# Towards an Online Service for Learning Computational Thinking using Scientific Workflows

Aravind Mohan, Shiyong Lu, Ke Zhang
Wayne State University
Detroit, MI, USA
{amohan, shiyong, bb2145}@wayne.edu

*Abstract*—**Computational thinking has recently been widely recognized as a fundamental skill that should be cultivated for everyone and in every field. Although there is an increasing interest in research in teaching and learning computational thinking in recent years, an engaging, effective online learning system is yet to be built for teaching, learning, and applying computational thinking online. To this end, we have developed DATAVIEW, an online social learning system for learning computational thinking concepts and skills online. The main contributions of this paper are: 1) we developed a new, effective online learning model for computational thinking based on our previous widely-applied R2D2 model. A signature characteristic of this new model is being interactive and learner-centered, thus "i" is carried out through the entire learning experiences, which provides teaching facilities to instructors and interactional tools among instructors and learners; 2) we implemented, validated, and refined iR2D2 in our DATAVIEW computational thinking online service; 3) We propose to use the DBR (design-based-research) approach to study the relationship between technology and teaching in the context of computational thinking, generating research results and findings applicable to online teaching in other domains as well.**

***Keywords- computational thinking; eLearning; scientific workflow management systems; DATAVIEW;***

## I. INTRODUCTION

Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be carried out effectively by an information-processing agent [1]. In recent years, computational thinking has been widely recognized as a fundamental skill that should be considered as important as the traditional 3R (Reading, Writing, Arithmetic) and cultivated for everyone and in every field [19, 27]. First, long before computers were invented, human had already started to recognize that many problems could be solved computationally. They also used computational thinking skills to solve many computational problems, either by paper and pencils, abacus, or just by bare mental minds, for example, the puzzle of "Tower of Hanoi" [1] and the travelling salesman problem [2]. After all, humans compute and they are the first-generation "computers". Second, as the printing press brought about 3R's in education, computers are bringing about computational thinking people of all ages. While computers mechanize and automate many computations and thus liberate humans from many of the routine computational activities, they also create an era of information-driven society, in which human needs to carry out many information-oriented tasks on a daily basis. Finally, today, computational thinking is not just about computers, just as music is not just about pianos. Computational thinking is not just for computer scientists, it is critical for all of us to contribute and succeed in every field, and to carry our daily lives more efficiently and effectively.

Although computational thinking is named for its extensive use of computer science techniques, it is not just computer programming. Like reading and writing, computational thinking is a fundamental skill that can be used by everyone in the world in every field [7]. Computational thinking includes the following characteristics: 1) analyzing and logically organizing data; 2) data modeling, data abstractions, and simulations; 3) formulating problems such that computers may assist; 4) identifying, testing, and implementing possible solutions; 5) automating solutions via algorithmic thinking; 6) generalizing and applying this process to other problems. Many universities such as Carnegie Mellon University, and big companies such as Microsoft and Google have invested to promote computational thinking through additional curriculum to support student learning. There is an increasing interest in education research on teaching and learning computational thinking in recent years. Repenning et al. [8] proposed to integrate the scalable game design in high and middle school. In [9], the author introduced the computational thinking to the biology class. However, a model for effectively teaching computational thinking online is still lacking. In this paper we address the limitation by proposing our new "iR2D2" model that provides a platform to teach computational thinking online in an interactive and learner-centered manner.

Scientific workflow management systems provides a visual learning tool for computational thinking in which the learners can visually manipulate various fundamental concepts of computing, such as functions and their composition like a craft. As a result, a scientific workflow tool transforms the science of computing into an art of composition. In the meanwhile, the tool provides runtime execution and verification to make sense of the composition: a learner can execute a workflow to verify and see what their "craft" can compute. As indicated above, computational thinking is essentially a method to incorporate the computational problem solving methods into different disciplines. On the other hand, scientific workflow management systems integrate a comprehensive set of critical computational thinking concepts and provide a platform to solve the computational problems. Today's advancement in eScience has made scientific workflows not only very popular from data analytics point of DATAVIEW, but also have become an appealing tool for scientists, who might lack IT skills, to apply computational thinking to solve many scientific problems computationally [3, 4].

