n'
    scriptStr += '<!-- //<![CDATA[¥n'
    scriptStr += "var numQuestions = "
    scriptStr += str(len(self.questionElements))+";¥n"
    scriptStr += "var rawScore = 0;¥n"

    ~~~ 途中略 ~~~

    rawScoreStr += ""
    if (""
    for numtemp in range(numAnswer):
        rawScoreStr += "" %s >= %s & %s <= %s & "" % (
            varStr + "Answer" + str(numtemp),
            "key" + str(i) + "Answer" + str(numtemp) + "low",
            varStr + "Answer" + str(numtemp),
            "key" + str(i) + "Answer" + str(numtemp) +
"high")
        rawScoreStr = rawScoreStr[:-1] + ""
        {
            rawScore++;

```

正答欄の正解未記入や数値以外の入力に対して、その旨の警告表示をおこなう部分

正答に対する許容範囲欄に対し、数値以外の入力が行われている場合、その旨の警告表示をおこなう部分

renderJavascriptForWeb クラスでは、web形式での export を行った際に出力される Javascript 組み込み済みの HTML 出力をおこなう

文字列格納用の変数に、Javascript を自動生成していく部分

Javascript にて採点(正誤判断)を行う部分を自動生成している

```

}"""

~~~~ 途中略 ~~~~

function calcScore2()
{
    getAnswer();

    calcRawScore();
    actualScore = Math.round(rawScore / numQuestions * 100);
    document.getElementById("quizForm%s").submitB.disabled = true;
    alert("Your score is " + actualScore + "%")

}

~~~~ 途中略 ~~~~

def renderJavascriptForScorm(self):
    """
    Return an XHTML string for generating the javascript for scorm export
    """
    scriptStr = '<script type="text/javascript">¥n'
    scriptStr += '<!-- //<![CDATA[¥n'

    ~~~~ 途中略 ~~~~

    answerStr += """
doLMSSetValue("cmi.interactions.%s.id","%s");
doLMSSetValue("cmi.interactions.%s.type","performance");
""" % (unicode(i), quesId, unicode(i))
    numAnswer = 0
    corAns = []
    tolAns = []
    lmssetstres = ""

    ~~~~ 途中略 ~~~~

    answerStr += """
doLMSSetValue("cmi.interactions.%s.correct_responses.%s.pattern",
"%s");
""" % (unicode(i), unicode(numAnswer), corAns[numAnswer] +
"plus_minus" + tolAns[numAnswer])

    ~~~~ 途中略 ~~~~

    answerStr += """
doLMSSetValue("cmi.interactions.%s.student_response",%s);
""" % (unicode(i), lmssetstres[:-5])

    ~~~~ 途中略 ~~~~

def __calcScore(self):
    """
    Return a score for preview mode.
    """
    rawScore = 0

```

Javascript の関数が正解数の割合にて点数を計算する部分を自動生成している

renderJavascriptForScorm クラスでは、renderJavascriptForWeb クラスでの動作に加え、LMS との各種通信(情報格納)部分を含む Javascript を生成する。

doLMSSetValue により、LMS へ格納すべき情報を受け渡す。ここでは正答と許容誤差を文字列格納している。

同様に doLMSSetValue により、受験者の出した解答を LMS へ格納するよう受け渡す。

問題作成時のプレビュー時に Python によって採点(正誤判定)をおこない点数を返す関数。


```

numQuestion = len(self.questionElements)
score = 0
ngcheck = 0

for question in self.idevice.questions:
    ngcheck = 0
    for option in question.options:
        try:
            float(option.correcttole)
        except ValueError:
            option.correcttole = "0"
        try:
            if not ((float(option.correctmain) - float(option.correcttole))
<= float(option.ans) <= (float(option.correctmain) + float(option.correcttole))):
                ngcheck = 1

        ~~~ 以下略 ~~~

```

図 10 : %(install dir)%/exe/webui/scormmnblock.py (抜粋)

手順 6-1. %(install dir)%/exe/webui/ディレクトリに手順 5. で追加したモジュール(今回の例では **scormmnblock.py**) から呼び出されるモジュールを追加。ここでは今回の開発にあたって特徴的な部分を抜粋にて解説する。開発を行った全リストについては付録 2 を見よ。

本ファイルでは大きく次のような事項を実現している。

- ・ 正答と許容誤差のセットを複数設定できるように、解答欄を定義する

ScormMnOpElement クラスを呼び出す部分を作成した。

以下に、**scormmnelement.py** ファイル(抜粋) を記す。

```

class ScormMnElement(object):
    """
    ScormMnElement is responsible for a block of question.
    Used by ScormMnBlock
    == SCORM MultiNumerical Question
    """
    def __init__(self, index, idevice, question):
        """
        Initialize
        """
        self.index = index

        ~~~ 途中略 ~~~

        i = 0
        for option in question.options:
            self.options.append(ScormMnOpElement(i,
                                                    question,
                                                    self.id,

```

追加した解答欄の数に応じて
ScormMnOpElement クラスを呼び
出し **options** リストに追加する。

```

                                option,
                                idevice))

        i += 1

        ~~~ 以下略 ~~~

```

図 11 : %(install dir)/exe/webui/scormmnelement.py (抜粋)

手順 6-2. %(install dir)/exe/webui/ディレクトリに手順 5. で追加したモジュール(今回の例では **scormmnblock.py**) から呼び出されるモジュールを追加。開発を行った全リストについては付録にて掲載することとし、ここでは今回の開発にあたって特徴的な部分を抜粋にて解説する。

本ファイルでは大きく次のような事項を実現している。

- ・%(install dir)/exe/webui/common.py モジュール内の **textInput** 関数を用いて、数値を入力する解答欄（正答欄と、正答に対する許容誤差欄）を作成した。

以下に、**scormmnopelement.py** ファイル(抜粋)を記す。

```

class ScormMnOpElement(object):
    """
    ScormMnOpElement is responsible for a block of option.  Used by
    ScormMnElement.
    == SCORM MultiNumerical Question InputCheck module,

        ~~~ 途中略 ~~~

    def process(self, request):
        """
        Process arguments from the web server.  Return any which apply to this
        element.

        ~~~ 途中略 ~~~

        ### fuku correctmain and correcttole
        if self.keyld+"correctmain"+unicode(self.index) in request.args:
            self.option.correctmain = request.args[self.keyld+"correctmain"+uni
code(self.index)][0]
        if self.keyld+"correcttole"+unicode(self.index) in request.args:
            self.option.correcttole = request.args[self.keyld+"correcttole"+unico
de(self.index)][0]
            出題者の正答や許容範囲、解答欄への学習
            者の記入内容を変数に格納している部分

        ### fuku answer input
        if self.keyld+"Answer"+unicode(self.index) in request.args:
            self.option.ans = request.args[self.keyld+"Answer"+unicode(self.in
dex)][0]

        ~~~ 途中略 ~~~

```

```

def renderEdit(self):
    """
    Returns an XHTML string for editing this option element
    """
    html = u"<tr><td align=¥\"left¥\"><b>%s</b>" % _("Instructions For
Answer( and Correct Answer)")

    ~~~ 途中略 ~~~

    html += common.richTextArea(self.answerId,
self.answerElement.field.content_w_resourcePaths,
package=this_package)
    html += "</td>"

    ### fuku Let's RenderEdit
    html += "</tr><tr><td>¥n"
    html += common.textInput(self.keyId+"correct main"+unicode(self.ind
ex),self.option.correctmain, 5)
    html += u"    &#177    "
    html += common.textInput(self.keyId+"correcttole"+unico de(self.ind
ex),self.option.correcttole, 5)

    ~~~ 途中略 ~~~

def renderView(self, preview=False):
    """
    Returns an XHTML string for viewing this option element
    """
    log.debug("renderView called")

    html = '<tr><td>'
    html += "<br>"
    html += common.textInput(self.keyId+"Answer"+unicode(self.index),
self.option.ans, 5)
    html += '</td><td>¥n'

    ~~~ 以下略 ~~~

```

問題に設置した各正答並びに
許容範囲欄を表示する部分

問題の表示時に、学習者の
解答欄を表示する部分

図 12 : %(install dir)/exe/webui/scormmnopement.py (抜粋)

3.4.4. 日本語化

日本語化用のファイル **exe_ja.po** を編集し、開発部分の日本語化を実施した。具体的には、下記、手順 7. のような追記を行い、**msgfmt** コマンドにて日本語メッセージオブジェクトを作成した。なお、ファイル内の追記場所に関しては、日本語化する前の英語メッセージのアルファベット順となっていたため、それに準じて追記場所を決定した。

手順 7. %(install dir)/exe/locale/ja/exe_ja.po へこれまで追加した新規開発モ

ジュールの日本語化部分を追記（抜粋）する。

以下に、exe_ja.po ファイル（抜粋）を記す。

```
#: exe/webui/scormmnelement.py:118
msgid "Add Answer Column"
msgstr "解答欄を追加"

#: exe/engine/scormmnidevice.py:126 exe/engine/scormmnidevice.py:186
msgid ""
"For each answer column of this ¥n"
"question, please mention contents and the explanatory note which you ¥n"
"should answer. In two blanks below , Enter a right answer (numerical value) ¥n"
"and a tolerance (numerical value).\"
msgstr ""
"この質問のそれぞれの解答欄に対して、答えるべき内容や注釈を記載してください。ま"
"た、下の2つの空欄には正しい答え（数値）と許容誤差（数値）を入れてください。\"

#: exe/engine/scormmnidevice.py:48 exe/engine/scormmnidevice.py:100
#: exe/webui/scormmnopement.py:100
msgid "Instructions For Answer( and Correct Answer)"
msgstr "解答への指示（並びに正答）\"

#: exe/webui/scormmnblock.py:127 exe/webui/scormmnblock.py:133 exe/web"
ui/scormmnblock.py:136
msgid "Please input correct answer(main) with numerical value.\"
msgstr "正しい解答を数値にて入力してください\"

#: exe/webui/scormmnblock.py:146 exe/webui/scormmnblock.py:149
msgid "Please input correct answer(tolerance) with numerical value.\"
msgstr \"正しい解答の許容誤差を数値にて入力してください(許容誤差の入力を省略す"
ると0(誤差を許さない)として扱われます)\"

#: exe/engine/scormmnidevice.py:247
msgid "SCORM MultiNumerical Question\"
msgstr \"SCORM 複数数値問題\"
```

図 13 : %(install dir)%/exe/locale/ja/exe_ja.po（抜粋）

3.5. SCORM 形式での出力結果

eXe から SCORM 出力を行った場合、次の図 14 のようなファイル構成で一つの zip ファイルが出力される。

imsmanifest.xml	...	マニフェストファイル
adlcp_rootv1p2.xsd	...	SCORM1.2共通のファイル
ims_xml.xsd	...	SCORM1.2共通のファイル
imsmd_rootv1p2p1.xsd	...	SCORM1.2共通のファイル
imscp_rootv1p1p2.xsd	...	SCORM1.2共通のファイル
<u>index.html</u>	...	<u>実際のテストコンテンツファイル</u>
base.css	...	index.htmlより読み込まれるスタイルシート
content.css	...	index.htmlより読み込まれるスタイルシート
common.js	...	index.htmlのhead部で読み込まれる共通JavaScript
APIWrapper.js	...	index.htmlで読み込まれるSCORM API呼び出し用JavaScript
SCOFuctions.js	...	index.htmlのhead部で読み込まれる共通JavaScript
fdl.html	...	GNUでのライセンスについての注意書き
imslrm.xml	...	マニフェストファイルから読み込まれる
libot_drag.js	...	マニフェストファイルから読み込まれる
他、gif、png、jpg等画像ファイル		

図 14 : eXe からの SCORM 出力物ファイル構成

eXe での SCORM 出力においては、上記のうち index.html だけが開発したモジュールによって出力時に生成されて出力される部分である。今回新規開発したモジュールも上記図 14 のファイル群のうち、この index.html だけを生成している。SCORM パッケージとして動作に必要となる Javascript は、共通的に用いられるものが common.js ファイルに収められ、LMS とのやり取りに用いられものは APIWrapper.js や SCOFuctions.js ファイルに収められる。これらに収められない独自の Javascript が index.html の中に埋め込まれた形で、eXe によって動的に作り出される。

ここではサンプルとして設問 1 問と解答欄二つを持った実際の SCORM 形式での zip 出力に書き出された index.html を掲載する。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>eXe</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- Created using eXe: http://exelearning.org -->
<style type="text/css">
@import url(base.css);
@import url(content.css);
</style>
<script type="text/javascript" src="common.js"></script>
</head>
<script type="text/javascript" src="APIWrapper.js"></script>
```

```

<script type="text/javascript" src="SCOFunctions.js"></script>
<body onload="loadPage()" onunload="unloadPage()"><div id="outer">
<div id="main">
<div id="nodeDecoration">
<p id="nodeTitle">
ホーム</p></div>
<div class="ScormMnIddevice" id="id0">
<script type="text/javascript">
<!-- //
var numQuestions = 1;           ←質問数が格納される
var rawScore = 0;
var actualScore = 0;
var question0Answer0;          } 学習者の記入した解答が格納される変数を準備
var question0Answer1;          }
var key0Answer0low = 21.0;      } 解答欄 1 の正答範囲 (low~high) が格納される
var key0Answer0high = 21.0;     }
var key0Answer1low = -12.0;     } 解答欄 2 の正答範囲 (low~high) が格納される
var key0Answer1high = -12.0;
function getAnswer()
{
    doLMSSetValue("cmi.interactions.0.id","key0b0");
    doLMSSetValue("cmi.interactions.0.type","performance");
    doLMSSetValue("cmi.interactions.0.correct_responses.0.pattern",
        "21plus_minus0");
    doLMSSetValue("cmi.interactions.0.correct_responses.1.pattern",
        "-12plus_minus0");

    question0Answer0 = document.getElementById("quizForm0").key
0b0Answer0.value;

    question0Answer1 = document.getElementById("quizForm0").key
0b0Answer1.value;

doLMSSetValue("cmi.interactions.0.student_response",question0Answer0+" "+
question0Answer1);
}
function calcRawScore(){
    if ( question0Answer0 &gt;= key0Answer0low &amp; question0Answer0 &lt;=
key0Answer0high &amp; question0Answer1 &gt;= key0Answer1low &amp; question0Ans
wer1 &lt;= key0Answer1high )
    {
        doLMSSetValue("cmi.interactions.0.result","correct");
        rawScore++;
    }
    else
    {
        doLMSSetValue("cmi.interactions.0.result","wrong");
    }
}

function calcScore2()
</pre>
<p>以下、一連の doLMSSetValue に<br/>よって、LMS へ情報を格納する</p>
<p>解答が正解かの判定と設問に対する<br/>累積の正解数カウントを行う</p>
</div>
<div data-bbox="508 905 534 921" data-label="Page-Footer">
<p>30</p>
</div>
```

```

    {
        computeTime(); // the student has stopped here.

        document.getElementById("quizForm0").submitB.disabled = true;

        getAnswer();

        calcRawScore();
        actualScore = Math.round(rawScore / numQuestions * 100);
        alert("Your score is " + actualScore + "%");

        doLMSSetValue( "cmi.core.score.raw", actualScore+"" );

        var mode = doLMSGetValue( "cmi.core.lesson_mode" );

        if ( mode != "review" && mode != "browse" ){
            if ( actualScore < 50 )
            {
                doLMSSetValue( "cmi.core.lesson_status", "failed" );
            }
            else
            {
                doLMSSetValue( "cmi.core.lesson_status", "passed" );
            }

            doLMSSetValue( "cmi.core.exit", "" );
        }

        exitPageStatus = true;

        doLMSCommit();

        doLMSFinish();

    }
//]]> -->
</script>
<form name="quizForm0" id="quizForm0" action="javascript:calcScore2();">
<div class="iDevice emphasis1">

<span class="iDeviceTitle">線形時不変システム</span>
<div class="iDevice_inner">
<div class="passrate" value="50"></div>
<div class="question">
<div id="taquestion0b0" class="block" style="display:block">ある線形時不変システムに時刻0で、ある信号を入力したとき、5、6サンプル時刻後の出力値はそれぞれ7、-4であった。また、10サンプル時刻以降の出力値は0であった。<br />
<br />
50サンプル時刻にこのシステムに先の入力信号の3倍を入力したとき、その入力から5、6サンプル時刻後のそれぞれの出力値はいくらとなるか。<br />
<br />
小数点以下第2位を四捨五入して記入しなさい。

```

累積の正解数カウントと全体の設問数より得点計算を行う

```

</div><br/>
<table><tr><td><br><input type="text" name="key0b0Answer0" id="key
0b0Answer0" value="" size="5" />
</td><td>
<div id="taoptionAnswer0q0b0" class="block" style="display:block"> 5 サンプル
時刻後の出力値を答えよ
</div></td></tr>
<tr><td><br><input type="text" name="key0b0Answer1" id="key0b0Ans
wer1" value="" size="5" />
</td><td>
<div id="taoptionAnswer1q0b0" class="block" style="display:block"> 6 サンプル
時刻後の出力値を答えよ
</div></td></tr>
</table></div>
<br/><input type="submit" name="submitB" value="解答を提出"/>
</div></div>
</form>
</div>
</div>
</div>
</body></html>

