ABSTRACT
Activities that require students to collaborate, share solutions, review each others’ work, or create materials explicitly for the use of other students have been shown to be beneficial not only to students’ learning of the material, but to their reflective, critical and creative skills. This paper presents a snap-shot of tools currently reported as being used to support such collaborative activities in Computer Science education. Basing our analysis on an extensive review of recent Computer Science Education literature, we categorize these tools according to both their form and use while identifying gaps, limitations and opportunities.

Keywords
Collaborative learning, educational technology, contributing pedagogies.

1. INTRODUCTION
A Contributing Student Pedagogy (CSP) has previously been defined as [1]: A pedagogy that encourages students to contribute to the learning of others and to value the contributions of others. This definition includes two necessary components:

- **Students contribute to the learning of others.** Such contributions may take a variety of forms, such as creating tutorial materials, demonstrations, worked examples, examination questions, providing feedback, etc.
- **Students value the contributions of others.** The students believe that the contributions of others are potentially useful in the context of the course.

Such pedagogies can be difficult to implement; for example: in large classes; when anonymity is important; or when the artifacts created by students need to be accessible to other students over an extended period of time. A previous survey [1] identified and classified several different approaches to CSP activities, noting that many instructors used technology to support them.

In this paper we explicitly address the use of tools that support CSP activities in Computer Science. We have performed a large-scale survey of recent Computer Science Education literature to identify cases where CSP is enabled by tools, identifying both a set of characteristics of such tools, as well as the types of activities that these are tools are used for.

This paper first provides an educational context for CSP activities. After presenting our revised definition of CSP, we describe our methodology, and discuss our findings with respect to tool characteristics and activity types. Our discussion considers trends, opportunities, limitations and gaps. A table summarizing all the tools discussed in the paper is given in the appendix.

2. BACKGROUND AND MOTIVATION
In a CSP activity, students create learning materials which are then shared with other students in the class. The resulting community can act to a greater or lesser degree like a “Community of Practice” (CoP), as described by Wenger [2]. We describe the characteristics of a CoP and some underlying learning theory, and discuss how these ideas relate to CSP.

2.1 Social Constructivism
Social constructivism is a well-established theory that deals with how learning occurs through social interactions, and it identifies with forms of knowledge that are dependent on culture and social context [3]. Constructivism holds that knowledge and meaning are generated individually, and that the mental models and understandings held by an individual are shaped by previous experience [4]. Mental models are built and consolidated through processes Piaget named assimilation and accommodation [5]. Assimilation involves absorbing new knowledge into an existing framework without changing the framework. This is the normal way of acquiring new knowledge, and this process serves to form links with existing knowledge and to reinforce the framework. However,
when confronted with an experience that contradicts the internal representation, the individual may be forced to change her mental model in order to accommodate the contradiction. Accommodation is only triggered when the learner recognizes a fault in her mental model.

Social constructivism recognizes the importance of social interaction in the learning process. In particular, encountering a divergent understanding of a concept is a challenge to defend ones own view, or to accept that your understanding is faulty. This challenge is more pronounced when the interaction involves a peer: “Finding yourself with a view on something that clashes with the view of a peer both tends to cause you to try to produce reasons to persuade them (and yourself), and to leave an internal marker that leads you to work on finding a resolution.” (p.289) [6]. Draper argues that peers are more effective in this respect than teachers, since teachers are more likely to elicit unthinking acceptance [6].

Wenger identifies the dual roles of participation and reification as central to the learning process and with the alignment of individuals’ own mental models with the communal learning task. Reification involves giving ideas a concrete form, such as writing them down or otherwise making an abstract concept take tangible form. We understand this as follows: Participation is to put yourself in a position where your understandings are sharpened against those of your peers. Reification provides the tangible objects that allow those understandings to be shared and scrutinized.

2.2 Communities of Practice

Wenger goes on to describe the way knowledge can be shaped and shared in “communities of practice” (CoPs). CoPs are “groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis” (p.4) [7].

Members of a CoP become informally bound together through the value they find in learning together. The value can be tangible (e.g. usable knowledge) or intangible (such as status within the community). CoPs can be large or small, long lived or of short duration, co-located or distributed. Membership can comprise people with similar background, knowledge or skills, or the membership can be heterogeneous. The community can operate within institutional boundaries, or can cross over organizations.

