Designing a blended course in android app development using 4C/ID

Citation for published version (APA):

Marcellis, M., Barendsen, E., & Van Merriënboer, J. (2018). Designing a blended course in android app development using 4C/ID. In *Proceedings - 18th Koli Calling Conference on Computing Education Research, Koli Calling 2018* Article a19 Association for Computing Machinery. https://doi.org/10.1145/3279720.3279739

Document status and date:
Published: 12/11/2018

DOI:
10.1145/3279720.3279739

Document Version:
Publisher's PDF, also known as Version of record

Document license:
Taverne

Designing a Blended Course in Android App Development using 4C/ID

Marco Marcellis
Amsterdam University of Applied Sciences
The Netherlands
m.m.c.m.marcellis@hva.nl

Erik Barendsen
Radboud University & Open University
The Netherlands
e.barendsen@cs.ru.nl

Jeroen van Merriënboer
School of Health Professions Education, Maastricht University
The Netherlands
j.vanmerrienboer@maastrichtuniversity.nl

ABSTRACT

Developing an Android app is a complex skill that is difficult for students to master because many constituent skills have to be processed simultaneously, which might cause a cognitive overload. Cognitive Load Theory (CLT) offers a framework to analyze and reduce the cognitive load, but the question is how to apply CLT in a systematic way during course design. The Four-Component Instructional Design (4C/ID) model incorporates CLT and offers additional instructional methods that improve learning. This paper reports on a case study applying 4C/ID to develop a blended course in Android app development. The following 4C/ID components were designed using the Ten Steps to Complex Learning (Ten Steps) approach: learning tasks, supportive information and procedural information. Each designed component or part thereof was categorized as face-to-face or online, for both lecturer and student, resulting in a blended learning design. The Ten Steps approach proved to be valuable in designing a blended course in Android app development. Our case study will be the starting point of further research.

CCS CONCEPTS

• **Applied computing → E-learning**; *Distance learning*; Collaborative learning;

KEYWORDS

Android app development, Blended Learning, 4C/ID

ACM Reference Format:
Marco Marcellis, Erik Barendsen, and Jeroen van Merriënboer. 2018. Designing a Blended Course in Android App Development using 4C/ID. In *Proceedings of Koli Calling 2018.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Developing an Android app is a complex skill that involves many constituent skills. These constituent skills have to be processed simultaneously, which might cause a cognitive overload that would make it difficult for students to master developing an Android app. This cognitive overload is called high element interactivity in Cognitive Load Theory (CLT). CLT relates to the amount of information that working memory can hold at one time [2, 14, 15]. CLT offers a framework to analyze and reduce the cognitive load of learning activities and has been previously applied in computing education [8, 9, 11, 16].

The question remains as to how to apply CLT in a *systematic* way during course design. A complicating factor is that applying CLT frees up resources but does not automatically ensure that freed resources are devoted to learning [22]. The Four-Component Instructional Design (4C/ID) model incorporates CLT and offers additional instructional methods that improve learning. The 4C/ID model is acknowledged as one of the best instructional models [10] and has been applied in several domains to several kinds of courses, i.e. e-learning, blended learning, instructor-led [7, 13, 17, 23]. However, using this model to design a blended course in higher computing education is a novelty.

This case study describes how to design a blended course in Android app development using the 4C/ID model. We describe the background of the model and illustrate the design of the course by highlighting the main design steps. For us, the course design will be the start of a research project investigating the application of the 4C/ID model in software development education, as well as the cognitive load of the resulting learning activities.

2 THE FOUR-COMPONENT INSTRUCTIONAL DESIGN MODEL

According to the 4C/ID model, a course design for a complex skill can always be described as four interrelated components: (1) Learning Tasks, (2) Supportive Information, (3) Procedural Information and (4) Part-task Practice. Learning tasks are the backbone of the course design. Learning tasks are real-life, authentic tasks that have to deal with a whole task, that is, a task that a professional might encounter in real life. Supportive information consists of systematic approaches to problem solving (SAPs) and presents domain models. It helps students to acquire the nonrecurrent constituent skills. Procedural information consists of rules or procedures to carry out the recurrent constituent skills. Part-task practice gives students additional practice in acquiring recurrent constituent skills that need to become fully automated [20].