```

図 15：サンプル問題の SCORM 出力物に含まれる index.html

3.6. 開発モジュールの使用例と動作確認

本節では、開発した新規作成モジュールの具体的な利用方法について示す。今回開発したテスト作成モジュールの特徴は、正解値に範囲指定が設定可能な数値解を扱え、なおかつその解答欄を複数追加できる点にある。こういった問題種別にマッチする問題としては、

- 計算過程で誤差が発生する問題、
- 図表などから値を読み取る関係で誤差が生じる問題、
- 問題文章において、いくつかの小問があり、それぞれに答えさせる問題、
- 解答を求めると複数の解が出てくる問題、

などが挙げられる。今回は開発した問題作成モジュールが活用できる具体例として、eXe で次のような問題を作成した。問題作成から SCORM 形式への出力までの詳細な手順については付録 1 に示した。

図1-aに示すトランジスタ増幅回路において、この回路のトランジスタの I_B - V_{BE} 特性、 I_C - I_B 特性及び I_C - V_{CE} 特性がそれぞれ図1-b～図1-dで示されるとき、動作点のコレクタ電流 I_C は、解答欄（ア）ミリアンペアであり、コレクターエミッタ間の電圧 V_{CE} は、解答欄（イ）ボルトである。

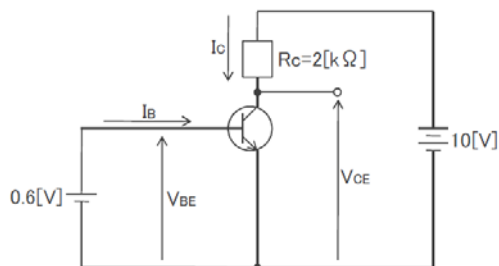


図1-a

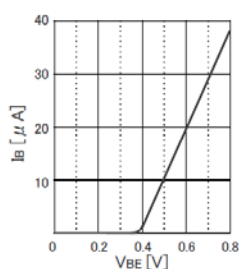


図1-b

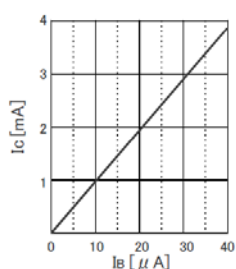


図1-c

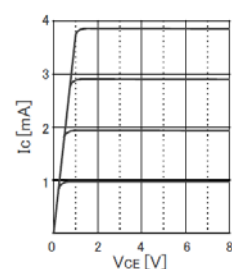


図1-d

図16：問題作成例として使用した問題内容

本来、eXe はコンテンツエディタとしての強力な機能を備えている。テスト作成時の問題文章は、簡易なワープロレベルの書式設定が可能であり、太字、下線、センタリング、ナンバリング、表組み、などが GUI 操作によって容易に行える。また、画像ファイルの添付だけでなく、数式入力方法として LaTeX ソースを画像変換し問題内に配置する機能などもあり、表現能力は非常に高い。図17に eXe での問題作成中の画面を掲載する。

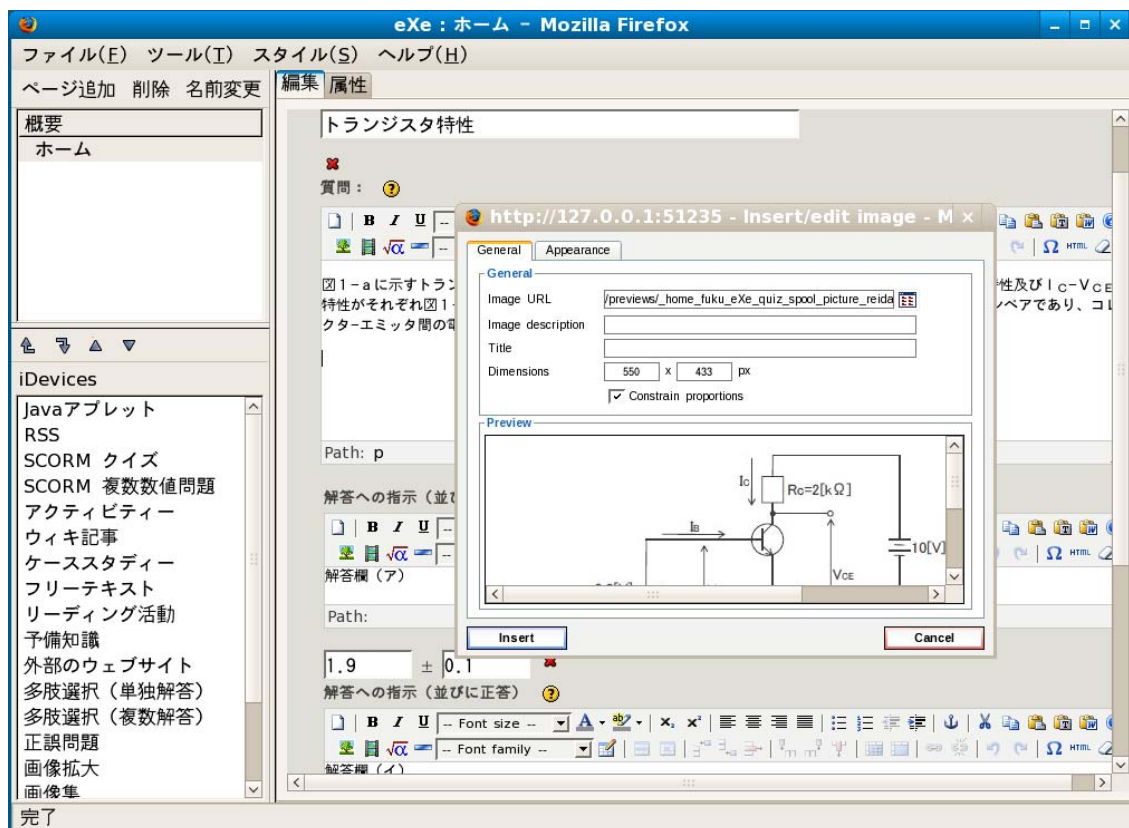


図 17：eXe での問題作成中の画面

次に、作成した問題を SCORM 形式で出力し、Blackboard Learning System、Moodle の各 LMS に読み込ませ動作確認を行った様子を記す。上記問題では、解答欄（ア）は 1.9 ± 0.1 （つまり、 $1.8 \sim 2.0$ が正解範囲）、解答欄（イ）は 6.2 ± 0.2 （ $6.0 \sim 6.4$ が正解範囲）を正解とした。この問題に対しそれぞれの LMS で正しく採点が行われることを、正解範囲の閾値を中心に以下の表にて確認をおこなった。本問題における有効数字は 2 桁であるが、今回の動作確認では、小数点以下 10 桁までチェックを行った。

表 4：作成した SCORM モジュールの各 LMS での動作確認表

	学習者の解答		Blackboard Learning System	Moodle
	解答欄(ア) 正解範囲 1.8～2.0	解答欄(イ) 正解範囲 6.0～6.4		
本来正解と 判断すべき 組み合わせ	1.9	6.2	○(正常動作)	○(正常動作)
	1.8	6.0	○(正常動作)	○(正常動作)
	1.8	6.4	○(正常動作)	○(正常動作)
	2.0	6.0	○(正常動作)	○(正常動作)
	2.0	6.4	○(正常動作)	○(正常動作)
本来不正解と 判断すべき 組み合わせ	1.7999999999	6.0	×(正常動作)	×(正常動作)
	1.7999999999	6.4	×(正常動作)	×(正常動作)
	2.0000000001	6.0	×(正常動作)	×(正常動作)
	2.0000000001	6.4	×(正常動作)	×(正常動作)
	1.8	5.9999999999	×(正常動作)	×(正常動作)
	2.0	5.9999999999	×(正常動作)	×(正常動作)
	1.8	6.4000000001	×(正常動作)	×(正常動作)
	2.0	6.4000000001	×(正常動作)	×(正常動作)

注：○は SCORM モジュールが正解と採点、×は不正解と採点したことを表す

表 4 を見てみると、本来正解とすべき解答値の組合せは全て正解と採点され、不正解とすべき解答値の組み合わせは全て不正解と採点できており、正常な動作状況が確認できた。

各 LMS への学習者受験情報等のデータ格納状況については、3.7 節 出力モジュールと LMS との連動にて詳述する。

3.7. 出力モジュールと LMS との連動

新規開発した拡張モジュールにて出力した SCORM モジュールを Blackboard Learning System ならびに Moodle に格納し、動作させたところ次のような形で LMS 側に情報が格納され表示された。

3.7.1. Blackboard Learning System

Blackboard Learning System では、SCORM パッケージが LMS への通信で格納する情報のうち、以下のようなかたちで講師側から情報の閲覧が可能である。

- ・ 得点 cmi.core.score.raw 値の例：100
- ・ 状態 cmi.core.lesson_status 値の例：passed
- ・ 合計時間 cmi.core.session_time 値の例：00:02:33.26

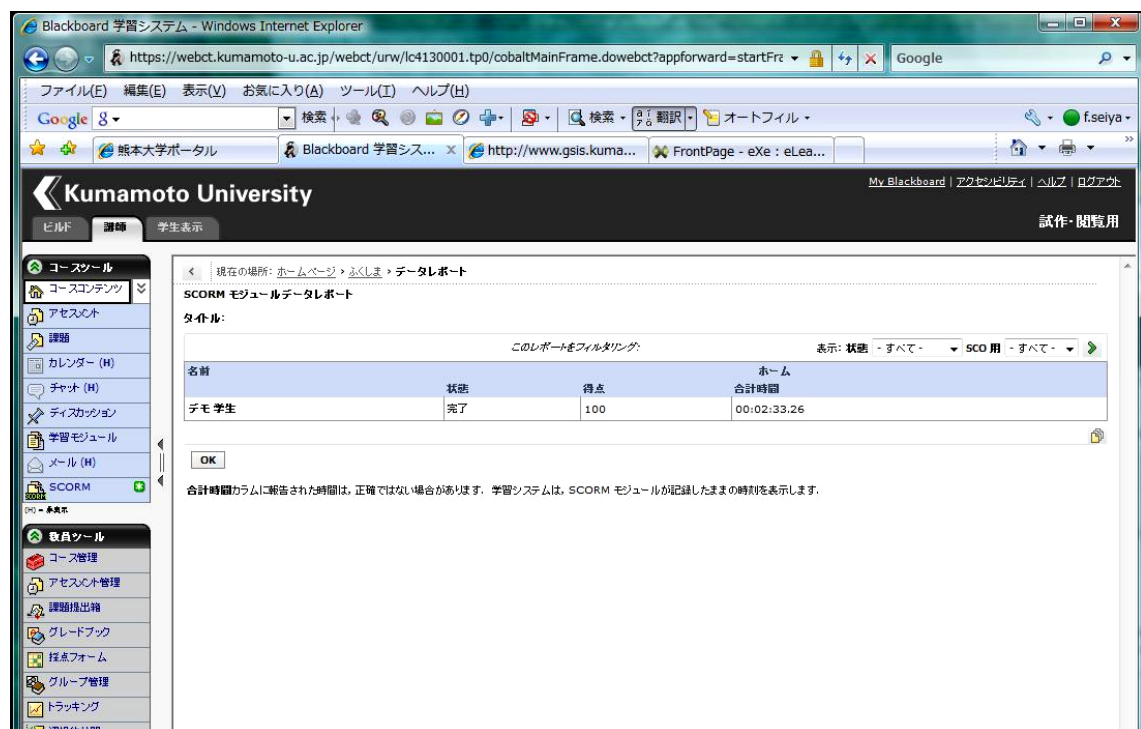


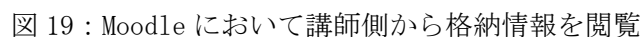
図 18 : Blackboard Learning System において講師側から格納情報を閲覧

Blackboard Learning System での特徴は、3.4.3. 新規開発モジュールの追加の手順 5. において図 9 として記した多くの LMS 格納情報のうち、講師側から閲覧できる項目が非常に少ない点である。ただし、これは SCORM モジュール利用時にどこまでの情報を講師や受験者に見せるかという LMS 側アプリケーションの設計思想であるので、開発モジュールとしては今回の開発物のように、たとえ現状の LMS で情報閲覧ができなくても、情報を格納しておくように作成しておくべきである。

3.7.2. Moodle

Moodle では、SCORM パッケージが LMS への通信で格納する情報のうち、以下のよう

- 実評点 `cmi.core.score.raw` 値の例: 100
- ステータス `cmi.core.lesson_status` 値の例: passed
- 時間 `cmi.core.session_time` 値の例: 00:00:08.92
- 正答 `cmi.interactions.X.correct_responses.Y.pattern`
値の例: 21plus_minus0.5 (21±0.5 の意味)
- 問題 ID `cmi.interactions.X.id` 値の例: keyXb0
- 正誤結果 `cmi.interactions.X.result` 値の例: correct
- 学習者の答え `cmi.interactions.X.student_response`
値の例: 21 -12 (解答欄 1 が 21、解答欄 2 が -12 の意味)
- 問題タイプ `cmi.interactions.X.type` 値の例: performance



37

においては、LMS に格納する情報のうち正答情報を格納する項目 (cmi.interactions.X.correct_responses.Y.pattern) は複数の欄を設けられる (X にはその問題に割り当てた通し番号等が入り、Y の部分には解答としての通し番号等が入る)。しかし学習者の解答については、1 つの問題に対し記入する項目が 1 つしかない。そのため、今回の開発においては、学習者の入力した複数の解答は単純に空白で区切った形で 1 つの項目に収まるようにして併記するようにしている。

4. まとめと考察

本研究では、現在の e ラーニングを取り巻く自然科学系へのテストツールの対応状況改善に向け、SCORM 出力が可能なオープンソース XHTML エディタ eXe のテスト作成機能の拡張を行った。今回の拡張では、正解とする数値を範囲指定できる出題を可能とし、更に複数の解答欄を設けられるテスト問題作成モジュールを開発した。この開発によって、自然科学系をはじめとする広い分野において、SCORM を利用した数値問題が利用可能となり、学習の理解度を把握するための手段を増やすことができた。

特に、実装されている SCORM での出題機能が択一式問題のみであった eXe を機能強化した側面だけでなく、現状の SCORM 形式問題作成ツールにおいて対応ツールが非常に少ない計算問題や数値問題に対応したことの意義は大きい。つまり、今回 SCORM テストコンテンツとして数値問題を GUI 操作で作成可能としたことで、eXe が本来持っていた多くのコンテンツ作成機能を更に便利にし、同時に eXe を SCORM 対応の数値問題作成ツールとして希少なツールという位置付けとすることができた。

本研究の成果としてもう一つ重要な点は、オープンソースとして公開されながら、その解説資料が少なかった eXe のアプリケーション構造について明らかにできたところにある。特に、テスト問題作成モジュールに関する構造については、実際のモジュール開発を例に詳細にソースを紹介する事ができたため、これからの eXe の更なる発展においてその指針を残すことができた。

今回開発のモジュールに対し、今後さらに改善を行うべき部分として、以下に課題を挙げてみると、

- ・現在は複数解答欄の全ての解答欄が正解の場合のみ、問題としての正解としている。複数解答欄の正解割合に応じて点数を付与する機能を実装する。
- ・現在の複数解答欄への答え方は、問題文章の指定に従ってそれぞれの問いに対応する答えを書くことになるが、順不同で正解とする問題にも対応させる。
- ・自然科学系に強い出題モジュールとするために、問題文中に埋め込んだ変数を予め指定した範囲でランダムに発生させ、更にそのランダムに発生した数値を用いて採点を行えるようにする。

といった点が挙げられる。これらのうち、複数解答に対する部分点設定機能や、順

不同採点機能の実装についてはそれほど難しくない。3 つ目のランダム数値による出題採点については、かなり動的な問題作成機能となり、python 言語のみでなく javascript も含めた複雑な開発作業になる。しかしこの機能まで実装されると、講師側は問題のテンプレートとなる部分をオーサリングツールで作成するだけで、実際の出題時にランダムに数値や解答の異なる問題を出題でき、講師側の問題作成作業負担を軽減できるうえ、解答の暗記や学習者同士の解答情報交換による不正にも対処することができる。恐らく現在世の中に出回っている SCORM 出力対応の問題の中でも、自然科学系への対応が最も強力なツールとなってくる。これらの対応を今後も継続し公開していくことで SCORM 対応の問題作成モジュールがより便利なものとなり、同時に本研究を契機に多くの開発者たちが、対応できる問題のバリエーションを増やしていくことを期待する。

謝辞

本研究を進めるにあたってお世話になりました多くの方々に、ここに感謝の意を表します。

常に温かい目で一つ一つ目の前のステップを上るための最大限のアドバイスを頂きました松葉龍一准教授に最大限の謝意を表します。研究期間延長の願いを心に留め最後まで修了に向けて進めようと配慮頂きました先生のお力がなければ、ここまで辿り着くことはなかったかと思います。また、私とソースコードとの睨み合いを正しい方向に向かわせ、注意すべきポイントを適宜アドバイス頂きました中野裕司教授、私の耐え得る最大限のプレッシャーを ID 理論に基づいて絶妙に与えて頂きました鈴木克明教授に心から感謝致します。先生方の知見の深さと懐の広さがなければ、本研究の今後の広がりを追求できなかったと思います。そして、全てのお名前を記すことはできませんが、本専攻におきまして合宿、発表を通じ研究の進展に有意なアドバイスを常に頂きました本専攻の多くの先生方に感謝申し上げます。

