

## **iLOG: A Framework for Automatic Annotation of Learning Objects with Empirical Usage Metadata**

**L.D. Miller**, *Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA*

*lmille@cse.unl.edu*

**Leen-Kiat Soh**, *Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA*

**Ashok Samal**, *Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA*

**Gwen Nugent**, *Center for Research on Children, Youth, Families & Schools, University of Nebraska-Lincoln, Lincoln, NE, USA*

**Abstract.** Learning objects (LOs) are digital or non-digital entities used for learning, education or training commonly stored in repositories searchable by their associated metadata. Unfortunately, based on the current standards, such metadata is often missing or incorrectly entered making search difficult or impossible. In this paper, we investigate automating metadata generation for SCORM-complaint LOs based on user interactions with the LO and static information about LOs and users. Our framework, called the Intelligent Learning Object Guide (iLOG), involves real-time tracking of each user sessions (an LO Wrapper), offline data mining to identify key attributes or patterns on how the LOs have been used as well as characteristics of the users (MetaGen), and the selection of these findings as metadata. Mechanisms used in the data mining include data imputation via clustering, association rule mining, and feature selection ensemble. This paper describes the methodology of automatic annotation, presents the results on the evaluation and validation of the algorithms, and discusses the resulting metadata. We have deployed our iLOG implementation for five LOs in introductory computer science topics and collected data for over 1400 sessions. We demonstrate that iLOG successfully tracks user interactions that can be used to automate the generation of meaningful empirical usage metadata for different stakeholder groups including learners and instructors, LO developers, and researchers.

**Keywords.** SCORM learning objects, empirical usage metadata, association rule mining, feature selection, data imputation

### **INTRODUCTION**

Learning objects (LOs) are defined as any entity, digital or non-digital, that may be used for learning, education, or training (LTSC, 2002). One of the most commonly used standards is the Sharable Content Object Reference Model (SCORM) developed by the Advanced Distributed Learning (ADL) initiative (ADL, 2003). SCORM-compliant LOs (hereafter LOs) commonly include (1) a tutorial component which explains the content, (2) a practice component where students can work through examples on the content, (3) and an assessment component which evaluates whether students have learned the content. The tutorial and practice components can provide active learning and feedback which promote higher student learning (Nugent, et al., 2009). Thus, an obvious approach is to augment traditional or online education programs with self-contained

LOs relevant to segments of the course. This could be done by using LOs in online repositories (e.g., MERLOT ([www.merlot.org](http://www.merlot.org)), Maricopa ([mcli.maricopa.edu](http://mcli.maricopa.edu)), Careo ([www.ucalgary.ca/commons/careo](http://www.ucalgary.ca/commons/careo)), ARIADNE ([www.ariadne-eu.org](http://www.ariadne-eu.org)), etc.). However, in order for LOs to be useful, instructors must be able to efficiently locate suitable LOs for the course without the need for an exhaustive manual search. The learning object metadata (LOM) standard (LTSC, 2002) was originally designed to facilitate searching for LO. LOM provides 80+ metadata fields intended to describe the content of the LO. Unfortunately, there are several problems with the LOM standard (Brooks, et al., 2005): (1) fields are often omitted by authors, (2) lacks semantic interoperability, (3) is not supported in learning management systems (LMSs), (4) only allows for a single metadata instance per LO. There has been considerable work on automatic metadata generation to support search such as the harvesting and validation systems provided by the ARIADNE foundation, but this only addresses the first problem. Furthermore, there is need for additional metadata to support different groups such as learners, instructors, LO designers and researchers (Ravasio, 2003; Kosba et al., 2007). Ravasio (2003) discusses the need for teacher- and student-friendly metadata. Metadata should support designers making revisions to the LOs and also support users trying to find and access them. According to Kosba et al. (2007), different kinds of metadata are required for learners and instructors in an online course. Instructors need to know about problems that learners are having with LOs. Metadata for instructors helps identify problems by providing high-level analyses based on user interactions. Providing such analyses also reduces cognitive and communication overhead for instructors. On the other hand, learners often require personalized help on LO content. Metadata for learners helps by providing advice on completing the LOs. Providing such advice also improves student satisfaction with the LOs.

McCalla (2004) propose an alternative theory of metadata based on user interactions designed to address all the above problems. This ecological approach includes static information about users (e.g., cognitive, affective, social characteristics), user's perspective on content, and interactions with the LOs (e.g., clicks, time spent, etc.). The main difference between the ecological approach and LOM is that *the ecological approach focuses on descriptive metadata mined from user interactions which gradually accumulate rather than prescriptive metadata from content developers* (Brooks et al., 2005). The Intelligent Learning Object Guide (iLOG) framework proposed in this paper implements this ecological approach theory.

First, iLOG allows for automatic annotation of empirical usage metadata (hereafter metadata). Such metadata is generated based on actual user interactions with the LOs and any static information. There are two types of static information in iLOG. Static user information includes demographic surveys, motivation and self-efficacy tests, placement exams, prerequisite quizzes, etc. Static LO information includes LO difficulty scores, evaluation ratings from previous semesters, configuration details, etc. The combination of user interactions and static information allows for metadata to be created which balance both user interactions which are more objective and the static information which is more often used for search.

Second, iLOG allows for semantic interoperability which also addresses the need for additional metadata. This is done by using data mining techniques including data imputation, feature selection, and association rule mining to generate metadata with an unambiguous meaning based on first-order logic. In this way, iLOG is similar to the "pragmatic web" described in McCalla (2004). In particular, we are interested capturing user interactions with learning objects and mining them for metadata to improve LO usability for different groups (e.g., learners and instructors, LO designers, and researchers). Learners could use the metadata as guidelines for success in the course. For example, the metadata could indicate that previous learners who did not go through the exercises had a much higher tendency to fail the LO. Additionally, LO designers could use the metadata to determine whether the LO is effectively presenting content to the users. For example, female students tended to perform poorly on the assessment component of the LOs in introductory computer science (CS) courses (Riley et al., 2009). Obviously, the LO designer should correct this gender bias by revising one or more LO components. Finally, researchers could use the metadata as initial exploration

to support subsequent, in-depth analysis. For example, a researcher might want to investigate why learners with high motivation who spent more than the average time on the assessment tended to fail the LOs in introductory CS courses (Riley et al., 2009). The researcher may find that such learners become discouraged when they cannot find the correct answers immediately. Finally, researchers could use the metadata as initial exploration to support subsequent, in-depth analysis. For example, a researcher might want to investigate why learners with high motivation who spent more than the average time on the assessment tended to fail the LOs in introductory CS courses (Riley et al., 2009). The researcher may find that such learners become discouraged when they cannot generate the correct answers immediately.

Third, iLOG overcomes both the lack of LMS support and the single metadata instance per LO using the LO Wrapper which can be added to existing LOs. The LO Wrapper logs all the user interactions to an external database without support from the LMS. Note that our user base consists of both instructors and learners. In the rest of the paper, “users” is used to refer to both. Also, the LO Wrapper adds metadata to LOs using a generation format which supports multiple metadata instances per LO.

In the following, we first present related work to our research in terms of metadata and automatic metadata generation for learning objects. We also look more specifically at annotated metadata derived from association rules mined from data and summarize the relationship between metadata and the LO repository. Second, we describe in detail our methodology for iLOG, focusing on the automation: from tracking to metadata generation. On tracking, we present the module called the LO Wrapper, that can be embedded with standard learning management systems (LMSs) that administer SCORM-compliant LOs. On metadata generation, we present MetaGen and its three key steps: data imputation, feature selection ensemble, and association rule mining. Third, in the RESULTS section, we provide two discussions, one on the validation of the algorithms used in the automation, and the other on the validation of annotated metadata from user interactions with online learning objects. Finally, we provide a high-level summary of the iLOG framework and discuss future improvements to its components.

We have created several LOs on introductory computer science (CS) concepts. Each of these LOs contained (1) a tutorial, (2) exercises and (3) assessment components consistent with the organizational guidelines given in Thompson (2005). Preliminary work for automatic metadata annotation from user interactions collected from these LOs was encouraging (Riley et al., 2009). The iLOG prototype was able to generate metadata with both high confidence for each of the learning objects considered. From these results, we were able to gain useful insights for the usage properties for different types of students. Further, Nugent et al. (2009) used the iLOG system prototype to evaluate instructional strategies, specifically, the impact of active learning and feedback on student learning. In this work, we have expanded the iLOG system with many refinements for automatic metadata generation. We also provide a rigorous validation for the metadata generated using iLOG based on three metrics for learning object metadata (Ochoa, 2008). The metadata is *complete* because iLOG considers all relevant features when creating association rules from user interactions. Second, the metadata is *accurate* because iLOG provides the confidence values for all the metadata. Third, the metadata has *provenance* because iLOG system updates the metadata on existing LOs based on new user interactions.

## RELATED WORK

In this section, we first provide existing work on learning object (LO) metadata standards. Second, we discuss two different approaches for automatic metadata generation based on (1) the LO content and (2) user interactions with the LO. The iLOG framework adopts the second approach. Finally, we describe existing work on automating LO selection from repositories.