978-1-4673-7281-7/15 $31.00 © 2015 IEEE
DOI 10.1109/SCC.2015.54

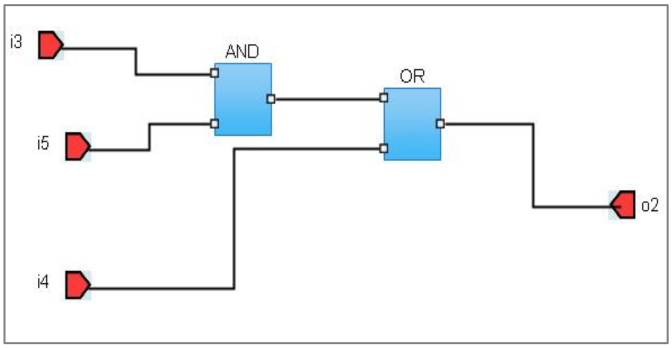


**Fig. 1. (a) A sample boolean logic workflow in DATAVIEW.**

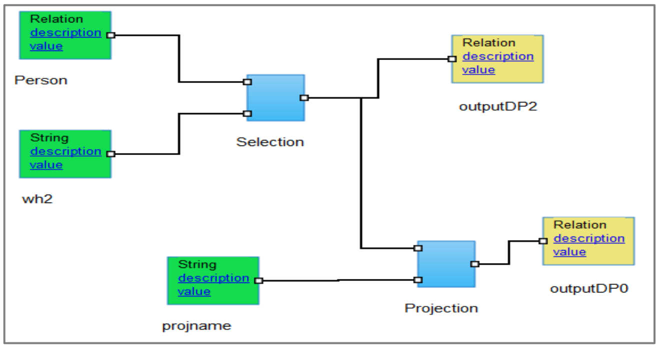


**Fig. 1. (b) A sample relational algebra workflow in DATAVIEW.**

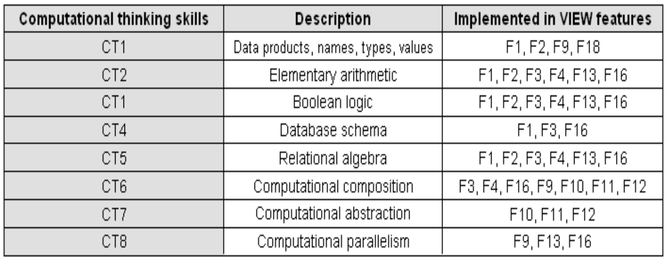
The visual and graph-style workflows, can naturally capture and represent the complex data analysis pipelines formulated in the minds of data scientists.

On the other hand, scientific workflows can also be used as a tool to teach computational thinking as a fundamental skill to undergraduate students, who are not necessarily majoring in computer science. In Fig 1, we first use a motivating example to see what a scientific workflow looks like and to appreciate the appeal of the scientific workflow approach from the perspective of instructors who teach computational thinking skills to undergraduate students.

Although the advancement of the eLearning technologies has a good reach among the student learning curriculum in various institutions, it lacks a learner centered interactive model and a tool to help the students in the whole learning process. Towards this end, in this paper we propose to extend the existing R2D2 model [27] to iR2D2 as an online teaching model for computational thinking. The working hypothesis is that the proposed iR2D2 model will maximize the knowledge transfer flow from instructors to learners while benefiting from the state-of-the-art advantages of today's online teaching: high accessibility for anytime, anywhere, and everybody. Our proposed method has the following three objectives: 1) we developed an effective online teaching model for computational thinking based on the well-applied R2D2 model. A signature characteristic of this newly extended model is being interactive and learner-centered, thus "i" is carried out through the whole learning experiences, which provides teaching facilities to the instructors and interactional tools among the instructors and the learners. 2) we implemented, validated, and refined iR2D2 in our DATAVIEW computational thinking online service. With this new model, DATAVIEW will provide a platform to teaching classes online, in which an instructor can demonstrate some computational thinking activities and enable learners to join and observe those activities in real-time. 3) We propose to use the DBR (design-based-research) approach to study the relationship between technology and teaching in the context of computational thinking, producing general research results that will be applicable to other online teaching domains as well. The DBR approach has the appealing feature such that the process of design, implementation, and evaluation will be repeated iteratively until intervention is effective.

## II. System Overview

We have developed our online learning system called DATAVIEW (http://dataview.org) that shows the feasibility of learning computational thinking through scientific workflows – data pipelines that visually chain different computational thinking modules together. As a Web-based service, no installation, upgrade, and maintenance are required for end users. One can use DATAVIEW simply via an Internet browser, such as Firefox and IE. In addition to elementary arithmetic, DATAVIEW has already supported the learning of Boolean algebra and relational algebra as key computational thinking concepts. In DATAVIEW, Boolean algebra expressions are supported as visual scientific workflows. Boolean algebra manipulates truth values (true and false) with their logical operations (AND, OR, NOT). It is the basis for logic, programming, and knowledge management. As shown in Fig 1a, an expression $o2 = (i3 \wedge i5) \vee i4$ can be visually shown as a scientific workflow. A student in the Digital Systems course can add arbitrary inputs and Boolean operators (AND, OR, NOT, NAND, NOR, XOR, NXOR, IMPLY, and EQUIV) and then chain them together into a workflow. The workflow can then be executed to calculate the final output, which can be verified against the mental calculation result of the student. *Relational algebra* is the basis for modern database management systems. In DATAVIEW, each database query is implemented as a visual scientific workflow that represents a relational algebra expression of the query. Tables in the database can be easily browsed in a Microsoft spreadsheet fashion. A learner can easily drag and drop multiple tables into the design panel and then chain them with relational algebra operators (SELECT, PROJECT, INSERTSECTION, UNION, MINUS, RENAME, JOIN, etc.) into a workflow. A learner can then run the workflow (query) and the query result will then be powered to the learner as a spreadsheet-style table. As an example, we show in Fig 1b, a scientific workflow created by the student in Database Systems 1 course at Wayne State University that calculates relational algebra expression $\pi_{\text{Name}}$ ($\sigma_{\text{Hobby/Stamps'}}$ (Person)), which takes an input table Person (Id, Name,



| Computational thinking skills | Description | Implemented in VIEW features |
|---|---|---|
| CT1 | Data products, names, types, values | F1, F2, F9, F18 |
| CT2 | Elementary arithmetic | F1, F2, F3, F4, F13, F16 |
| CT1 | Boolean logic | F1, F2, F3, F4, F13, F16 |
| CT4 | Database schema | F1, F3, F16 |
| CT5 | Relational algebra | F1, F2, F3, F4, F13, F16 |
| CT6 | Computational composition | F3, F4, F16, F9, F10, F11, F12 |
| CT7 | Computational abstraction | F10, F11, F12 |
| CT8 | Computational parallelism | F9, F13, F16 |