また、ここまで苦楽を共にして最後までやり遂げる仲間意識を共有させて頂いた先輩方、同期生の方々そして後輩の皆さんに感謝致します。そして仕事と学習の両立の中で、最も犠牲となった家庭において最後まで協力をしてくれた家族ならびに親族の皆さんに感謝致します。

最後に e ラーニングコンテンツ作成に向け、高機能かつ多様な出力形式に対応したこのツール eXe をオープンソースとして公開し、これまで発展させてきた eXe Project のメンバーを始めとする世界中の多くの皆様に感謝します。

参考文献・参考サイト

- [1] 鈴木克明, 詳説 インストラクショナルデザイン e ラーニングファンダメンタル, 日本イーラーニングコンソーシアム, 2004.
- [2] William Horton 著, 日本コンサルタントグループラーニングセンター訳, e ラーニング導入読本 教育担当者のための WBT マニュアル, 日本コンサルタントグループ, 2001.
- [3] 赤堀侃司, 教育工学への招待 教育の問題解決の方法論, ジャストシステム出版部, 2002.
- [4] 市川尚・高橋暁子・鈴木克明, 複数の制御構造の適用と学習のための統合型ドリルシェル「ドリル工房」の開発, 日本教育工学会論文誌, 32(2), pp.157-168, 2008.
- [5] Hot Potatoes, <http://hotpot.uvic.ca/>
- [6] 鈴木克明, 教材設計マニュアルー独学を支援するためにー, 北大路書房, 2002.
- [7] 池田央, テストの科学 試験にかかわるすべての人に, 日本文化科学社, 1992.
- [8] 安浪誠祐, 教材作成ツール Hot Potatoes を用いた WebCT 用のテストの作成について, (http://www.webct.jp/c2005/proc/p2_yasunami_doc.pdf), 2005.
- [9] Moodle, <http://moodle.org/>
- [10] Blackboard Learning System, <http://www.blackboard.com/>
- [11] WebClass, <http://www.webclass.jp/>
- [12] アメリカ国防総省 ADL, <http://www.adlnet.gov/>
- [13] 日本イーラーニングコンソーシアム公開 SCORM 2004 日本語版, <http://www.elc.or.jp/aen/content/japan/act2005/>
- [14] 日本イーラーニングコンソーシアム公開 SCORM Version1.2 日本語版, <http://www.elc.or.jp/cgi-bin/csvmail/kigyouscormdownload.html>
- [15] IMS Global Learning Consortium (QTI フォーマット) <http://www.imsglobal.org/question/>
- [16] JISC CETIS (centre for educational technology & interoperability standards), http://wiki.cetis.ac.uk/QTI_Training_Guide
- [17] eXe eLearning XHTML editor 公式サイト (eXe project), <http://exelearning.org/>
- [18] IMS Global Learning Consortium, <http://www.imsglobal.org/>
- [19] Python, <http://www.python.org/>

- [20] 日本 Python ユーザ会, <http://www.python.jp/Zope/>
- [21] Alex Martelli 著 (株)クイープ訳, Python クイックリファレンス, オライリー・ジャパン.
- [22] Fedora Project, <http://fedoraproject.org/>
- [23] 篠原竜雄, SCORM 準拠の e ラーニング・コンテンツ開発の実際, IBM 社 ProVISIONFall12002No. 35,
http://www-06.ibm.com/jp/provision/no35/pdf/35_cpr2.pdf

付録

付録 1 新規開発モジュールによるテスト問題作成手順

実際に、開発したモジュールを使用する際の手順は以下のようになる。

- ・作成モジュールの選択

画面左側の iDevices 一覧より、「SCORM 複数数値問題」を選択する。

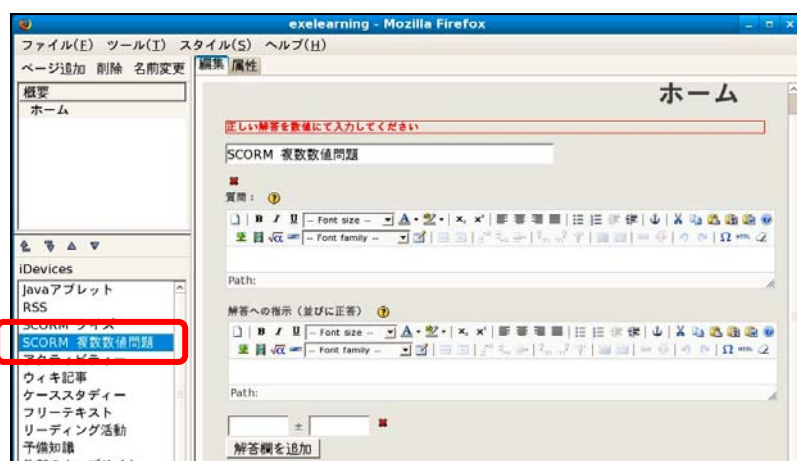


図 20：問題作成 iDevice 選択

- ・タイトル、問題文章、各解答部の文章、正答並びに許容誤差の記入

タイトルを書き換え、質問欄に問題文章を記入する。また解答欄への指示の欄へ解答を書くにあたっての指示を記入する。その下に2つならんでいる空欄に正答と許容誤差を記入する。

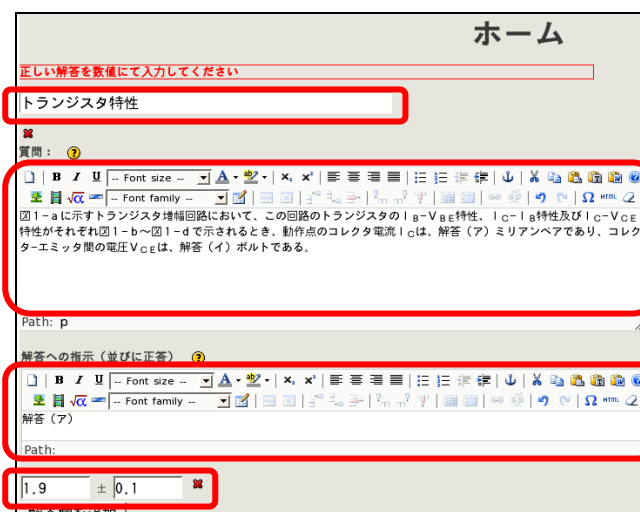


図 21：問題文章等の入力

問題文章中に図を張り付ける場合には、ツールバー内のアイコンをクリックし、添付したい画像を選択する。

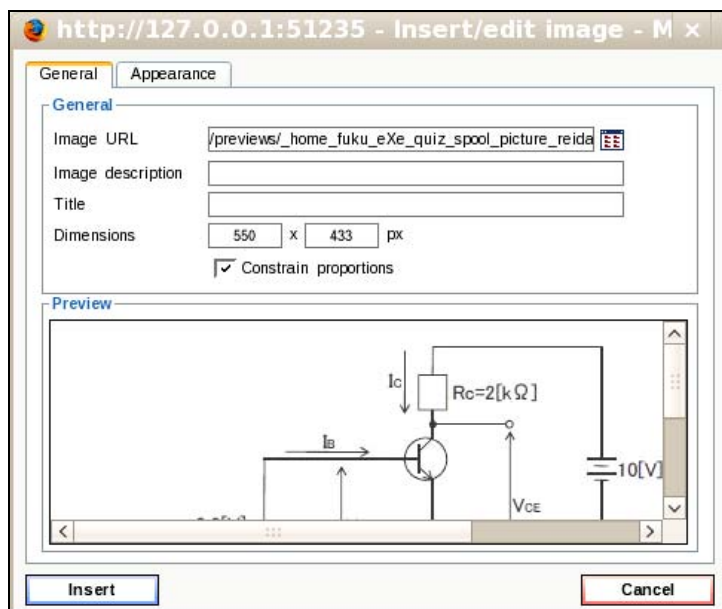


図 22：添付画像選択画面

- ・複数解答欄の追加

「解答欄を追加」のボタンを押すことによって、解答欄を増やすことができるので、そこへも解答の際の指示文章と、正答ならびに許容誤差の数値を記入する。

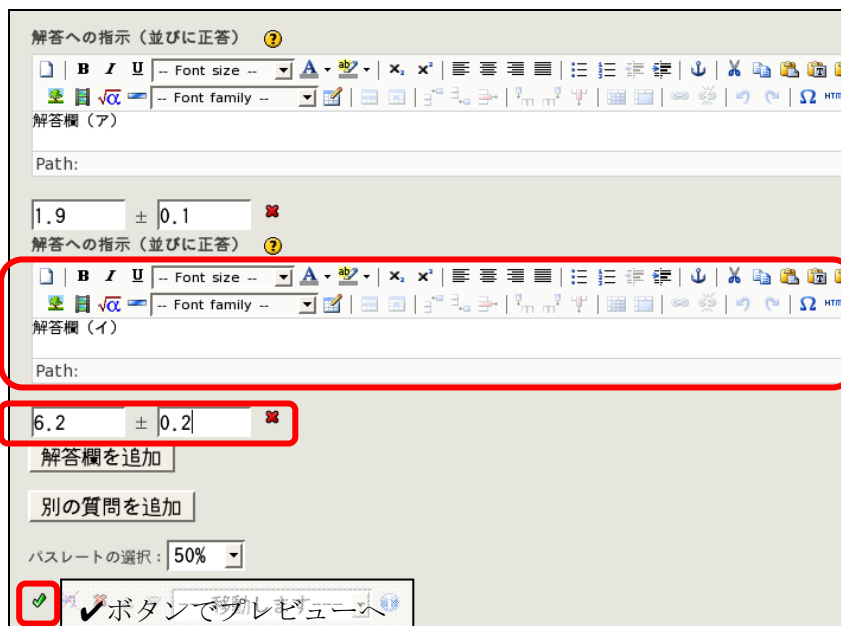


図 23：追加された解答欄へ解答指示と正答を入力

- ・プレビューにて動作確認

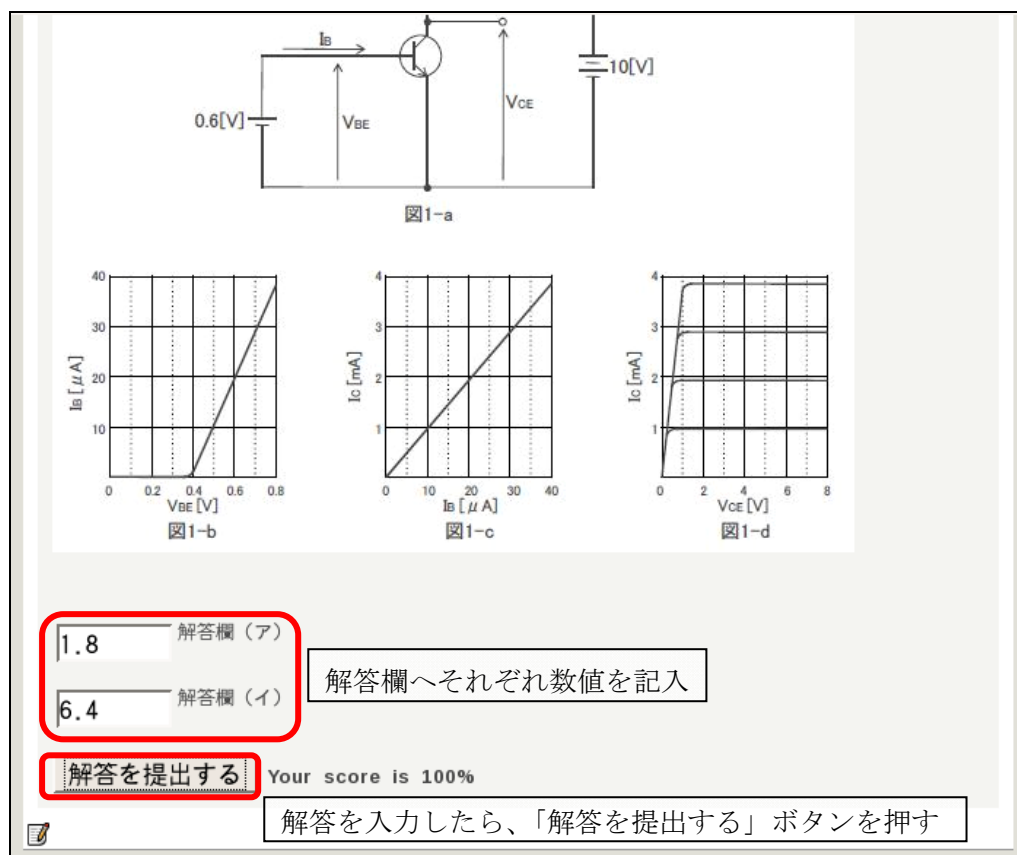


図 24 : プレビュー画面にて動作確認

- ・ SCORM 出力

画面上部のメニューより「ファイル」→「エクスポート」→「SCORM 1.2」を選ぶことにより、SCORM 形式での保存が可能となる。

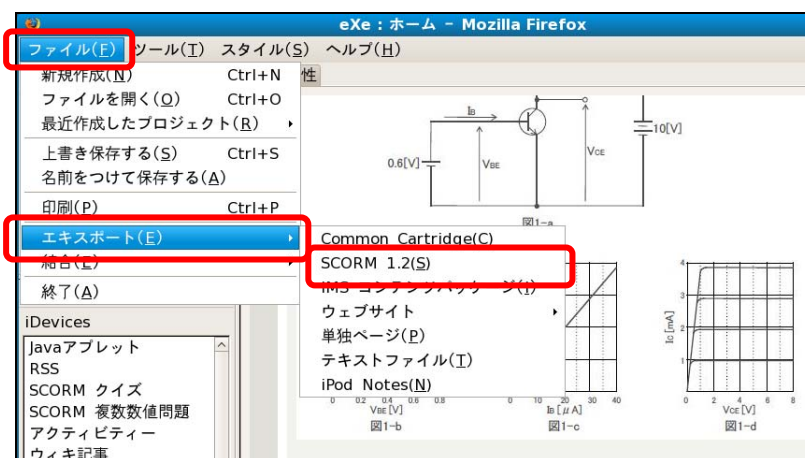


図 25 : SCORM 形式への出力

付録 2 既存ソースへの追記内容

ファイル 1 ～ ファイル 3 まで（太字下線部分が追記箇所）

ファイル 1 : %(install dir)%/exe/engine/idevicestore.py

```
# =====
# eXe
# Copyright 2004-2006, University of Auckland
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
# =====
"""
The collection of iDevices available
"""

from exe.engine import persist
from exe.engine.idevice import Idevice
from exe.engine.field import TextAreaField, FeedbackField
from nevow.flat import flatten

import imp
import sys
import logging

log = logging.getLogger(__name__)

# =====
class IdeviceStore:
    """
    The collection of iDevices available
    """
    def __init__(self, config):
        """
        Initialize
        """
        self._nextIdeviceId = 0
        self.config = config
        self.extended = []
        self.generic = []
        self.listeners = []

    def getNewIdeviceId(self):
        """
        Returns an iDevice Id which is unique
        """
        id_ = unicode(self._nextIdeviceId)
```

```

        self._nextdeviceId += 1
        return id_

    def getIdevices(self):
        """
        Get the idevices which are applicable for the current node of
        this package
        In future the idevices which are returned will depend
        upon the pedagogical template we are using
        """
        return self.extended + self.generic

    def delGenericIdevice(self, idevice):
        """
        Delete a generic idevice from idevicestore.
        """
        self.generic.remove(idevice)

    def register(self, listener):
        """
        Register a listener who is interested in changes to the
        IdeviceStore.
        Created for IdevicePanels, but could be used by other objects
        """
        self.listeners.append(listener)

    def addIdevice(self, idevice):
        """
        Register another iDevice as available
        """
        log.debug("IdeviceStore.addIdevice")
        # idevice prototypes need to be in edit mode
        idevice.edit = True
        self.generic.append(idevice)
        for listener in self.listeners:
            listener.addIdevice(idevice)

    def load(self):
        """
        Load iDevices from the generic iDevices and the extended ones
        """
        log.debug("load iDevices")
        idevicesDir = self.config.configDir/'idevices'
        if not idevicesDir.exists():
            idevicesDir.mkdir()
        self.__loadExtended()
        self.__loadGeneric()

    def __loadExtended(self):
        """
        Load the Extended iDevices (iDevices coded in Python)
        """
        # self.__loadUserExtended()

        from exe.engine.freetextidevice import FreeTextIdevice
        from exe.engine.multimediaidevice import MultimediaIdevice
        from exe.engine.reflectionidevice import ReflectionIdevice
        from exe.engine.casestudyidevice import CaseStudyIdevice