## Learning object metadata

There has been a considerable interest involving metadata for LOs. Several standards for specifying what metadata to include with LOs have been generated; the most commonly used is the IEEE LTSC Learning Object Metadata (LOM) standard (LTSC, 2002). Friesen (2004) provides a brief description for the organization of metadata consistent with the LOM standard. The LOM standard for metadata is the most widely accepted, but it is far from perfect as alluded to earlier. Other work has also cited deficiencies in the LOM standard. First, it lacks metadata on the quality of learning objects as judged by the users (Vargo, 2003). Second, according to Polsani (2003), one of the functional requirements for LOs is accessibility. This is done by tagging the LO with metadata so it can be stored and referenced in a repository. However, current metadata standards do not require the content developer to provide all the metadata. This often leads to omitted metadata that minimizes accessibility for the LOs. Friesen (2004) conducted an international survey on the implementation of the LOM standard and found that much of the metadata was not provided by human users making it difficult to incorporate into the LO design and meta-tagging processes. Finally, Cardinaels (2006) discussed the need for confidence in the precision or accuracy of metadata for a specific LO and for context-aware metadata. Recently, a study was conducted on human-generated, metadata using the LOM standard (Cechinel, 2009) in which students used an annotation tool to enter the metadata. The results show a high percentage error from students on entering the correct metadata—as much as 25% on some metadata sections—despite years of refinement to metadata standards and annotation tools. Thus, even with the LOM standard, there is still a need for automating the generation of metadata which is more efficient, less costly and more-consistent than human processing (Roy, 2008).

An alternative to automatic metadata generation is forcing the developers to provide the metadata. Bailey (2006) discusses learning activity nuggets which contain specific elements such as subject area, level of difficulty, prerequisite skills, environment, etc. Nuggets have a set of metadata which is automatically populated when a nugget is created using the online editor. However, this approach does not follow any standard and is unlikely to be widely adopted.

## Automating metadata generation

There has been considerable work in the last ten years involving automating metadata generation for LOs. The first common approach focuses on the LO content (Cardinaels, 2005; Gasevic, 2005; Javanovic, 2006; Saini, 2006; Zouaq 2007a; Zouaq 2007b; Roy 2008). This approach first populates an ontology using the content and then generates metadata from the ontology. The main advantage for ontology (Javanovic, 2006) is convenience in searching through LO repositories. Semantic web reasoning could search for LOs with content of a certain type using context ontology, dealing with a certain topic using a domain ontology, or with a certain level of granularity using a structural ontology. The ontologies used in this approach are either (1) provided by the content developer (Gasevic, 2005; Javanovic, 2006; Saini, 2006) or (2) populated using natural language processing to extract keywords, word dependencies, and concepts from the LO content (Zouaq, 2007a; Zouaq, 2007b; Roy, 2008).

Cardinaels (2005) discusses an automatic indexing framework which generates metadata for SCORM-based LOs. This framework consists of two components. First, the context-based indexer generates metadata based on when the LO is used. Second, the object-based indexer generates metadata based on organization of the learning object (e.g., types of files included). In a case study this framework was able to automatically generate metadata fields included with SCORM. However, these fields were limited to metadata about the *structure* of the LO.

TANGRAM (Jovanovic 2006) uses an ontology approach for the metadata. It employs structural and context ontologies for storing the content for the learning objects. It also maintains ontologies for learner paths and models. The learner models are generated from initial questionnaires. TANGRAM allows a content developer to upload new LOs to a repository, automatically tagging them with high-quality metadata, search the LO repository, and compose a new LO using components from existing LOs. However, the majority of the metadata required for annotating the LO must first be supplied manually by the content author. After this is done, TANGRAM automatically annotates the LO components and integrates the LO into the repository. This annotation is consistent with IEEE LOM standard. The main difference between TANGRAM and iLOG is that, in TANGRAM, user interactions play no part in the automatic generation of the metadata. Initially, the metadata is generated based on sample metadata supplied by the developer. *Subsequent user interactions with the LO are stored on a separate ontology and never used to revise the metadata.*

Roy (2008) uses an algorithm to identify concepts in the text for learning objects in three different subjects (physics, biology and geography). The algorithm distinguishes between outcome concepts necessary for the learning goal and prerequisite concepts which the user must understand before the outcome concepts. Concepts are extracted using a shallow parsing approach for identifying verbs used in definitions (e.g., defined, derived, called, etc.). The algorithm uses a three-layered hierarchical knowledge base. First, the term layer stores lexical terms (i.e., keywords). Second, the concept ontology contains the relationships between domain-specific concepts. Third, the topic layer organizes the concepts, discussed for each topic, based on the learning requirements for the institution. The algorithm uncovered many of the same concepts that were also manually observed by human experts. The automatic annotation tool adds these concepts in a machine comprehensible format compliant with the IEEE LOM standard. *This algorithm computes the metadata using only the LO content whereas iLOG uses both the LO content and user interactions to compute the metadata.*

Finally, Saini (2006) provides an algorithm for the automatic classification of the LO into an ontology based on the LO content. This method uses a semi-supervised algorithm based on expectation maximization (EM) where the keywords available in the ontology are used for bootstrapping the classification of the LOs.

## Metadata annotation

There has been previous work on metadata annotation based on user interactions with the LOs (Minaei-Bidgoli et al., 2004; Brooks et al., 2005; Etchells, 2006; Wolpers, 2006; Castro, 2007; Garcia, 2009a; Garcia, 2009b; Segura, 2009; Liu, 2009). The general underlying paradigm is mining the stored user interactions into suitable metadata (Castro, 2007). In particular, association rule miners are often used for automatic metadata generation (Minaei-Bidgoli, 2004; Garcia, 2009a; Garcia, 2009b; Segura, 2009) because they provide human-comprehensible metadata and an evaluation metric. However, other algorithms have been tried including Bayesian belief networks (Liu, 2009) and artificial neural networks (Etchells, 2006). Here we review these algorithms briefly and distinguish them from iLOG.

The Learning Online Network with Computer-Assisted Personalized Approach (LON-CAPA) (Minaei-Bidgoli, 2004) employs association rule mining to describe user interactions with online course work. The mining contrast rules (MCR) algorithm in LON-CAPA computes a set of conjunctive, contrast rules based on (1) user features, such as GPA and gender, (2) problem features, such as the assignment difficulty, and (3) user/problem interactions, such as the number of attempts and time spent on the assessment. The MCR computes association rules based on whether the user passed/failed the LO and each rule has both a rule support and confidence value associated with it. This is very similar to the association rule miner component

in the iLOG system. However, there is no provision in MCR for replacing missing values. Further, MCR assumes that all features are potentially relevant to the association rules. *Thus, MCR requires hand-tuning to avoid being swamped with less interesting rules based on irrelevant features. The imputation and feature selection components in the iLOG system handle both eventualities.*

The iHelp LMS (Brooks et al., 2005) is based on the ecological approach discussed earlier (McCalla, 2004). This system uses semantic web-based middleware to generate small packages of user interactions called “events”. Events are marked up using a resource description framework schema. This is done by mapping each question/answer pair to a particular concept in the domain and education objectives ontology. Different questions/answers represent different levels of learner understanding (e.g., knowledge, cognition, etc.). The iHelp LMS also tracks user interaction data such as which LO were read, how long and in what order, etc. The iLOG system tracks user interactions on individual pages in the LO allowing for higher resolution metadata. For iHelp, such events and user interaction data are used by software agents for analysis and action—specifically choosing the next LO present to the learners. As alluded to earlier, iLOG generates metadata for different groups (e.g., learners and instructors, LO designers, and researchers). Brooks et al.(2005) concede that modifying the LMS to track user interactions hurts interoperability. The iLOG system embeds tracking in the LO Wrapper which allows for much more flexibility in the LMS used for SCORM-compliant LOs (e.g., Blackboard, Moodle, etc.).

The CAM framework (Wolpers, 2007) intercepts user interactions with many applications such as the web browser. These user interactions are converted into metadata which are stored on an external database. This approach is very similar to the wrapper described for the iLOG system. Both intercept user interactions and send them to an external database. In CAM, the transmission is one-way because metadata never leaves the database. However, in iLOG the transmission is two-way. *The metadata generated by iLOG is automatically annotated to the existing LOs making it more widely available to different groups including the learners and instructors.*

Garcia (2009a) uses the Apriori algorithm for association rule mining on user interactions with e-learning courses. This algorithm provides recommendations to the instructors based on the association rules. It employs reinforcement learning based on instructor responses and expert evaluation. Unlike iLOG, this system requires the active assistance of the instructor during metadata annotation.

Segura (2009) combined the clustering technique with association rule mining on LOs from several repositories. First, this method clusters all the LOs based on the LOM metadata included in each LO (i.e., metadata used as features for clustering). Second, it used the Apriori association rule miner, separately, on the metadata in each cluster. This method is similar to the imputation component in iLOG. Both use clustering algorithm to create partitions where missing feature-value pairs can be filled in from similar values. However, the iLOG imputation uses a more complex hierarchical approach than the K-Means clustering algorithm. The need for additional complexity in our solution is due to the absence of the correct number of clusters as that number cannot be pre-determined, rendering simpler solution such as the K-Means clustering algorithm in appropriate. As discussed later, dynamic tree cut is used to choose the correct set of clusters among those created by hierarchical clustering. Further, as discussed later in the ‘Data Imputation’ section, iLOG can decide to not use clustering at all when increasing amounts of missing data make the clusters less viable.