**Fig. 2. (a) A list of core computational thinking skills.**

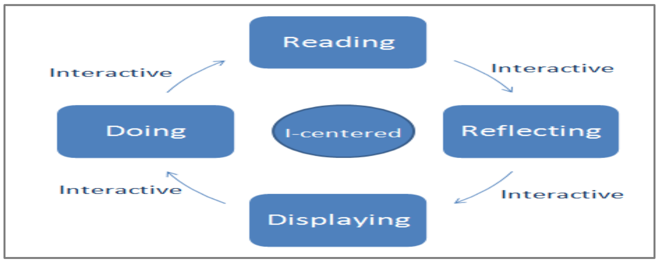


**Fig. 2. (b) *i*R2D2 Model.**

Address, Hobby), selects those persons whose hobbies are “stamps”, and then prints out their names.

In this workflow, green boxes represent inputs, yellow boxes represent outputs, and blue boxes represent computational procedures.

More complex workflows (data pipelines) are formed by connecting the output of one computational box to the input of another computational box using a data link. In the workflow, the Selection box has two inputs: the Person table, and the Selection condition “Hobby=’stamps’”, while the Projection box also has two inputs: the output table of the Selection box, and the Projection attribute list “Name”. Finally, the output of the Selection box and the output of the Projection box are each linked to a result container, which is used to contain and display the corresponding calculation result. In the workflow, the output of the Selection box feeds as both the input of the Projection box and the input for the outputDP2 box. Therefore, common sub expressions or values are easy to represent in a workflow diagram. Of course, the power of scientific workflows goes beyond calculating relational algebra and relational algebra expressions that process structured data. In its most general form, a box can be an arbitrary computational procedure written in an arbitrary programming language, and a data item can be a structure table or a spreadsheet, but can also be a semi-structured document, or an unstructured dataset file. In DATAVIEW, we support such general forms of workflows and provide a framework that not only enables a student to design workflows, but also provides a runtime system to execute her scientific workflows on our dedicated DATAVIEW server or on the Cloud. As an ongoing work, we are improving our DATAVIEW system to be an online social learning and teaching tool for computational thinking skills. DATAVIEW features a service-oriented architecture consisting of six loosely coupled subsystems:

- A Web Portal for DATAVIEW. Via the Web Portal, a scientist can design and modify workflows, present data products and provenance, and manage subsystems.
- A Workflow Engine to schedule, execute, and manage workflows.
- A Workflow Monitor to display system status and handle exceptions.
- A Task Manager to schedule, execute, and manage tasks (i.e., primitive workflows).
- A Provenance Manager to store and query workflow provenance.
- A Data Product Manager to store and manage datasets.

Below we have summarized the most distinguishing features of DATAVIEW system:

1. F1: DATAVIEW features the first uniform workflow model [5, 30], in which workflows are the only building blocks. In DATAVIEW, tasks are primitive workflows and all workflow constructs do not discriminate workflows from tasks. Such a model greatly simplifies workflow design, in which a workflow designer only needs to compose complex workflows from simpler ones without the need to first encapsulate workflows to tasks or vice versa during the composition process. Such simplification can greatly decrease learning curves for novices.
2. F2: DATAVIEW has a powerful workflow composition power in which workflow constructs are fully compositional one another with arbitrary levels [5]. This often results in DATAVIEW workflows that are more concise and efficient to execute, which can be hard to model in other workflow systems.
3. F3: DATAVIEW features a pure dataflow-based workflow language SWL [5], including the dataflow counterparts of control flow-style constructs, such as conditional and loop. Existing workflow languages often require both control flow and dataflow constructs, resulting in complex or even obscure semantics and non-trivial workflow design.
4. F4: DATAVIEW supports the cloud Map Reduce programming model not only at the job level, but also at the workflow level [25]. Therefore, one can apply the Map and Reduce constructs on an arbitrary workflow with arbitrary number of times. As a result, DATAVIEW can process nested lists of data products in parallel using multiple runs of a workflow.
5. F5: DATAVIEW features a collectional data model [6] that supports not only traditional primitive data types, such as integer, float, double, boolean, char, string, but also files, relations, hierarchical collections (hierarchical key-value pairs) to support parallel processing of data collections.

6. F6: DATAVIEW supports a high-level graph-based provenance query language OPQL. In most cases, users can formulate lineage queries easily without the need of writing recursive queries or knowing the underlying database schema [3].
7. F7: DATAVIEW features the first service-oriented architecture that conforms to the reference architecture for scientific workflow management systems (SWFMSs) [3].