```



```

from exe.engine.truefalsedevice import TrueFalseDevice
# converting ImageWithTextDevice -> FreeTextDevice:
#from exe.engine.imagewithtextdevice import ImageWithTextDevice
from exe.engine.wikipediaiddevice import WikipediaDevice
from exe.engine.attachmentiddevice import AttachmentDevice
from exe.engine.titledevice import TitleDevice
from exe.engine.galleryiddevice import GalleryDevice
from exe.engine.clozeiddevice import ClozeDevice
from exe.engine.flashwithtextdevice import FlashWithTextDevice
from exe.engine.externalurliddevice import ExternalUrlDevice
from exe.engine.imagemagnifieriddevice import ImageMagnifierDevice
# converting Maths IDevice -> FreeTextDevice:
#from exe.engine.mathiddevice import MathDevice
from exe.engine.multichoiceiddevice import MultipleChoiceDevice
from exe.engine.rssiddevice import RssDevice
from exe.engine.multiselectiddevice import MultipleSelectDevice
from exe.engine.appletiddevice import AppletDevice
from exe.engine.flashmovieiddevice import FlashMovieDevice
from exe.engine.quiztestiddevice import QuizTestDevice
from exe.engine.scormmniddevice import ScormMnDevice

```

追記

```

self.extended.append(FreeTextDevice())

self.extended.append(MultipleChoiceDevice())

self.extended.append(ReflectionDevice())

self.extended.append(CasestudyDevice())
self.extended.append(TrueFalseDevice())

defaultImage = unicode(self.config.webDir/"images"/"sunflowers.jpg")

# converting ImageWithTextDevice -> FreeTextDevice:
#self.extended.append(ImageWithTextDevice(defaultImage))

self.extended.append(ImageMagnifierDevice(defaultImage))

defaultImage = unicode(self.config.webDir/"images"/"sunflowers.jpg")
defaultSite = 'http://%s.wikipedia.org/' % self.config.locale
self.extended.append(WikipediaDevice(defaultSite))
self.extended.append(AttachmentDevice())
self.extended.append(GalleryDevice())
self.extended.append(ClozeDevice())
self.extended.append(FlashWithTextDevice())
self.extended.append(ExternalUrlDevice())
# converting Maths IDevice -> FreeTextDevice:
#self.extended.append(MathDevice())
self.extended.append(MultipleMediaDevice())
self.extended.append(RssDevice())
self.extended.append(MultipleSelectDevice())
self.extended.append(AppletDevice())
self.extended.append(FlashMovieDevice())
self.extended.append(QuizTestDevice())
self.extended.append(ScormMnDevice())

# generate new ids for these iDevices, to avoid any clashes
for iddevice in self.extended:
    iddevice.id = self.getNewDeviceId()

```

追記

```

def __loadUserExtended(self):
    """
    Load the user-created extended iDevices which are in the iddevices

```

```

directory
"""

idevicePath = self.config.configDir/'idevices'
log.debug("load extended iDevices from "+idevicePath)

if not idevicePath.exists():
    idevicePath.makedirs()
sys.path = [idevicePath] + sys.path

# Add to the list of extended idevices
for path in idevicePath.listdir("*.py"):
    log.debug("loading "+path)
    moduleName = path.basename().splitext()[0]
    module = __import__(moduleName, globals(), locals(), [])
    module.register(self)

# Register the blocks for rendering the idevices
for path in idevicePath.listdir("*.block.py"):
    log.debug("loading "+path)
    moduleName = path.basename().splitext()[0]
    module = __import__(moduleName, globals(), locals(), [])
    module.register()

def __loadGeneric(self):
    """
    Load the Generic iDevices from the appdata directory
    """
    genericPath = self.config.configDir/'idevices'/'generic.data'
    log.debug("load generic iDevices from "+genericPath)
    if genericPath.exists():
        self.generic = persist.decodeObject(genericPath.bytes())
        self.__upgradeGeneric()
    else:
        self.__createGeneric()

# generate new ids for these iDevices, to avoid any clashes
for idevice in self.generic:
    idevice.id = self.getNewIdeviceId()

def __upgradeGeneric(self):
    """
    Upgrades/removes obsolete generic idevices from before
    """
    # We may have two reading activities,
    # one probably has the wrong title,
    # the other is redundant
    readingActivitiesFound = 0
    for idevice in self.generic:
        if idevice.class_ == 'reading':
            if readingActivitiesFound == 0:
                # Rename the first one we find
                idevice.title = x_(u"Reading Activity")
                # and also upgrade its feedback field from using a simple
                # string, to a subclass of TextAreaField.
                # While this will have been initially handled by the
                # field itself, and if not, then by the genericidevice's
                # upgrade path, this is included here as a possibly
                # painfully redundant safety check due to the extra
                # special handling of generic idevices w/ generic.dat
                for field in idevice.fields:
                    if isinstance(field, FeedbackField):
                        # must check for the upgrade manually, since

```

```

# persistence versions not used here.
# (but note that the persistence versioning
# will probably have ALREADY happened anyway!)
if not hasattr(field,"content"):
    # this FeedbackField has NOT been upgraded:
    field.content = field.feedback
    field.content_w_resourcePaths = field.content
    field.content_wo_resourcePaths = field.content
else:
    # Destroy the second
    self.generic.remove(iddevice)
    readingActivitiesFound += 1
    if readingActivitiesFound == 2:
        break
self.save()

def __createGeneric(self):
    """
    Create the Generic iDevices which you get for free
    (not created using the iDevice editor, but could have been)
    Called when we can't find 'generic.data', generates an initial set of
    free/builtin idevices and writes the new 'generic.data' file
    """
    from exe.engine.genericidevice import GenericIdevice

    readingAct = GenericIdevice(_(u"Reading Activity"),
                                u"reading",
                                _(u"University of Auckland"),
                                x_(u""<p>The Reading Activity will primarily
be used to check a learner's comprehension of a given text. This can be done
by asking the learner to reflect on the reading and respond to questions about
the reading, or by having them complete some other possibly more physical task
based on the reading.</p>"""),
                                x_(u""<p>Teachers should keep the following "
                                "in mind when using this iDevice: </p>"
                                "<ol>"
                                "<li>"
                                "Think about the number of "
                                "different types of activity "
                                "planned for your resource that "
                                "will be visually signalled in the "
                                "content. Avoid using too many "
                                "different types or classification "
                                "of activities otherwise learner "
                                "may become confused. Usually three "
                                "or four different types are more "
                                "than adequate for a teaching "
                                "resource."
                                "</li>"
                                "<li>"
                                "From a visual design "
                                "perspective, avoid having two "
                                "iDevices immediately following "
                                "each other without any text in "
                                "between. If this is required, "
                                "rather collapse two questions or "
                                "events into one iDevice. "
                                "</li>"
                                "<li>"
                                "Think "
                                "about activities where the "
                                "perceived benefit of doing the "
                                "activity outweighs the time and "
                                "effort it will take to complete "

```

```

        "the activity. "
        "</li>"
        "</ol>"))
        readingAct.emphasis = Idevice.SomeEmphasis
        readingAct.addField(TextAreaField(_(u"What to read"),
        _(u""Enter the details of the reading including reference details. The
        referencing style used will depend on the preference of your faculty or
        department.""")))
        readingAct.addField(TextAreaField(_(u"Activity"),
        _(u""Describe the tasks related to the reading learners should undertake.
        This helps demonstrate relevance for learners.""")))

        readingAct.addField(FeedbackField(_(u"Feedback"),
        _(u""Use feedback to provide a summary of the points covered in the reading,
        or as a starting point for further analysis of the reading by posing a question
        or providing a statement to begin a debate.""")))

        self.generic.append(readingAct)

        objectives = GenericIdevice(_(u"Objectives"),
        u"objectives",
        _(u"University of Auckland"),
        _(u""Objectives describe the expected outcomes of the learning and should
        define what the learners will be able to do when they have completed the
        learning tasks."""),
        u"")
        objectives.emphasis = Idevice.SomeEmphasis

        objectives.addField(TextAreaField(_(u"Objectives"),
        _(u""Type the learning objectives for this resource.""")))
        self.generic.append(objectives)

        preknowledge = GenericIdevice(_(u"Preknowledge"),
        u"preknowledge",
        "",
        "",
        _(u""Prerequisite knowledge refers to the knowledge learners should already
        have in order to be able to effectively complete the learning. Examples of
        pre-knowledge can be: <ul>
        <li>Learners must have level 4 English </li>
        <li>Learners must be able to assemble standard power tools </li></ul>
        """), u"")
        preknowledge.emphasis = Idevice.SomeEmphasis
        preknowledge.addField(TextAreaField(_(u"Preknowledge"),
        _(u""Describe the prerequisite knowledge learners should have to effectively
        complete this learning.""")))
        self.generic.append(preknowledge)

        activity = GenericIdevice(_(u"Activity"),
        u"activity",
        _(u"University of Auckland"),
        _(u""An activity can be defined as a task or set of tasks a learner must
        complete. Provide a clear statement of the task and consider any conditions
        that may help or hinder the learner in the performance of the task."""),
        u"")
        activity.emphasis = Idevice.SomeEmphasis
        activity.addField(TextAreaField(_(u"Activity"),
        _(u""Describe the tasks the learners should complete.""")))
        self.generic.append(activity)

        self.save()

    def __createReading011(self):
        """

```

Create the Reading Activity 0.11

We do this only once when the user first runs eXe 0.11

"""

from exe.engine.genericidevice import GenericIdevice

readingAct = GenericIdevice(_(u"Reading Activity 0.11"),

u"reading",

_(u"University of Auckland"),

x_(u""<p>The reading activity, as the name

suggests, should ask the learner to perform some form of activity. This activity should be directly related to the text the learner has been asked to read.

Feedback to the activity where appropriate, can provide the learner with some reflective guidance.</p>"""),

x_(u"Teachers should keep the following "

"in mind when using this iDevice: "

""

""

"Think about the number of "

"different types of activity "

"planned for your resource that "

"will be visually signalled in the "

"content. Avoid using too many "

"different types or classification "

"of activities otherwise learner "

"may become confused. Usually three "

"or four different types are more "

"than adequate for a teaching "

"resource."

""

""

"From a visual design "

"perspective, avoid having two "

"iDevices immediately following "

"each other without any text in "

"between. If this is required, "

"rather collapse two questions or "

"events into one iDevice. "

""

""

"Think "

"about activities where the "

"perceived benefit of doing the "

"activity outweighs the time and "

"effort it will take to complete "

"the activity. "

""

""))

readingAct.emphasis = Idevice.SomeEmphasis

readingAct.addField(TextAreaField(_(u"What to read"),

_(u""Enter the details of the reading including reference details. The referencing style used will depend on the preference of your faculty or department."""))

readingAct.addField(TextAreaField(_(u"Activity"),

_(u""Describe the tasks related to the reading learners should undertake. This helps demonstrate relevance for learners."""))

readingAct.addField(FeedbackField(_(u"Feedback"),

_(u""Use feedback to provide a summary of the points covered in the reading, or as a starting point for further analysis of the reading by posing a question or providing a statement to begin a debate."""))

objectives = GenericIdevice(_(u"Objectives"),

u"objectives",

_(u"University of Auckland"),

```

_(u"""Objectives describe the expected outcomes of the learning and should
define what the learners will be able to do when they have completed the
learning tasks.""",
    u""")
    objectives.emphasis = Idevice.SomeEmphasis

    objectives.addField(TextAreaField(_(u"Objectives"),
_(u"""Type the learning objectives for this resource.""")))
    self.generic.append(objectives)

    preknowledge = GenericIdevice(_(u"Preknowledge"),
    u"preknowledge",
    "",
    _(u"""Prerequisite knowledge refers to the knowledge learners should already
have in order to be able to effectively complete the learning. Examples of
pre-knowledge can be: <ul>
<li>         Learners must have level 4 English </li>
<li>         Learners must be able to assemble standard power tools </li></ul>
"""), u""")
    preknowledge.emphasis = Idevice.SomeEmphasis
    preknowledge.addField(TextAreaField(_(u"Preknowledge"),
_(u"""Describe the prerequisite knowledge learners should have to effectively
complete this learning.""")))
    self.generic.append(preknowledge)

    activity = GenericIdevice(_(u"Activity"),
    u"activity",
    _(u"University of Auckland"),
    _(u"""An activity can be defined as a task or set of tasks a learner must
complete. Provide a clear statement of the task and consider any conditions
that may help or hinder the learner in the performance of the task."""),
    u""")
    activity.emphasis = Idevice.SomeEmphasis
    activity.addField(TextAreaField(_(u"Activity"),
_(u"""Describe the tasks the learners should complete.""")))
    self.generic.append(activity)

    self.save()

def save(self):
    """
    Save the Generic iDevices to the appdata directory
    """
    idevicesDir = self.config.configDir/'idevices'
    if not idevicesDir.exists():
        idevicesDir.mkdir()
    fileOut = open(idevicesDir/'generic.data', 'wb')
    fileOut.write(persist.encodeObject(self.generic))

# =====

```

ファイル 2 : %(install dir)%/exe/engine/ package.py

```

# =====
# eXe
# Copyright 2004-2006, University of Auckland
# Copyright 2006-2008 eXe Project, http://eXeLearning.org/
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or

```

```

# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
# =====
"""
Package represents the collection of resources the user is editing
i.e. the "package".
"""

import logging
import time
import zipfile
import re
from xml.dom import import minidom
from exe.engine.path import Path, TempDirPath, toUnicode
from exe.engine.node import import Node
from exe.engine.genericidevice import GenericIdevice
from exe.engine.persist import import Persistable, encodeObject, ¥
                                decodeObject, decodeObjectRaw
from exe import import globals as G
from exe.engine.resource import import Resource
from twisted.persisted.styles import import Versioned, doUpgrade
from twisted.spread.jelly import import Jellyable, Unjellyable
from exe.engine.beautifulsoup import import BeautifulSoup
from exe.engine.field import import Field

log = logging.getLogger(__name__)

def clonePrototypeldevice(title):
    idevice = None

    for prototype in G.application.ideviceStore.getIddevices():
        if prototype.get_title() == title:
            log.debug('have prototype of:' + prototype.get_title())
            idevice = prototype.clone()
            idevice.edit = False
            break

    return idevice

def burstIdevice(idev_type, i, node):
    # given the iDevice type and the BeautifulSoup fragment i, burst it:
    idevice = clonePrototypeldevice(idev_type)
    if idevice is None:
        log.warn("unable to clone " + idev_type + " idevice")
        freetext_idevice = clonePrototypeldevice('Free Text')
        if freetext_idevice is None:
            log.error("unable to clone Free Text for " + idev_type
                    + " idevice")
            return
        idevice = freetext_idevice

    # For idevices such as GalleryImage, where resources are being attached,
    # the idevice should already be attached to a node before bursting it open:
    node.addIdevice(idevice)

```

```

idevice.burstHTML(i)
return idevice

def loadNodesIdevices(node, s):
    soup = BeautifulSoup(s)
    body = soup.find('body')

    if body:
        idevices = body.findAll(name='div',
                                attrs={'class' : re.compile('Idevice$') })
        if len(idevices) > 0:
            for i in idevices:
                # WARNING: none of the idevices yet re-attach their media,
                # but they do attempt to re-attach images and other links.

                if i.attrMap['class']=="activityIdevice":
                    idevice = burstIdevice('Activity', i, node)
                elif i.attrMap['class']=="objectivesIdevice":
                    idevice = burstIdevice('Objectives', i, node)
                elif i.attrMap['class']=="preknowledgeIdevice":
                    idevice = burstIdevice('Preknowledge', i, node)
                elif i.attrMap['class']=="readingIdevice":
                    idevice = burstIdevice('Reading Activity', i, node)
                # the above are all Generic iDevices;
                # below are all others:
                elif i.attrMap['class']=="RssIdevice":
                    idevice = burstIdevice('RSS', i, node)
                elif i.attrMap['class']=="Wikipedialdevice":
                    # WARNING: Wiki problems loading images with accents, etc:
                    idevice = burstIdevice('Wiki Article', i, node)
                elif i.attrMap['class']=="ReflectionIdevice":
                    idevice = burstIdevice('Reflection', i, node)
                elif i.attrMap['class']=="GalleryIdevice":
                    # WARNING: Gallery problems with the popup html:
                    idevice = burstIdevice('Image Gallery', i, node)
                elif i.attrMap['class']=="ImageMagnifierIdevice":
                    # WARNING: Magnifier missing major bursting components:
                    idevice = burstIdevice('Image Magnifier', i, node)
                elif i.attrMap['class']=="AppletIdevice":
                    # WARNING: Applet missing file bursting components:
                    idevice = burstIdevice('Java Applet', i, node)
                elif i.attrMap['class']=="ExternalUrlIdevice":
                    idevice = burstIdevice('External Web Site', i, node)
                elif i.attrMap['class']=="ClozeIdevice":
                    idevice = burstIdevice('Cloze Activity', i, node)
                elif i.attrMap['class']=="FreeTextIdevice":
                    idevice = burstIdevice('Free Text', i, node)
                elif i.attrMap['class']=="CasestudyIdevice":
                    idevice = burstIdevice('Case Study', i, node)
                elif i.attrMap['class']=="Multichoiceldevice":
                    idevice = burstIdevice('Multi-choice', i, node)
                elif i.attrMap['class']=="MultiSelectIdevice":
                    idevice = burstIdevice('Multi-select', i, node)
                elif i.attrMap['class']=="QuizTestIdevice":
                    idevice = burstIdevice('SCORM Quiz', i, node)
                elif i.attrMap['class']=="TrueFalseIdevice":
                    idevice = burstIdevice('True-False Question', i, node)
                追記 elif i.attrMap['class']=="ScormMnldevice":
idevice = burstIdevice('SCORM MultiNumerical Question', i, node)
                else:
                    # NOTE: no custom idevices burst yet,
                    # nor any deprecated idevices. Just burst into a FreeText:
                    log.warn("unburstable idevice " + i.attrMap['class'] +
                            "; bursting into Free Text")