2.1 Cognitive load

The cognitive load associated with learning tasks is controlled by organizing the learning tasks into simple to complex task classes. The first task class typically consists of the simplest version of the

Koli Calling 2018, November 2018, Finland

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

complex skill that a professional might encounter in real life. A learning task in a simpler task class has less element interactivity than a learning task in a more complex task class. Learning tasks in the same task class have the same level of element interactivity. The cognitive load associated with supportive information is controlled by only presenting the information to the students which is necessary to complete a particular task class. In addition, the information is presented before the students start working on the learning tasks. The cognitive load associated with procedural information is controlled by offering the procedural information just-in-time. There is no need to offer the procedural information in advance, hence the procedural information is delivered to the students together with the learning tasks. Part-task practice automates the recurrent constituent skills which makes performing the learning task easier and hence reduces the cognitive load [21].

2.2 Instructional methods

The learning tasks include induction, which enables students to construct cognitive schemas of how to approach problems in a domain and of how this domain is organized. Supportive information includes elaboration, which integrates new information with cognitive schemas already available in memory. Elaboration and induction are responsible for constructing cognitive schemas. Procedural information includes rule formation and is, in combination with strengthening, which is included in Part-task practice, responsible for rule automation [21].

2.3 Ten Steps to Complex Learning

The Ten Steps to Complex Learning (Ten Steps) approach offers practical guidelines to apply the 4C/ID model [21]. Table 1 shows the relation between the 4C/ID model and the Ten Steps approach.

Table 1: 4C/ID and Ten Steps

4C/ID Components	Ten Steps
Learning Tasks	1. Design Learning Tasks
	2. Design Performance Assessments
	3. Sequence Learning Tasks
Supportive Information	4. Design Supportive Information
	5. Analyze Cognitive Strategies
	6. Analyze Mental Models
Procedural Information	7. Design Procedural Information
	8. Analyze Cognitive rules
	9. Analyze Prerequisite Knowledge
Part-task Practice	10. Design Part-task Practice

3 COURSE DESIGN

3.1 Context

The Amsterdam University of Applied Sciences offers a minor in Mobile Development. As part of this minor, students are taught how to develop an Android app. Students are required to have basic programming skills, but they do not need to have experience in Android app development. Android app development is taught to different levels of students, both part-time and full-time, from different streams, different years and different countries. To cope with this diversity, flexibility is crucial. Flexibility is achieved by using online learning combined with face-to-face learning, also known as blended learning [5, 12].

3.2 4C/ID components

Learning tasks, supportive information and procedural information are always needed to design a course. However, part-task practice is only needed when a high level of automaticity is required [20]. In the course design, part-task practice was omitted because a high level of automaticity is not required in an introductory Android app development course.

The following 4C/ID components were designed using the Ten Steps approach: learning tasks, supportive information and procedural information. These components were used to create a blended course design.

3.3 Learning tasks

The following steps from the Ten Steps approach were performed to design the learning tasks: (1) design learning tasks, (2) design performance assessments and (3) sequence learning tasks. First, the learning tasks need to vary from each other just as they would in the real world. The course design accomplishes this by varying the app category: shopping, games, education, et cetera. Learning tasks are organized in simple to complex task classes. Second, the learning tasks need to be authentic and need to be performed using a real development tool, namely Android Studio.

3.3.1 Task classes. Task classes were designed using conditions to vary the complexity of the task classes. This method is called the simplifying conditions approach [21]. The following conditions were chosen to design the task classes: complexity of user interface, number of activities, and complexity of data layer. These conditions were used to design seven task classes from simple to complex. Table 2 presents these task classes, which gradually increase the cognitive load.

Table 2: Task classes

Task class	User Interface	Screens	Data layer
1	Simple UI	1	POJO
2	Gesture		
3		>1	
4			Local Storage
5			Remote Data Source
6	Fragment		
7	ViewModel		

The conditions were found using a skill hierarchy (Figure 1), which was made by decomposing the skill of Android app development with the aid of several professional Android developers from different companies.