The Eliminating and Optimized (EOS) Selection algorithm (Liu, 2009) is designed to select a suitable set of LOs for users from a repository. EOS uses a Bayesian Belief network to compare the user features collected from the survey data with user features including gender, year of student, major, reading level, etc. The learning object features include pedagogical objective, environment, expected reading level, etc. The network is trained on the collected survey data and LO features subjectively specified for each LO. The network computes a different weight for each combination of user and LO features. These combinations are metadata used to select LOs for each user based on the specific user features from the survey. Both

EOS and iLOG select which features are relevant for the metadata. EOS considers combinations in the network with a significantly high weight, while iLOG employs feature selection to choose the subset of features most relevant to the assessment. The LO features for EOS incorporate some aggregate information from previous users (e.g., duration the LO access, the number of help requests, assessment result for the user, etc.). However, the emphasis is on the survey results from the user and the aggregate information is not automatically collected. The iLOG system also employs surveys, but there is a greater emphasis on evaluating user interactions which are automatically collected from the database. Additionally, the Bayesian Belief network used in EOS can only compare user and LO features. The association rule miner in the iLOG system compares all the features (i.e., both user and LO) together.

Finally, EtcHELLS (2006) discusses finding usage features in LOs to predict student final grades. A fuzzy inductive reasoning (FIR) is used for feature selection and a neural network for orthogonal search-based rule extraction (OSRE). This approach uses one feature selection algorithm, rather than the ensemble approach used by iLOG, which could result in fewer relevant features identified compared to the ensemble. Further, the neural network requires hand-tweaking to prevent overfitting.

## LO repositories

There has been considerable work on organizing LOs into repositories. Vargo (2003) suggested organizing repositories using levels of learning objects. Level 1 refers to single page, Level 2 to a lesson, Level 3 refers to a collection of Level 2 objects, (e.g., a course), and Level 4 refers to a set of courses that leads to a certificate. Unfortunately, such an organization has yet to be adopted. At present, existing LOs are stored in repositories such as Campus Alberta Repository of Educational Objects, Federal Government Resources for Educational Excellence, FreeFoto, Maricopa Learning Exchange, Merlot, Wisconsin Online Resource Center (Nash, 2005). These repositories are searchable based on LO metadata. However, there are three problems with searching for learning objects (Nash, 2005). First, the LOs are not interchangeable due to size, or inconsistent languages. Many also have a cultural bias. Second, there is an inconsistent classification scheme. Specifically, the learning levels for LOs (K-12 through graduate) are not specified. Third, the quality for LOs is highly variable in terms of production, classification, etc. Tompsett (2005) discusses how it can be very difficult for developers to create new courses from LOs stored in repositories. This is due to the difficulty of finding a set of LOs which integrate together well while still covering all the topics in the course. There is some existing work in helping developers to select LOs from repositories (Karampiperis & Sampson, 2004; Broisin; 2005).

Karampiperis & Sampson (2004) gives an algorithm to automatically select LOs from a repository by emulating the human-based LO selection process. This is done by training a classifier on the metadata for LOs selected by the developer over a small-scale test. The downside to this approach is that it requires all the LOs to use the same set of metadata. Additionally, this approach can only be used to find LOs similar to those originally selected by the developer. Thus, this approach will not replace the need for developers to search LO repositories.

Broisin (2005) gives a service oriented architecture consisting of three layers: (1) learning management system (LMS) to deliver courseware, (2) learning object repository (LOR) to manage LOs and (3) mediation layer which bridges the LMS and LOR. This architecture automatically extracts a variety of metadata from the LMS and updates the LO in the repository. This includes general metadata (such as the title), semantics metadata (such as the science type and main discipline), pedagogical metadata (such as the role of the user), and technical metadata (such as the required operating system). On the surface, this approach is similar to that used by iLOG. However, the metadata supplied by this architecture is based entirely on the LO content. There is no consideration for creating metadata from user interactions with the LO as in iLOG. Thus, the

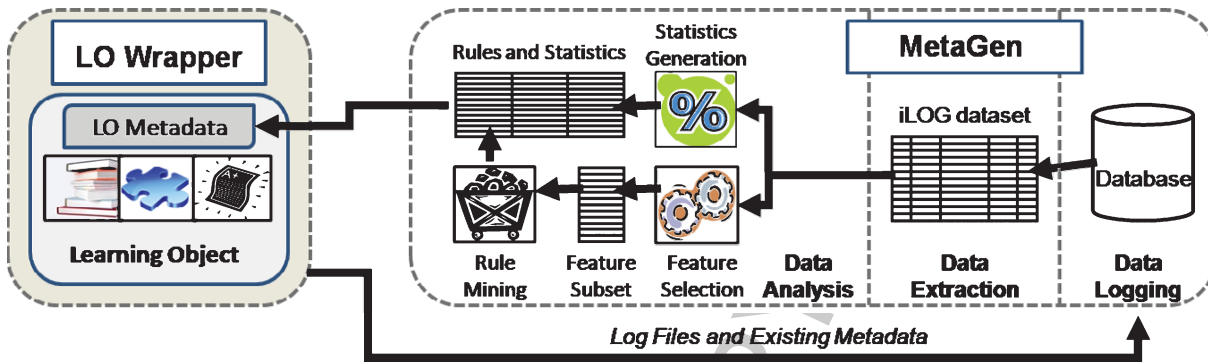


Fig. 1. The Intelligent Learning Object Guide (iLOG) framework (from Riley et al., 2009).

performance of the LO is not considered in the metadata making it more difficult for developers to choose suitable LOs from the repository.

## METHODOLOGY

In this section, we describe the two halves of the iLOG framework (see Fig. 1) for automatic LO annotation. First, the **LO Wrapper** surrounds existing LOs and intercepts user interactions between the user and the LMS. These user interactions are logged to the external iLOG database. The wrapper also appends metadata generated by the iLOG framework to the existing LOs. Second, the **MetaGen** system is used by iLOG for automatic generation of metadata. MetaGen first extracts user interaction and static user/LO data into a self-contained dataset. MetaGen then analyzes the dataset using feature selection and rule mining components to create rules which are used as LO metadata. The iLOG framework adheres to both the SCORM (Dodds, 2001) and LOM (LTSC, 2002) standards. For existing SCORM-compliant LOs, using the iLOG framework only requires adding the LO Wrapper. This can be done automatically using an upload script that we have developed which automatically copies the LO Wrapper and any annotated metadata to the LO's zipped file. None of the other files in the LO are altered in any way. The LOs can then be uploaded to any SCORM-complaint LMS. The LO Wrapper automatically stores the user interactions in real-time. MetaGen runs offline, but is fully automatic and can be run whenever new metadata is required.

## Automation

Generally, tracking user interactions with the LOs requires modifications to the LMS where the LOs are displayed to users. The downside to this approach is that it requires non-standard modifications to the LMS which, as discussed in Brooks et al. (2005), severely restrict the interoperability for these LOs and the potential user-base both of which are inconsistent with the SCORM standard. Thus, iLOG provides LOs with their own capability for tracking user interactions. First, this allows the LO to be deployed seamlessly using any existing SCORM-compliant LMS. Second, interested parties could then access the LOs directly to obtain stored user interactions. Miller et al. (2008) proposes adding this capability to the SCORM 2.0 project. However, this capability does not currently exist. Instead, we use the LO Wrapper which can be easily integrated into any SCORM-compliant LO.

The LO Wrapper uses the Easy Shareable Content Object (SCO) Adapter for SCORM 1.2 (<http://www.ostyn.com/standards/demos/SCORM/wraps/easyscoadapterdoc.htm#license>). The SCO



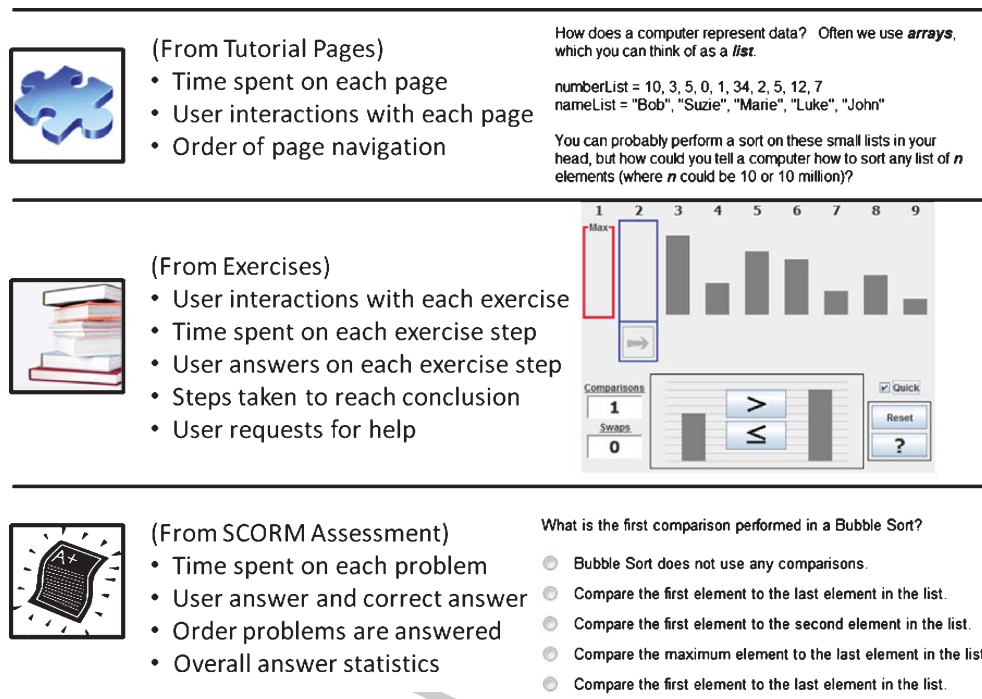


Fig. 2. User interactions captured using the LO Wrapper with corresponding example of LO content.