```



```

        idevice = burstIdevice('Free Text', i, node)

    else:
        # no idevices listed on this page,
        # just create a free-text for the entire page:
        log.warn("no idevices found on this node, bursting into Free Text.")
        idevice = burstIdevice('Free Text', i, node)

    else:
        log.warn("unable to read the body of this node.")

def test_for_node(html_content):
    # to see if this html really is an exe-generated node
    exe_string = u"<!-- Created using eXe: http://exelearning.org -->"
    if html_content.decode('utf-8').find(exe_string) >= 0:
        return True
    else:
        return False

def loadNode(pass_num, resourceDir, zippedFile, node, doc, item, level):
    # populate this node
    # 1st pass = merely unzipping all resources such that they are available,
    # 2nd pass = loading the actual node idevices.
    titles = item.getElementsByTagName('title')
    node.setTitle(titles[0].firstChild.data)
    node_resource = item.attributes['identifierref'].value
    log.debug("** level + ' ' + titles[0].firstChild.data + '->' + item.attributes['identifierref'].value")

    for resource in doc.getElementsByTagName('resource'):
        if resource.attributes['identifier'].value == node_resource:
            for file in resource.childNodes:
                if file.nodeName == 'file':
                    filename = file.attributes['href'].value

                    is_exe_node_html = False
                    if filename.endswith('.html') ¥
                    and filename != "fdl.html" ¥
                    and not filename.startswith("galleryPopup"):
                        # fdl.html is the wikipedia license, ignore it
                        # as well as any galleryPopups:
                        is_exe_node_html = ¥
                        test_for_node(zippedFile.read(filename))

                    if is_exe_node_html:
                        if pass_num == 1:
                            # 2nd pass call to actually load the nodes:
                            log.debug("loading idevices from node: ' + filename)
                            loadNodesIdevices(node, zippedFile.read(filename))
                        elif filename == "fdl.html" or ¥
                        filename.startswith("galleryPopup."):
                            # let these be re-created upon bursting.
                            if pass_num == 0:
                                # 1st pass call to unzip the resources:
                                log.debug("ignoring resource file: ' + filename)
                        else:
                            if pass_num == 0:
                                # 1st pass call to unzip the resources:
                                try:
                                    zipinfo = zippedFile.getinfo(filename)
                                    log.debug("unzipping resource file: '
                                        + resourceDir/filename )
                                    outFile = open(resourceDir/filename, "wb")
                                    outFile.write(zippedFile.read(filename))

```

```

        outFile.flush()
        outFile.close()
    except:
        log.warn('error unzipping resource file: '
                + resourceDir/filename )

        #####
        # WARNING: the resource is now in the resourceDir,
        # BUT it is NOT YET added into any of the project,
        # much less to the specific idevices or fields!
        # Although they WILL be saved out with the project
        # upon the next Save.
        #####

    break

# process this node's children
for subitem in item.childNodes:
    if subitem.nodeName == 'item':
        # for the first pass, of unzipping only, do not
        # create any child nodes, just cruise on with this one:
        next_node = node
        if pass_num == 1:
            # if this is actually loading the nodes:
            next_node = node.createChild()
        loadNode(pass_num, resourceDir, zippedFile, next_node,
                doc, subitem, level+1)

def loadCC(zippedFile, filename):
    """
    Load an IMS Common Cartridge or Content Package from filename
    """
    package = Package(Path(filename).namebase)
    xmldoc = minidom.parseString( zippedFile.read('imsmanifest.xml'))

    organizations_list = xmldoc.getElementsByTagName('organizations')
    level = 0
    # now a two-pass system to first unzip all applicable resources:
    for pass_num in range(2):
        for organizations in organizations_list:
            organization_list = organizations.getElementsByTagName(
                'organization')
            for organization in organization_list:
                for item in organization.childNodes:
                    if item.nodeName == 'item':
                        loadNode(pass_num, package.resourceDir, zippedFile,
                                package.root, xmldoc, item, level)

    return package

# =====
class DublinCore(Jellyable, Unjellyable):
    """
    Holds dublin core info
    """

    def __init__(self):
        self.title = ""
        self.creator = ""
        self.subject = ""
        self.description = ""
        self.publisher = ""
        self.contributors = ""
        self.date = ""
        self.type = ""
        self.format = ""
        self.identifier = ""

```

```

        self.source = ""
        self.language = ""
        self.relation = ""
        self.coverage = ""
        self.rights = ""

    def __setattr__(self, name, value):
        self.__dict__[name] = toUnicode(value)

class Package(Persistable):
    """
    Package represents the collection of resources the user is editing
    i.e. the "package".
    """
    persistenceVersion = 9
    nonpersistent      = ['resourceDir', 'filename']
    # Name is used in filenames and urls (saving and navigating)
    _name              = ""
    tempFile           = False # This is set when the package is saved as a temp copy file
    # Title is rendered in exports
    _title             = ""
    _author            = ""
    _description       = ""
    _backgroundImg     = ""
    # This is like a constant
    defaultLevelNames = [x_(u"Topic"), x_(u"Section"), x_(u"Unit")]

    def __init__(self, name):
        """
        Initialize
        """
        log.debug(u"init " + repr(name))
        self._nextIdDeviceId = 0
        self._nextNodeId    = 0
        # For looking up nodes by ids
        self._nodeIdDict    = {}

        self._levelNames    = self.defaultLevelNames[:]
        self.name            = name
        self._title         = u""
        self._backgroundImg = u""
        self.backgroundImgTile = False

        # Empty if never saved/loaded
        self.filename       = u""

        self.root           = Node(self, None, _(u"Home"))
        self.currentNode    = self.root
        self.style          = u"default"
        self.isChanged      = False
        self.iddevices      = []
        self.dublinCore     = DublinCore()
        self.scolinks       = False
        self.license        = "None"
        self.footer         = ""

        # Temporary directory to hold resources in
        self.resourceDir = TempDirPath()
        self.resources = {} # Checksum-[Resource(),...]

# Property Handlers

```

```

def set_name(self, value):
    self._name = toUnicode(value)
def set_title(self, value):
    self._title = toUnicode(value)
def set_author(self, value):
    self._author = toUnicode(value)
def set_description(self, value):
    self._description = toUnicode(value)

def get_backgroundImg(self):
    """Get the background image for this package"""
    if self._backgroundImg:
        return "file://" + self._backgroundImg.path
    else:
        return ""

def set_backgroundImg(self, value):
    """Set the background image for this package"""
    if self._backgroundImg:
        self._backgroundImg.delete()

    if value:
        if value.startswith("file://"):
            value = value[7:]

        imgFile = Path(value)
        self._backgroundImg = Resource(self, Path(imgFile))
    else:
        self._backgroundImg = u"

def get_level1(self):
    return self.levelName(0)
def set_level1(self, value):
    if value != "":
        self._levelNames[0] = value
    else:
        self._levelNames[0] = self.defaultLevelNames[0]

def get_level2(self):
    return self.levelName(1)
def set_level2(self, value):
    if value != "":
        self._levelNames[1] = value
    else:
        self._levelNames[1] = self.defaultLevelNames[1]

def get_level3(self):
    return self.levelName(2)
def set_level3(self, value):
    if value != "":
        self._levelNames[2] = value
    else:
        self._levelNames[2] = self.defaultLevelNames[2]

# Properties

name = property(lambda self:self._name, set_name)
title = property(lambda self:self._title, set_title)
author = property(lambda self:self._author, set_author)
description = property(lambda self:self._description, set_description)

backgroundImg = property(get_backgroundImg, set_backgroundImg)

```

```

level1 = property(get_level1, set_level1)
level2 = property(get_level2, set_level2)
level3 = property(get_level3, set_level3)

def findNode(self, nodeId):
    """
    Finds a node from its nodeId
    (nodeId can be a string or a list/tuple)
    """
    log.debug(u"findNode" + repr(nodeId))
    node = self._nodeIdDict.get(nodeId)
    if node and node.package is self:
        return node
    else:
        return None

def levelName(self, level):
    """
    Return the level name
    """
    if level < len(self._levelNames):
        return _self._levelNames[level]
    else:
        return _(u"?????")

def save(self, filename=None, tempFile=False):
    """
    Save package to disk
    pass an optional filename
    """
    self.tempFile = tempFile
    # Get the filename
    if filename:
        filename = Path(filename)
        # If we are being given a new filename...
        # Change our name to match our new filename
        name = filename.splitpath()[1]
        if not tempFile:
            self.name = name.basename().splittest()[0]
    elif self.filename:
        # Otherwise use our last saved/loaded from filename
        filename = Path(self.filename)
    else:
        # If we don't have a last saved/loaded from filename,
        # raise an exception because, we need to have a new
        # file passed when a brand new package is saved
        raise AssertionError(u'No name passed when saving a new package')
    # Store our new filename for next file|save, and save the package
    log.debug(u"Will save %s to: %s" % (self.name, filename))
    if tempFile:
        self.nonpersistant.remove('filename')
        oldFilename, self.filename = self.filename, unicode(self.filename)
        try:
            filename.safeSave(self.doSave, _('SAVE FAILED!¥nLast succesful save is %s.'))
        finally:
            self.nonpersistant.append('filename')
            self.filename = oldFilename
    else:
        # Update our new filename for future saves
        self.filename = filename
        filename.safeSave(self.doSave, _('SAVE FAILED!¥nLast succesful save is %s.'))
        self.isChanged = False

```

```

        self.updateRecentDocuments(filename)

def updateRecentDocuments(self, filename):
    """
    Updates the list of recent documents
    """
    # Don't update the list for the generic.data "package"
    genericData = G.application.config.configDir/'idevices'/'generic.data'
    if genericData.isfile() or genericData.islink():
        if Path(filename).samefile(genericData):
            return
    # Save in recentDocuments list
    recentProjects = G.application.config.recentProjects
    if filename in recentProjects:
        # If we're already number one, carry on
        if recentProjects[0] == filename:
            return
        recentProjects.remove(filename)
    recentProjects.insert(0, filename)
    del recentProjects[5:] # Delete any older names from the list
    G.application.config.configParser.write() # Save the settings

def doSave(self, fileObj):
    """
    Actually performs the save to 'fileObj'.
    """
    zippedFile = zipfile.ZipFile(fileObj, "w", zipfile.ZIP_DEFLATED)
    try:
        for resourceFile in self.resourceDir.files():
            zippedFile.write(unicode(resourceFile.normpath()),
                             resourceFile.name.encode('utf8'), zipfile.ZIP_DEFLATED)

            zinfo = zipfile.ZipInfo(filename='content.data',
                                     date_time=time.localtime()[0:6])
            zinfo.external_attr = 0100644 << 16L
            zippedFile.writestr(zinfo, encodeObject(self))
    finally:
        zippedFile.close()

def extractNode(self):
    """
    Clones and extracts the currently selected node into a new package.
    """
    newPackage = Package('NoName') # Name will be set once it is saved..
    newPackage.title = self.currentNode.title
    newPackage.style = self.style
    newPackage.author = self.author
    newPackage._nextNodeId = self._nextNodeId
    # Copy the nodes from the original package
    # and merge into the root of the new package
    self.currentNode.copyToPackage(newPackage)
    return newPackage

@staticmethod
def load(filename, newLoad=True, destinationPackage=None):
    """
    Load package from disk, returns a package.
    """
    if not zipfile.is_zipfile(filename):
        return None

    zippedFile = zipfile.ZipFile(filename, "r")

    try:

```

```

        # Get the jellied package data
        toDecode = zippedFile.read(u"content.data")
    except KeyError:
        log.info("no content.data, trying Common Cartridge/Content Package")
        newPackage = loadCC(zippedFile, filename)
        newPackage.tempFile = False
        newPackage.isChanged = False
        newPackage.filename = Path(filename)

    return newPackage

# Need to add a TempDirPath because it is a nonpersistant member
resourceDir = TempDirPath()

# Extract resource files from package to temporary directory
for fn in zippedFile.namelist():
    if unicode(fn, 'utf8') != u"content.data":
        outFile = open(resourceDir/fn, "wb")
        outFile.write(zippedFile.read(fn))
        outFile.flush()
        outFile.close()

try:
    newPackage = decodeObjectRaw(toDecode)
    G.application.afterUpgradeHandlers = []
    newPackage.resourceDir = resourceDir
    G.application.afterUpgradeZombies2Delete = []

    if newLoad:
        # provide newPackage to doUpgrade's versionUpgrade() to
        # correct old corrupt extracted packages by setting the
        # any corrupt package references to the new package:

        log.debug("load() about to doUpgrade newPackage %s"
                  + newPackage._name + "%s" % repr(newPackage) )
        if hasattr(newPackage, 'resourceDir'):
            log.debug("newPackage resourceDir = "
                      + newPackage.resourceDir)
        else:
            # even though it was just set above? should not get here:
            log.error("newPackage resourceDir has NO resourceDir!")

        doUpgrade(newPackage)

        # after doUpgrade, compare the largest found field ID:
        if G.application.maxFieldId >= Field.nextId:
            Field.nextId = G.application.maxFieldId + 1

    else:
        # and when merging, automatically set package references to
        # the destinationPackage, into which this is being merged:

        log.debug("load() about to merge doUpgrade newPackage %s"
                  + newPackage._name + "%s" % repr(newPackage)
                  + " INTO destinationPackage %s"
                  + destinationPackage._name + "%s" % repr(destinationPackage))

        log.debug("using their resourceDirs:")
        if hasattr(newPackage, 'resourceDir'):
            log.debug("    newPackage resourceDir = "
                      + newPackage.resourceDir)
        else:
            log.error("newPackage has NO resourceDir!")

```

```

        if hasattr(destinationPackage, 'resourceDir'):
            log.debug("    destinationPackage resourceDir = "
                    + destinationPackage.resourceDir)
        else:
            log.error("destinationPackage has NO resourceDir!")

        doUpgrade(destinationPackage,
                  isMerge=True, preMergePackage=newPackage)

        # after doUpgrade, compare the largest found field ID:
        if G.application.maxFieldId >= Field.nextId:
            Field.nextId = G.application.maxFieldId + 1

    except:
        import traceback
        traceback.print_exc()
        raise

    if newPackage.tempFile:
        # newPackage.filename was stored as it's original filename
        newPackage.tempFile = False
    else:
        # newPackage.filename is the name that the package was last loaded from
        # or saved to
        newPackage.filename = Path(filename)

    # Let idevices and nodes handle any resource upgrading they may need to
    # Note: Package afterUpgradeHandlers *must* be done after Resources'
    # and the package should be updated before everything else,
    # so, prioritize with a 3-pass, 3-level calling setup
    # in order of: 1) resources, 2) package, 3) anything other objects
    for handler_priority in range(3):
        for handler in G.application.afterUpgradeHandlers:

            if handler_priority == 0 and ¥
            repr(handler.im_class)=="<class 'exe.engine.resource.Resource'>":
                # level-0 handlers: Resource
                handler()

            elif handler_priority == 1 and ¥
            repr(handler.im_class)=="<class 'exe.engine.package.Package'>":
                # level-1 handlers: Package (requires resources first)
                if handler.im_self == newPackage:
                    handler()
                else:
                    log.warn("Extra package object found, " ¥
                        + "ignoring its afterUpgradeHandler: " ¥
                        + repr(handler))

            elif handler_priority == 2 and ¥
            repr(handler.im_class)!="<class 'exe.engine.resource.Resource'>" ¥
            and ¥
            repr(handler.im_class)!="<class 'exe.engine.package.Package'>":
                # level-2 handlers: all others
                handler()

    G.application.afterUpgradeHandlers = []

    num_zombies = len(G.application.afterUpgradeZombies2Delete)
    for i in range(num_zombies-1, -1, -1):
        zombie = G.application.afterUpgradeZombies2Delete[i]
        # now, the zombie list can contain nodes OR resources to delete.
        # if zombie is a node, then also pass in a pruning parameter..
        zombie_is_node = False