## III. Model

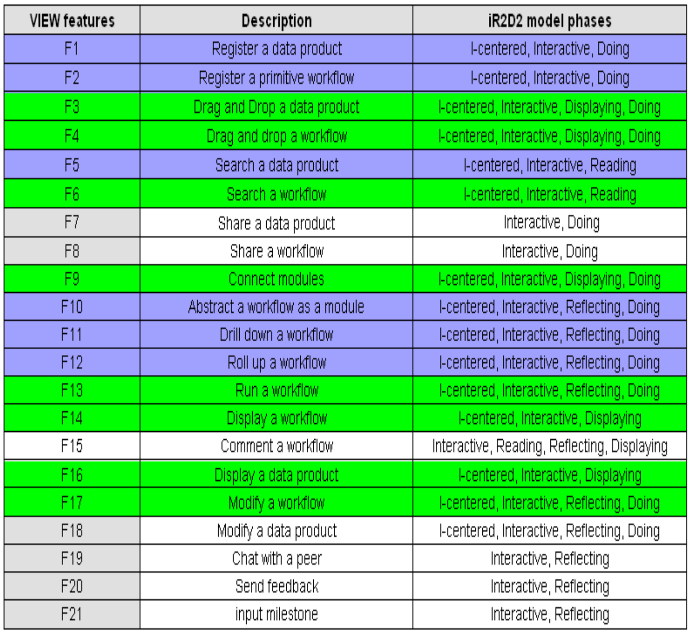
The original R2D2 model (Read, Reflect, Display, and Do) [19, 27] was designed specifically for addressing varied student learning preferences, diverse backgrounds, and experiences, and generational differences concerning learning technologies. The model is particularly helpful in the trend of moving from lecture-dominated classes to learner-centered learning. We propose to extend R2D2 to ***i*R2D2** as shown in Fig 2b**,** highlighting interactiveness throughout the learning environments, and to tailor ***i*R2D2** to the context of teaching, learning, and applying computational thinking skills. The core computational thinking skills are listed in Fig 2a.

The goal of *i*R2D2 is to meet the following challenges of teaching and learning computational thinking skills online:

- **Challenge 1, more responsive:** Utilizing the automatic features, the online learning system will provide appropriate feedback, comments, suggestions, and other responses to learners immediately, based on their online learning behaviors and performances.
- **Challenge 2, more engaging:** The online learning system will help engage learners by connecting learners with similar interests, backgrounds, skill levels, etc., based on learner profiles.
- **Challenge 3, more dynamic:** As a learner progresses in the learning process, how can the learning system dynamically align her with more advanced content and connect her to more advanced learning groups? How can the system dynamically evolve with the society's learning need and learning styles?

We initially target the following four computational thinking learning outcomes:

1. **Learning outcome 1**: students will have the knowledge of computational thinking concepts to recognize that a problem can be solved computationally;
2. **Learning outcome 2:** students will have the ability to formulate a problem in a form that is solvable computationally;
3. **Learning outcome 3:** students will have the capability to come up with a computational solution to a problem;
4. **Learning outcome 4:** students will have the skills to apply computational thinking to improve the efficiency of their daily lives.



| VIEW features | Description | iR2D2 model phases |
|---|---|---|
| F1 | Register a data product | I-centered, Interactive, Doing |
| F2 | Register a primitive workflow | I-centered, Interactive, Doing |
| F3 | Drag and Drop a data product | I-centered, Interactive, Displaying, Doing |
| F4 | Drag and drop a workflow | I-centered, Interactive, Displaying, Doing |
| F5 | Search a data product | I-centered, Interactive, Reading |
| F6 | Search a workflow | I-centered, Interactive, Reading |
| F7 | Share a data product | Interactive, Doing |
| F8 | Share a workflow | Interactive, Doing |
| F9 | Connect modules | I-centered, Interactive, Displaying, Doing |
| F10 | Abstract a workflow as a module | I-centered, Interactive, Reflecting, Doing |
| F11 | Drill down a workflow | I-centered, Interactive, Reflecting, Doing |
| F12 | Roll up a workflow | I-centered, Interactive, Reflecting, Doing |
| F13 | Run a workflow | I-centered, Interactive, Reflecting, Doing |
| F14 | Display a workflow | I-centered, Interactive, Displaying |
| F15 | Comment a workflow | Interactive, Reading, Reflecting, Displaying |
| F16 | Display a data product | I-centered, Interactive, Displaying |
| F17 | Modify a workflow | I-centered, Interactive, Reflecting, Doing |
| F18 | Modify a data product | I-centered, Interactive, Reflecting, Doing |
| F19 | Chat with a peer | Interactive, Reflecting |
| F20 | Send feedback | Interactive, Reflecting |
| F21 | input milestone | Interactive, Reflecting |

**Fig. 3. Computational thinking features in DATAVIEW.**

Although we target the above mentioned computational thinking learning outcomes, there is more than just four learning outcomes that a learner can be benefited from computational thinking skills. In our literature survey, we found that the computational thinking is very new in the research but the future of computational thinking skills is very bright as many companies like Microsoft and Google already started to invest money in this direction as they want humans to think like machines, by doing so humans can understand machines in a better way and hence solve many computational problems in an efficient manner. These outcomes not only emphasize the understanding of computational thinking concepts, but also apply them to solve real-life problems. The audience not only includes computer science major students, but also non-computer science learners. These skills, when applied, can not only improve computer programming, but also improve the efficiency of daily activities, such as scheduling, prioritization, multitasking, abstraction, and automation.