adapter provides a direct interface with the SCORM API the LMS uses for displaying the LO. This connection to the SCORM API updates the LO Wrapper when pages are displayed to the user and also provides information about the assessment component. The LO Wrapper also uses existing web technologies including JavaScript and PHP to create a bridge between the LO and an external database. Using this bridge, the wrapper transmits user interactions to the database (in a separate location) and metadata back to the LO. This bridge requires a connection to the Internet, but this is generally not an issue because such a connection is also required for most LMSs. To address privacy concerns, personal information (e.g., user names, demographic information, etc.) are stored in an external, password-protected database behind a Firewall. Data transmitted over the internet is encrypted and metadata annotated to the LOs are based on aggregated user interactions containing no personal information.

Figure 2 summarizes the user interactions *automatically* captured by the LO Wrapper with corresponding examples of LO content for each component in iLOG LOs (i.e., tutorial, exercise, and assessment). In the tutorial, the wrapper captures user interactions with each page by hooking into the mouse events in the hypertext markup language (HTML) for LO pages in SCORM-compliant LOs. From these mouse events, the wrapper can deduce the type of user interactions. For example, the wrapper can distinguish between users scrolling down a page in the LO or clicking on an external hyperlink. The wrapper stores such user interaction events in collections in the JavaScript. Additionally, the LO Wrapper is notified by the interface with the SCORM API when new LO pages are loaded in the tutorial. The wrapper updates collections in the JavaScript such as user interaction navigations along with the time spent on each page. In the exercise, the LO Wrapper uses a direct interface with the exercise to collect user interactions from inside the embedded exercise. The wrapper provides an interface for exercises written in Flash and for those written as Java Applets. This interface allows the wrapper to collect user interactions about specific steps in the exercise. For example, the wrapper can obtain the time spent on the first sorting step in Fig. 2 and whether or not the

user got the correct answer. The wrapper also stores the order steps are taken to reach the end of the exercise and any steps that prompt user requests for help. In the assessment, the LO Wrapper stores information received from the SCORM API on each assessment problem. This includes the name of the problem, time spent on the problem, type of the problem (e.g., multiple-choice), correct answer, user's answer, weight of the problem, etc.

MetaGen runs on a server external to the LMS. Currently, MetaGen is run offline to generate new metadata after LOs are deployed to the LMS. First, missing feature-value pairs are filled in using a data imputation component (discussed below). Second, MetaGen automatically computes empirical usage metadata using ensemble feature selection and association rule mining component. All components are discussed in more detail below. Further, MetaGen has modest resource requirements.

The runtime for MetaGen on the iLOG dataset with 104 feature and 1455 records is 3 minutes on an Intel Quad Core running at 2.5 GHz with 6.00 GB RAM. The runtime scales linearly with the number of records so a dataset with 2910 records takes 6 minutes. The size of the iLOG dataset is 834 KB.

## MetaGen components

The MetaGen framework uses three separate modules for automatic metadata annotation: (1) data logging, (2) data extraction, and (3) data analysis. First, the data logging module of MetaGen integrates data from three sources: (1) static LO data, (2) static user data, and (3) user interactions from the LO Wrapper. Next, the data extraction module creates the iLOG dataset from the external database. Each record in the dataset corresponds to a particular user-LO session with a *label* (i.e., class value) to estimate the learning outcome based on whether the user passed/failed the LO. First, this module uses the *Data Imputation* component to fill in the missing feature-value pairs for the records. Second, this module uses the *Feature Selection Ensemble* component to select the most relevant features. Finally, this module uses the *Association Rule Miner* component to generate the metadata. Such metadata and any usage statistics specified by the content developer (e.g., placement exams, demographic information, etc.) are annotated using XML format to the LO zip file.

We next discuss all three important components in the MetaGen framework: (1) Data Imputation from the data extraction module, (2) Feature Selection Ensemble from the data analysis module and (3) Association Rule Miner also from the data analysis module.

### *Data imputation*

In our previous work (Riley et al., 2009) we discovered there were some records in the iLOG datasets which contained missing feature-value pairs. For our LOs, users are required to complete the LO from beginning to end. Specifically, each user must go through all the tutorial pages and answer all the assessment pages. While a user is not forced to interact with the exercises, skipping exercises results in zeros for corresponding user interactions rather than unknown values. Thus, we can be confident that missing feature-value pairs are the result of non-deterministic problems with the database bridge (e.g., temporary loss of connection) rather than users deliberately skipping harder-to-understand concepts.

Many such records contain both missing and present feature-value pairs. The missing feature-value pairs make it difficult to use this record for feature selection and rule mining. However, simply discarding any record with missing feature-value pairs wastes a considerable amount of potentially interesting metadata. We would like to utilize such records in data analysis rather than preprocessing to remove all records with missing feature-value pairs. To facilitate this, we have added a Data Imputation component to the MetaGen framework which fills in the missing feature-value pairs in the dataset records.

The Data Imputation component uses the Cluster-Impute algorithm which employs (1) hierarchical clustering, (2) dynamic tree cuts, and (3) linear regression classifier to fill in the missing feature-value pairs. Cluster-Impute employs a hierarchical clustering (Johnson, 1967) separately on both the records and the features. This clustering consists of an agglomerative approach which starts with clusters containing a single record. These clusters are merged together progressively until there is only a single cluster containing all the records. Then, Cluster-Impute uses a dynamic tree cut algorithm (Langfelder et al., 2008) to choose a set of clusters which best represents the natural cluster shapes in the dataset. We specify the minimum number of records (= 5) and attributes (= 3) in each cluster for dynamic tree cut based on previous work (Miller et al., 2011). These values are small because we do not want to impose a large cluster size minimum as this could skew the clustering results by forcing dissimilar records into the same clusters. Finally, our algorithm trains a linear regression classifier (Hand et al., 2001) on the clusters to impute the missing values. This is done by fitting a linear model to the records or features in each cluster and using the model with the least error to predict the missing values. This type of imputation exploits both the cluster membership and any high correlation between the features.

However as the number of missing values in the dataset increase cluster membership becomes less useful. Therefore, the missing threshold determines which type of imputation is used by Cluster-Impute. If the number of missing values is reasonable (i.e., less than threshold) then the missing values are imputed using the clusters as described above. On the other hand, if the clusters contain records with mostly missing values (i.e., greater than threshold) then we cannot create viable clusters so we instead use the linear regression model to impute the missing values from all the records together (without the benefit of cluster membership and feature correlation). We chose 50% as the minimum threshold based on previous work (Miller et al., 2011) showing that clusters are not viable with more than 50% missing values. The Cluster-Impute algorithm is validated later in the RESULTS section.

### ***Feature selection ensemble***

Features in the iLOG dataset are collected from many different types of user interactions with the learning objects (see Fig. 2 for examples from each LO component). For the iLOG dataset, not all the features collected are equally relevant to the label which estimates the learning outcome based on whether the user passed/failed the LO. The inclusion of features irrelevant to the label often degrades the classification model (Hand, et al., 2001). Unfortunately, we do not know which features are relevant in the iLOG dataset when running MetaGen. As a result, MetaGen uses *feature selection* algorithms to choose the relevant subset of features used for the entire iLOG dataset.

The MetaGen feature selection component uses feature selection (FS) algorithms from the Weka software library (Witten & Frank, 2005). There are two different types of feature selection algorithms in Weka: (1) subset evaluation and (2) feature evaluation. The *key difference* between the two types is the way they evaluate the feature. The subset evaluation algorithms evaluate the features together, while the feature evaluation algorithms evaluate them separately. Subset evaluation algorithms use a search algorithm to find subsets of features and each uses a distinct fitness function to evaluate the subsets. Features are added to subsets only if they improve fitness. After searching, the algorithm returns the *subset of features with the highest fitness*. On the other hand, feature evaluation algorithms evaluate each feature separately. Each feature evaluation algorithm uses a distinct fitness function to individually evaluate the features. After evaluation, the algorithm returns *all the features each with a score based on the fitness values*.

