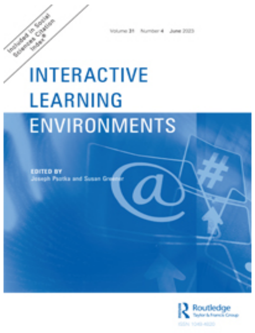


Interactive Learning Environments

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/nile20

A patterns based approach for the design of educational technologies

Sridhar Chimalakonda & Kesav V. Nori

To cite this article: Sridhar Chimalakonda & Kesav V. Nori (2023) A patterns based approach for the design of educational technologies, Interactive Learning Environments, 31:4, 2114-2133, DOI: 10.1080/10494820.2021.1875000

To link to this article: https://doi.org/10.1080/10494820.2021.1875000

INTERACTIVE LEARNING ENVIRONMENTS
2023, VOL. 31, NO. 4, 2114–2133
https://doi.org/10.1080/10494820.2021.1875000

A patterns based approach for the design of educational technologies*

Sridhar Chimalakonda [a] and Kesav V. Nori[b]

[a]Department of Computer Science & Engineering, Indian Institute of Technology, Tirupati, India; [b]Software Engineering Research Center, International Institute of Information Technology Hyderabad, Gachibowli, India

ABSTRACT

Despite rapid advances, modeling a variety of instructional designs to support variations in teaching and learning during the design of educational technologies is still an open challenge. In this paper, we propose a patterns based approach for the design of educational technologies to address this challenge. This is in contrast with existing literature that focuses either on patterns in education or in software, and not both. The core idea of our approach is to leverage patterns for modeling instructional design knowledge and to connect it with patterns in software architecture. We discuss different categories of patterns in instructional design. We then present the notion of *Pattern-Oriented Instructional Design* (POID) as a way to model instructional design as a connection of patterns (*ProcessPattern*, *ContentPattern*) and integrate it with *Pattern-Oriented Software Architecture* (POSA) based on fundamental principles in software engineering. We demonstrate our approach through adult literacy case study (287 million learners, 22 Indian Languages and a variety of instructional designs). The results of our approach (both web and mobile versions) are available at *http://rice.iiit.ac.in* and were transferred to *National Literacy Mission Authority* of *Government of India* for further proliferation.

ARTICLE HISTORY
Received 2 April 2020
Accepted 18 December 2020

KEYWORDS
Adult literacy; educational technologies; instructional design; modeling; patterns; software engineering

1. Background & introduction

Instructional Design[1] plays a fundamental role in the field of technology enhanced learning as an underlying and complex discipline often involving multiple perspectives and connotations (Reigeluth, 2013a, 2013b; Reigeluth & Carr-Chelman, 2009). Berger et al. defined instructional design as a "systematic development of instructional specifications using learning and instructional theory to ensure the quality of instruction" (Berger & Kam, 1996). Merrill has defined instructional design as the practice of creating "instructional experiences which make the acquisition of knowledge and skill more efficient, effective, and appealing" (Merrill, 2012). In essence, the need to have an instructional design basis for educational technologies has been emphasized in the literature by several researchers (Duffy & Jonassen, 2013; Garzotto et al., 2009; Goodyear et al., 2004; Laurillard, 2013a, 2013b; Reid, 2014).

On the other hand, the need to support different aspects of teaching and learning in multiple contexts has led to a variety of instructional design models (Brown & Green, 2015; Reigeluth, 2013a, 2013b; Reigeluth & Carr-Chelman, 2009). This is primarily because there is no "one-size-fits-

CONTACT Sridhar Chimalakonda ✉ ch@iittp.ac.in

*A part of this work was carried out during first author's doctoral thesis at IIIT hyderabad, India (Chimalakonda, 2017) and a preliminary version of the work during its initial days is published as a 3-page short paper in ICALT 2014 (Chimalakonda & Nori, 2014).

all" solution for all instructional design needs. Gustafson et al. present a continuum of instructional strategies from instructor-centered to student-centered (Gustafson & Branch, 1997). The ADDIE model involving *analysis*, *design*, *development*, *implementation* and *evaluation* phases is one of the most commonly used instructional design model that is applicable in several contexts (Kruse, 2002). Gagne's series of nine learning events (Gagne & Briggs, 1974) has laid foundation for several instructional design models such as Dick's instructional model (Dick et al., 2001) and Merrill's first principles of instruction (Merrill, 2012). Millwood summarizes over 25 learning theories in a concept map connecting the different facets of instructional design (Millwood, 2014). Gibbons (2013)'s work discusses eight views of instructional design. Mizoguchi et al. has done a comprehensive survey of learning theories from instructional technology perspective and modeled them using ontologies (Mizoguchi et al., 2007).

In addition, the changing landscape of educational technologies also requires supporting variations in instructional designs for different stakeholders (Reigeluth, 2013b) presenting a strong need for *systematic modeling of instructional design*. In the literature, there are two ways of modeling instructional designs during design of educational technologies (i) Educational modeling languages (EMLs) and (ii) *patterns*.

Educational modeling languages (EMLs) such as poEML (Caeiro et al., 2014) and Web COLLAGE (Villasclaras-FernáNdez et al., 2013) have emerged as a way to model different aspects of instructional design along with many other approaches (Botturi et al., 2008; Martínez-Ortiz et al., 2007). A hierarchical framework for integrating open educational resources at different levels of granularity was presented in Sampson and Zervas (2014). IMS-Learning Design emerged as a standard for learning design (Consortium, I. G. L., 2003) and then focus shifted to tools such as LAMS (Dalziel, 2003) and LDSE (Laurillard et al., 2013) that aim to support teachers. Despite this rapid progress, many researchers have pointed to several shortcomings of modeling and reusing instructional design such as complexity of authoring, lack of adequate tool support, interoperability (Larson & Lockee, 2019; Neumann et al., 2009). Our analysis of the state-of-the-art in learning design also revealed that the goal of end-to-end automation (from pedagogy to technology) without support for variations and tools has resulted in too generic, too complex specifications leading to slow progress and is in line with the literature (Bodily et al., 2019; Gropper, 2017; Laurillard, 2013a). Based on the variations that are possible for instructional designs, an open research challenge is *to facilitate the design of educational technologies with support for variations in instructional designs*.

In a different line of research for modeling instructional design, researchers have proposed patterns as a mechanism to capture best practices of teaching and learning (Bergin et al., 2012; Borchers, 2001). The core idea of *patterns* and *pattern languages* is the encapsulation, modeling and delivery of expert's knowledge and best practices to novices in a discipline. *Patterns are primarily concerned with the idea of finding recurring solutions to recurring problems in a certain context* (Alexander et al., 1977). Essentially, patterns are derived from experiences and provide abstract representations at different levels of granularity expressed in different ways by different experts in different communities (Meszaros & Doble, 1998).

The Pedagogical Patterns Project provides a large collection of patterns focusing on pedagogy and does not explicitly consider the use of technology and software patterns (Bergin et al., 2012). The E-LEN project is another initiative aimed at providing pedagogically-informed technology and experiences as pattern languages for new institutions mainly focusing on learning management systems (Avgeriou et al., 2003). Patterns for person centered e-learning were proposed in Derntl (2005) based on action research but are not designed for scale and variety. Pattern languages for different aspects of educational design (Goodyear & Retalis, 2010) also ignored the notion of Technology Enhanced Learning (TEL). Patterns for person centered e-learning were proposed in Derntl (2005) based on action research but are not designed for scale and variety. Pattern languages for different aspects of educational design (Goodyear & Retalis, 2010) also ignored the notion of TEL. Laurillard (2012) has created pedagogical patterns using a design science approach. Another direction was to mine patterns derived from practitioner workshops and document them for use (Mor, 2014). Despite these directions, the focus of existing work is on mining and documenting patterns from pedagogy perceptive largely ignoring technology aspects. Even in works that considered technology aspects (Derntl & Motschnig-Pitrik, 2004), there is a huge gap between domain (instructional design) and technology patterns.

In our analysis of literature, we have observed that patterns and pattern languages are still not widely used either in instructional design or in TEL because of several reasons including:

- Existing patterns are mainly driven from a pedagogical perspective rather than their *structure* and have minimal support for design of educational technologies (Bergin et al., 2012; Borchers, 2001; Mor, 2014; Patterns, 2014).
- Existing approaches focus on patterns and pattern languages either for *communication* or *engineering* purposes and not both (Chimalakonda, 2017).
- Lack of bridge between instructional design patterns and software patterns (Chimalakonda, 2017).
- Patterns are not used to facilitate reuse and reduce effort during design of educational technologies (Chimalakonda & Nori, 2020).

With this background, the research goal of this paper is to leverage the potential of domain (instructional design) and technology patterns for modeling a scale and variety of instructional design variations during design of educational technologies and demonstrate the variations for the adult literacy case study. Our work fundamentally deviates from the state-of-the-art as we focus on not one instructional design but on a family of similar but distinct instructional designs and also connect the domain and software patterns.

In the next section, we discuss the case study that is used for evaluation.

2. Methodology & case study

To achieve the research goals set out in this paper and to evaluate the proposed approach, we use the standard qualitative case study method (Creswell & Poth, 2017). In this section, we present the adult literacy case study used for evaluation.

The research work presented in this paper has stemmed from one of the hard challenges in education and specifically adult literacy in India. As per the report from UNESCO (UNESCO, 2014), there are around 287 million Indians, who are beyond the age of schooling, speak their language, but cannot read or write and these people are spread across 22 official Indian Languages. The National Literacy Mission Authority of India (NLMA)[2] of Government of India has devised several initiatives to address this challenge in the past several decades. The use of ICTs such as radio forums and satellite televisions for adult literacy dates back to as early as 1976 (Dighe, 2010; Farrell, 2004; Patel, 2002). Computer and laptop based solutions for literacy were experimented by Tata Consultancy Services (TCS) (TCS, 2020) and TaraAkshar (Akshar+, 2020). TCS, an Indian Software Consultancy Services Company has been contributing to adult literacy since 2001 (TCS, 2020). They have developed *e*Learning Systems for 9 Indian Languages and also for languages like *Urdu*, *Moore*, *Spanish* and *Arabic* (TCS, 2020). An approach based on same language subtitling for songs delivered on television increased motivation of learners (Kothari, 2008) but focused only on reading skills and is not interactive. A landscape research review of mobiles for reading outlines the need for extensive further research to validate the effectiveness of mobile technologies (Wagner et al., 2014).