```



```

        if isinstance(zombie, Node):
            zombie_is_node = True

        if zombie_is_node:
            zombie.delete(pruningZombies=True)
        else:
            zombie.delete()
        del zombie
    G.application.afterUpgradeZombies2Delete = []

    newPackage.updateRecentDocuments(newPackage.filename)
    newPackage.isChanged = False
    return newPackage

def cleanUpResources(self):
    """
    Removes duplicate resource files
    """
    # Delete unused resources.
    # Only really needed for upgrading to version 0.20,
    # but upgrading of resources and package happens in no particular order
    # and must be done after all resources have been upgraded

    # some earlier .elp files appear to have been corrupted with
    # two packages loaded, *possibly* from some strange extract/merge
    # functionality in earlier eXe versions?
    # Regardless, only the real package will have a resourceDir,
    # and the other will fail.
    # For now, then, put in this quick and easy safety check:
    if not hasattr(self, 'resourceDir'):
        log.warn("cleanUpResources called on a redundant package")
        return

    existingFiles = set([fn.basename() for fn in self.resourceDir.files()])
    usedFiles = set([reses[0].storageName for reses in self.resources.values()])
    for fn in existingFiles - usedFiles:
        (self.resourceDir / fn).remove()

def findResourceByName(self, queryName):
    """
    Support for merging, and anywhere else that unique names might be
    checked before actually comparing against the files (as will be
    done by the resource class itself in its _addOurselvesToPackage() )
    """
    foundResource = None
    queryResources = self.resources
    for this_checksum in queryResources:
        for this_resource in queryResources[this_checksum]:
            if queryName == this_resource.storageName:
                foundResource = this_resource
                return foundResource
    return foundResource

def upgradeToVersion1(self):
    """
    Called to upgrade from 0.3 release
    """
    self._nextNodeId = 0
    self._nodeIdDict = {}

    # Also upgrade all the nodes.
    # This needs to be done here so that draft gets id 0
    # If it's done in the nodes, the ids are assigned in reverse order

```

```

draft = getattr(self, 'draft')
draft._id = self._regNewNode(draft)
draft._package = self
setattr(self, 'editor', Node(self, None, _(u"iDevice Editor")))

# Add a default idevice to the editor
idevice = GenericIdevice("", "", "", "", "")
editor = getattr(self, 'editor')
idevice.parentNode = editor
editor.addIdevice(idevice)
def superReg(node):
    """Registers all our nodes
    because in v0 they were not registered
    in this way"""
    node._id = self._regNewNode(node)
    node._package = self
    for child in node.children:
        superReg(child)
superReg(self.root)

def _regNewNode(self, node):
    """
    Called only by nodes,
    stores the node in our id lookup dict
    returns a new unique id
    """
    id_ = unicode(self._nextNodeId)
    self._nextNodeId += 1
    self._nodeIdDict[id_] = node
    return id_

def getNewIdeviceId(self):
    """
    Returns an iDevice Id which is unique for this package.
    """
    id_ = unicode(self._nextIdeviceId)
    self._nextIdeviceId += 1
    return id_

def upgradeToVersion2(self):
    """
    Called to upgrade from 0.4 release
    """
    getattr(self, 'draft').delete()
    getattr(self, 'editor').delete()
    delattr(self, 'draft')
    delattr(self, 'editor')
    # Need to renumber nodes because idevice node and draft nodes are gone
    self._nextNodeId = 0
    def renumberNode(node):
        """
        Gives the old node a number
        """
        node._id = self._regNewNode(node)
        for child in node.children:
            renumberNode(child)
    renumberNode(self.root)

def upgradeToVersion3(self):
    """

```

```

Also called to upgrade from 0.4 release
"""
self._nextIddeviceId = 0

def upgradeToVersion4(self):
    """
    Puts properties in their place
    Also called to upgrade from 0.8 release
    """
    self._name = toUnicode(self.__dict__['name'])
    self._author = toUnicode(self.__dict__['author'])
    self._description = toUnicode(self.__dict__['description'])

def upgradeToVersion5(self):
    """
    For version 0.11
    """
    self._levelNames = self.levelNames
    del self.levelNames

def upgradeToVersion6(self):
    """
    For version 0.14
    """
    self.dublinCore = DublinCore()
    # Copy some of the package properties to dublin core
    self.title = self.root.title
    self.dublinCore.title = self.root.title
    self.dublinCore.creator = self._author
    self.dublinCore.description = self._description
    self.scolinks = False

def upgradeToVersion7(self):
    """
    For version 0.15
    """
    self._backgroundImg = ""
    self.backgroundImgTile = False

def upgradeToVersion8(self):
    """
    For version 0.20, alpha, for nightlies r2469
    """
    self.license = 'None'
    self.footer = ""
    self.iddevices = []

def upgradeToVersion9(self):
    """
    For version >= 0.20.4
    """
    if not hasattr(self, 'resources'):
        # The hasattr is needed, because sometimes, Resource instances are upgraded
        # first and they also set this attribute on the package
        self.resources = {}
    G.application.afterUpgradeHandlers.append(self.cleanUpResources)

# =====

```

ファイル 3 : %(install dir)/exe/webui/builtinblocks.py

```

# =====
# eXe
# Copyright 2004–2006, University of Auckland
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
# =====
"""
BuiltInBlocks imports the iDevice blocks which are built-in (i.e. not
plugins) to eXe
"""

from exe.webui.freetextblock import FreeTextBlock
from exe.webui.genericblock import GenericBlock
from exe.webui.multichoiceblock import MultichoiceBlock
from exe.webui.reflectionblock import ReflectionBlock
from exe.webui.casestudyblock import CasestudyBlock
from exe.webui.truefalseblock import TrueFalseBlock
from exe.webui.imagewithtextblock import ImageWithTextBlock
from exe.webui.wikipediablock import WikipediaBlock
from exe.webui.attachmentblock import AttachmentBlock
from exe.webui.galleryblock import GalleryBlock
from exe.webui.clozeblock import ClozeBlock
from exe.webui.flashwithtextblock import FlashWithTextBlock
from exe.webui.externalurlblock import ExternalUrlBlock
from exe.webui.imagemagnifierblock import ImageMagnifierBlock
from exe.webui.mathblock import MathBlock
from exe.webui.multimedialblock import MultimediaBlock
from exe.webui.rssblock import RssBlock
from exe.webui.multiselectblock import MultiSelectBlock
from exe.webui.appletblock import AppletBlock
from exe.webui.flashmovieblock import FlashMovieBlock
from exe.webui.quiztestblock import QuizTestBlock
from exe.webui.scormmnblock import ScormMnBlock
# =====

```

追記

付録3 新規作成モジュール群【ファイル1～ファイル4までの4ファイル】

ファイル1: %(install dir)/exe/engine/scormmnidevice.py

```

# =====
# eXe
# Copyright 2004–2006, University of Auckland
# Copyright 2004–2008 eXe Project, http://eXeLearning.org/
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or

```

```

# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
# =====
"""
A ScormMn Idevice is one built up from ....
"""

import logging
from exe.engine.persist import Persistable
from exe.engine.idevice import Idevice
from exe.engine.translate import lateTranslate
from exe.engine.field import TextAreaField
import re

log = logging.getLogger(__name__)

# =====
class AnswerOption(Persistable):
    """
    A ScormMnQuestion is built up of question and AnswerOptions. Each
    answerOption can be rendered as an XHTML element
    """

    def __init__(self, question, idevice, answer=""):
        """
        Initialize
        """
        self.question = question
        self.idevice = idevice

        self.answerTextArea = TextAreaField(x_(u'Instructions For Answer( and Correct Answer)'),
                                             self.question._optionInstruc,
                                             answer)
        self.answerTextArea.idevice = idevice

        self.correctmain = ""
        self.correcttote = ""
        self.ans = ""

    def getResourcesField(self, this_resource):
        """
        implement the specific resource finding mechanism for this iDevice:
        """
        # be warned that before upgrading, this iDevice field could not exist:
        if hasattr(self, 'answerTextArea') and hasattr(self.answerTextArea, 'images'):
            for this_image in self.answerTextArea.images:
                if hasattr(this_image, '_imageResource') and this_resource == this_image._imageResource:
                    return self.answerTextArea

        return None

    def getRichTextFields(self):
        """

```

AnswerOption(複数解答欄の class)を初期化する際に、正答と許容範囲、学習者解答を格納する変数を作成する

```

Like getResourcesField(), a general helper to allow nodes to search
through all of their fields without having to know the specifics of each
iDevice type.
"""
fields_list = []
if hasattr(self, 'answerTextArea'):
    fields_list.append(self.answerTextArea)

return fields_list

def upgrade_setIddevice(self, iddevice, question):
    """
    While some of this might typically be done in an automatic upgrade
    method called from in increased persistence version, the problem
    with that approach is that the iddevice was not previously stored,
    and cannot easily be gotten at that stage of operation.

    Rather than making such an upgrade method more messy than necessary,
    this method allows the parent ScormMnQuestion to merely set
    itself on each of its AnswerOptions during its own upgrade.

    Helps upgrade to somewhere before version 0.25 (post-v0.24),
    taking the old unicode string fields,
    and converting them into a image-enabled TextAreaFields:
    """
    self.iddevice = iddevice
    self.question = question
    self.answerTextArea = TextAreaField(x_(u'Instructions For Answer( and Correct Answer)'),
                                        self.question._optionInstruc,
                                        self.answer)
    self.answerTextArea.iddevice = self.iddevice

# =====
class ScormMnQuestion(Persistable):
    """
    A ScormMnQuestion is built up of question and AnswerOptions.
    """

    persistenceVersion = 3

    def __init__(self, iddevice, question=""):
        """
        Initialize
        """
        self.iddevice = iddevice

        self.options = []
        self._questionInstruc = x_(u""Enter the question stem.
The quest should be clear and unambiguous. Avoid negative premises
as these can tend to be ambiguous.""")
        self._optionInstruc = x_(u""For each answer column of this
question, please mention contents and the explanatory note which you
should answer. In two blanks below , Enter a right answer (numerical value)
and a tolerance (numerical value).""")

        self.questionTextArea = TextAreaField(x_(u'Question:'),
                                              self._questionInstruc, u'')
        self.questionTextArea.iddevice = self.iddevice

        self.addOption()

    # Properties
    questionInstruc = lateTranslate('questionInstruc')
    optionInstruc = lateTranslate('optionInstruc')

```

```

def addOption(self):
    """
    Add a new option to this question.
    """
    self.options.append(AnswerOption(self, self.iddevice))

def getResourcesField(self, this_resource):
    """
    implement the specific resource finding mechanism for this iDevice:
    """
    # be warned that before upgrading, this iDevice field could not exist:
    if hasattr(self, 'questionTextArea') and hasattr(self.questionTextArea, 'images'):
        for this_image in self.questionTextArea.images:
            if hasattr(this_image, '_imageResource') and this_resource == this_image._imageResource:
                return self.questionTextArea

    for this_option in self.options:
        this_field = this_option.getResourcesField(this_resource)
        if this_field is not None:
            return this_field

    return None

def getRichTextFields(self):
    """
    Like getResourcesField(), a general helper to allow nodes to search
    through all of their fields without having to know the specifics of each
    iDevice type.
    """
    fields_list = []
    if hasattr(self, 'questionTextArea'):
        fields_list.append(self.questionTextArea)

    for this_option in self.options:
        fields_list.extend(this_option.getRichTextFields())

    return fields_list

def upgradeToVersion1(self):
    """
    Upgrades to v 0.13
    """
    self._optionInstruc = x_(u""For each answer column of this
    question, please mention contents and the explanatory note which you
    should answer. In two blanks below , Enter a right answer (numerical value)
    and a tolerance (numerical value).""")

    def upgradeToVersion2(self):
        """
        Upgrades to v 0.13
        """
        self._questionInstruc= x_(u""Enter the question stem.
        The quest should be clear and unambiguous. Avoid negative premises
        as these can tend to be ambiguous.""")

    def upgrade_setIddevice(self, iddevice):
        """
        While some of this might typically be done in an automatic upgrade
        method called from in increased persistence version, the problem

```

with that approach is that the idevice was not previously stored, and cannot easily be gotten at that stage of operation.

Rather than making such an upgrade method more messy than necessary, this method allows the parent ScormMnQuestionIdevice to merely set itself on each of its ScormMnQuestions during its own upgrade.

Helps upgrade to somewhere before version 0.25 (post-v0.24), taking the old unicode string fields, and converting them into a image-enabled TextAreaFields:

```
self.idevice = idevice
self.questionTextArea = TextAreaField(x_(u'Question:'),
                                      self._questionInstruc,
                                      self.question)
self.questionTextArea.idevice = self.idevice
```

```
# and then, need to propagate the same upgrades
# down through each of the options:
for option in self.options:
    option.upgrade_setIdevice(self.idevice, self)
```

```
# =====
class ScormMnIdevice(Idevice):
    """
    A ScormMnIdevice Idevice is one built up from question and <<Answer Box>>
    """
    persistenceVersion = 8

    def __init__(self):
        """
        Initialize
        """
        Idevice.__init__(self,
                        x_(u"SCORM MultiNumerical Question"),
                        x_(u"Seiya Fukushima"),
                        x_(u""Unlike the MCQ the SCORM quiz is used to test
the learners knowledge on a topic without providing the learner with feedback
to the correct answer. The quiz will often be given once the learner has had
time to learn and practice using the information or skill.
"""), u"", "question")
        self.isQuiz = True
        self.emphasis = Idevice.SomeEmphasis
        self.score = -1
        self.isAnswered = True
        self.passRate = "50"
        self.questions = []
        self.addQuestion()
        self.systemResources += ["common.js", "libot_drag.js"]

    def addQuestion(self):
        """
        Add a new question to this iDevice.
        """
        self.questions.append(ScormMnQuestion(self))

    def getResourcesField(self, this_resource):
        """
        implement the specific resource finding mechanism for this iDevice:
        """
        for this_question in self.questions:
            this_field = this_question.getResourcesField(this_resource)
```

ScormMnIdevice class の初期化部分において、iDevice(eXe におけるコンテンツ作成機能をこう呼ぶ)の名前や作成者情報を格納する


```

        if this_field is not None:
            return this_field

        return None

    def getRichTextFields(self):
        """
        Like getResourcesField(), a general helper to allow nodes to search
        through all of their fields without having to know the specifics of each
        iDevice type.
        """
        fields_list = []

        for this_question in self.questions:
            fields_list.extend(this_question.getRichTextFields())

        return fields_list

    def burstHTML(self, i):
        """
        takes a BeautifulSoup fragment (i) and bursts its contents to
        import this idevice from a CommonCartridge export
        """
        # SCORM QuizTest Idevice:
        title = i.find(name='span', attrs={'class' : 'iDeviceTitle' })
        self.title = title.renderContents().decode('utf-8')

        inner = i.find(name='div', attrs={'class' : 'iDevice_inner' })

        passrate = inner.find(name='div', attrs={'class' : 'passrate' })
        self.passRate = passrate.attrMap['value'].decode('utf-8')

        # copied and modified from Multi-Select:

        sc_questions = inner.findAll(name='div', attrs={'class' : 'question'})
        if len(sc_questions) < 1:
            # need to remove the default 1st question
            del self.questions[0]

        for question_num in range(len(sc_questions)):
            if question_num > 0:
                # only created with the first question, add others:
                self.addQuestion()

            question = sc_questions[question_num]

            questions = question.findAll(name='div', attrs={'class' : 'block',
                'id' : re.compile('^taquestion') })
            if len(questions) == 1:
                # ELSE: should warn of unexpected result!
                inner_question = questions[0]
                self.questions[question_num].questionTextArea.content_wo_resourcePaths ¥
                    = inner_question.renderContents().decode('utf-8')
                # and add the LOCAL resource paths back in:
                self.questions[question_num].questionTextArea.content_w_resourcePaths ¥
                    =
            self.questions[question_num].questionTextArea.MessageResourceDirsIntoContent( ¥

        self.questions[question_num].questionTextArea.content_wo_resourcePaths
        self.questions[question_num].questionTextArea.content ¥
            = self.questions[question_num].questionTextArea.content_w_resourcePaths

        options = question.findAll(name='div', attrs={'class' : 'block',
            'id' : re.compile('^taoptionAnswer') })

```

```

        answers = question.findAll(name='input', attrs={'type' : 'radio'})
        if len(options) < 1:
            # need to remove the default 1st option
            del self.questions[question_num].options[0]

        for option_loop in range(0, len(options)):
            if option_loop >= 1:
                # more options than created by default:
                self.questions[question_num].addOption()

self.questions[question_num].options[option_loop].answerTextArea.content_wo_resourcePaths ¥
    = options[option_loop].renderContents().decode('utf-8')
    # and add the LOCAL resource paths back in:

self.questions[question_num].options[option_loop].answerTextArea.content_w_resourcePaths ¥
    =
self.questions[question_num].options[option_loop].answerTextArea.MessageResourceDirsIntoContent( ¥

self.questions[question_num].options[option_loop].answerTextArea.content_wo_resourcePaths)
    self.questions[question_num].options[option_loop].answerTextArea.content ¥
    =
self.questions[question_num].options[option_loop].answerTextArea.content_w_resourcePaths
    # and finally, see if this is a correct answer:
    this_answer = answers[option_loop].attrMap['value']
    if this_answer == "0":
        # then this option is correct:
        self.questions[question_num].options[option_loop].isCorrect ¥
            = True
        # and SCORM quiz also has an overall correctAnswer;
        # since it only allows one answer, this must be it:
        self.questions[question_num].correctAns = option_loop

def upgradeToVersion2(self):
    """
    Upgrades the node from 1 (v0.5) to 2 (v0.6).
    Old packages will loose their icons, but they will load.
    """
    log.debug(u"Upgrading iDevice")
    self.emphasis = Idevice.SomeEmphasis

def upgradeToVersion3(self):
    """
    Upgrades the node from 1 (v0.6) to 2 (v0.7).
    Change icon from 'multichoice' to 'question'
    """
    log.debug(u"Upgrading iDevice icon")
    self.icon = "question"

def upgradeToVersion4(self):
    """
    Upgrades v0.6 to v0.7.
    """
    self.lastIdevice = False

def upgradeToVersion5(self):
    """
    Upgrades to exe v0.10
    """