3.3.2 Scaffolding. In the first task class, the students build simple single screen apps, but even the first learning task of the first task class is too difficult for students to perform without any support. Therefore, the students get considerable support in the first

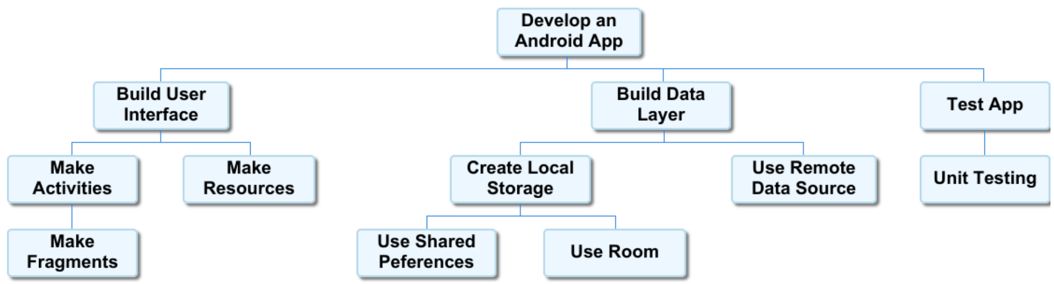

Figure 1: Skill hierarchy

learning task, and this support gradually fades in the learning tasks that follow. The completion strategy was used to design this so-called scaffolding [19, 21].

The completion strategy uses task support to build scaffolding. Task support consists of three elements: goal state, given state and solution. In the Android app development course, the given state is the requirements of the Android app including how the user interface should look (design). The goal state is an Android app that runs without errors and complies with the required functionality. The solution is the steps to get from the given to the goal state.

In the first learning task of each task class, the students receive the given and goal states including a complete solution, which is called a worked-out example. In the second learning task, the students only have access to a partial solution, which they have to complete. This learning task is called a completion task. In the last learning task, the learners receive no task support; they have to find the solution by themselves. This is called a conventional task.

3.4 Supportive information

The step "design supportive information" from the Ten Steps approach was performed to design the supportive information. A modeling example is used to teach students SAPs and is highly effective in terms of learning [3, 6, 18]. A modeling example means that the lecturer performs the SAPs and explains why particular steps or decisions are made. For example, the lecturer shows how to implement local storage. When the lecturer is finished, the students apply the modeling example themselves to promote retention. Each task class includes a modeling example.

To teach students the domain models, students are divided into small groups and are asked to answer guiding questions using resources that were provided. These guiding questions and resources give the students the opportunity to elaborate on the domain models. To further improve elaboration, cognitive feedback that the students receive from the lecturer is included in the course design. Cognitive feedback means that students can compare their own problem solving, reasoning and decision making with those of the lecturer. Performance assessments are used to give cognitive feedback to the students. The skill hierarchy is used to define performance objectives, that is, descriptions of what the students must be able to do after the course. Performance objectives foster reflection and provide the basis for assessing the students.

3.5 Procedural information

The step "design procedural information" from the Ten Steps approach was performed to design the procedural information. Students are given corrective feedback to help them develop rules. Corrective feedback helps students to recognize errors and form rules to prevent them. Android Studio provides corrective feedback using a code analyzer called Lint. Lint analyzes the code and gives feedback about errors and potential improvements.

3.6 Blended learning design

Each designed component or part thereof was categorized as face-to-face or online, for both lecturer and student, resulting in a blended learning design. The modeling example was made part of the face-to-face activities because SAPs in the Android app development domain change so often that a recorded demo would be quickly outdated. The students can apply the modeling example and give answers to questions in small groups using several resources, which are linked to online tutorials or documentation. Face-to-face activities also include performance assessments, which allow the students to receive cognitive feedback. The students perform the learning tasks online individually, which gives them the opportunity to work at their own pace, place and time. The students receive procedural information along with the learning tasks. The students can consult this information when needed. They can also get online guidance from other students and the lecturer by asking questions. In addition, the modeling example is available online and can be consulted. Table 3 shows the blended learning design based on the 4C/ID components.