For the last 12 years, Chimalakonda et al. have proposed several technological aids to support adult literacy in India. Chimalakonda and Nori (2012) proposed a pattern for modeling instructional material however, they did not consider the idea of pattern variations and did not have an implementation. An ontology based framework for modeling instruction design is discussed in Chimalakonda and Nori (2013c). *EasyAuthor* was proposed as an authoring tool for modeling instructional material based on ontologies (Chimalakonda & Nori, 2013b). A 2-page summary of the interdisciplinary approach for design of educational technologies for adult literacy is presented in Chimalakonda and Nori (2013a). The application of software product lines to accelerate the development of eLearning Systems in adult literacy is presented in Chimalakonda (2017) and Chimalakonda and Nori (2020). In their preliminary 3-page short paper, Chimalakonda and Nori (2014) have proposed the idea of integrating patterns in instructional design and software to facilitate development of educational technologies. However, the paper did not explain patterns either in domain or in software. In addition, the proposed approach did not have an implementation or evaluation.

Teaching adults requires a different pedagogy (Merriam, 2001). Merriam defines an adult learner as someone who has (i) an independent view of learning (ii) extensive prior experience (iii) dynamic learning goals (iv) focuses on application of knowledge for practical purposes and (v) gets motivated for internal factors (Merriam, 2001). NLMA has researched for decades and proposed Improved Pace and Content of Learning (IPCL) as a uniform methodology for teaching 3*R*s -- *R*eading, w*R*iting and a*R*ithmetic to adult illiterates in India (DAE, 2003).

The NLMA as domain expert for adult literacy in India mandates that any technology-based solution for adult literacy across all 22 official Indian languages must use the *uniform methodology* as the basis and customize the instructional design including process, content, learner styles and other aspects as per varied local needs of languages and regions across India. Based on this mandate, the instructional material (called as *primer* and printed as a physical textbook) based on IPCL for different languages is subsequently produced by 32 State Resource Centers (SRCs)[3] across India. Each of the primer is divided into three parts and each part is divided into approximately 9 to 12 lessons and each lesson has content focusing on 3Rs. It is estimated that there are around 1000 primer variations produced by 32 SRCs across 22 official languages by customizing the base instructional design. In this context, the research challenge is:

How to facilitate the design of eLearning Systems for flexible instructional design variations for the scale and variety of adult literacy in India?

3. A patterns based approach

Based on critical analysis of the literature, Buschmann et al. propose two major uses of patterns (i) *capturing, documenting and communicating experience through a set of abstractions and common vocabulary* and (ii) *construction of systems from predefined set of components and variants* (Buschmann et al., 2007, 1996). In this paper, we apply the idea of patterns for modeling instructional design and its variants based on NLMA's IPCL and *e*Learning Systems based on TCS' field-tested *e*Learning Systems. We also look at patterns as a central way to encapsulate commonly understood knowledge of experts (instructional designers, software architects) as a base for design of educational technologies. Figure 1 presents the core architecture of the proposed patterns based approach. This architecture stems from fundamental principles in software engineering and integrates multiple architecture styles. This architecture shows a holistic perspective (top-down and bottom-up) and tries to integrate patterns in both domain as well as software through five layers. We now briefly explain different layers of our approach.

- *Instructional design/methodologies.* Most of the educational technologies today require huge effort from teachers in configuring the technologies rather than on focusing instruction (Laurillard et al., 2013). The first layer from the top in Figure 1 focuses on providing a pedagogical foundation for design of educational technologies. We rely on well-established principles and practices from a pedagogical perspective in this paper and specifically focus on how we can structure these practices towards systematic design of educational technologies. For adult literacy case study, we rely on IPCL methodology (DAE, 2003) as a pedagogical foundation. This pedagogy is further integrated with other commonly accepted approaches such as Merrill's principles of instruction (Merrill, 2012) from instructional process perspective and Bloom's taxonomy from learning perspective.

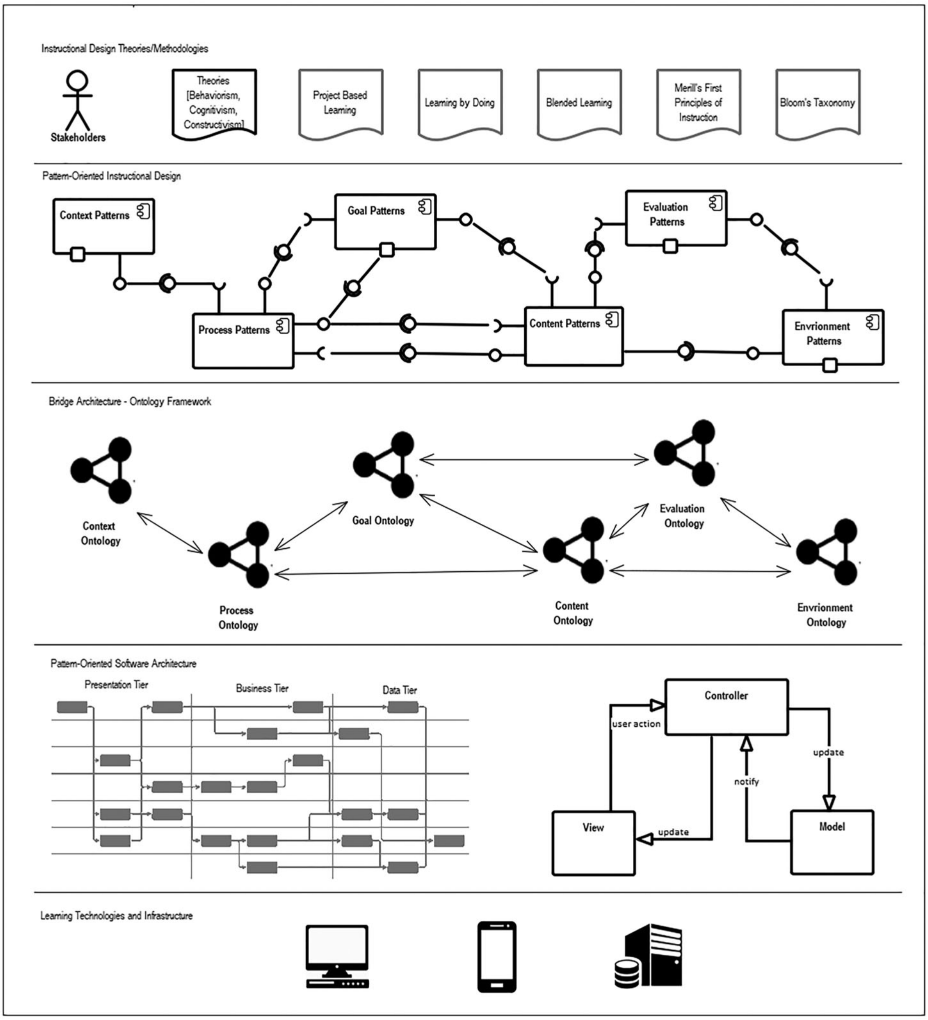


Figure 1. Architecture of patterns based approach.

- *Pattern-oriented instructional design.* The goal of this layer is to model instructional design using patterns (Section 4).
- *An ontology based modeling framework.* This layer acts a bridge between domain and software platforms. The key purpose of this bridge layer is to connect from instructional design (domain) to educational technologies (software) through a common interface. We used ontologies to represent instructional knowledge based on patterns and to connect with the rest of the software architecture (Chimalakonda & Nori, 2013c).
- *Pattern-oriented software architecture.* In their seminal work that inspired successful use of patterns in software engineering, Buschmann et al. emphasize that the primary purpose of architectural patterns is to create a fundamental structural organization schema for software systems (Buschmann et al., 1996). In our approach, instructional architecture patterns derived from POID and software architecture patterns that drive POSA provide this structure for instructional design and educational technologies. We use common interfaces to bridge these different types of patterns at different levels of abstraction *(instructional architecture patterns → instructional design patterns → software architecture patterns → design patterns)*. Essentially, POSA

represents the architecture of educational technologies. In the fourth layer and on the left hand side of Figure 1, we have a three-tier architecture of a TEL system that implements two important architectural patterns from Buschmann et al. (1996) --(i) *From Mud to Structure (Layers)* allows controlled decomposition of overall system (ii) *Model-View-Controller (MVC)* (Krasner & Pope, 1988) and several design patterns (*Strategy*, *Composite*, *Factory*). On the right hand side is an MVC architectural pattern that is derived from instructional architecture patterns and instructional design patterns in POID (Section 5).

- *Technologies, platform and infrastructure.* This layer provides the delivery platform with a set of technologies and tools. In this layer, a set of technologies and tools are designed to support the proposed architecture and facilitate the semi-automatic development of *e*Learning Systems.

We will elaborate POID and POSA in further sections.

4. Pattern-oriented instructional design

Domain engineering is a critical activity in software engineering to address complexity, reuse and evolution needs of software systems (Taylor et al., 2009). Modeling and structuring *domain* is a fundamental step that acts as a basis towards facilitating reuse using a patterns based approach. In this paper, we take a cue from Apel et al. (2013) and consider *domain* as an area of knowledge (i) that covers the desired requirements of the systems (ii) includes a set of concepts and terminology understood by practitioners and (iii) includes the knowledge of how to build software systems in that area.

We propose Pattern-Oriented Instructional Design as *a domain engineering activity* towards design of educational technologies based on instructional design. Based on pedagogical inputs, POID aims at designing a solution in the problem domain using the language of instructional designers and teachers. The purpose of this layer is two-fold (i) *to structure instructional design for reuse* (ii) *to facilitate flexible modeling of instructional design variations*. We base the design of POID on POSA (Buschmann et al., 2007) to model instructional design aspects as patterns. Figure 2 shows a high level diagram of how POID corresponds to POSA with different levels of granularity. For example, on the left hand side, we have design patterns at the bottom and then architecture patterns. On similar lines, we have instructional design patterns at the bottom and instructional architecture patterns on top of them.