The proposed *i*R2D2 model illustrated in Fig 2b has four phases plus intensive interactiveness throughout all phases:

1. **Interactiveness & user-centeredness:** This signature feature of this model focuses on different types of interactions necessary for effective and engaging learning: (a) interactions between learners and the content being learned, (b) interactions among learners in the online community, and learners with self-selected and/or system matched peer mentors, (c) learners with automatic, dynamic feedback provided by DATAVIEW. Examples of activities

include, but are not limited to, online chatting, messaging, product or process sharing, commenting, and the like. This highly interactive, learner-centered model aims to engage new learners, retain existing learners, and facilitate knowledge sharing, knowledge construction, critical thinking, and exchanges among community members. This feature is critical for the new generations of learners [20], and social and collaborative learners.

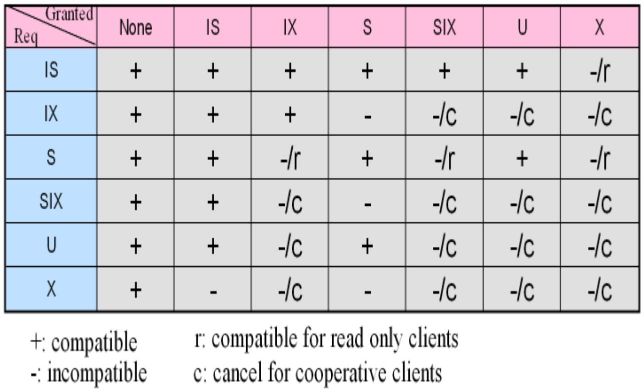
2. **Reading:** This phase mainly focuses on methods and tools to help learners obtain knowledge through tasks such as online readings, e-learning explorations, and listening to audio lectures.
3. **Reflecting:** This phase focuses on reflective activities such as online blogs, reflective writing, and self-check or review activities and self-testing examinations.
4. **Displaying:** This phase focuses on the visual representations of the content with activities such as virtual tours, timelines, animations, and concept maps.
5. **Doing:** This phase focuses on what the learners can do with the content in terms of hands-on activities, including executions, simulations, scenarios, authentic cases, problem solving, and more.

In addition to the above features, our system also contains a detailed documentation with various videos and slides to prepare and orient novice learners to get started, walk them through with a set of pre-designed **step-by-step exercises**. For the phase Doing, we will prepare a set of workflow examples in the system, with which a learner can immediately interact by editing, trying, and running. Lowering the learning curve and smoothing the progression of levels will be the key to integrate all phases of our iR2D2 model.

## IV. IMPLEMENTATION

While individual creation and execution of scientific workflows develops innovation and creativity in personal computational thinking, many studies show that a collaborative environment may greatly simulate learning interests, encourage knowledge sharing and construction, and improve learner engagement. Our proposed iR2D2 model is implemented in DATAVIEW, an online social learning system for computational thinking. In our DATAVIEW system, we focus on supporting teaching and learning the core computational thinking skills listed in Fig 2a. These skills are supported by a set of DATAVIEW system features as listed in Fig 3 Features in green have already been fully implemented; features in blue have been partially implemented; while non-colored features will be implemented in the future. All features are designed and implemented using our proposed iR2D2 model using the mapping provided in Fig 2a and Fig 3.

.



| Granted<br>Req | None | IS | IX | S | SIX | U | X |
|---|---|---|---|---|---|---|---|
| IS | + | + | + | + | + | + | -/r |
| IX | + | + | + | - | -/c | -/c | -/c |
| S | + | + | -/r | + | -/r | + | -/r |
| SIX | + | + | -/c | - | -/c | -/c | -/c |
| U | + | + | -/c | + | -/c | -/c | -/c |
| X | + | - | -/c | - | -/c | -/c | -/c |

+: compatible r: compatible for read only clients
-: incompatible c: cancel for cooperative clients

**Fig. 4. Locking based coordination matrix.**

One of the key features that we implemented is "workflow sharing" (DATAVIEW feature F8), with which an instructor can share an example workflow with a class of students. Workflow sharing will also be supported between students so that they not only learn from the instructor but also from each other. There are two modes of sharing: 1) **public sharing**, in which a user will share a workflow with all the users of DATAVIEW; or 2) **private sharing**, in which a user will share a workflow with another individual user or another group of individual users. If a workflow is shared, then all participants of the workflow can edit and run the workflow, and if desirable, they can share it with other users. In the meanwhile, an instructor can also create a virtual session in which sharing is prohibited among students and is only supported between the instructor and students. Such a non-sharing session can be used for the evaluation of students, so that they can perform their learning tasks independently.

While individual creation and execution of scientific workflows develops innovation and creativity in personal computational thinking, many studies show that a collaborative environment may greatly simulate learning interests, encourage knowledge sharing and construction, improve learner engagement and academic achievements [16, 17, 18, 19, 21]. To support collaborative learning and doing, we developed our collaboration and coordination mechanisms so that multiple users can edit and modify a common workflow simultaneously. In this mode, learners will usually work on the same building blocks (sub-workflow, tasks, and data channels) simultaneously. Interdependencies among building blocks add more complexity to this problem.