Neither type of feature selection algorithm is intrinsically superior to the other. Rather, each type specializes on finding different kinds of relevant features based on the feature-value pairs in the dataset. The iLOG dataset contains different kinds of features collected from user interactions. Thus, to improve feature selection in MetaGen we employ an *ensemble* of feature selection algorithms. In an ensemble approach,

Table 1

Feature selection algorithms in MetaGen ensemble. Type denotes whether the algorithm evaluates features together (i.e., SUBSET) or separately (i.e., FEATURE)

FS Algorithm	Type	Summary of fitness function
CfsSubsetEval	SUBSET	Correlation statistic
ClassifierSubsetEval	SUBSET	Naive Bayes classifier
ConsistencySubsetEval	SUBSET	Consistency statistic
CostSensitiveSubsetEval	SUBSET	Correlation statistic with cost matrix
FilteredSubsetEval	SUBSET	Correlation statistic on subset of records
WrapperSubsetEval	SUBSET	Naive Bayes classifier with cross validation
ChiSquaredAttributeEval	FEATURE	Chi-squared statistic
ReliefFAttributeEval	FEATURE	RELIEF estimate for features
SymmetricalUncertAttributeEval	FEATURE	Symmetrical uncertainty estimate on label
CostSensitiveAttributeEval	FEATURE	RELIEF estimate for features with cost matrix

multiple algorithms are run on the same dataset and each contributes to the final decision of which features are relevant. The MetaGen ensemble currently employs 10 different feature selection algorithms summarized in Table 1. This table includes a summary of the fitness function used for each algorithm. A description of the individual algorithms is outside the scope of this paper. Interested readers should consult (Guyon & Elisseeff, 2003) for more details.

The ensemble combines the relevant features chosen by all the algorithms using a voting scheme. The subset evaluation algorithms vote for all the features in their subset. On the other hand, the feature evaluation algorithms vote for features based on the individual score for the feature computed by that algorithm. After all the votes are tallied, the ensemble chooses the relevant features which have votes from the majority of the algorithms. This approach allows the ensemble to leverage the strengths of multiple feature selection algorithms and insures highest percentage of relevant features is found. The ensemble is validated in the RESULTS section.

### **Association rule mining**

Association rules show feature values that co-occur frequently in a given dataset. Identifying or mining these rules allows one to gain insights to how features behave in relevance to each other. The Association Rule Mining component in the iLOG Framework uses the Tertius algorithm that is a top-down rule discovery system based on first-order logic representation (Flach & Lachiche, 2001). The main advantage of Tertius over other rule discovery systems, such as Apriori, is the confirmation function used to evaluate the rules. This function balances both novelty and support for potential rules. This function allows the use of background knowledge when computing the rules. In the iLOG framework, this background knowledge consists of the assessment score for the student (i.e., pass/fail). The use of background knowledge allows Tertius to be applied to supervised learning tasks such as concept learning. In iLOG, concept learning consists of finding features-values which, considered together, are most relevant to the assessment score. This differs from the feature selection ensemble. Tertius considers individual, feature-value pairs for separate features, while the ensemble considers all the feature-value pairs for a single feature to decide whether the entire feature is relevant. Tertius uses a top-down search algorithm when creating the association rules (Flach & Lachiche, 2001). The search first starts with an empty rule. Next, Tertius iteratively refines the rule by adding new-attribute values. Tertius continues to refine the rule as long as such refinements increase

the confirmation value. Finally, Tertius adds the rule and restarts the search to create new rules. Tertius ends when no additional rules can be created with sufficient confirmation values based on the threshold given in Flach and Lachiche (2001). Afterwards, it returns the set of rules along with their confirmation values which are used as annotated metadata in the iLOG framework.

The Tertius algorithm used in Association Rule Mining component only operates on nominal-valued features (Deltour, 2001). This is because of the implementation of the non-redundant refinement operator. As a result, the numeric features from the Data Imputation component must be converted from numeric- to nominal-valued features. The iLOG framework uses the multi-interval discretization method proposed in Fayyad (1993) that uses an information entropy minimization heuristic to convert the features. We validate the Tertius algorithm in the RESULTS section.

## RESULTS

In this section, we first provide a validation for all three MetaGen components used for automatic metadata generation. We then discuss the suitability of the iLOG metadata separately for both learners and instructors.

### Validation for MetaGen components

To demonstrate the effectiveness of Metagen, we provide a rigorous validation for all three MetaGen components: (1) Data Imputation, (2) Feature Selection Ensemble, and (3) Association Rule Mining. This validation includes analysis of results for all three components run separately on a mix of the iLOG dataset and synthetic datasets.

#### *Data imputation*

The Data Imputation component is used to fill in missing feature-value pairs for records in the iLOG dataset. The goal of validation for this component was to determine if the missing values are imputed correctly. We use the iLOG dataset to validate this component because it contains a wide variety of different features for imputation.

First, the iLOG dataset is pre-processed to remove all the records containing missing feature-value pairs so we can determine which features are imputed correctly. We refer to this version as the iLOG *complete* dataset. Next, a certain percentage of the total, remaining feature-value are selected uniformly at random and marked as *missing*. Finally, we run the Cluster-Impute algorithm to fill in all the missing feature-value pairs and compare them to the original, correct feature-value pairs. The Data Imputation accuracy is measured as the ratio of the number of missing feature-value pairs correctly imputed over the total, missing feature-value pairs. However, the chances that the exact, numeric feature value will be computed are very small. Thus, we measure whether the two feature-value pairs are approximately equal using a heuristic based on statistical methods for computing equivalency (Wellek, 2002). Based on the parameters in Wellek (2002), the imputed feature-value is considered to be correct *if it is within the acceptance interval of one standard deviation measured of the correct feature value*.

Table 2 gives the accuracy for Cluster-Impute on the iLOG complete dataset with varying amounts of total, missing-feature values. It also gives the results for the two one-sided tests for equivalence (TOST) (Wellek, 2002). Note that many methods in statistics (e.g., *t*-test, ANOVA, Kolmogorov-Smirnov, etc) are designed to show that two samples are sufficiently different by rejecting the null hypothesis that they are the same. TOST works the opposite way; it shows two samples are sufficiently equivalent by rejecting the null hypothesis that they are different. The results show that Cluster-Impute achieves high imputation accuracy

Table 2

Cluster-Impute results on iLOG dataset. The missing percentage (Miss) of feature-value pairs is given along with the imputation accuracy and the results of the TOST equivalence test

Dataset	Miss (%)	Accuracy	Hypothesis	<i>p</i> -value
iLOG Complete	5	0.7921	reject	<0.0001
iLOG Complete	10	0.8180	reject	<0.0001
iLOG Complete	15	0.8260	reject	<0.0001
iLOG Complete	20	0.7905	reject	<0.0001

even when 20% of the total feature-value pairs are missing. Note that in our tests the iLOG framework has approximately a 2% miss rate. However, we are still interested in higher values up to 20% because users may have less reliable Internet connections. Further, there is statistical significance (at *p*-value < 0.0001, epsilon 0.36) that the imputed feature-value pairs are equivalent to the correct pairs. These results indicate that the Cluster-Impute algorithm is able to correctly impute missing feature-value pairs even in datasets with a wide variety of different features.

### **Feature selection ensemble**

Results in Riley et al. (2009) show that the iLOG dataset contains *both* relevant and irrelevant features. However, we cannot be certain which features in the iLOG dataset are relevant—i.e., we do not have the “ground-truth”—because they are based on the real-world user interactions with the learning objects. Thus, our validation strategy is based on simulated datasets which resemble the iLOG dataset specifically exemplifying this property of having a mixture of relevant and irrelevant features. These synthetic datasets were created using RDG1 generator in Weka (Witten & Frank, 2005). The RDG1 generator uses a decision list to create records with feature-value pairs consistent with rules based on the labels. Interested readers should consult (Witten & Frank, 2005) for more information. For these datasets, we can specify the exact number of relevant and irrelevant features. Thus, we have the full information on which features are relevant/irrelevant allowing us to evaluate the ensemble.

Furthermore, this evaluation consists of two parts. First, we justify the need for a feature selection ensemble. This is done by showing that feature selection algorithms identify different subsets of relevant features on the same datasets. None of the algorithms are intrinsically superior because none identify all the relevant features by themselves. Thus, we could find more relevant features by combining the results using an ensemble. Second, we demonstrate that an ensemble *effectively* combines the results to identify the relevant features despite the presence of irrelevant features. This is done by comparing the number of relevant features found for individual algorithms and the ensemble.

The same 30 synthetic datasets are used for both parts of the feature selection validation. All these datasets contain 100 records with 20 total features each. However, they contain varying numbers of relevant and irrelevant features. Datasets D1-D10 contain 5 relevant and 15 irrelevant features, D11-D20 contain 10 of each, and D21-D30 contain 15 relevant and 5 irrelevant features. Thus, we also consider the effects of datasets with a greater percentage of relevant/irrelevant features on the algorithms. For part one of the validation we use the chi-square test on a contingency table to evaluate whether the relevant features selected are dependent on individual feature selection algorithms. For part two of the validation we use ANOVA contrasts to compare the performance of the individual algorithms with the ensemble.