Figure 3 shows an overview of POID. The core instructional design requirements stem from different kinds of instructors and learners, which are then used as input to compose different patterns such as *goals*, *process*, *content* and so on to create a specific instructional design. *A POID is an integration of patterns at instructional architecture level and instructional design patterns towards designing a specific instructional design for a specific set of educational requirements*. Here, we consider instructional architectural patterns as high-level organizing structures that address recurring high-level problems in instructional design. These patterns are essentially an integration of instructional design patterns, which focus on a specific aspect such as goals or process. Examples of these patterns and variations are provided in Section 4.1 and Section 4.2.

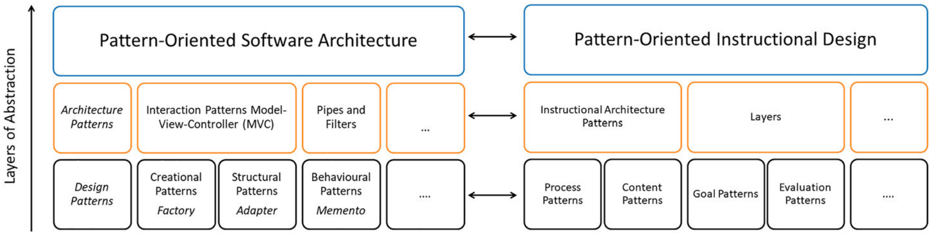


Figure 2. Comparision of POSA and POID.

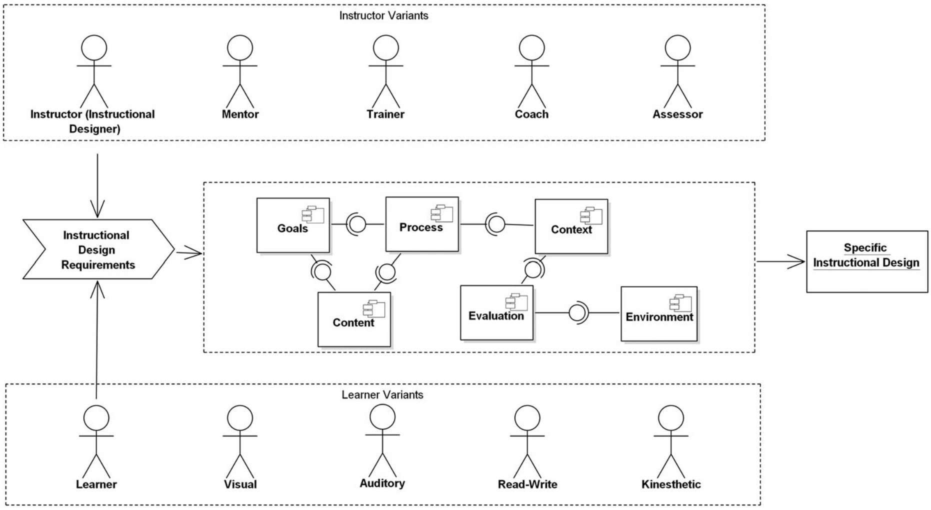
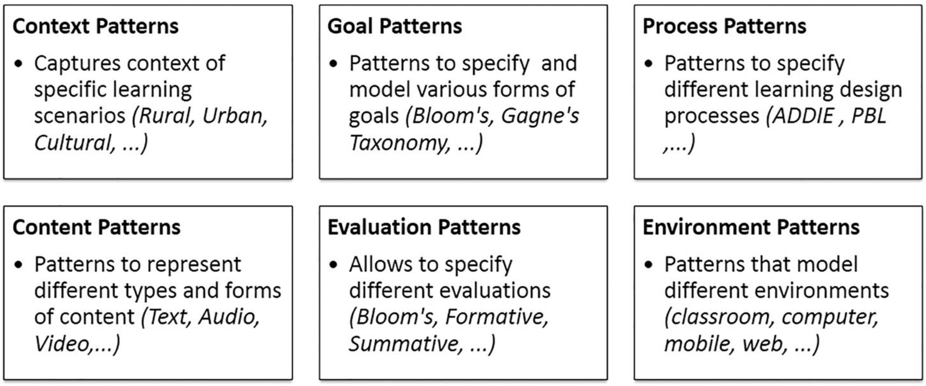


Figure 3. An overview of Pattern-Oriented Instructional Design.

The POID process starts by understanding instructional design and then progresses towards a detailed analysis of the instructional design to identify patterns in the domain. Then these patterns are continuously refined and relationships between them are identified and established. Figure 4 shows a classification of patterns in POID into categories of *Context*, *Goals*, *Process*, *Content*, *Evaluation* and *Environment*. This list is primitive and is designed such that it can be adapted and extended by instructional designers to support evolution. Each of these categories have patterns that aim to address a particular aspect of instructional design. For example, *Goals* in instructional design can be represented in various forms such as revised Bloom‘s taxonomy or Gagne's learning outcomes, and each of them can have their associated patterns, from which the teacher or instructional designer chooses patterns for their particular context. In our approach, every pattern provides and requires a set of interfaces that clearly define the boundaries of that pattern and how that pattern communicates with the rest of the patterns.

Figure 1 shows two possible instructional architecture patterns; one that integrates goals and evaluations (goal provides an interface ↔ evaluation requires an interface). The core idea of this pattern is to provide a flexible architecture that allows changing of goals and evaluations in an instructional design with less effort. Another possible pattern could be an integration of processes


Context Patterns
- Captures context of specific learning scenarios *(Rural, Urban, Cultural, ...)*

Goal Patterns
- Patterns to specify and model various forms of goals *(Bloom's, Gagne's Taxonomy, ...)*

Process Patterns
- Patterns to specify different learning design processes *(ADDIE , PBL ,...)*

Content Patterns
- Patterns to represent different types and forms of content *(Text, Audio, Video,...)*

Evaluation Patterns
- Allows to specify different evaluations *(Bloom's, Formative, Summative, ...)*

Environment Patterns
- Patterns that model different environments *(classroom, computer, mobile, web, ...)*

Figure 4. A classification of patterns in instructional design.

and content. This kind of abstraction leads to POID, which in itself is a technology independent solution in the language of problem domain. However, patterns provide a basic structure of a generic solution, which have to be further adapted and implemented for a specific context a basic structure of a generic solution to a family of problems (Buschmann et al., 2007).

In this paper, we rely on our 12 year-long experience in adult literacy case study to discover patterns. *We also conducted a workshop in collaboration with TCS and NLMA to discover our patterns with the community towards a consensus* .[4] The workshop consisted of directors of SRCs representing officers who are responsible for creating instructional process and material for teaching adult illiterates based on local requirements. Specifically, we introduced our patterns for instructional process and content and incorporated their feedback. We also held interviews with the directors of SRCs to figure out possible variations in technology for adult literacy. In addition to our experience, we also rely on commonly accepted approaches such as revised Bloom‘s taxonomy (Anderson et al., 2001) and Merrill‘s first principles of instruction (Merrill, 2012) as a basis for the proposed patterns.

We briefly explain two instructional design patterns in the next section.

4.1. A pattern for modeling instructional processes

We present *ProcessPattern* as a basic structure for modeling instructional process as shown in Figure 5. The pattern named *pasi* is organized as a *lesson* and denotes *p*lays, *a*cts, *s*cenes and *i*nstructions which are derived from Chimalakonda (2017). The number and order of the plays, acts, scenes, instructions is not strictly fixed. For example, the first *play*, *act* and *scene* focus on providing motivation to the learner and the last *instruction* might be a summary of what has been learnt so far in a particular lesson. Figure 6 shows few examples of *acts* and some *scenes*. In this example, there are several *acts* each having its respective goals, and consisting of specific *scenes* and further *instructions*. For example, *Act2* introduces new sounds through familiar words or phrases and *Act7* focuses on providing learners with exercises.

This *ProcessPattern* has several sources of variation for instruction process. The variations can be the *number* of plays, acts, scenes, instructions; the *order* of them, the specific play, act, scene or

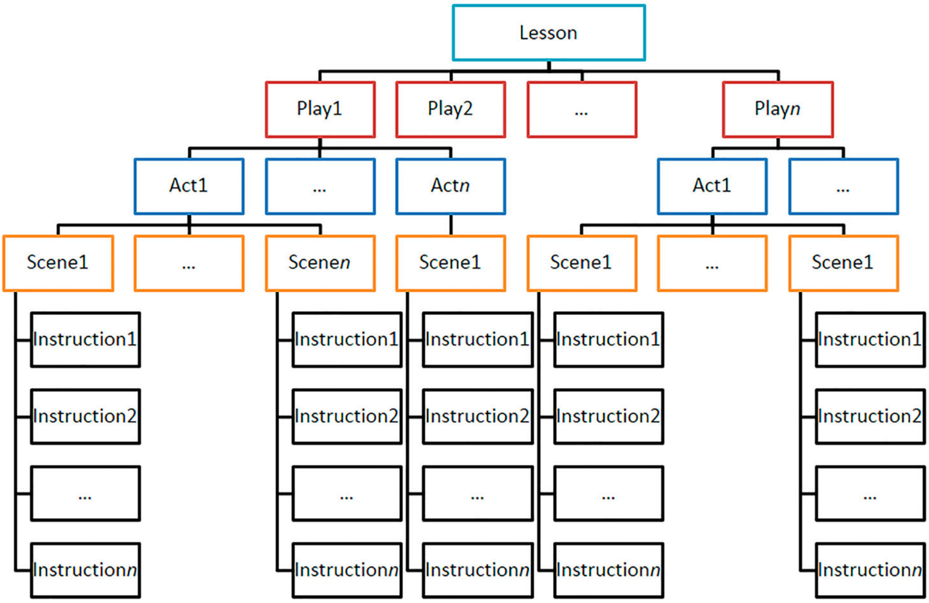


Figure 5. Structure of instructional process.