4 DISCUSSION

This paper illustrates how a blended course in Android app development was designed using the 4C/ID model. This model proves to be very helpful in making a blended course design; the 4C/ID components and their parts can be used to decide if they should be offered online or face-to-face. However, the design of a blended course should do more than just enhance the face-to-face activities with online activities. The two parts should reinforce each other, which would make the course more appealing and effective. Blended learning is about face-to-face interactions and technologically-mediated interactions between students, lecturers and learning resources [1].

Table 3: Blended learning design

	Face-to-Face		Online	
	Lecturer	Students (small group)	Lecturer	Student (individual)
Learning tasks				works on tasks
			gives guidance	gives and receives guidance
Supportive information	performs modeling example	apply modeling example		consults modeling example
		answering questions using resource links		consults resources
	gives cognitive feedback (peer assessment)			
Procedural information				gets corrective feedback (code analyzer)
				consults information

Students who only perform the learning tasks online would not benefit from the advantages of blended learning. The online activities would, in that case, only enhance the face-to-face activities. Thus, the students are divided into small groups during face-to-face activities. The students get acquainted and are more willing to help each other with the learning tasks during the online activities. We argue that this online guidance could further reduce cognitive load compared to just online or face-to-face courses designed with the 4C/ID model. In addition, with blended learning we can engage the students and offer a meaningful learning experience, which improves learning [4].

4.1 Ten Steps approach

As part of the Ten Steps approach, a skill hierarchy was designed, which was valuable in getting an overview of the complex skill of developing an Android app. However, defining this skill hierarchy was time-consuming: many companies were consulted and many iterations were needed. The skill hierarchy was also helpful in finding conditions to define the simple to complex task classes that are necessary to prevent the overload. However, defining these task classes was also a time-consuming process. In general, applying the Ten Steps approach takes time but it is helpful and gives detailed guidelines about how to reduce the load and improve learning. A limitation of the Ten Steps approach is that the process of making a blended learning design is not explicitly guided.

4.2 Future work

Future research will examine how to expand the Ten Steps approach with guidelines for designing a blended course. Research will also be conducted on methods to construct the skill hierarchy and task classes in a more systematic way. Also the cognitive load for the each of resulting components will be studied. As a first step, the Android app development course has been developed, resulting in an online learning environment [1]. Figure 2 shows a screenshot of the online learning environment. In addition, we believe that the Ten Steps approach could also be applied in other fields of computing education.

REFERENCES

[1] Ana-Maria Bliuc, Peter Goodyear, and Robert A. Ellis. 2007. Research focus and methodological choices in studies into students' experiences of blended learning in higher education. *The Internet and Higher Education* 10, 4 (2007), 231–244. DOI: http://dx.doi.org/10.1016/J.IHEDUC.2007.08.001

[2] O. Chen, G. Woolcott, and J. Sweller. 2017. Using cognitive load theory to structure computer-based learning including MOOCs. *Journal of Computer Assisted Learning* 33, 4 (8 2017), 293–305. DOI: http://dx.doi.org/10.1111/jcal.12188

[3] J. Frerejean, J. L.H. van Strien, P. A. Kirschner, and S. Brand-Gruwel. 2018. Effects of a modelling example for teaching information problem solving skills. *Journal of Computer Assisted Learning* (2018). DOI: http://dx.doi.org/10.1111/jcal.12276

[4] D. R. (D. Randy) Garrison. 2017. *E-learning in the 21st century : a community of inquiry framework for research and practice.* Routledge, New York. 202 pages.