**Validate need for ensemble.** Table 3 gives the relevant feature counts for all 10 feature selection algorithms used in the Ensemble run, independently, on the 30 synthetic datasets. It also shows the number

Table 3

The number of times each feature was identified as relevant by the feature selection algorithms and the total number of times each feature was actually relevant

FS Algorithm	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
CfsSubsetEval	1	0	1	6	5	1	5	3	1	0
ClassifierSubsetEval	6	9	6	4	15	8	9	9	8	4
ConsistencySubsetEval	1	1	2	7	7	3	5	4	3	2
CostSensitiveSubsetEval	3	1	5	3	6	1	6	4	0	0
FilteredSubsetEval	5	3	3	2	8	4	6	5	2	3
WrapperSubsetEval	8	5	5	7	16	7	10	6	7	5
ChiSquaredAttributeEval	1	0	1	1	5	1	5	3	1	0
ReliefFAttributeEval	8	4	7	5	13	8	10	9	6	6
SymmetricalUncertAttributeEval	9	4	4	3	15	9	10	9	6	6
CostSensitiveAttributeEval	6	7	7	5	9	4	8	5	7	4
<b>TIMES RELEVANT</b>	<b>16</b>	<b>14</b>	<b>14</b>	<b>10</b>	<b>23</b>	<b>13</b>	<b>15</b>	<b>12</b>	<b>15</b>	<b>12</b>
	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
CfsSubsetEval	1	2	2	1	2	4	2	1	0	0
ClassifierSubsetEval	6	10	6	8	5	11	9	6	15	7
ConsistencySubsetEval	2	1	3	5	2	6	3	5	1	6
CostSensitiveSubsetEval	3	1	2	2	2	4	2	1	2	1
FilteredSubsetEval	4	4	4	4	3	4	5	1	3	3
WrapperSubsetEval	7	8	3	10	6	8	4	2	9	5
ChiSquaredAttributeEval	1	2	2	1	2	5	2	1	0	0
ReliefFAttributeEval	7	7	5	12	4	13	10	2	7	8
SymmetricalUncertAttributeEval	9	8	8	12	3	12	10	6	12	7
CostSensitiveAttributeEval	5	6	4	9	5	8	6	4	8	5
<b>TIMES RELEVANT</b>	<b>14</b>	<b>16</b>	<b>12</b>	<b>19</b>	<b>12</b>	<b>19</b>	<b>17</b>	<b>12</b>	<b>18</b>	<b>17</b>

of times each feature was relevant over all the datasets. Overall, there is considerable variation in the counts for the number of relevant features found. In fact, a chi-square test on the resulting contingency table provides evidence (with  $p < 0.0001$ ) that the features selected are not independent of the feature selection algorithms. Recall that feature selection algorithms use different fitness functions and specialize on finding different kinds of features. Some of the algorithms are more conservative (e.g., CfsSubsetEval) and some are more aggressive (e.g., SymmetricalUncertFeatureEval). However, *none of the algorithms are able to identify all the relevant attributes in the synthetic datasets*. Further, the feature selection algorithms each choose different subsets of relevant features. Taken together, this provides motivation for using an ensemble approach for feature selection which can combine the different subsets to identify as many relevant features as possible and more accurately.

**Validate ensemble results.** Table 4 gives the number of relevant features found by the individual algorithms and the ensemble on each of the 30 synthetic datasets. From the results, we observe that the ensemble finds many more relevant features than the individual algorithms. In fact, ANOVA contrasts comparing (1) the ensemble with the subset evaluation algorithms, (2) with the feature evaluation algorithms, (3) and with all the algorithms together provides evidence (with  $p < 0.0001$ ) that the ensemble achieves superior results in terms of identifying the correct features. This demonstrates that the ensemble is capable of merging the results from the individual algorithms to identify a larger subset of relevant features. Further, the

number of relevant features found on most synthetic datasets is very close the actual number: 5 on datasets D1-D10, 10 on D11-D20, and 15 on D21-D30. The varying number of relevant and irrelevant features has little impact on the ensemble because it utilizes both conservative and aggressive feature selection algorithms. However, we observe that the ensemble also misidentify a greater number of irrelevant features as relevant. In the current design of the iLOG framework, these irrelevant features would be eventually filtered out by the rule mining process (as rules with irrelevant features will likely lead to low coverage and confidence). Nevertheless, we realize the need to balance the *effectiveness* and *efficiency* of our ensemble algorithm and will address this issue in our future work.

## Validation for metadata results

Here we demonstrate the suitability of the annotated metadata created using the MetaGen framework. First, we give examples of actual metadata generated from the iLOG dataset (i.e., association rules) for all three of the groups alluded to earlier: learners and instructors, LO designers, and researchers. Second, we give a general discussion of the annotated metadata particularly from an instructional support perspective and a student pedagogical perspective.

Overall, this metadata is the result of all three MetaGen components evaluated on the iLOG dataset from the iLOG deployment (Riley, et al., 2009). This dataset contains records with user interactions and static user and LO data from four introductory computer sciences courses using five separate learning objects. In total there are over 1400 distinct student/LO sessions. Each session contains 104 features comprising 55 user interaction-based features, 45 static user data features, and 4 static LO data features.

There are three different groups who can benefit from annotated metadata (learners and instructors, LO designers, and researchers). These separate groups have very different needs for metadata (Ravasio, 2003; Kosba et al., 2007) so we have collaborated with educational experts to validate the usefulness of metadata for all groups. Table 5 contains examples of the actual metadata generated from the iLOG dataset for the different courses along with the confidence (Conf.) provided by Tertius (Flach & Lachiche, 2001).

First, metadata for the “learner & instructor” group generally involves user interactions with the assessment—unsurprising, but useful for validating that iLOG metadata is consistent with expectations. More interesting to learners and instructors are the connections between the assessment and user interactions on other LO components or static user data. Specifically, users who spend more time on each tutorial page and a reasonable amount of time on the assessment tend to pass the LO whereas users who are not required to take the course (for their major) and spend a considerable amount of time on the assessment tend to fail the LO. With such metadata, both learners and instructors can make pedagogical changes to promote better learning outcomes for the course. For example, learners could spend more time on the tutorials whereas instructors could provide additional scaffolding for students who are not required to take the course.

Second, metadata for the “LO design” group focuses more on the static user data. Recall that such static user data are on a Likert scale (1 = strongly disagree, 5 = strongly agree). As expected, students who give low ratings to the LOs or strongly prefer the professor tend to also do worse on the LOs. More interesting to content developers are the connections between static user data. Specifically, students with sufficient math background tend to pass regardless of whether the LO was easy to use whereas students who gave the LOs a low overall rating tended to fail. Such metadata demonstrates what static user data affects the learning outcome for the LO (i.e., pass/fail on the assessment) and to some extent serves as an outcome predictor. Content developers can then emphasize particular static user data when making refinements to the LOs. For example, the developer could increase the difficulty of mathematical expressions in the LOs or investigate specific complaints from the users which may have contributed to the poor learning outcome.



Table 4

Relevant features found on 30 synthetic datasets. Datasets D1-D10 have 5 relevant features, D11-D20 have 10 relevant features, and D21-D30 have 15 relevant features. The results show that some algorithms are more aggressive than others

FS Algorithm	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
CfsSubsetEval	3	1	2	1	2	1	1	2	1	1
ClassifierSubsetEval	4	3	3	5	3	3	4	5	2	5
ConsistencySubsetEval	3	1	3	3	3	3	0	3	2	1
CostSensitiveSubsetEval	2	2	2	2	1	1	2	2	1	1
FilteredSubsetEval	2	2	1	3	2	1	4	4	1	1
WrapperSubsetEval	3	3	2	4	3	4	5	5	2	3
ChiSquaredAttributeEval	3	1	2	1	1	1	1	2	1	2
ReliefFAttributeEval	4	3	4	4	4	4	4	5	3	2
CostSensitiveAttributeEval	4	4	2	3	3	4	3	5	2	3
OneRAttributeEval	3	4	3	5	3	2	2	2	1	2
<b>AVERAGE</b>	<b>3.1</b>	<b>2.4</b>	<b>2.4</b>	<b>3.1</b>	<b>2.5</b>	<b>2.4</b>	<b>2.6</b>	<b>3.5</b>	<b>1.6</b>	<b>2.1</b>
ENSEMBLE	5	5	5	5	4	5	5	5	4	5
	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
CfsSubsetEval	0	0	3	2	0	1	1	1	2	0
ClassifierSubsetEval	5	0	7	6	7	7	8	2	5	1
ConsistencySubsetEval	0	0	3	6	0	4	1	1	3	0
CostSensitiveSubsetEval	2	0	2	2	2	2	2	1	2	0
FilteredSubsetEval	0	0	2	4	1	2	1	2	6	2
WrapperSubsetEval	3	0	5	5	3	3	3	0	9	1
ChiSquaredAttributeEval	0	0	3	2	0	1	0	0	2	0
ReliefFAttributeEval	5	4	3	6	4	5	5	3	5	3
CostSensitiveAttributeEval	8	5	7	7	7	4	6	4	8	3
OneRAttributeEval	3	0	5	5	4	7	3	0	5	1
<b>AVERAGE</b>	<b>2.6</b>	<b>0.9</b>	<b>4</b>	<b>4.5</b>	<b>2.8</b>	<b>3.6</b>	<b>3</b>	<b>1.4</b>	<b>4.7</b>	<b>1.1</b>
ENSEMBLE	10	7	9	10	10	9	9	6	9	4
	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30
CfsSubsetEval	1	2	1	2	1	1	1	1	2	1
ClassifierSubsetEval	8	7	9	7	10	2	10	7	8	8
ConsistencySubsetEval	2	6	3	1	0	1	1	5	6	4
CostSensitiveSubsetEval	3	2	2	3	2	1	0	1	2	2
FilteredSubsetEval	3	4	3	4	5	1	5	3	5	2
WrapperSubsetEval	8	9	5	8	7	1	10	6	10	8
ChiSquaredAttributeEval	1	2	1	2	1	0	0	1	2	1
ReliefFAttributeEval	7	8	7	6	8	7	8	7	6	7
CostSensitiveAttributeEval	6	7	8	7	7	7	8	7	8	5
OneRAttributeEval	8	6	8	8	3	0	4	8	6	11
<b>AVERAGE</b>	<b>4.7</b>	<b>5.3</b>	<b>4.7</b>	<b>4.8</b>	<b>4.4</b>	<b>2.1</b>	<b>4.7</b>	<b>4.6</b>	<b>5.5</b>	<b>4.9</b>
ENSEMBLE	13	14	14	15	14	9	15	14	14	15