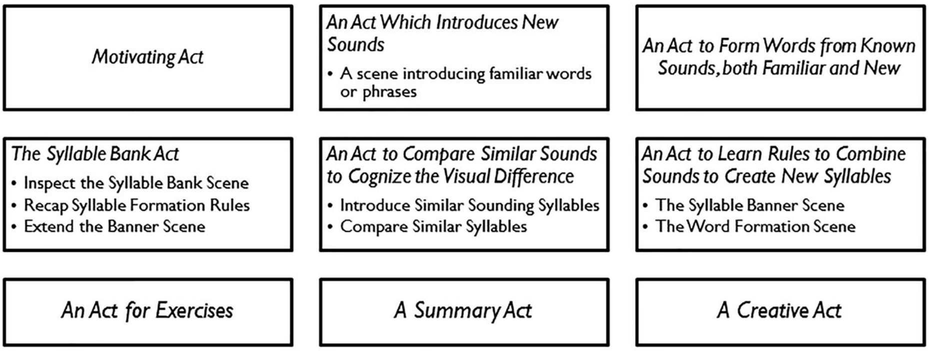

Motivating Act
An Act Which Introduces New Sounds
• A scene introducing familiar words or phrases
An Act to Form Words from Known Sounds, both Familiar and New
The Syllable Bank Act
• Inspect the Syllable Bank Scene
• Recap Syllable Formation Rules
• Extend the Banner Scene
An Act to Compare Similar Sounds to Cognize the Visual Difference
• Introduce Similar Sounding Syllables
• Compare Similar Syllables
An Act to Learn Rules to Combine Sounds to Create New Syllables
• The Syllable Banner Scene
• The Word Formation Scene
An Act for Exercises
A Summary Act
A Creative Act

Figure 6. Example *acts* in adult literacy instructional design.

instruction, the content used and other aspects of instruction providing customization for scale and variety. An instance could be *Act2* which could be changed to teach new sounds to different learners by using their names as the basis.

While the *ProcessPattern* provides a goal-driven structure for modeling a pedagogy at a high-level, it misses several details. It is here where we leverage Merrill's analysis of existing instructional design models (Merrill, 2012) that led to the fundamental principles of *activation*, *application*, *integration demonstration* and *task-orientation*. Each of these principles (activities) are repeatedly used in a specific order in the instructional process to fulfill goals. Generally, the *ProcessPattern* involves some or all of these principles at different levels of granularity but the application of these principles becomes explicit for tasks at *instruction* level. So, for example *Scene1* of *Act2* introduces words that are familiar to the learners essentially involving activation principle whereas learners have to use application principle in *Scene2* of *Act5* to form new words from existing syllables and so on.

We used this pattern to represent a commonly accepted instructional process based on IPCL methodology (DAE, 2003) for adult literacy case study (Chimalakonda & Nori, 2017), which could be instantiated thousands of times for all Indian languages and variants. The details of the instructional process variants for multiple Indian languages is presented in Section 6.

4.2. A pattern for modeling instructional material

Instructional material is at the center of instructional design and we discovered a pattern for content. Figure 7 shows the structure of *ContentPattern* with progression from simple facts to be remembered to foundations in the subject. *Facts* are essentially irrefutable observations in the subject and *cases* are observed based on abstraction of similar *facts*. A further attempt to identify similar *cases* can lead

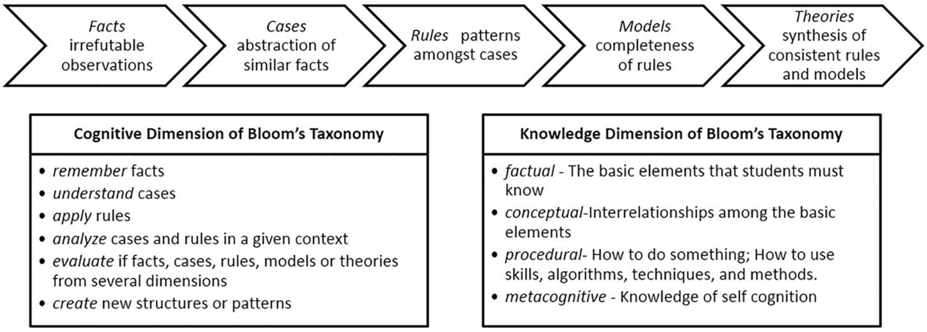

Facts irrefutable observations
Cases abstraction of similar facts
Rules patterns amongst cases
Models completeness of rules
Theories synthesis of consistent rules and models
Cognitive Dimension of Bloom's Taxonomy
• remember facts
• understand cases
• apply rules
• analyze cases and rules in a given context
• evaluate if facts, cases, rules, models or theories from several dimensions
• create new structures or patterns
Knowledge Dimension of Bloom's Taxonomy
• factual - The basic elements that students must know
• conceptual-Interrelationships among the basic elements
• procedural- How to do something; How to use skills, algorithms, techniques, and methods.
• metacognitive - Knowledge of self cognition

Figure 7. *ContentPattern* and mapping to *cognitive*, *knowledge* dimensions of Bloom's taxonomy

to a set of *rules* that can be applied to learn the content. *Models* and *theories* provide a rationle for learners to reason about the knowledge that was gained and during its application. This pattern is further mapped to cognitive and knowledge dimensions of Bloom's revised taxonomy. For example, *facts* can be mapped to the *remember* and *rules* at *apply* level. Similarly, different levels of knowledge in the form of *facts*, *cases*, *rules*, *models* and *theories* can be mapped to *factual*, *conceptual*, *procedural* and *metacognitive* levels of knowledge dimension of Bloom's revised taoxnomy. The foundations of this pattern during its formative stages were presented in Chimalakonda and Nori (2012).

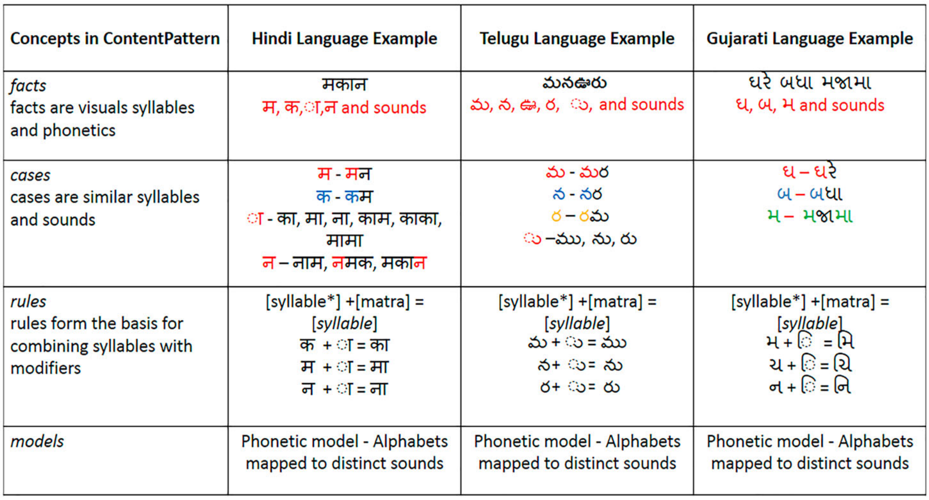
We have applied this pattern for *content* based on IPCL and scientific method for adult literacy case study. Figure 8 shows instances of the *ContentPattern* in *Hindi*, *Telugu* and *Gujarati* languages along with variations.

The *ProcessPattern* and *ContentPattern* can be construed as an instructional architecture pattern. In this pattern, teaching is structured as a set of *plays*, *acts*, *scenes* and *instructions* delivering *facts*, *cases*, *rules*, *models* and *theories* (Chimalakonda, 2017).

5. Pattern-oriented software architecture

Software architecture deals with most of the design decisions concerned with *structure*, *behavior*, *interaction*, *qualities*, and *implementation* of a software system (Taylor et al., 2009). In this paper, we are interested in identifying the variants that are possible in these aspects to support the desired requirements of scale and variety in educational technologies. In general, architectural patterns and reference architectures provide a baseline and a set of guidelines for creating specific software architectures tapping the variabilities of multiple related systems in a particular domain (Taylor et al., 2009). Morch lists three common ways of tailoring software applications (Mørch, 1997):

Customizing the application by configuring a set of pre-defined options ranging from configuring themes in the user interface to turning off or turning on features of the existing system. For example, an instructor might configure the theme of the *e*Learning System based on local context and culture, evaluation in terms of multiple choice or fill in the blanks and so on. In *integration*, users can have configuration options to integrate functionality that is outside the system. For example, a teacher might want to integrate learning management system such as *Moodle* into the current system. This might require tweaking of components, extending interfaces and fixing interoperability issues. A GLUE-architecture was proposed to support integration of tools into virtual environments



Concepts in ContentPattern	Hindi Language Example	Telugu Language Example	Gujarati Language Example
facts facts are visuals syllables and phonetics	मकान म, क,◌ा,न and sounds	మనఊరు మ, న, ఊ, ర, ◌ు, and sounds	ઘરે બધા મજામા ઘ, બ, મ and sounds
cases cases are similar syllables and sounds	म - मन क - कम ◌ा - का, मा, ना, काम, काका, मामा न – नाम, नमक, मकान	మ - మర న - నర ర – రమ ◌ు –ము, ను, రు	ઘ – ઘરે બ – બધા મ – મજામા
rules rules form the basis for combining syllables with modifiers	[syllable*] +[matra] = [*syllable*] क + ◌ा = का म + ◌ा = मा न + ◌ा = ना	[syllable*] +[matra] = [*syllable*] మ + ◌ు = ము న+ ◌ు= ను ర+ ◌ు= రు	[syllable*] +[matra] = [*syllable*] મ + િ = મિ ચ + િ = ચિ ન + િ = નિ
models	Phonetic model - Alphabets mapped to distinct sounds	Phonetic model - Alphabets mapped to distinct sounds	Phonetic model - Alphabets mapped to distinct sounds

Figure 8. Example of *ContentPattern* instances for *Hindi*, *Telugu* and *Gujarati* languages.

(Alario-Hoyos et al., 2013). When additional new features have to be added, the system has to be *extended* by adding new code, re-writing and sometimes even re-designing some parts of the system. For example, if a new accreditation rule requires the instructional goals to conform to a particular standard, then the system designed for extensibility should have a basic module for evaluation which could be extended to add/remove features required as per new accreditation requirements.

In this paper, we address variability in instructional design as well as in software through the idea of patterns. The core idea of POSA is to create software architecture patterns and map to patterns in POID. Architecture patterns can further consist of design patterns with each of them addressing variability at different levels of granularity. We briefly provide examples for both architecture and design patterns below.