In such an environment, learners have to trust that any conflict will be detected properly and resolved cooperatively. Our survey revealed that lacking coordination often leads to duplicated work or conflicting actions [29]. How to coordinate among learners is extremely important for scientists to trust a technology-based collaboration system. Existing coordination schemes, however, are not sufficient for workflow

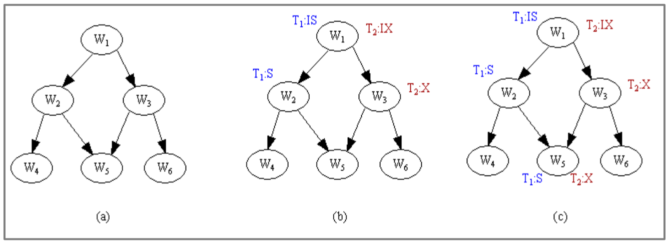


**Fig. 5. Workflow tree structure.**

co-design for two important reasons: 1) they are mainly for non-cooperative transactions, thus lacking cooperative primitives for conflict resolution; and 2) they are not oriented for workflows with arbitrary-level hierarchical structures and possible sharing of sub-workflows. In our system, we support a trust-oriented coordination mechanism, towards improving system responsiveness and group awareness (knowing what learners have done and are expected to do) in a virtual world. This is achieved by system-level support of automatic conflict detection and resolution via human cooperation. The key insight is that via coordination, many conflicts can be either avoided or resolved by collaborators, who, based on trust on the system and peers, can control the pace of their interactions with the system and selection of alternative benevolent actions when conflicts do occur. In Fig 4, we illustrate our idea of a locking-based coordination matrix, in which IS, IX, SIX represent intent shared, exclusive and intent shared exclusive locks, respectively; S, X, U represent shared, exclusive, and update locks, respectively; each entry represents the compatibility between granted lock and requesting lock, or provides a cooperation action.

Our scheme provides several major advantages over traditional concurrency control-based coordination in several significant ways. First, when a write-read conflict occurs, a scientist may still proceed if she is willing to enter a read-only mode (instead of waiting until the write-lock is released). Second, when a write-write conflict occurs, a scientist can choose to wait, or cancel the current update operation and try to update again at a later time. Finally, an option of notification can be issued to synchronize among collaborators. Based on such a coordination matrix, we design and implement various locking-based coordination algorithms. For example, to lock a workflow, the system will first check if the requested lock already exists in the granted lock list of the workflow. If not found, the algorithm will check the compatibility between the requested lock and the existing locks. This process will be performed recursively along parent workflows, if existed. If not compatible, three possibilities exist. 1) If the coordination matrix entry is "-", the requested lock will be put in the waiting list of the workflow. 2) If the compatibility matrix entry is "-/r", the algorithm will prompt the user to choose to either wait or enter into a read-only session. 3) If the compatibility matrix entry is "-/c", the algorithm will prompt the user to choose either to wait or cancel the current operation. As an ongoing work, we are currently designing algorithms for workflow lock release, construct locking, construct lock release, and so on based on the above proposed coordination matrix. We assume that the workflow under construction has a tree-like structure, that is, each sub-workflow has one single parent. In practice, however, a sub workflow may be shared by different parts of a workflow. As shown in Fig 5.(a), sub workflow W5 is a shared by sub workflows W2 and W3. Fig 5.(b) supposes that one learner (T1) attempts to read W2 (IS lock on W1 and S lock on W2), and another learner (T2) intends to write on W3 (IX lock on W1 and X lock on W3). As a result, T1 can read W2 including W5, and T2 can write W3 including W5. Such a read and write conflict, shown in Fig 5(c), however, cannot be detected. One possible solution is to require that, a lock on a workflow can only be granted if all its descendant sub workflows can be granted the same lock. In our example, the read/write conflict can be detected at W5. This approach, however, introduces additional overhead (more locks) for shared workflows. We will investigate more comprehensive locking problems and solutions, by balancing between the number of locks and the performance of the coordination scheme. Another feature that is implemented in DATAVIEW is "comment a workflow" (DATAVIEW feature F15). After a learner creates a workflow, DATAVIEW will allow other users to comment on the workflow. This is a very important feature of interactions and discussions among users. Comments for a workflow include descriptions, questioning and answers, suggesting alternative ways, criticizing, etc.

Finally, we also implemented the social tagging functionality so that a learner can add tags to a workflow or a data product. These tags serve not only to group workflows and data products as a form of categorization, but also to help find items in the future by other learners [28]. We will also develop mentor recommendation mechanisms based on the interests of two learners u and v, using their respective tag vocabularies $T_u$ and $T_v$. The interest similarity between two users u and v is characterized by min(sim(u, v), sim(v, u)) where sim(u, v) is defined as:

$$sim(u,v) = \frac{\sum_{t_1 \in T_u, t_2 \in T_v} sim(t_1, t_2)}{|T_u|}$$

Here, sim($t_1$, $t_2$) denote the Leacock-Chodorow similarity between tags $t_1$ and $t_2$ and $|T_u|$ denote the size of vocabulary $T_u$. These new features, including those go beyond Fig 3, will be then used to validate and refine our proposed iR2D2 model.

## V. An Overview on DBR Approach

In this paper, we propose to use the DBR (design-based research) approach [22, 23, 24] to study the relationships between technology and teaching targeted for computational thinking learners and to evaluate the effectiveness of our proposed iR2D2 model. The DBR approach allows us to identify if these features are indeed helpful in helping the learning of computational thinking, and if any, what additional enhancements can be made. As a preliminary research, we are planning to conduct constant formative evaluations throughout the design, development and implementation processes as per DBR general guidelines, while validating and refining the iR2D2 model within the DATAVIEW system. We expect that our study will help answer three fundamental questions: 1) Does DATAVIEW help instructors to effectively teach computational thinking skills in the online learning system? 2) Do the functions provided by DATAVIEW support a rich, enjoyable user experience for both instructors and learners? If so, how? 3) What additional features and functions are favorable or needed to further promote the teaching and learning of computational thinking skills in an online social learning community?