[5] Linda De George-Walker and Mary Keeffe. 2010. Self-determined blended learning: a case study of blended learning design. *Higher Education Research & Development* 29, 1 (2 2010), 1–13. DOI: http://dx.doi.org/10.1080/07294360903277380

[6] Vincent Hoogerheide, Sofie M.M. Loyens, and Tamara van Gog. 2014. Comparing the effects of worked examples and modeling examples on learning. *Computers in Human Behavior* 41 (12 2014), 80–91. DOI: http://dx.doi.org/10.1016/J.CHB.2014.09.013

[1] http://www.android-development.app/

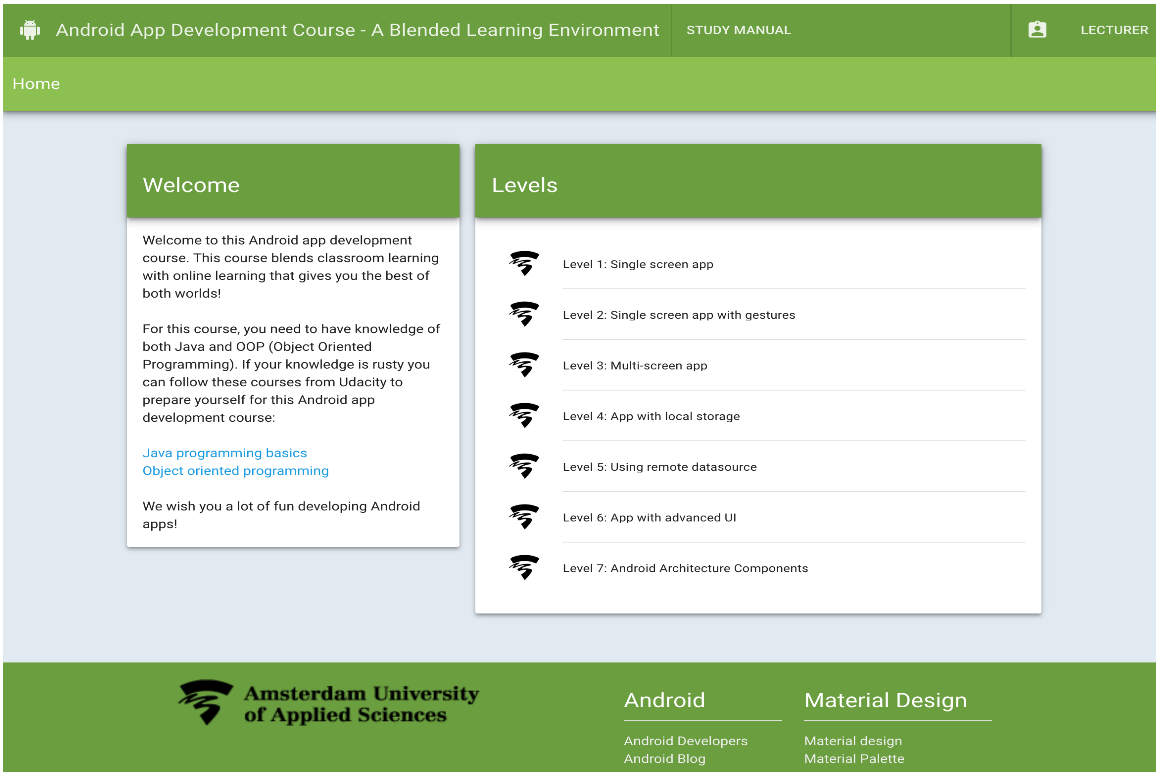

Figure 2: Online learning environment

[7] Charlotte Larmuseau, Jan Elen, and Fien Depaepe. 2018. The influence of students' cognitive and motivational characteristics on students' use of a 4C/ID-based online learning environment and their learning gain. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge - LAK '18*. ACM Press, New York, New York, USA, 171–180. DOI : http://dx.doi.org/10.1145/3170358.3170363

[8] Rainalee Mason. 2012. *Designing introductory programming courses : the role of cognitive load*. Ph.D. Dissertation. Southern Cross University. https://epubs.scu.edu.au/theses/307

[9] Rainalee Mason, Carolyn Seton, and Graham Cooper. 2016. Applying cognitive load theory to the redesign of a conventional database systems course. *Computer Science Education* 26, 1 (1 2016), 68–87. DOI : http://dx.doi.org/10.1080/08993408.2016.1160597

[10] M. David Merrill. 2002. First principles of instruction. *Educational Technology Research and Development* 50, 3 (2002), 43–59. DOI : http://dx.doi.org/10.1007/BF02505024