We illustrate the idea of POSA through a commonly used pattern called Model-View-Controller (MVC) (Krasner & Pope, 1988). The key purpose of MVC is to facilitate separation of the interactive application into three parts or components to efficiently address changing requirements. Models represent the underlying application domain knowledge and act as core structure for Views and Controllers. Instructional design is the underlying domain in this paper and forms the basis for models. The representation of this model itself can vary based on how the domain is modeled. Views primarily focus on the user interface, graphical elements and what the users view as part of the system. A typical user interface consists of hierarchical views and the data for these views is fed from the model behind them. This is quite useful as a model can have many views associated with it facilitating variability from in user interfacesa user interface perspective. Consider a scenario where an *e*Learning System uses the same model but can be viewed with several user interface metaphors. The variabilities can range from simple color or themes to complex modifications emerging from instructional design. The order of these views itself can be a source of variability. *Controller* component of MVC pattern acts as the bridge between models and views.

It is not uncommon to integrate several architectural patterns while designing a software application to address varied requirements (Buschmann et al., 2007). For example, Figure 9 shows a simple overview of how MVC pattern can be integrated into a layered architectural pattern. Here the user interface, business logic and data are separated into three layers Presentation, Business and Domain. MVC pattern is spread across *presentation* and *business* layers. The model part of the MVC pattern is part of the domain strongly mapped to data parts of the domain i.e. pattern-oriented instructional design. A simple way to vary the application is to change these data parts in the POID without changing the interfaces exposed to POSA. Even MVC itself uses *Observer* pattern to notify views and controllers, providing a variation point. Composite pattern is commonly associated for constructing user interface elements in a hierarchical way and exposing only the top level view for the entire architecture leaving flexibility to change user interface elements. The domain layer emphasizes the strong role of domain in this architecture than just traditional databases. The data is mapped to patterns in the domain as this can allow traceability of changes from domain to software. A 10-step process for MVC pattern from Buschmann et al. (1996) along with potential variabilities is shown below:

(1) Separate human-computer interaction from core functionality
inputs, output behaviors, accessor functions

(2) Implement the change-propagation mechanism
publish-subscribe pattern, specific implementations

(3) Design and implement the views
appearance of views, display procedures, parameterized views, multiple draw methods

(4) Design and implement the controllers
specific behaviours for user actions, event handling

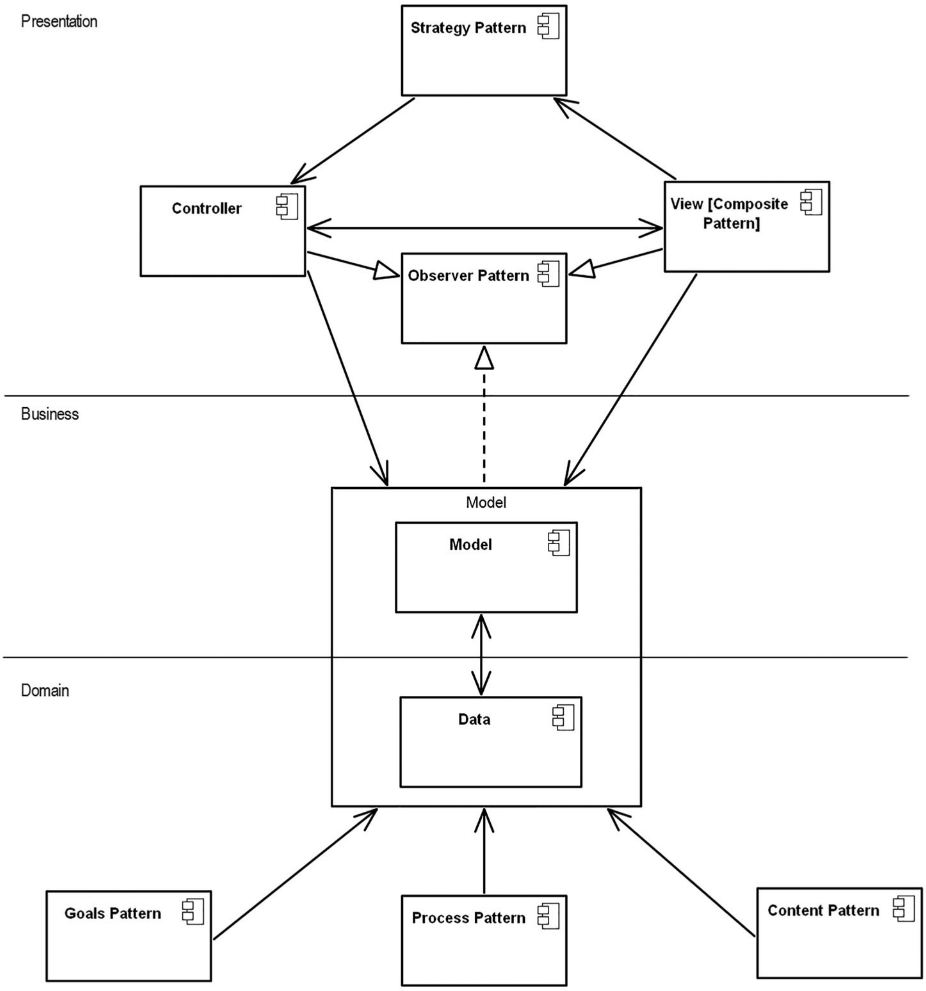


Figure 9. Layered Model View Controller Pattern. **Domain Patterns** – *Goals Pattern*, *Process Pattern*, *Content Pattern*; **Software Patterns** – *Strategy Pattern*, *Observer Pattern*, *Composite Pattern*.

(5) Design and implement the view-controller relationship
initializations for which factory method pattern could be used, hierarchy of views and controllers
(6) Implement the set-up of MVC
initializations, events
(7) Dynamic view creation
components for managing views
(8) Pluggable controllers
different controllers
(9) Infrastructure for hierarchical views and controllers
composite pattern, chain of responsibility pattern
(10) Further decoupling from system dependencies
bridge pattern, higher levels of abstraction

In the next section, we discuss an implementation of our approach that uses patterns as the base for instructional designs and educational technologies.

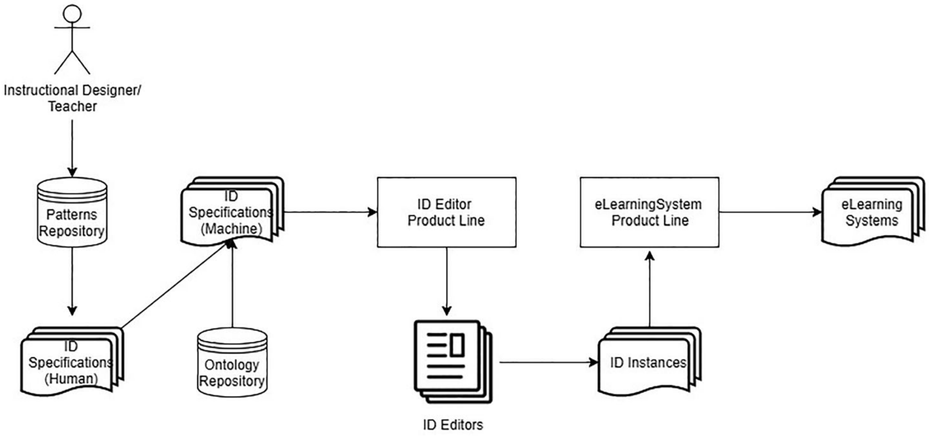
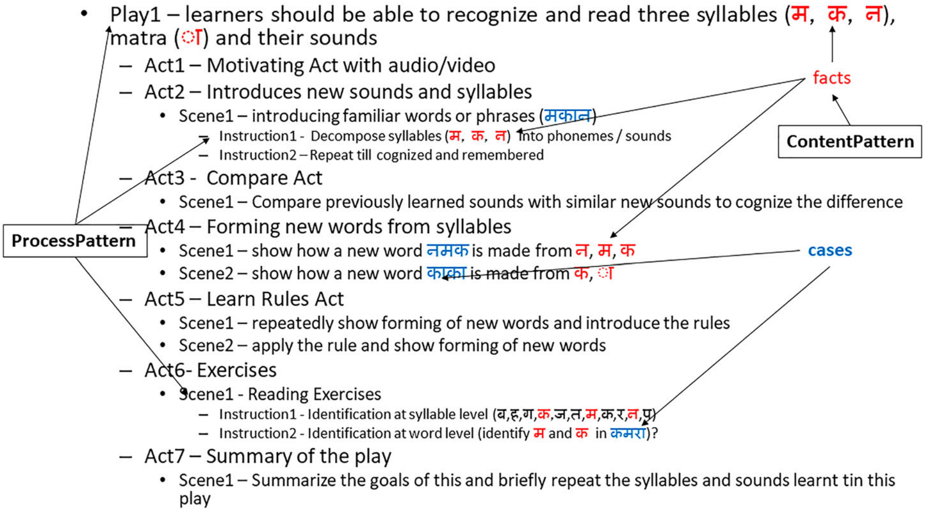

Instructional Designer/ Teacher
Patterns Repository
ID Specifications (Machine)
ID Editor Product Line
eLearningSystem Product Line
eLearning Systems
ID Specifications (Human)
Ontology Repository
ID Editors
ID Instances

Figure 10. Overview of implementation of pattern-oriented approach.