In our research efforts, pre-tests and post-tests will be implemented as a future work to assess learning outcomes and thus to identify the specific learning outcomes in the DATAVIEW experiences. Qualitative data will be collected through online survey, semi-structured interviews, and selected focus groups to help understand how the iR2D2 model as implemented in the DATAVIEW system may have contributed to the learning outcomes, as well as motivation, engagement, and community building and retention of learning.

Our evaluation will focus on the learning outcomes listed in section 3, and will be based on the data that we will collect from our home departments and partnering universities and beyond, as DATAVIEW reaches out to more user groups. The goal is to provide insights regarding the effectiveness of our learning tool and to increase our understanding of the relationship between learning and technology in the context of learning computational thinking, and to inform the design of such technology. We will collect a variety of quantitative and qualitative evaluation data, including task-performance measures (e.g., graded lab reports) to evaluate computational thinking skills of the students before and after the study period. In addition, we will collect post questionnaire data to measure the participants' opinions regarding the adoptability of DATAVIEW as well as understanding of their learning experiences. We will also conduct post interviews to understand the participants' perceptions of learning computational thinking skills using DATAVIEW. In summary, we intend to study if the learning system is effective, decreases learning difficulty, and facilitates the adoption of our tool for learning computational thinking skills.

## VI. Related Work

In the past two decades, workflow technology has been used successfully in many enterprises for business process automation and reengineering [10]. According to the Workflow Management Coalition [11], a workflow is "the computerized facilitation or automation of a business process, in whole or part, and concerned with the automation of procedures where documents, information, or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal [12]." Recently, workflow technology has been increasingly used in managing and delivering e-learning services more efficiently and effectively [13, 14]. Workflow technology allows building e-learning systems that offer the right tasks at the right point of time to the right person along with relevant resources needed to perform these tasks.

ShareFast is a workflow-based e-learning system in the domain of design engineering education [32]. Applying Bruner's theory [15], ShareFast uses workflows to provide the whole picture of the learning process and tasks to provide the details of individual learning steps. The system automatically tracks a learner's behavior. By doing so, the captured information is used by the instructor to improve learning materials, which can shorten students' learning duration. The University of Southern Queensland uses workflow technology to model the whole learning process as four sub-workflows: teaching sub-workflow, leaning sub-workflow, admin sub-workflow, and infrastructure sub-workflow. A case study has been conducted to show that users of the e-learning system have improved their learning performance than on-campus and traditional distance students. Flex-eL [13] is a flexible e-learning environment built upon the workflow technology. The system enables teachers to design and develop process-centric courses and monitor students' progress, and students to learn at their own pace while following the guidelines and checkpoints as part of the course processes designed by teachers. Flex-eL provides flexible learning pathways and might help the realization of the virtual university vision. However, none of this work focuses on teaching computational thinking skills. Moreover, existing workflow-based e-learning systems u*se* *control-flow* oriented workflow structures to enhance e-learning experiences and performance, these control-flow oriented workflows are not suitable for teaching "computational thinking" skills as they cannot model data flows and functions explicitly. Our proposed DATAVIEW system uses *dataflow* oriented scientific workflows to represent "computational thinking" concepts. In contrast to existing systems, workflows are part of the infrastructure itself, and cannot be changed by learners, while in DATAVIEW, scientific workflows are learning as well as teaching materials of "computational thinking" concepts, and thus can created, modified, executed, and shared by learners and instructors to transfer their computational thinking skills. To our best knowledge, this work is the first one to address the lack of a teaching model for computational thinking and apply the visual tool of scientific workflow to teach and learn computational thinking skills.

## VII. Conclusions And Future Work

In this paper, we first developed a new effective online learning model for computational thinking based on our previous widely-applied R2D2 model. A signature characteristic of this new model is being interactive and learner-centered thus "i" is carried out through the entire learning experiences, which provides teaching facilities to instructors and interactional tools among instructors and learners. Second, we implemented, validated, and refined iR2D2 in our DATAVIEW computational thinking online service. Third, we proposed to use the DBR (design-based-research) approach to study the relationship between technology and teaching in the context of computational thinking, generating research results and findings applicable to online teaching in other domains as well. As ongoing work, we are currently improving our DATAVIEW system by developing new features and conducting several case studies, including some tutorials in three courses in computer science department to demonstrate the effectiveness of our learning technologies and the strength of our DBR approach.

## Acknowledgment

This work is supported by U.S. National Science Foundation under ACI-1443069 and is based upon work supported in part by the National Science Foundation under Grant No. 0910812.