[11] Briana B. Morrison, Lauren E. Margulieux, Barbara Ericson, and Mark Guzdial. 2016. Subgoals Help Students Solve Parsons Problems. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16*. ACM Press, New York, New York, USA, 42–47. DOI : http://dx.doi.org/10.1145/2839509.2844617

[12] Ingrid A.E. Spanjers, Karen D. Könings, Jimmie Leppink, DaniÃńlle M.L. Verstegen, Nynke de Jong, Katarzyna Czabanowska, and Jeroen J.G. van Merriënboer. 2015. The promised land of blended learning: Quizzes as a moderator. *Educational Research Review* 15 (6 2015), 59–74. DOI : http://dx.doi.org/10.1016/J.EDUREV.2015.05.001

[13] Astrid Pratidina Susilo, Jeroen van Merriënboer, Jan van Dalen, Mora Claramita, and Albert Scherpbier. 2013. From Lecture to Learning Tasks: Use of the 4C/ID Model in a Communication Skills Course in a Continuing Professional Education Context. *The Journal of Continuing Education in Nursing* 44, 6 (2013), 278–284. DOI : http://dx.doi.org/10.3928/00220124-20130501-78

[14] John Sweller. 2010. Element Interactivity and Intrinsic, Extraneous, and Germane Cognitive Load. *Educ Psychol Rev* 22 (2010), 123–138. DOI : http://dx.doi.org/10.1007/s10648-010-9128-5

[15] John Sweller. 2011. Cognitive Load Theory. *Psychology of Learning and Motivation* 55 (1 2011), 37–76. DOI : http://dx.doi.org/10.1016/B978-0-12-387691-1.00002-8

[16] John Sweller and John. 2016. Cognitive Load Theory and Computer Science Education. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16*. ACM Press, New York, New York, USA, 1–1. DOI : http://dx.doi.org/10.1145/2839509.2844549

[17] Ellen te Pas, Margreet Wieringa-de Waard, Bernadette Snijders Blok, Henny Pouw, and Nynke van Dijk. 2016. Didactic and technical considerations when developing e-learning and CME. *Education and Information Technologies* 21, 5 (9 2016), 991–1005. DOI : http://dx.doi.org/10.1007/s10639-014-9364-2

[18] Tamara Van Gog, Fred Paas, and XX. 2004. Process-Oriented Worked Examples: Improving Transfer Performance Through Enhanced Understanding *. *Instructional Science* 32 (2004), 83–98. https://search.proquest.com/docview/740302287?pq-origsite=gscholar

[19] Jeroen J. G. van Merriënboer and Paul Ayres. 2005. Research on cognitive load theory and its design implications for e-learning. *Educational Technology Research and Development* 53, 3 (9 2005), 5–13. DOI : http://dx.doi.org/10.1007/BF02504793

[20] Jeroen J. G. van Merriënboer, Richard E. Clark, and Marcel B. M. de Croock. 2002. Blueprints for Complex Learning: The 4C/ID-Model. (2002). DOI : http://dx.doi.org/10.2307/30221150

[21] Jeroen J. G. van Merriënboer and Paul Arthur Kirschner. 2018. *Ten steps to complex learning : a systematic approach to four-component instructional design*. Routledge, New York. 399 pages.

[22] Jeroen J. G. van Merriënboer, Paul A. Kirschner, and Liesbeth Kester. 2003. Taking the Load Off a Learner's Mind: Instructional Design for Complex Learning. *Educational Psychologist* 38, 1 (2003), 5–13. DOI : http://dx.doi.org/10.1207/S15326985EP3801{_}2

[23] Mieke Vandewaetere, Dominique Manhaeve, Bert Aertgeerts, Geraldine Clarebout, Jeroen J G Van Merriënboer, and Ann Roex. 2015. 4C/ID in medical education: How to design an educational program based on whole-task learning: AMEE Guide No. 93. *Medical Teacher* 37, 1 (2015), 4–20. DOI : http://dx.doi.org/10.3109/0142159X.2014.928407