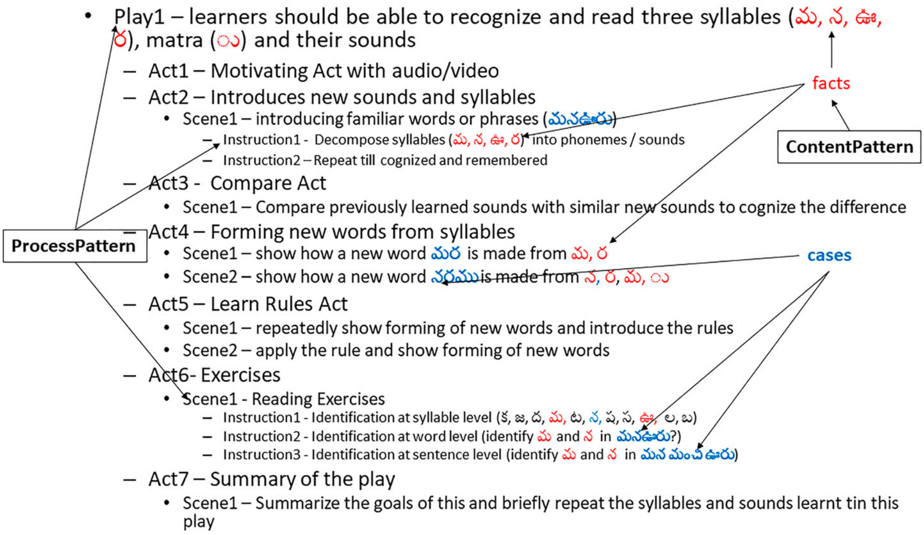
6. Evaluation

The primary goal of the research presented in this manuscript is to facilitate design of educational technologies for variations in instructional design. We followed a two-pronged approach for evaluation (i) implementation of a platform for modeling instructional design variations (Figure 10[5]) (ii) use of the platform for modeling instructional design variations for 8 Indian languages (Figures 11–14). In addition, we transferred the platform and approach to NLMA for on-field implementation and further proliferation. Chimalakonda and Nori (2013b) have initially designed an authoring tool named *EasyAuthor* to help non-technical teachers to create educational content based on patterns through ID*ont* framework (Chimalakonda & Nori, 2013c). This tool has wizards that help teachers to create different aspects of instructional design (context, goals, process, content, evaluation and environment). Each of these wizards are based on patterns for corresponding aspects of the instructional design. This essentially connects different components of educational technologies to instructional design. However, this tool did not address the need to create a set of



- Play1 – learners should be able to recognize and read three syllables (म, क, न), matra (◌ा) and their sounds
  - Act1 – Motivating Act with audio/video
  - Act2 – Introduces new sounds and syllables
    - Scene1 – introducing familiar words or phrases (मकान)
      - Instruction1 - Decompose syllables (म, क, न) into phonemes / sounds
      - Instruction2 – Repeat till cognized and remembered
  - Act3 - Compare Act
    - Scene1 – Compare previously learned sounds with similar new sounds to cognize the difference
  - Act4 – Forming new words from syllables
    - Scene1 – show how a new word नमक is made from न, म, क
    - Scene2 – show how a new word काका is made from क, ◌ा
  - Act5 – Learn Rules Act
    - Scene1 – repeatedly show forming of new words and introduce the rules
    - Scene2 – apply the rule and show forming of new words
  - Act6- Exercises
    - Scene1 - Reading Exercises
      - Instruction1 - Identification at syllable level (ब,ह,ग,क,ज,त,म,क,र,न,प)
      - Instruction2 - Identification at word level (identify म and क in कमरा)?
  - Act7 – Summary of the play
    - Scene1 – Summarize the goals of this and briefly repeat the syllables and sounds learnt tin this play

Figure 11. Part of an example *play* in *Hindi* language based on *ProcessPattern*(*pasi*).



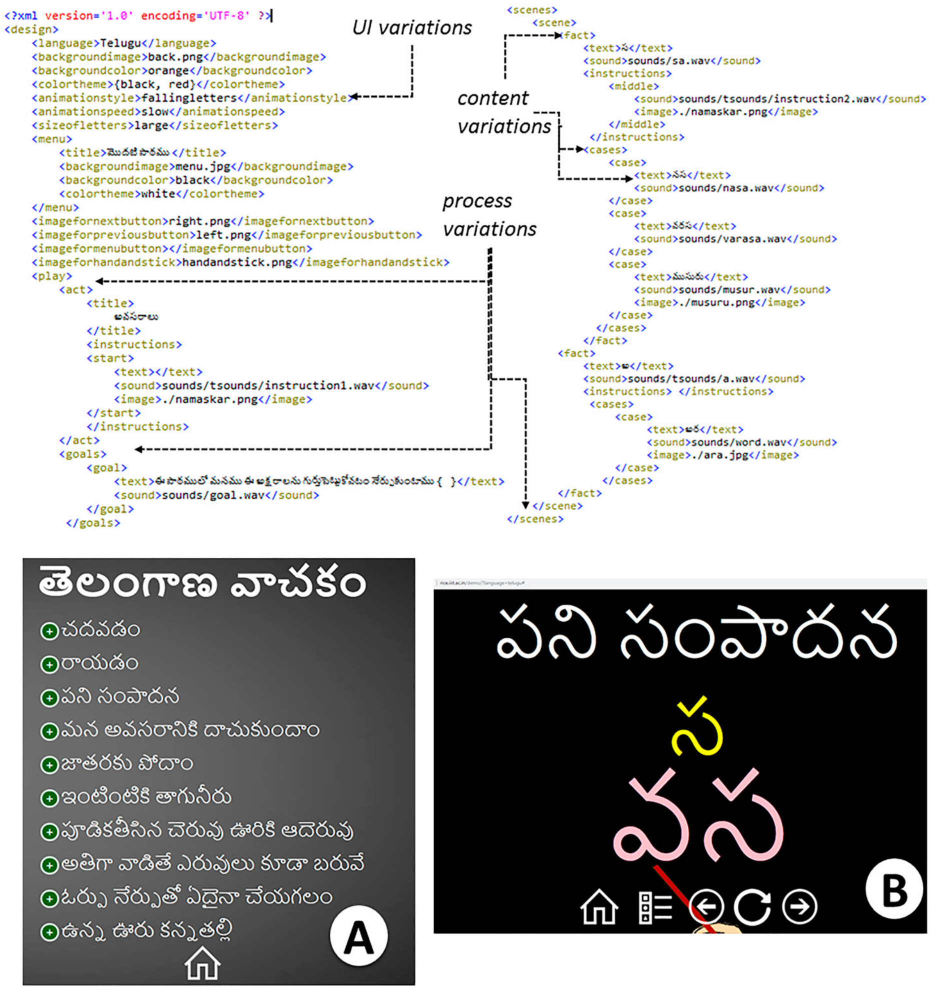
- Play1 – learners should be able to recognize and read three syllables (మ, న, ఊ, ర), matra (◌ం) and their sounds
  - Act1 – Motivating Act with audio/video
  - Act2 – Introduces new sounds and syllables
    - Scene1 – introducing familiar words or phrases (మనఊరు)
      - Instruction1 - Decompose syllables (మ, న, ఊ, ర) into phonemes / sounds
      - Instruction2 – Repeat till cognized and remembered
  - Act3 - Compare Act
    - Scene1 – Compare previously learned sounds with similar new sounds to cognize the difference
  - Act4 – Forming new words from syllables
    - Scene1 – show how a new word మర is made from మ, ర
    - Scene2 – show how a new word నరము is made from న, ర, మ, ◌ం
  - Act5 – Learn Rules Act
    - Scene1 – repeatedly show forming of new words and introduce the rules
    - Scene2 – apply the rule and show forming of new words
  - Act6- Exercises
    - Scene1 - Reading Exercises
      - Instruction1 - Identification at syllable level (క, జ, ద, మ, ట, న, ష, స, ఊ, ల, బ)
      - Instruction2 - Identification at word level (identify మ and న in మనఊరు?)
      - Instruction3 - Identification at sentence level (identify మ and న in మన మంచి ఊరు)
  - Act7 – Summary of the play
    - Scene1 – Summarize the goals of this and briefly repeat the syllables and sounds learnt tin this play

Figure 12. Part of an example *play* in *Telugu* language based on *ProcessPattern* (*pasi*).

authoring tools based on a variety of instructional designs. For example, there is a need for *EasyAuthor1* mapping to an *InstructionalDesignModel1*, *EasyAuthor2* for *InstructionalDesignModel2* and so on. To address this need of automating the development of *EasyAuthor(N)* tools for varied *InstructionalDesignModel(N)*, we have developed a software product line approach based on ontologies as detailed in Chimalakonda (2017) and Chimalakonda and Nori (2020).

Figure 10 shows an overview of the implementation of our approach. Here, an instructional designer first decides the instructional design model for a set of courses that have to be taught. For example, the instructional designer may choose to use "project based learning" as a strategy to teach several courses. He creates a set of patterns for aspects such as *goals*, *process*, *content* either from scratch or by customizing existing patterns to suit the specific instructional design model. However, this model is mainly aimed at different stakeholders such as teachers, instructional designers, evaluators, policy makers and so on. These instructional design specifications have to be converted to specifications that are machine-processable such that automation is possible. To achieve this, ontologies are used to concretely represent the patterns (Chimalakonda, 2017). These specifications are read by a tool called *ID Editor Product Line* that semi-automatically generates an "*ID Editor*" that essentially allows a teacher to create a specific instance of instructional design. This *ID Editor* is driven by the idea of connecting patterns in POID and POSA. The *ID Instances* are then used as input to another product line for generating *e*Learning Systems. Technically, the patterns are represented using ontologies as an OWL file, which are instantiated for specific instance of the instructional design and eventually processed by tools to generate the *e*Learning Systems consisting of user interface for adult literary in India. The technical details of these product lines is beyond the scope of this paper but are available elsewhere Chimalakonda (2017) and Chimalakonda and Nori (2020).

Based on the process pattern presented in Section 4.1 and structure in Figure 5, we show a fractional part of an example *play* for the *Hindi* language in Figure 11. In this example, there are several *acts* each having its respective goals, and consisting of specific *scenes* and further *instructions*. For example *Act4* deals with the goal of teaching how to form new words from syllables with two *scenes* illustrating how new words are formed from already learnt syllables. This *ProcessPattern* has several sources of variation for instruction process. The variations can be the *number* of plays, acts, scenes, instructions; the *order* of them, the specific play, act, scene or instruction, the content used and other aspects of instruction providing customization for scale and variety. We



```
<?xml version='1.0' encoding='UTF-8' ?>
<design>
    <language>Telugu</language>
    <backgroundimage>back.png</backgroundimage>
    <backgroundcolor>orange</backgroundcolor>
    <colortheme>{black, red}</colortheme>
    <animationstyle>fallingletters</animationstyle>
    <animationspeed>slow</animationspeed>
    <sizeofletters>large</sizeofletters>
    <menu>
        <title>మొదటి పాఠము</title>
        <backgroundimage>menu.jpg</backgroundimage>
        <backgroundcolor>black</backgroundcolor>
        <colortheme>white</colortheme>
    </menu>
    <imagefornextbutton>right.png</imagefornextbutton>
    <imageforpreviousbutton>left.png</imageforpreviousbutton>
    <imageformenubutton></imageformenubutton>
    <imageforhandandstick>handandstick.png</imageforhandandstick>
    <play>
        <act>
            <title>
                అవసరాలు
            </title>
            <instructions>
            <start>
                <text></text>
                <sound>sounds/tsounds/instruction1.wav</sound>
                <image>./namaskar.png</image>
            </start>
            </instructions>
        </act>
        <goals>
            <goal>
                <text>ఈ పాఠములో మనము ఈ అక్షరాలను గుర్తుపెట్టుకోవటం నేర్చుకుంటాము {  }</text>
                <sound>sounds/goal.wav</sound>
            </goal>
         </goals>
```

```
<scenes>
    <scene>
        <fact>
            <text>స</text>
            <sound>sounds/sa.wav</sound>
            <instructions>
                <middle>
                    <sound>sounds/tsounds/instruction2.wav</sound>
                    <image>./namaskar.png</image>
                </middle>
             </instructions>
            <cases>
                <case>
                    <text>నస</text>
                    <sound>sounds/nasa.wav</sound>
                </case>
                <case>
                    <text>వరస</text>
                    <sound>sounds/varasa.wav</sound>
                </case>
                <case>
                    <text>ముసురు</text>
                    <sound>sounds/musur.wav</sound>
                    <image>./musuru.png</image>
                </case>
              </cases>
            </fact>
        <fact>
            <text>అ</text>
            <sound>sounds/tsounds/a.wav</sound>
            <instructions> </instructions>
             <cases>
               <case>
                   <text>అర</text>
                   <sound>sounds/word.wav</sound>
                   <image>./ara.jpg</image>
               </case>
             </cases>
        </fact>
    </scene>
</scenes>
```