```

```

self._upgradeldeviceToVersion1()

def upgradeToVersion6(self):
    """
    Upgrades to v0.12
    """
    self._upgradeldeviceToVersion2()
    self.systemResources += ["common.js", "libot_drag.js"]

def upgradeToVersion7(self):
    """
    Upgrades to v0.14
    """
    # Note: the following routine doesn't appear to exist anymore,
    # so now that the persistence version is finally upgrading to 7,
    # (and then, actually on to 8) this is no longer works, go figure!
    #####
    #self._upgradeldeviceToVersion3()
    #####

    self.isQuiz = True

def upgradeToVersion8(self):
    """
    Upgrades to somewhere before version 0.25 (post-v0.24)
    Taking the ScormMnQuestions' old unicode string fields,
    and converting them into a image-enabled TextAreaFields:
    """
    for question in self.questions:
        question.upgrade_setldevice(self)

## =====
def register(iddeviceStore):
    #""Register with the iddeviceStore""
    #iddeviceStore.extended.append(ScormMnldevice())

```

ファイル 2: %(install dir)%/exe/engine/scormmnblock.py

```

# =====
# eXe
# Copyright 2004-2006, University of Auckland
# Copyright 2004-2008 eXe Project, http://eXeLearning.org
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
# =====
"""
ScormMnBlock can render and process ScormMnldevices as XHTML
"""

```

```

import logging
from exe.webui.block import Block
from exe.webui.scormmnelement import ScormMnElement
from exe.webui import common
log = logging.getLogger(__name__)

# =====
class ScormMnBlock(Block):
    """
    ScormMnBlock can render and process ScormMnIdevices as XHTML
    """
    def __init__(self, parent, iddevice):
        """
        Initialize a new Block object
        """
        Block.__init__(self, parent, iddevice)
        self.iddevice = iddevice
        self.questionElements = []
        self.message = False
        if not hasattr(self.iddevice, 'undo'):
            self.iddevice.undo = True

        i = 0
        for question in iddevice.questions:
            self.questionElements.append(ScormMnElement(i, iddevice,
                                                         question))
            i += 1

    def process(self, request):
        """
        Process the request arguments from the web server
        """
        Block.process(self, request)

        is_cancel = common.requestHasCancel(request)

        if ("addQuestion"+unicode(self.id)) in request.args:
            self.iddevice.addQuestion()
            self.iddevice.edit = True
            # disable Undo once a question has been added:
            self.iddevice.undo = False

        if "passrate" in request.args and not is_cancel:
            self.iddevice.passRate = request.args["passrate"][0]

        for element in self.questionElements:
            element.process(request)

        if ("action" in request.args and request.args["action"][0] == "done"
            or not self.iddevice.edit):
            self.iddevice.isAnswered = True
            # remove the undo flag in order to reenale it next time:
            if hasattr(self.iddevice, 'undo'):
                del self.iddevice.undo

        ### fuku input correctmain(correct main answer check)
        for question in self.iddevice.questions:
            for option in question.options:
                if option.correctmain == "":

```

正答欄の正解未記入や数値以外の
入力に対して制御をおこなう部分

```

        self.iddevice.edit = True
    else:
        try:
            float(option.correctmain)
        except ValueError:
            self.iddevice.edit = True
        except TypeError:
            self.iddevice.edit = True

    ### fuku input correcttole(correct answer tolerance check)
    for question in self.iddevice.questions:
        for option in question.options:
            if not option.correcttole == "":
                try:
                    float(option.correcttole)
                except ValueError:
                    self.iddevice.edit = True
                except TypeError:
                    self.iddevice.edit = True

    if "submitScore" in request.args ¥
    and not is_cancel:
        self.iddevice.score = self.__calcScore()

    if "title"+self.id in request.args ¥
    and not is_cancel:
        self.iddevice.title = request.args["title"+self.id][0]

def renderEdit(self, style):
    """
    Returns an XHTML string with the form element for editing this block
    """
    html = "<div class=¥"iDevice¥"
    ### fuku input correctmain(correct main answer check)
    for question in self.iddevice.questions:
        for option in question.options:
            if option.correctmain == "":
                html += common.editModeHeading(
                    _("Please input correct answer(main) with numerical value.))
            else:
                try:
                    float(option.correctmain)
                except ValueError:
                    html += common.editModeHeading(
                        _("Please input correct answer(main) with numerical value.))
                except TypeError:
                    html += common.editModeHeading(
                        _("Please input correct answer(main) with numerical value.))

    ### fuku input correcttole(correct answer tolerance check)
    for question in self.iddevice.questions:
        for option in question.options:
            if not option.correcttole == "":
                try:
                    float(option.correcttole)
                except ValueError:
                    html += common.editModeHeading(
                        _("Please input correct answer(tolerance) with numerical value.))
                except TypeError:
                    html += common.editModeHeading(
                        _("Please input correct answer(tolerance) with numerical value.))

```

正答に対する許容範囲
欄に対し、数値以外の
入力が行われていない
か判定、制御をおこな
う部分

正答欄の正解未記入や数値以外の入力に対して、そ
の旨の警告表示をおこなう部分

正答に対する許容範囲
欄に対し、数値以外の
入力が行われている場
合、その旨の警告表示
をおこなう部分

```

html += common.textInput("title"+self.id, self.iddevice.title)
html += u"<br/><br/>¥n"

for element in self.questionElements:
    html += element.renderEdit()

value = _("Add another Question")
html += "<br/>"
html += common.submitButton("addQuestion"+unicode(self.id), value)
html += "<br/><br/>" + _("Select pass rate: ")
html += "<select name=¥\"passrate¥\">¥n"
template = ' <option value=¥s0¥s>¥s0¥%</option>¥n'
for i in range(1, 11):
    if str(i)+ "0" == self.iddevice.passRate:
        html += template % (str(i), ' selected="selected"', str(i))
    else:
        html += template % (str(i), "", str(i))
html += "</select>¥n"
html += "<br /><br />" + self.renderEditButtons(undo=self.iddevice.undo)
html += "</div>¥n"
self.iddevice.isAnswered = True

return html

def renderView(self, style, preview=False):
    """
    Returns an XHTML string for viewing this block
    """
    html = u'<form name="quizForm¥s" id="quizForm¥s" ' % (
        self.iddevice.id, self.iddevice.id)
    html += u'action="javascript:calcScore2();">¥n'
    html += u'<div class="iDevice '
    html += u'emphasis'+unicode(self.iddevice.emphasis)+">¥n'
    html += u'¥n'
    html += u'<span class="iDeviceTitle">'
    html += self.iddevice.title+'</span>¥n'
    html += u'<div class="iDevice_inner">¥n'
    html += u'<div class="passrate" value="¥s"></div>¥n' % self.iddevice.passRate
    for element in self.questionElements:
        if preview:
            html += element.renderPreview() + "<br/>"
        else:
            html += element.renderView() + "<br/>"

    html += '<input type="submit" name="submitB" '
    html += 'value="¥s"/>¥n' % _(u"SUBMIT ANSWERS")
    html += "</div></div>¥n"
    html += '</form>¥n'

    return html

def renderJavascriptForWeb(self):
    """
    Return an XHTML string for generating the javascript for web export
    """
    scriptStr = '<script type="text/javascript">¥n'
    scriptStr += '<!-- //<![CDATA[¥n'
    scriptStr += "var numQuestions = "
    scriptStr += str(len(self.questionElements))+";¥n"
    scriptStr += "var rawScore = 0;¥n"
    scriptStr += "var actualScore = 0;¥n"

```

renderJavascriptForWeb クラスでは、web 形式での export を行った際に出力される Javascript 組み込み済みの HTML 出力をおこなう

文字列格納用の変数に、Javascript を自動生成していく部分

```

answerStr = ""function getAnswer()
{
varStrs = ""
keyStrs = ""
rawScoreStr = ""}
function calcRawScore(){¥n""

for element in self.questionElements:
    i = element.index
    varStr = "question" + str(i)
    keyStr = "key" + str(i)
    quesId = "key" + str(element.index) + "b" + self.id
    numOption = element.getNumOption()
    getEle = 'document.getElementById("quizForm%s")' % ¥
        self.iddevice.id
    chk = '%s.%s[i].checked%' (getEle, quesId)
    value = '%s.%s[i].value' % (getEle, quesId)
    numAnswer = 0
    corAns = []
    tolAns = []
    for option in element.question.options:
        corAns.append(option.correctmain)
        if option.correcttole == "":
            tolAns.append(u"0")
            keyStrs += "var key" + str(i) + "Answer" + str(numAnswer) + "low = " +
str(option.correctmain) + ";\n"
            keyStrs += "var key" + str(i) + "Answer" + str(numAnswer) + "high = " +
str(option.correctmain) + ";\n"
        else:
            tolAns.append(option.correcttole)
            keyStrs += "var key" + str(i) + "Answer" + str(numAnswer) + "low = " +
str(float(option.correctmain) - float(option.correcttole)) + ";\n"
            keyStrs += "var key" + str(i) + "Answer" + str(numAnswer) + "high = " +
str(float(option.correctmain) + float(option.correcttole)) + ";\n"
            varStrs += "var " + varStr + "Answer" + str(numAnswer) + ";\n"
            numAnswer += 1
    for numtemp in range(numAnswer):
        answerStr += ""
        %sAnswer%s = %s;
        "" % (varStr, str(numtemp), getEle + "." + quesId + "Answer" + str(numtemp) + ".value")

    rawScoreStr += ""
    if ("")
    for numtemp in range(numAnswer):
        rawScoreStr += "" %s >= %s & %s <= %s &"" % (
            varStr + "Answer" + str(numtemp),
            "key" + str(i) + "Answer" + str(numtemp) + "low",
            varStr + "Answer" + str(numtemp),
            "key" + str(i) + "Answer" + str(numtemp) + "high")
    rawScoreStr = rawScoreStr[:-1] + ""
    {
        rawScore++;
    }""

scriptStr += varStrs
scriptStr += keyStrs

scriptStr += answerStr

scriptStr += rawScoreStr

scriptStr += ""

}

```

Javascript にて採点
(正誤判断)を行う部分
を自動生成している

```

function calcScore2()
{
    getAnswer();

    calcRawScore();
    actualScore = Math.round(rawScore / numQuestions * 100);
    document.getElementById("quizForm%s").submitB.disabled = true;
    alert("Your score is " + actualScore + "%")
}

//]]> -->
</script>¥n"" % self.iddevice.id

return scriptStr

def renderJavascriptForScorm(self):
    """
    Return an XHTML string for generating the javascript for scorm export
    """
    scriptStr = '<script type="text/javascript">¥n'
    scriptStr += '<!-- //¥n'
    scriptStr += "var numQuestions = "
    scriptStr += unicode(len(self.questionElements))+";¥n"
    scriptStr += "var rawScore = 0;¥n"
    scriptStr += "var actualScore = 0;¥n"
    answerStr = """function getAnswer()
{
    varStrs = ""
    keyStrs = ""
    rawScoreStr = ""
    function calcRawScore(){¥n"""

    for element in self.questionElements:
        i = element.index
        varStr = "question" + unicode(i)
        keyStr = "key" + unicode(i)
        quesId = "key" + unicode(element.index) + "b" + self.id
        numOption = element.getNumOption()
        getEle = 'document.getElementById("quizForm%s")' % ¥
            self.iddevice.id
        chk = '%s.%s[i].checked%' (getEle, quesId)
        value = '%s.%s[i].value' % (getEle, quesId)
        answerStr += """
        doLMSSetValue("cmi.interactions.%s.id","%s");
        doLMSSetValue("cmi.interactions.%s.type","performance");
        """ % (unicode(i), quesId, unicode(i))
        numAnswer = 0
        corAns = []
        tolAns = []
        lmssetstres = ""
        for option in element.question.options:
            corAns.append(option.correctmain)
            if option.correcttole == "":
                tolAns.append(u"0")
                keyStrs += "var key" + unicode(i) + "Answer" + unicode(numAnswer) + "low = " +
                    unicode(option.correctmain) + ";¥n"
                keyStrs += "var key" + unicode(i) + "Answer" + unicode(numAnswer) + "high = " +
                    + unicode(option.correctmain) + ";¥n"
            else:
                tolAns.append(option.correcttole)
                keyStrs += "var key" + unicode(i) + "Answer" + unicode(numAnswer) + "low = " +
                    unicode(float(option.correctmain) - float(option.correcttole)) + ";¥n"
</pre>
</div>
<div data-bbox="627 133 872 174" data-label="Text">
<p>Javascript の関数が正解数の割合にて点数を計算する部分を自動生成している</p>
</div>
<div data-bbox="577 267 874 333" data-label="Text">
<p>renderJavascriptForScorm クラスでは、renderJavascriptForWeb クラスでの動作に加え、LMS との各種通信(情報格納)部分を含む Javascript を生成する。</p>
</div>
<div data-bbox="507 905 534 921" data-label="Page-Footer">
<p>80</p>
</div>
```



```

        keyStrs += "var key" + unicode(i) + "Answer" + unicode(numAnswer) + "high = "
+ unicode(float(option.correctmain) + float(option.correcttole)) + ";\n"
        varStrs += "var " + varStr + "Answer" + unicode(numAnswer) + ";\n"
        answerStr += ""
        doLMSSetValue("cmi.interactions.%s.correct_responses.%s.pattern",
            "%s");
        "" % (unicode(i), unicode(numAnswer), corAns[numAnswer] + "plus_minus" +
tolAns[numAnswer])
        numAnswer += 1
        for numtemp in range(numAnswer):
            answerStr += ""
            %sAnswer%s = %s;
            "" % (varStr, unicode(numtemp), getEle + "." + quesId + "Answer" + unicode(numtemp) +
".value")

            lmssetstres += varStr + 'Answer' + unicode(numtemp) + '+' + '+'
            answerStr += ""
            doLMSSetValue("cmi.interactions.%s.student_response",%s);
            "" % (unicode(i), lmssetstres[:-5])

            rawScoreStr += ""
            if (""
            for numtemp in range(numAnswer):
                rawScoreStr += "" %s >= %s & %s <= %s & "" % (
                    varStr + "Answer" + unicode(numtemp),
                    "key" + unicode(i) + "Answer" + unicode(numtemp) + "low",
                    varStr + "Answer" + unicode(numtemp),
                    "key" + unicode(i) + "Answer" + unicode(numtemp) + "high")
                rawScoreStr = rawScoreStr[:-1] + ""
            {
                doLMSSetValue("cmi.interactions.%s.result","correct");
                rawScore++;
            }
            else
            {
                doLMSSetValue("cmi.interactions.%s.result","wrong");
            }
            "" % (unicode(i), unicode(i))

        scriptStr += varStrs
        scriptStr += keyStrs
        scriptStr += answerStr
        scriptStr += rawScoreStr
        scriptStr += ""
    }

    function calcScore2()
    {
        computeTime(); // the student has stopped here.
        ""
        scriptStr += ""
        document.getElementById("quizForm%s").submitB.disabled = true;
        "" % (self.idevice.id)
        scriptStr += ""
        getAnswer();

        calcRawScore();

        actualScore = Math.round(rawScore / numQuestions * 100);
        ""
        scriptStr += 'alert("Your score is " + actualScore + "%")'
        scriptStr += ""

        doLMSSetValue( "cmi.core.score.raw", actualScore+"" );

        var mode = doLMSGetValue( "cmi.core.lesson_mode" );

```

doLMSSetValue により、LMS へ格納すべき情報を受け渡す。ここでは正答と許容誤差を文字列格納している。

同様に doLMSSetValue により、受験者の出した解答を LMS へ格納するよう受け渡す。

```

        if ( mode != "review" && mode != "browse" ){
            if ( actualScore < %s )
            {
                doLMSSetValue( "cmi.core.lesson_status", "failed" );
            }
            else
            {
                doLMSSetValue( "cmi.core.lesson_status", "passed" );
            }

            doLMSSetValue( "cmi.core.exit", "" );
        }

        exitPageStatus = true;

        doLMSCommit();

        doLMSFinish();
    }
//]]> -->
</script>¥n"" % self.iddevice.passRate

    return scriptStr

def renderPreview(self, style):
    """
    Returns an XHTML string for previewing this block
    """
    html = u"<div class=¥"iDevice "
    html += u"emphasis"+unicode(self.iddevice.emphasis)+¥" "
    html += u"ondblclick=¥"submitLink('edit'," + self.id + ", 0);¥">¥n"
    html += u'<img alt="" class="iDevice_icon" '
    html += u"src=¥"/style/" + style + "/icon_" + self.iddevice.icon
    html += ".gif¥" />¥n"
    html += u"<span class=¥"iDeviceTitle¥">"
    html += self.iddevice.title + "</span>¥n"
    html += u'<div class="iDevice_inner">¥n'

    for element in self.questionElements:
        html += element.renderPreview() + "<br/>"

    html += '<input type="submit" name="submitScore"'
    html += ' value="%s"/> ' % _("Submit Answer")

    if not self.iddevice.score == -1:
        message = "Your score is " + unicode(self.iddevice.score) + "%"
        html += "<b>" + message + "</b><br/>"

    self.iddevice.score = -1

    html += u"</div>¥n"
    html += self.renderViewButtons()
    html += u"</div>¥n"
    return html

def __calcScore(self):
    """
    Return a score for preview mode.
    """