## References

1. http://en.wikipedia.org/wiki/Tower_of_Hanoi
2. http://en.wikipedia.org/wiki/Travelling_salesman_problem
3. Cui Lin, Shiyong Lu, Xubo Fei, Artem Chebotko, Darshan Pai, Zhaoqiang Lai, Farshad Fotouhi, and Jing Hua, "A Reference Architecture for Scientific Workflow Management Systems and the DATAVIEW SOA Solution," IEEE Transactions on Services Computing, vol.2, no. 1, pp. 77-92, 2009.
4. Terence L. van Zyl, Anwar Vahed, Graeme McFerren, Derek Hohls: Earth Observation Scientific Workflows in a Distributed Computing Environment. T. GIS, vol.16, no. 2, pp. 233-248, 2012.
5. Xubo Fei and Shiyong Lu, "A Dataflow-Based Scientific Workflow Composition Framework," IEEE Transactions on Services Computing, vol.5, no. 1, pp. 45-58, 2012.
6. Xubo Fei and Shiyong Lu, "A Collectional Data Model for Scientific Workflow Composition," IEEE International Conference on Web Services, pp. 567-574, 2010.
7. Jeannette M. Wing, "Computational Thinking," Communications of the ACM, vol.49, no. 3, pp. 33-35, 2006.
8. A. Repenning, D. Webb, A. Ioannidou, "Scalable Game Design and the Development of a Checklist for Getting Computational Thinking into Public Schools," ACM technical symposium on Computer science education, pp. 265-269, 2010.
9. H. Qin, "Teaching computational thinking through bioinformatics to biology students," ACM technical symposium on Computer science education, , pp. 188-191, 2009.
10. Shiyong Lu, Arthur Bernstein, and Philip Lewis, "Automatic Workflow Verification and Generation", Theoretical Computer Science, vol.353, no. (1-3), pp. 71-92, 2006.
11. Workflow Management Coalition. http://www.wfmc.org/
12. D. Hollinsworth. "The Workflow Reference Model", Workflow Management Coalition," TC00-1003, 1994.
13. Joe Lin, Charley Ho, Wasim Sadiq, Maria E. Orlowska, "Using Workflow Technology to Manage Flexible e-Learning Services," Educational Technology & Society, vol.5, no. 4, pp. 116-123, 2002.
14. Jianming Yong: Workflow-based e-learning platform. CSCWD 2005:1002-1007
15. Bruner, J., "The process of Education", Harvard University Press, Cambridge, MA, 1960.
16. Bernard, Robert M., and Beatriz Rojo de Rubalcava, "Collaborative online distance learning," Issues for future practice and research, Distance Education, vol.21, no. (7), pp. 260-277, 2000.
17. Hamm Mary and Adams Dennis, "The Collaborative Dimensions of Learning," Ablex Publishing, Norwood, NJ, 1992.
18. Jonassen, David, Mark Davidson, Mauri Collins, John Campbell, and Brenda Bannan Haag, "Constructivism and computer-mediated communication in distance education," The American Journal of Distance Education, vol.9, no. (2), pp. 7-26, 1995.
19. Bonk, Curtis J., and Ke Zhang, "Empowering Online Learning: 100+ Activities for Reading, Reflecting, Displaying, and Doing," San Francisco, CA: Jossey-Bass, 2008.
20. Zhang, K. & Bonk, C. J., "Generational learners and e-learning technology," H. Yang & S. C-Y. Yuen (eds.). Handbook of Research on Practices and Outcomes in E-Learning: Issues and Trends, pp. 76-92. Hershey, PA: IGI Publishing, 2009.
21. Zhang, K., and Toker, S., "Stimulating critical thinking in a virtual learning community with instructor moderations and peer reviews," Knowledge Management & E-Learning: An International Journal, vol.3, no. (4), pp. 534-547, 2011.
22. Wang, F., and Hannafin, M. J., "Design-based research and technology-enhanced learning environments," Educational Technology Research and Development, vol.53, no. (4), pp. 5-23, 2005.
23. Barab, S., and Squire, K., "Design-based research: Putting a stake in the ground," The Journal of the learning sciences, vol.13, no. (1), pp. 1-14, 2004.
24. Joseph, D., "The practice of design-based research: Uncovering the interplay between design, research, and the real-world context," Educational Psychologist, vol.39, no. (4), pp. 235-242, 2004.
25. Xubo Fei, Shiyong Lu, and Cui Lin, "A MapReduce-Enabled Scientific Workflow Composition Framework," IEEE International Conference on Web Services (ICWS), pp.663-670, Los Angeles, CA, 2009.
26. B.L. Berg, "Qualitative Research Methods for the Social Sciences," Boston: Allyn and Bacon, 1989.
27. Bonk, C. J., and K. Zhang, "Introducing the R2D2 model: Online learning for the diverse learners of this world," Distance Education vol.27, no. (2), pp. 249-264, 2006.
28. Scott A. Golder and Bernardo A. Huberman, "Usage patterns of collaborative tagging systems," J. Information Science (JIS) vol.32, no. (2), pp. 198-208, 2006.
29. Shiyong Lu and Jia Zhang, "Collaborative Scientific Workflows Supporting Collaborative Science," International Journal of Business Process Integration and Management, vol.5, no. (2), pp. 185-199, 2011.
30. A. Mohan, et al., "Addressing the Shimming Problem in Big Data Scientific Workflows," in Proc. of the 2014 IEEE International Conference on Services Computing (SCC'14), pp. 347-354.
31. D. Ruan, et al., "A User-Defined Exception Handling Framework in the VIEW Scientific Workflow Management System," in Proc. of the 2012 IEEE International Conference on Services Computing (SCC'14), pp. 274-281.
32. Kazuo Hiekata, Hiroyuki Yamato, Piroon Rojanakamolsan and Wataru Oishi, "A Framework for Design Engineering Education with Workflow-based e-Learning System," JSW vol.2, no. (4), pp. 88-95, 2007.