తెలంగాణ వాచకం

- చదవడం
- రాయడం
- పని సంపాదన
- మన అవసరానికి దాచుకుందాం
- జాతరకు పోదాం
- ఇంటింటికి తాగునీరు
- పూడికతీసిన చెరువు ఊరికి ఆదెరువు
- అతిగా వాడితే ఎరువులు కూడా బరువే
- ఒర్పు నేర్పుతో ఏదైనా చేయగలం
- ఉన్న ఊరు కన్నతల్లి

పని సంపాదన

స

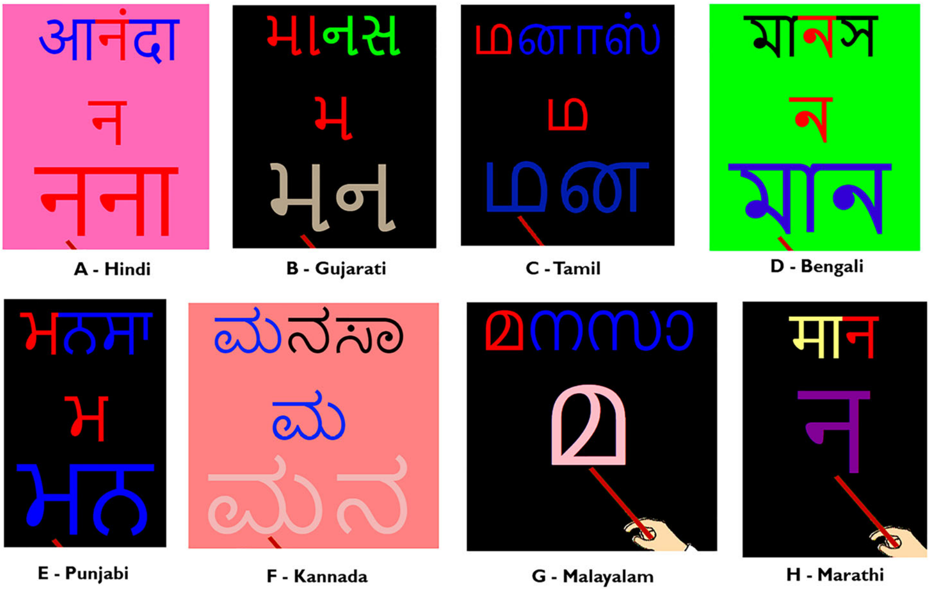
వస

Figure 13. Excerpt from an instance of customized instructional design with process and content pattern for *Telugu* language, table of contents (a) and a concrete animation (b).

have also used the same base to facilitate instructional process and content variations in *Telugu* language (Figure 12).

Figure 13 shows a fragment of custom instructional design instance for *Telugu* language. This instance is created from an instructional design schema that covers aspects such as *context*, *goals*, *process*, *content* (Chimalakonda, 2017). The figure shows three possible variations for instructional design. UI variations include properties such as *background*, *animationstyle*, *animationspeed* for customizing user interface of the eLearning System. Process and content variations are based on *process* and *content* pattern and modifications to these patterns are parsed by the *eLearningSystem Product Line* tool to eventually generate the eLearning System. For example, teaching through new words and syllables can be done by modifying *cases* and *facts* in the custom instructional design instance, which is represented as XML. Figure 13(a) shows the entire table of contents generated based on textbook content for adult literacy using the customized instructional design instance and Figure 13(b) shows the custom animation that is generated from the tool based on the same instructional design instance.

Figure 14 shows the content variations that are generated using our platform for 8 Indian languages [A-Hindi, B-Gujarati, C-Tamil, D-Bengali, E-Punjabi, F-Kannada, G-Malayalam, H-Marathi].



A - Hindi B - Gujarati C - Tamil D - Bengali

E - Punjabi F - Kannada G - Malayalam H - Marathi

Figure 14. Content variations in 8 Languages [A-Hindi, B-Gujarati, C-Tamil, D-Bengali, E-Punjabi, F-Kannada, G-Malayalam, H-Marathi] generated from the platform based on proposed approach.

Owing to the nature of our research work closely connected to the idea of solving societal challenges using computing, the implementation of our research work is made available at http://rice.iiit.ac.in[6] and the work has been transferred to NLMA, Govt. of India for further proliferation. The mobile version of the generated software is deployed on *Google Play Store* and is available at https://bit.ly/2T3DQRU. In addition, our work is also listed in the official websites of Department of Adult Education of *Govt. of Telangana* at http://tslma.nic.in/ and State Resource Center, *Govt. of Telangana* at http://srctelangana.com/.

7. Discussion & limitations

This paper is an attempt to integrate research on patterns and patterns-based approaches in software engineering and instructional design to facilitate design of educational technologies that have a pedagogical basis and support systematic reuse. There are many ways of improving our work.

Even though the patterns based approach presented in this paper has addressed the concerns of scale and variety during design of educational technologies and had practical impact for adult literacy in India, there are several limitations of our approach. Firstly, there is no way to claim that the patterns we presented are the only possible patterns for adult literacy instructional design nor they are comprehensive. In each category of patterns such as goals, process, content and so on, there is possibility of a variety of patterns in multiple contexts. We have realized that it is possible to map the instructional process and material in *primer* to *ProcessPattern* and *ContentPattern* only if instructors are trained in NLMA's IPCL methodology, Merrill's first principles of instruction and Bloom's revised taxonomy. However, in practice this is seldom the case as most of the adult literacy teachers are not fully trained and it is practically hard to find well-trained instructors for all official languages across India.

The proposed approach assumes that patterns can be composed in many ways to generate specific instructional designs. While this may be possible with compatible patterns such as *ProcessPattern* and *ContentPattern*, it is hard to compose other patterns in a coherent way, especially if patterns emerge from differing contexts, which is noted in the literature as well Derntl (2005). The effort for customizing the patterns and facilitating their composition can sometimes damage the reuse advantage of patterns. By design, patterns evolve over a period of time after feedback from the *e*Learning Systems based on them are deployed on the field. There is a need to figure out better mechanisms to update educational technologies based on a set of patterns if the base patterns change. Integration of domain and software patterns is currently done by researchers who have in-depth background of both instructional design and software. However, when we transferred the approach to NLMA, we realized that a software expert is required to guide the integration of patterns.

8. Conclusions & future work

We emphasized that lack of instructional design base is a critical challenge for design of educational technologies. So is the challenge of facilitating reuse and supporting a variety of instructional designs. While there is extensive work on patterns for instructional design and technology enhanced learning, the main focus has been either on domain patterns (instructional design) or on technology patterns, and not integrating both. To address these concerns, we presented a patterns-based approach that integrates patterns in instructional design and educational technologies. The crux of this approach is to create a structure of solution in instructional design and integrate it with a solution in technology based on patterns. This approach has its roots in fundamental architecture principles in software engineering. Based on these principles, we presented an architecture that integrates *Pattern-Oriented Instructional Design* that is driven by instructional design methodologies and *Pattern-Oriented Software Architecture* that drives the design of educational technologies. The essence of our approach is to systematically model different aspects of instructional design (*goals*, *process*, *content*) using *patterns* such as *GoalsPattern*, *ProcessPattern* (plays, acts, scenes, instructions) and *ContentPattern* (facts, cases, rules, models and theories). We demonstrated the application of our approach to model patterns in adult literacy case study in India. We then provided an implementation of our approach that generates instructional design authoring tools based on patterns. Finally, we see this paper as a first step that addresses some challenges during design of educational technologies through solutions in software engineering. We see the following directions for further research:

- Discovering new patterns to support the breadth of instructional designs, and figuring out mechanisms to support evolution of existing patterns.
- Modeling notations for representing patterns and supporting techniques and tools for handling patterns throughout life cycle.
- Formal approaches for composing patterns in domain, software and between domain and software.
- Approaches for validating assembly of patterns and their integration.
- Applying our approach to beyond adult literacy for schooling and other forms of education, both for facilitating patterns in instructional design and software.

Notes

1. We consider *instructional design as an underlying structure consisting of different aspects of instruction such as goals, process, content aimed at (i) providing a base for quality of instruction (ii) facilitating design of educational technologies*
2. http://nlm.nic.in
3. http://www.nlm.nic.in/dir.htm
4. Meeting held at TCS, Hyderabad, India in 2011 and further at NLMA, Delhi, India in 2016
5. http://rice.iiit.ac.in
6. https://git.io/fjx7Z

Acknowledgements

We thank Tata Consultancy Services (TCS) for initial inputs, National Literacy Mission Authority (NLMA) for considering our work at national level, and Govt. of Telangana who are the early adopters of the proposed work.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Sridhar Chimalakonda http://orcid.org/0000-0003-0818-8178

References

Akshar+, T. (2020). Literacy programmes in india, digital literacy in india, tara akshar+.