```

問題作成時のプレビュー時に Python
によって採点(正誤判定)をおこない
点数を返す関数。

```

rawScore = 0
numQuestion = len(self.questionElements)
score = 0
ngcheck = 0

for question in self.iddevice.questions:
    ngcheck = 0
    for option in question.options:
        try:
            float(option.correcttole)
        except ValueError:
            option.correcttole = "0"
        try:
            if not ((float(option.correctmain) - float(option.correcttole)) <= float(option.ans)
<= (float(option.correctmain) + float(option.correcttole))):
                ngcheck = 1
            except ValueError:
                ngcheck = 1
            log.info("userAns " + unicode(option.ans) + ": "
                    + "correctans_from" + unicode(float(option.correctmain) -
float(option.correcttole)) + "to" + unicode(float(option.correctmain) + float(option.correcttole)))
            if ngcheck == 0:
                rawScore += 1

    if numQuestion > 0:
        score = rawScore * 100 / numQuestion

    for question in self.iddevice.questions:
        for option in question.options:
            option.ans = ""

return score

# =====
"""Register this block with the BlockFactory"""
from exe.engine.scormmnidevice import ScormMnIdevice
from exe.webui.blockfactory import g_blockFactory
g_blockFactory.registerBlockType(ScormMnBlock, ScormMnIdevice)
# =====

```

ファイル 3: %(install dir)/exe/webui/scormmnelement.py

```

# =====
# eXe
# Copyright 2004-2006, University of Auckland
# Copyright 2004-2008 eXe Project, http://eXeLearning.org
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
# =====

```

```

"""
ScormMnElement is responsible for a block of question.
Used by ScormMnBlock for SCORM MultiNumerical Question
"""

import logging
from exe.webui.scormmnopolement import ScormMnOpElement
from exe.webui import common
from exe.webui.element import TextAreaElement

log = logging.getLogger(__name__)

# =====
class ScormMnElement(object):
    """
    ScormMnElement is responsible for a block of question.
    Used by ScormMnBlock
    == SCORM MultiNumerical Question
    """
    def __init__(self, index, iddevice, question):
        """
        Initialize
        """
        self.index = index
        self.id = unicode(index) + "b" + iddevice.id
        self.iddevice = iddevice

        # to compensate for the strange unpickling timing when objects are
        # loaded from an elp, ensure that proper iddevices are set:
        if question.questionTextArea.iddevice is None:
            question.questionTextArea.iddevice = iddevice
        self.questionElement = TextAreaElement(question.questionTextArea)
        self.question = question

        self.questionId = "question"+self.id
        self.questionElement.id = self.questionId

        self.options = []

        self.keyId = "key" + self.id

        i = 0
        for option in question.options:
            self.options.append(ScormMnOpElement(i,
                                                    question,
                                                    self.id,
                                                    option,
                                                    iddevice))

            i += 1

    def process(self, request):
        """
        Process the request arguments from the web server
        """
        log.info("process " + repr(request.args))
        if self.questionId in request.args:
            self.questionElement.process(request)

        if ("addOption"+unicode(self.id)) in request.args:
            self.question.addOption()
            self.iddevice.edit = True

```

追加した解答欄の数に応じて
ScormMnOpElement クラスを呼び
出し **options** リストに追加する。

```

        # disable Undo once an option has been added:
        self.iddevice.undo = False

    if "action" in request.args and request.args["action"][0] == self.id:
        # before deleting the question object, remove any internal anchors:
        for q_field in self.question.getRichTextFields():
            q_field.ReplaceAllInternalAnchorsLinks()
            q_field.RemoveAllInternalLinks()
        self.iddevice.questions.remove(self.question)
        # disable Undo once a question has been deleted:
        self.iddevice.undo = False

    for element in self.options:
        element.process(request)

def renderEdit(self):
    """
    Returns an XHTML string with the form element for editing this element
    """
    html = u"<div class=%s%iDevice%s">%n"
    html += common.submitImage(self.id, self.iddevice.id,
                               "/images/stock-cancel.png",
                               _("Delete question"))
    html += self.questionElement.renderEdit()

    html += u"<table width =%s100%%%s">"
    html += u"<tbody>"

    for element in self.options:
        html += element.renderEdit()

    html += u"</tbody>"
    html += u"</table>%n"
    value = _(u"Add Answer Column")
    html += common.submitButton("addOption"+unicode(self.id), value)
    html += u"<br />"
    html += u"</div>%n"

    return html

def renderPreview(self):
    """
    Returns an XHTML string for previewing this element
    """
    return self.renderView(preview=True)

def renderView(self, preview=False):
    """
    Returns an XHTML string for viewing this element
    """
    html = u""

    html += "<div class=%squestion%s">%n"
    if preview:
        html += self.questionElement.renderPreview()
    else:
        html += self.questionElement.renderView()
    html += "<br/>%n"

    html += "<table>"
    for element in self.options:
        if preview:

```

```

        html += element.renderPreview()
    else:
        html += element.renderView()
    html += "</table>"

    html += "</div>¥n"

    return html

def getNumOption(self):
    """
    return the number of options
    """
    return len(self.question.options)

# =====

```

ファイル 4: %(install dir)%/exe/webui/scormmnopelement.py

```

# =====
# eXe
# Copyright 2004-2006, University of Auckland
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
# =====
"""
ScormMnOpElement is responsible for a block of option. Used by
ScormMnElement.

import logging
from exe.webui import common
from exe.webui.element import TextAreaElement

log = logging.getLogger(__name__)
# =====
class ScormMnOpElement(object):
    """
    ScormMnOpElement is responsible for a block of option. Used by
    ScormMnElement.
    == SCORM MultiNumerical Question InputCheck module,
    but with enough subtle variations that you'd better pay attention :-))
    """
    def __init__(self, index, question, questionId, option, ideoice):
        """
        Initialize
        """
        self.index = index

```

```

self.id          = unicode(index) + "q" + questionId
self.question    = question
self.questionId  = questionId
self.option      = option
self.answerId    = "optionAnswer" + unicode(index) + "q" + questionId
self.keyId       = "key" + questionId
self.iDevice     = iDevice
self.checked     = False

# to compensate for the strange unpickling timing when objects are
# loaded from an elp, ensure that proper iDevices are set:
# (only applies to the image-embeddable ones, not FlashElement)
if option.answerTextArea.iDevice is None:
    option.answerTextArea.iDevice = iDevice
self.answerElement = TextAreaElement(option.answerTextArea)
self.answerElement.id = self.answerId

if not hasattr(self.iDevice, 'undo'):
    self.iDevice.undo = True

def process(self, request):
    """
    Process arguments from the web server.  Return any which apply to this
    element.
    """
    log.debug("process " + repr(request.args))

    is_cancel = common.requestHasCancel(request)

    if self.answerId in request.args and not is_cancel:
        self.answerElement.process(request)
        """ 出題者の正答や許容範囲、解答欄への学習者の記入内容を変数に格納している部分 """

    ### fuku correctmain and correcttole
    if self.keyId + "correctmain" + unicode(self.index) in request.args:
        self.option.correctmain = request.args[self.keyId + "correctmain" + unicode(self.index)][0]
    if self.keyId + "correcttole" + unicode(self.index) in request.args:
        self.option.correcttole = request.args[self.keyId + "correcttole" + unicode(self.index)][0]

    ### fuku answer input
    if self.keyId + "Answer" + unicode(self.index) in request.args:
        self.option.ans = request.args[self.keyId + "Answer" + unicode(self.index)][0]

    if "action" in request.args and request.args["action"][0] == self.id:
        # before deleting the option object, remove any internal anchors:
        for o_field in self.option.getRichTextFields():
            o_field.ReplaceAllInternalAnchorsLinks()
            o_field.RemoveAllInternalLinks()
        self.question.options.remove(self.option)
        # disable Undo once an option has been deleted:
        self.iDevice.undo = False

def renderEdit(self):
    """
    Returns an XHTML string for editing this option element
    """
    html = u"<tr><td align='left'><b>%s</b>" % _("Instructions For Answer( and Correct Answer)")
    html += common.elementInstruc(self.question.optionInstruc)

    html += u"</td><td>%n"
    html += u"</td></tr><tr><td colspan=2>%n"

```

```

# rather than using answerElement.renderEdit(),
# access the appropriate content_w_resourcePaths attribute directly,
# since this is in a customised output format
# (in a table, with an extra delete-option X to the right)

this_package = None
if self.answerElement.field_iddevice is not None ¥
and self.answerElement.field_iddevice.parentNode is not None:
    this_package = self.answerElement.field_iddevice.parentNode.package

html += common.richTextArea(self.answerId,
    self.answerElement.field.content_w_resourcePaths,
    package=this_package)

html += "</td>"

### fuku Let's RenderEdit
html += "</tr><tr><td>¥n"
html += common.textInput(self.keyId+"correctmain"+unicode(self.index), self.option.corre
ctmain, 5)
html += u"    &#177    "
html += common.textInput(self.keyId+"correcttole"+unicode(self.index), self.option.corre
cttole, 5)

html += common.submitImage(self.id, self.iddevice.id,
    "/images/stock-cancel.png",
    _(u"Delete option"))

html += "</td></tr>¥n"

return html

def renderPreview(self):
    """
    Returns an XHTML string for previewing this option element
    """
    return self.renderView(preview=True)

def renderView(self, preview=False):
    """
    Returns an XHTML string for viewing this option element
    """
    log.debug("renderView called")

    html = '<tr><td>'
    html += "<br>"
    html += common.textInput(self.keyId+"Answer"+unicode(self.index), self.option.ans, 5)
    html += '</td><td>¥n'

    if preview:
        html += self.answerElement.renderPreview()
    else:
        html += self.answerElement.renderView()
    html += "</td></tr>¥n"

    return html

# =====

```

問題に設置した各正答並びに
許容範囲欄を表示する部分

問題の表示時に、学習者の
解答欄を表示する部分

付録 4 日本語化追記内容

(太字下線部分が追記箇所、無関係の部分は途中省略有)

ファイル : %(install dir)%/exe/locale/ja/exe_ja.po

```
# EXELearning Japanese PO file
# Copyright (C) 2004-2006 EXELearning.org
# This file is distributed under the same license as the EXE package.
# eXe Project <exe@exelearning.org>
#
msgid ""
msgstr ""
"Project-Id-Version: eXe Learning¥n"
"Report-Msgid-Bugs-To: ¥n"
"POT-Creation-Date: 2008-08-04 09:50+1200¥n"
"PO-Revision-Date: 2007-09-03 19:22+1200¥n"
"Last-Translator: Jim Tittsler <jim@exelearning.org>¥n"
"Language-Team: Jun Iwata <iwata@matsue-ct.ac.jp>¥n"
"MIME-Version: 1.0¥n"
"Content-Type: text/plain; charset=UTF-8¥n"
"Content-Transfer-Encoding: 8bit¥n"
"X-EXE-Language: 日本語¥n"
"X-Poedit-SourceCharset: utf-8¥n"
"X-Generator: Pootle 0.10.1¥n"

#: exe/engine/appletidevice.py:68 exe/engine/appletidevice.py:189
msgid ""
" <p>If the applet you're adding was generated ¥n"
"by one of the programs in this drop down, please select it, ¥n"
"then add the data/applet file generated by your program. </p>¥n"
"<p>eg. For Geogebra applets, select geogebra, then add the .ggb file that ¥n"
"you created in Geogebra.</p>"
msgstr ""
" <p>追加するアプレットがこのドロップダウンにあるプログラムの一つを使って作成 ¥n"
"されている場合、それを選択し、その後、あなたのプログラムで作成した ¥n"
"データ／アプレットファイルを追加してください。 </p>¥n"
"<p>例： Geogebra アプレットを追加する場合、「geogebra」を選択し、¥n"
"あなたが Geogebra で作成した .ggb ファイルを追加してください。 </p>"

#: exe/xului/mainpage.py:283 exe/xului/mainpage.py:849
#: exe/xului/mainpage.py:978 exe/xului/mainpage.py:1034
#: exe/xului/mainpage.py:1055
#, python-format
msgid ""
"¥"%s¥" already exists.¥n"
"Please try again with a different filename"
msgstr ""
"¥"%s¥"は、既に存在しています。¥n"
"違ったファイル名で再度お試しください"

~~~~ 途中略 ~~~

#: exe/webui/testquestionelement.py:122 exe/webui/element.py:1766
msgid "Add another Option"
msgstr "別のオプションを追加"

#: exe/webui/scormmnelement.py:118
msgid "Add Answer Column"
msgstr "解答欄を追加"
```

追記

#: exe/webui/quiztestblock.py:112 exe/webui/multiselectblock.py:91
msgid "Add another Question"
msgstr "別の質問を追加"

#: exe/webui/casestudyblock.py:113
msgid "Add another activity"
msgstr "別のアクティビティーを追加"

~~~~ 途中略 ~~~~

#: exe/xului/mainpage.py:903 exe/xului/mainpage.py:945  
#: exe/xului/mainpage.py:1012  
#, python-format  
msgid "Folder name %s already exists. Please choose another one or delete existing one then try again."  
msgstr "フォルダー名 %s は既に存在しています。他の名前にするか、現存するフォルダを削除し、再度試してください。"

追記

#: exe/engine/scormmnidevice.py:126 exe/engine/scormmnidevice.py:186  
msgid ""  
"For each answer column of this %n"  
"question, please mention contents and the explanatory note which you %n"  
"should answer. In two blanks below , Enter a right answer (numerical value) %n"  
"and a tolerance (numerical value)."  
msgstr ""  
"この質問のそれぞれの解答欄に対して、答えるべき内容や注釈を記載してください。ま  
た、下の2つの空欄には正しい答え（数値）と許容誤差（数値）を入れてください。"

#: exe/webui/externalurlblock.py:81  
msgid "Frame Height:"  
msgstr "フレームの高さ : "

~~~~ 途中略 ~~~~

#: exe/engine/clozeidevice.py:116 exe/engine/truefalseidevice.py:165
#: exe/engine/truefalseidevice.py:392 exe/webui/common.py:330
msgid "Instructions"
msgstr "ガイド"

追記

#: exe/engine/scormmnidevice.py:48 exe/engine/scormmnidevice.py:100
#: exe/webui/scormmnopement.py:100
msgid "Instructions For Answer(and Correct Answer)"
msgstr "解答への指示（並びに正答）"

#: exe/engine/clozeidevice.py:274
msgid "Instructions For Learners"
msgstr "学習者への指示"

~~~~ 途中略 ~~~~

#: exe/webui/editorpane.py:195  
msgid "Please enter<br />an idevice name."  
msgstr "<br />iDevice の名前を記入してください。"

追記

#:exe/webui/scormmnblock.py:127exe/webui/scormmnblock.py:133exe/webui  
/scormmnblock.py:136  
msgid "Please input correct answer(main) with numerical value."

msgstr "正しい解答を数値にて入力してください"

#: exe/webui/scormmnblock.py:146 exe/webui/scormmnblock.py:149

追記

msgid "Please input correct answer(tolerance) with numerical value."

msgstr "正しい解答の許容誤差を数値にて入力してください(許容誤差の入力を省略すると0(誤差を許さない)として扱われます)"

#: exe/webui/element.py:1359

#, fuzzy

msgid "Please select a .flv file."

msgstr "フォーラムを選択してください"

~~~~ 途中略 ~~~~

#: exe/engine/truefalseidevice.py:144

msgid "True-False Question"

msgstr "正誤問題"

#: exe/engine/scormmnidevice.py:247

msgid "SCORM MultiNumerical Question"

追記

msgstr "SCORM 複数数値問題"

#: exe/engine/truefalseidevice.py:146

msgid ""

"True/false questions present a statement where ¥n"

"the learner must decide if the statement is true. This type of question works ¥n"

"well for factual information and information that lends itself to either/or ¥n"

"responses."

msgstr ""

"「正誤問題」は、記述された内容が正しいか違っているか学習者に判断させるための ¥n"

"質問形式です。この種類の質問形式は、事実に基づいた情報や、正しいか間違っ¥n"

"いるかの判断がしやすい情報を利用する際に効果があります。"

~~~~ 途中略 ~~~~

#~ msgid "Strong emphasis"

#~ msgstr "強調 (最大) "

#~ msgid "Unable to connect to "

#~ msgstr "接続できません"

#~ msgid "some help tip here."

#~ msgstr "ヒントがあります"