Table 5  
Association rules for the iLOG dataset

Group	Course	Conf.	Association rule
Learner & instructor	101	0.33	LoEasyToUse > 1 and assessmentTotalSeconds = [2,164] → assessmentPassFail = <b>pass</b>
		0.32	requiredCourse = No and assessmentTotalSeconds > 164 → assessmentPassFail = <b>fail</b>
	150	0.63	exerciseTotalClicks < 2 and assessmentTotalSeconds = [2,273] → assessmentPassFail = <b>pass</b>
		0.62	assessmentAvgClicksPerPage < 2.5 → assessmentPassFail = <b>fail</b>
	155	0.51	tutorialMinTimePerPage > 4 and assessmentTotalSeconds = [260,281] → assessmentPassFail = <b>pass</b>
		0.49	assessmentAvgTimePerPage < 3.4 → assessmentPassFail = <b>fail</b>
	183	0.46	OverallRatingOfLO > 2 and tutorialTotalClicks ≥ 2 → assessmentPassFail = <b>pass</b>
	0.42	tutorialMinClicksPerPage > 1 and assessmentMaxClicksPerPage < 3 → assessmentPassFail = <b>fail</b>	
LO Design	101	0.32	highestMath > Calculus I and LoEasyToUse ≥ 1 → assessmentPassFail = <b>pass</b>
		0.29	currentSEConfidenceInCSAbilityNum ≤ 1 → assessmentPassFail = <b>fail</b>
	150	0.67	LOValuableAdditionToCourse > 2 and expectToTakeMoreCSCourses = 4 → assessmentPassFail = <b>pass</b>
		0.64	LOHelpedMeUnderstandTopic ≤ 3 and currentSEConfToPerformWellInClass ≤ 3 → assessmentPassFail = <b>fail</b>
	155	0.47	requiredCourse = Yes and currentSEConfidenceInCSAbility > 1 → assessmentPassFail = <b>pass</b>
		0.48	OverallRatingOfLO < 2 → assessmentPassFail = <b>fail</b>
	183	0.45	LOHelpedMeUnderstandTopic > 1 and tutorialTotalClicks ≥ 2 → assessmentPassFail = <b>pass</b>
		0.41	LOMoreInterestingThanProf ≤ 1 → assessmentPassFail = <b>fail</b>
Researcher	101	0.31	currentSEConfidenceInCSAbility > 1 and assessmentAvgTimePerPage = [2,15] → assessmentPassFail = <b>pass</b>
		0.30	currentSEConfidenceInCSAbility < 1 → assessmentPassFail = <b>fail</b>
	150	0.66	currentSEConfToPerformWellInClass > 3 and exerciseTotalClicks < 2 → assessmentPassFail = <b>pass</b>
		0.63	LoEasyToUse ≤ 3 and currentAvgMSE ≤ 3.3 → assessmentPassFail = <b>fail</b>
	155	0.50	requiredCourse = No and currentAvgSE ≥ 1.4 → assessmentPassFail = <b>pass</b>
		0.45	Difficulty ≥ 2.1 and OverallRatingOfLO < 1.5 → assessmentPassFail = <b>fail</b>
	183	0.50	OverallRatingOfLO ≥ 1.5 and currentAvgMSE ≥ 2.8 → assessmentPassFail = <b>pass</b>
		0.43	currentAvgMSE < 2.8 → assessmentPassFail = <b>fail</b>

Finally, metadata for the “researcher” group focuses more on student learning. As expected, students with low motivation and self-efficacy tend to do worse on the LOs. More interesting to researchers are the connections between self-reported student learning and the user interaction data. Specifically, students confident in CS abilities do not need the exercises to pass whereas students who feel the LOs are harder to use and have lower motivation and self-efficacy tend to fail the LOs. Researchers can use such metadata as empirical support for learning research in affective factors for online LOs. As alluded to earlier, this metadata provides initial exploration to support subsequent, in-depth analysis.

In order to provide a validation of the appropriateness of the metadata for use by learners and instructors, results were reviewed by a faculty member from the College of Education. This education expert focused on association rules with the highest confidence. In general, it was found that variables traditionally associated with higher learning were represented in the association rules. In particular, time spent on various sections of the LO, including the assessment, was predictive of pass/fail on the LO assessment. The level of interactivity, as represented by the number of clicks on sections of the LO, was also predictive of learning. This result clearly supports the value of active learning, which is a well researched instructional strategy (Astrachen, et al., 2002; Nugent, et al., 2009). Students evaluative rating of the LO, as determined by a Likert scale, was also a key variable, and supports research showing the relationship between student attitudes and achievement (Alsop & Watts, 2003; Koballa & Glynn, 2007). Students' self-efficacy, as represented by perceptions of their confidence in their computer science knowledge and attitudes, sense of academic preparation for the particular computer science course, and grade expectation, was reflected in the association rules. Another attitudinal variable represented was student motivation, which tapped their motivation to learn more about computer science, their interest in the content area of the course, and their expectation to take more computer science courses. In summary, the association rules predicting students' score on the LO assessment encapsulated key variables which research has shown to be predictive of student learning including motivation and self-efficacy.

These results also support earlier research using an educational statistics regression approach to identify variables which predicted student learning on the LO assessment. Combining data across course and LOs, it was found that there were differences in the LOs in terms of student learning and that more time spent on the LO and the use of active learning strategies contributed to greater learning (Nugent, et al., 2009).

## CONCLUSIONS

The traditional classroom approach involving the textbook and lectures has several significant problems motivating the use of online education programs. Learning objects (LO) are small, self-contained lessons which are often used in such programs. LOs are commonly stored in searchable repositories to facilitate reuse. Course developers search a repository for suitable LOs based on the LO metadata. Unfortunately, based on the current standards, such metadata is often missing or incorrectly entered making searching difficult or impossible. In this paper, we investigate automating metadata generation and annotation for SCORM-compliant LOs based on user interactions with the LO and static information about LOs and users. We present the Intelligent Learning Object Guide (iLOG) framework which implements the ecological approach for metadata based on user interactions. We discuss how the proposed framework addresses problems with the current metadata standard. This framework consists of two components. First, the LO Wrapper logs user interactions with the LO to an external database and updates the LOs with new metadata. Second, the MetaGen system generates metadata automatically based on user interactions and static information. To accomplish this, MetaGen extracts and analyzes a dataset using three separate components. The data imputation component is used to fill in any missing feature-value pairs in the dataset. This component uses Cluster-Impute, a novel algorithm presented here which employs hierarchical clustering, dynamic tree cuts and linear regression to fill in missing feature-value pairs based similar, known pairs. Next, MetaGen employs a feature selection ensemble component to select the subset of features most relevant to the learning goal (e.g., helping students pass the assessment). Finally, MetaGen uses the association rule miner component to create rules based on only the relevant features in the dataset. Such metadata is annotated using XML format and written automatically to the corresponding LO's zipped file. We also provide a rigorous validation for all three components and for metadata generated from real-world datasets using MetaGen. The MetaGen components are validated using a mix of real-world and synthetic datasets. We

include the results for a study where over 1400 sessions were collected from four introductory computer science courses using five separate learning objects. Each session included 104 features comprising 55 user interaction-based features, 45 static user data features, and 4 static LO data features. From this dataset, iLOG has generated a considerable amount of metadata annotated to the LOs. We have also provided a discussion of how this metadata can be used by different groups including learner and instructors, LO designers, and researchers. This discussion included empirical support and expert analysis for our pedagogical claims. In particular, the iLOG framework shows how the ecological approach theory for metadata can address problems with the current metadata standard.

In the future, we intend to evaluate the iLOG framework on additional LOs deployed to larger group of students. This expanded deployment should allow iLOG to generate even more interesting metadata and should provide information on how learners and instructors utilize existing metadata. We would also like to branch out into different LO content areas. The current sets of LOs are designed based on introductory computer science topics. We would like to compare metadata from user interactions on these topics to metadata generated for LOs on different topics. Regarding the MetaGen system, the feature selection ensemble currently emphasizes only the relevant features. It finds all relevant features, but also some irrelevant features. We need to investigate how to reduce the number of irrelevant features found without adversely affecting the ensemble. Finally, our educational expert recommended that format of the metadata output be further simplified for greater usability by learners and instructors. While detailed metadata will be of interest to content developers and researchers, learners and instructors want clear and simplified information about what types of students will best benefit from the LOs and how the LO can most profitably be used.

## ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0632642 and an NSF GAANN fellowship. The authors would like to thank other members of the iLOG team for their programming and data processing work: Sarah Riley, Erica Lam, WayLoon Tan, Beth Neilsen, and Nate Stender.

## REFERENCES

- ADL, Advanced Distributed Learning. (2003). *DoD affirms SCORM's role in training transformation*. Retrieved July 18, 2005, from [www.adlnet.org](http://www.adlnet.org).
- Alsop, S., & Watts, M. (2003). Science education and affect. *International Journal of Science Education*, 25, 1043-1047.
- Astrachan, O., Duvall, R., Forbes, J., & Rodger, S. (2002). Active learning in small to large courses. ASEE/IEEE Frontiers in Education Conference.
- Bailey, C., Zalfan, M.T., & Davis, H.C., Fill, K., & Conole, G. (2006) Panning for gold: Designing pedagogically-inspired learning nuggets. *Educational Technology and Society*, 9(1), 113-122.
- Minaei-Bidgoli, B., Kortemeyer, G., & Punch, W. (2004). Mining feature importance: Applying evolutionary algorithms within a web-based educational system. *CITSA*, 1-6.
- Broisin, J., Vidal, P., Meire, M., & Duval, E. (2005). Bridging the gap between learning management systems and learning object repositories: Exploiting learning context information. *AICT/SAPIR/ELETE'05*, 478-483.
- Brooks, C., McCalla, G., & Winter, M. (2005). Flexible learning object metadata. *Applications of Semantic Web technologies for E-Learning (SW-EL)*, 1-8.

- Cardinaels, K., Duval, E., & Oliivié, H. (2006). A formal model of learning object metadata. *Innovative Approaches for Learning and Knowledge Sharing*, 74-87.
- Cardinaels, K., Meire, M., & Duval, E. (2005). Automating metadata generation: The simple indexing interface. *WWW*, 548-556.
- Castro, F., Vellido, A., Nebot, À., & Mugica, F. (2007). Applying data mining techniques to e-learning problems. *Evolution of Teaching and Learning Paradigms in Intelligent Environment*, 62, 183-221.
- Cechinel, C., Sánchez-Alonso, S., & Sicilia, M.Á. (2009). Empirical analysis of errors on human-generated learning objects metadata. *Metadata and Semantic Research*, 46(1), 60-70.
- Deltour, A. (2001). Tertiary extension to weka. Technical Report (CSTR-01-001). Department of Computer Science, University of Bristol.
- Dodds, P. (2001). The SCORM run-time environment. *Advanced Distributed Learning*, 1-85.
- Etchells, T., Nebot, A., Vellido, A., Lisboa, P., & Mugica, F. (2006). Learning what is important: Feature selection and rule extraction in a virtual course. *European Symposium on Artificial Neural Networks*, 401-406.
- Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *IJCAI*, 1022-1029.
- Flach, P.A., & Lachiche, N. (2001). Confirmation-guided discovery of first-order rules with Tertius. *Machine Learning*, 42(1), 61-95.
- Friesen, N. (2004). The international learning object metadata survey. Technical Evaluation Report. Athabasca University.
- García, E., Romero, C., Ventura, S., & Castro, C. (2009a). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Modeling and User-Adapted Interaction*, 19(1), 99-132.
- García, E., Romero, C., Ventura, S., & Castro, C. (2009b). Evaluating web based instructional models using association rule mining. UMAP'09.
- Gasevic, D., Jovanovic, J., & Boskovic, M. (2005). Ontologies for reusing learning object content. *ICALT*, 944-945.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *JMLR*, 3, 1157-1182.
- Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of Data Mining*. Cambridge, Massachusetts: The MIT Press.
- Jovanovic, J., Gasevic, D., & Devedzic, V. (2006). Ontology-based automatic annotation of learning content. *International Journal on Semantic Web & Information Systems*, 2(2), 91-119.
- Johnson, S. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3), 241-254.
- Karampiperis, P., & Sampson, D. (2004). Adaptive learning object selection in intelligent learning systems. *Journal of Interactive Learning Research*, 15, 389-407.
- Koballa, T., & Glynn, S. (2007). Attitudinal and motivational constructs in science learning. In S. Abell, & N. Lederman (Eds.), *Handbook of Research on Science Education* (pp. 75-102). Mahwah, New Jersey: Erlbaum.
- Kosba, E., Dimitrova, V., & Boyle, R. (2007). Adaptive feedback generation to support teachers in web-based distance education. *User Modeling and User-Adapted Interaction*, 17(4), 379-413.
- Langfelder, P., Zhang, B., & Horvath, S. (2008). Defining clusters from a hierarchical cluster tree: The dynamic tree cut package for R. *Bioinformatics*, 24(5), 719-720.
- LTSC. (2002). Draft standard for learning object metadata. IEEE LTSC Review Committee.
- Liu, J. (2009). *Individualized Selection of Learning Objects*. Saskatoon, Canada: Master of Science, University of Saskatchewan.
- McCalla, G. (2004). The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education*, 7, 1-23.
- Miller, L.D., Soh, L., Riley, S., & Samal, A. (2008). *Motivation for a Direct Access Tunnel between Content Developers and Learning Objects*, White Paper, Learning Education Training Systems Interoperability.
- Miller, L.D., Stender, N., Soh, L-K., & Kupzyk, K. (2011). Hierarchical clustering algorithm with dynamic tree cut for data imputation. Technical Report (TR-UNL-CSE-2011-0003). Department of Computer Science and Engineering, University of Nebraska, Lincoln.

- Minaei-Bidgoli, B., Tan, P.-N., & Punch, W. (2004). Mining interesting contrast rules for a web-based educational system. In *Machine Learning and Applications*, 320-327.
- Nash, S. (2005). Learning objects, learning object repositories, and learning theory: Preliminary best practices for online course. *Interdisciplinary Journal of Knowledge and Learning Objects*, 1, 217-228.
- Nugent, G., Kupzyk, K., Riley, S., Miller, L.D., Hostetler, J., Soh, L.-K., & Samal, A. (2009). Empirical usage metadata in learning objects. *FIE*, 1285-1290.
- Ochoa, X. (2006). Quality metrics for learning object metadata. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2006(1), 1004-1011.
- Pansanato, L., & Fortes, R. (2005). Strategies for automatic LOM metadata generating in a web-based CSCL tool. *WebMedia'05*, 1-8.
- Polsani, P. (2003). Use and abuse of reusable learning objects. *Journal of Digital Information*, 3(4), 1-10.
- Ravasio, P., Schluep, S., & Schar, G. (2003). Metadata in e-Learning systems: Focus on its value for the user. *International Conference on Multimedia and ICTs in Education*, 1-5.
- Riley, S., Miller, L.D., Soh, L.-K., Samal, A., & Nugent, G. (2009). Intelligent learning object guide (iLOG): A framework for automatic empirically-based metadata generation. *AIED*, 515-522.
- Roy, D., Sarkar, S., & Ghose, S. (2008). Automatic extraction of pedagogic metadata from learning content. *International Journal of Artificial Intelligence in Education*, 18(2), 97-118.
- Saini, P., Ronchetti, M., & Sona, D. (2006). Automatic generation of metadata for learning objects. *ICALT*, 275-279.
- Segura, A., Vidal, C., Menendez, V., Zapata, A., & Prieto, M. (2009). Exploring characterizations of learning object repositories using data mining techniques. *Metadata and Semantic Research*, 46(2), 215-225.
- Thompson, K., & Yonekura, F. (2005). Practical guidelines for learning object granularity from one higher education setting. *Interdisciplinary Journal of Knowledge and Learning Objects*, 1, 163-179.
- Tompsett, C. (2005). Reconfigurability: Creating new courses from existing learning objects will always be difficult! *Journal of Computer Assisted Learning*, 21(6), 440-448.
- Vargo, J., Nesbit, J.C., Belfer, K., & Archambault, A. (2003). Learning object evaluation: Computer-mediated collaboration and inter-rater reliability. *International Journal of Computers and Applications*, 25, 1-8.
- Wellek, S. (2002). *Testing Statistical Hypotheses of Equivalence* (1st ed.). Chapman & Hall/CRC.
- Witten, I., & Frank, E. (2005). Data mining: Practical machine learning tools and techniques. *Morgan Kaufmann*.
- Wolpers, M., Najjar, J., Verbert, K., & Duval, E. (2007). Tracking actual usage: The attention metadata approach. *Educational Technology & Society*, 10(3), 106-121.
- Zouaq, A., Nkambou, R., & Frasson, C. (2007a). Using a competence model to aggregate learning knowledge objects. *ICALT*, 836-840.
- Zouaq, A., Nkambou, R., & Frasson, C. (2007b). An integrated approach for automatic aggregation of learning knowledge objects. *Interdisciplinary Journal of Knowledge and Learning Objects*, 3, 135-162.