Alario-Hoyos, C., Bote-Lorenzo, M. L., GóMez-SáNchez, E., Asensio-PéRez, J. I., Vega-Gorgojo, G., & Ruiz-Calleja, A. (2013). Glue!: An architecture for the integration of external tools in virtual learning environments. *Computers & Education*, *60*(1), 122–137. https://doi.org/10.1016/j.compedu.2012.08.010

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). A pattern language: Towns, buildings, construction (center for environmental structure series).

Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths, J., & Wittrock, M. C. (2001). *A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives, abridged edition*. Longman.

Apel, S., Batory, D., Kästner, C., & Saake, G. (2013). Feature-oriented software product lines: concepts and implementation. Springer Science & Business Media.

Avgeriou, P., Papasalouros, A., Retalis, S., & Skordalakis, M. (2003). Towards a pattern language for learning management systems. *Educational Technology & Society*, *6*(2), 11–24.

Berger, C., & Kam, R. (1996). Definitions of instructional design. *Retrieved March*, *30*, 2020.

Bergin, J., Eckstein, J., Volter, M., Sipos, M., Wallingford, E., Marquardt, K., Chandler, J., Sharp, H., & Manns, M. L. (2012). *Pedagogical patterns: Advice for educators*. Joseph Bergin Software Tools.

Bodily, R., Leary, H., & West, R. E. (2019). Research trends in instructional design and technology journals. *British Journal of Educational Technology*, *50*(1), 64–79. https://doi.org/10.1111/bjet.2019.50.issue-1

Borchers, J. O. (2001). A pattern approach to interaction design. *Ai & Society*, *15*(4), 359–376. https://doi.org/10.1007/BF01206115

Botturi, L., Stubbs, S. T., & Global, I. (2008). *Handbook of visual languages for instructional design: Theories and practices*. Information Science Reference Hershey.

Brown, A. H., & Green, T. D. (2015). *The essentials of instructional design: Connecting fundamental principles with process and practice*. Routledge.

Buschmann, F., Henney, K., & Schimdt, D. (2007). *Pattern-oriented software architecture: on patterns and pattern language* (*Vol. 5*). John Wiley & Sons.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). Pattern-oriented software architecture volume 1.

Caeiro, M., Llamas, M., & Anido, L. (2014). Poeml: Modeling learning units through perspectives. *Computer Standards & Interfaces*, *36*(2), 380–396. https://doi.org/10.1016/j.csi.2013.08.009

Chimalakonda, S. (2017). *A software engineering approach for design of educational technologies* [PhD Thesis]. International Institute of Information Technology-Hyderabad.

Chimalakonda, S., & Nori, K. V. (2012). Towards a model driven elearning framework to improve quality of teaching. In *Technology for Education (T4E), 2012 IEEE Fourth International Conference on* (pp. 138–143). IEEE.

Chimalakonda, S., & Nori, K. V. (2013a). Designing technology for 287 million learners. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on* (pp. 197–198). IEEE.

Chimalakonda, S., & Nori, K. V. (2013b). Easyauthor: Supporting low computer proficiency teachers in the design of educational content for adult illiterates. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems* (pp. 649–654). ACM.

Chimalakonda, S., & Nori, K. V. (2013c). Idont: An ontology based educational modeling framework for instructional design. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on* (pp. 253–255). IEEE.

Chimalakonda, S., & Nori, K. V. (2014). A patterns-based approach for modeling instructional design and tel systems. In *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on* (pp. 54–56). IEEE.

Chimalakonda, S., & Nori, K. V. (2017). A pattern for modeling instructional process for design of elearning systems -- quality, scale and variety. *Indian Journal of Adult Education*, *78*(1).

Chimalakonda, S., & Nori, K. V. (2020). A family of software product lines in educational technologies. *Computing*, 102, 1–28. https://doi.org/10.1007/s00607-019-00772-x

Consortium, I. G. L. (2003). Ims learning design specification.

Creswell, J. W., & Poth, C. N. (2017). *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.

DAE (2003). *Handbook for developing IPCL material*. Directorate of Adult Education, India.

Dalziel, J. (2003). Implementing learning design: The learning activity management system (lams).

Derntl, M. (2005). *Patterns for person centered e-learning*. Citeseer.

Derntl, M., & Motschnig-Pitrik, R. (2004). Patterns for blended, person-centered learning: Strategy, concepts, experiences, and evaluation. In *Proceedings of the 2004 ACM symposium on Applied computing* (pp. 916–923). ACM.

Dick, W., Carey, L., & Carey, J. O. (2001). *The systematic design of instruction* (*Vol. 5*). Longman New York.

Dighe, A. (2010). Use of ICTs in literacy. *Indian Journal of Adult Education*.

Duffy, T. M., & Jonassen, D. H. (2013). *Constructivism and the technology of instruction: A conversation*. Routledge.

Farrell, G. (2004). ICT and literacy: who benefits? Experience from Zambia and India.

Gagne, R. M., & Briggs, L. J. (1974). *Principles of instructional design*. Holt, Rinehart & Winston.

Garzotto, F., Retalis, S., & Chapter, V. (2009). A critical perspective on design patterns for e-learning. *Learning Objects: Issues, Applications and Technologies* (pp. 346–372).

Gibbons, A. S. (2013). *An architectural approach to instructional design*. Routledge.

Goodyear, P., Avgeriou, P., Baggetun, R., Bartoluzzi, S., Retalis, S., Ronteltap, F., & Rusman, E. (2004). Towards a pattern language for networked learning. *University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science* (pp. 449–455).

Goodyear, P., & Retalis, S. (2010). *Technology-enhanced learning: Design patterns and pattern languages*. Sense Publishers.

Gropper, G. L. (2017). Instructional design:? science? technology? both? neither. *Educational Technology*, 40–52.

Gustafson, K. L., & Branch, R. M. (1997). *Survey of instructional development models*. ERIC.

Kothari, B. (2008). Let a billion readers bloom: Same language subtitling (sls) on television for mass literacy. *International Review of Education*, *54*(5-6), 773–780. https://doi.org/10.1007/s11159-008-9110-3

Krasner, G. E., & Pope, S. T. (1988). A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of Object Oriented Programming*, *1*(3), 26–49.

Kruse, K. (2002). Introduction to instructional design and the addie model. *Retrieved March*, *26*, 2020.

Larson, M. B., & Lockee, B. B. (2019). *Streamlined ID: A practical guide to instructional design*. Routledge.

Laurillard, D. (2012). *Teaching as a design science: Building pedagogical patterns for learning and technology*. Routledge.

Laurillard, D. (2013a). *Rethinking university teaching: A conversational framework for the effective use of learning technologies*. Routledge.

Laurillard, D. (2013b). *Teaching as a design science: Building pedagogical patterns for learning and technology*. Routledge.

Laurillard, D., Charlton, P., Craft, B., Dimakopoulos, D., Ljubojevic, D., Magoulas, G., Masterman, E., Pujadas, R., Whitley, E. A., & Whittlestone, K. (2013). A constructionist learning environment for teachers to model learning designs. *Journal of Computer Assisted Learning*, *29*(1), 15–30. https://doi.org/10.1111/jcal.2013.29.issue-1

Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J. L., & Fernandez-Manjon, B. (2007). Educational modeling languages. In *Computers and education* (pp. 27–40). Springer.

Merriam, S. B. (2001). Andragogy and self-directed learning: Pillars of adult learning theory. *New Directions for Adult and Continuing Education*, *2001*(89), 3–14. https://doi.org/10.1002/ace.3

Merrill, M. D. (2012). *First principles of instruction*. John Wiley & Sons.

Meszaros, G., & Doble, J. (1998). A pattern language for pattern writing. *Pattern Languages of Program Design*, *3*, 529–574.

Millwood, R. (2014). *The design of learner-centred, technology-enhanced education* [PhD thesis]. University of Bolton.

Mizoguchi, R., Hayashi, Y., & Bourdeau, J. (2007). Inside theory-aware and standards-compliant authoring system. In *SW-EL'07* (p. 18).

Mor, Y. (2014). *Practical design patterns for teaching and learning with technology*. Springer.

Mørch, A. (1997). Three levels of end-user tailoring: Customization, integration, and extension. *Computers and Design in Context*, 51–76.

Neumann, S., Klebl, M., Griffiths, D., Hernández-Leo, D., L. De la Fuente-Valentin, Hummel, H., Brouns, F., Derntl, M., & Oberhuemer, P. (2009). Report of the results of an ims learning design expert workshop.

Patel, I. (2002). *Information and communication technology and distance adult literacy education in India*. Institute of Rural Management Anand.

Patterns, P. (2014). The pedagogical patterns project.

Reid, P. (2014). Categories for barriers to adoption of instructional technologies. *Education and Information Technologies*, *19*(2), 383–407. https://doi.org/10.1007/s10639-012-9222-z

Reigeluth, C. M. (2013a). *Instructional-design theories and models: A new paradigm of instructional theory* (*Vol. 2*). Routledge.

Reigeluth, C. M. (2013b). *Instructional design theories and models: An overview of their current status*. Routledge.

Reigeluth, C. M., & Carr-Chelman, A. A. (2009). *Instructional-design theories and models: Building a common knowledge base* (*Vol. 3*). Education Review//Reseñas Educativas.

Sampson, D. G., & Zervas, P. (2014). A hierarchical framework for open access to education and learning. *International Journal of Web Based Communities*, *10*(1), 25–51. https://doi.org/10.1504/IJWBC.2014.058385

Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2009). *Software architecture: foundations, theory, and practice*. Wiley Publishing.

TCS (2020). Csr case study, computer based functional literacy.

UNESCO (2014). *Education for all global monitoring report 2013/4: teaching and learning: achieving quality for all*. United Nations Educational and Scientific and Cultural Organization.

Villasclaras-FernáNdez, E., HernáNdez-Leo, D., Asensio-PéRez, J. I., & Dimitriadis, Y. (2013). Web collage: An implementation of support for assessment design in cscl macro-scripts. *Computers & Education*, *67*, 79–97. https://doi.org/10.1016/j.compedu.2013.03.002

Wagner, D. A., Castillo, N. M., Murphy, K. M., Crofton, M., & Zahra, F. T. (2014). Mobiles for literacy in developing countries: An effectiveness framework. *Prospects*, *44*(1), 119–132. https://doi.org/10.1007/s11125-014-9298-x