Fundamental to all CoPs are: “a domain of knowledge, which defines a set of issues; a community of people who care about the domain, and a shared practice they are developing to be effective in their domain” [7].

An academic research community is an example of a community of practice. It has a domain of knowledge, a community of researchers who are active in the area, and conventions covering such things as research methodology, expectations of published papers, etc. that constitute a shared practice.

2.3 CoPs in the computing industry

Workplace practice in the computing industry is dominated by team work, and by ongoing learning within virtual communities, typically mediated by technical forums and tutorial blogs. Increasingly, CoPs are being actively cultivated by organizations as a means of managing business knowledge, practices and approaches [7].

Traditional academic teaching, with its focus on instructor-led transmission of established knowledge (as embodied in text books), is ill suited to preparing graduates for subsequent employment. This mismatch is usually addressed to some extent by a gradual transition toward more open-ended courses, but it is appropriate to explore alternative pedagogies and curriculums that incorporate relevant practices earlier and more consistently.

2.4 Communities of learning

The idea of having a class of students form a CoP, a vibrant community of actively engaged students working together for the purpose of learning course material, is therefore an attractive one. The domain of the CoP is the course content, the community is the students enrolled in the class, and the practice is their understanding of the course material.

Other mappings are possible. For example, a learning community draws together students from multiple courses and even subject areas, and can involve a fundamental restructuring of the curriculum [8]. However, such radical interventions are likely to prove challenging to implement in most institutions.

Our own work is strongly influenced by Collis and Moonen [9] who emphasize the process of learning by engaging students as co-creators of learning resources. The distinction between learning-as-acquiring knowledge and learning-as-a-process is succinctly articulated by Sfard [10]. Collis and Moonen’s approach incorporates social constructivism in a practical manner, retaining both content learning and process learning in a meaningful way.

3. OUR REFINEMENT OF THE CSP DEFINITION

The definition presented in Section 1 was very broad, an umbrella term for a multitude of examples of activities that allow for shared learning. Here we refine this definition to make it more specific, and to clearly focus on aspects of CSPs that distinguish them from other forms of collaborative learning.

In particular, we focus on the particular means by which students contribute to the learning of others, and the intention of the contributing student when they are doing so. We also consider the second aspect of the definition (students value the contributions of others) in terms of whether students learning from other students is an explicit intention of the instructor in the pedagogical design.

3.1 Artifacts

We believe that the explicit creation of a concrete, identifiable artifact is an important part of a CSP. The artifact should be created by one or more students, and be used by other students for their own learning. An artifact may also be created for the instructor (for example, an assessed exercise), but such artifacts are not considered unless they are a part of the CSP activity.

The case where students create an artifact and this artifact is not explicitly used by other students in their learning is therefore excluded from this definition. In addition, we exclude examples where the artifact created is not rich enough to explicitly be used by other students for learning; for example, the use of ‘clickers’ in lectures where the ‘artifact’ created by students is a numerical answer to a multiple choice question.

The creation of these concrete artifacts is useful for the instructor too. Hattie [11] emphasizes the importance of instructors getting feedback on how students are doing during the course. Requiring students to create artifacts for sharing with others allows the instructor to ‘look over the students’ shoulders’ to see the extent of their current understanding.
3.2 Intention

CSP as defined above may happen ‘by accident’. For example, students may informally share and discuss their draft assignments with friends – in this case, there is a concrete artifact, and friends will learn from seeing how the creator of the artifact has addressed a problem. However, this is unintentional collaborative learning, and not part of the instructor’s explicit pedagogical design.

Our refined definition of CSP takes into account both the explicit intention of the instructor (in defining the learning activities) and the intended audience for the artifact – that is, whether the students are aware that the artifacts they are creating will be used by other students.

We therefore exclude the informal sharing and discussion of artifacts that may happen during, for example, group work or pair programming. We also exclude occasions where an instructor may use a student’s answer to a question as an example of a model (or imperfect) solution, if the student was unaware that the instructor may do this.

3.3 Our definition

We focus on those activities where the specific intention of the instructor is that a student (or students) produces an artifact that is shared with at least one other student for the purposes of learning.

We therefore suggest a revised definition for the purposes of this paper:

Contributing Student Pedagogy (CSP): A pedagogy that requires students to produce artifacts for the purpose of contributing to other students’ learning, and encourages students to value these peer contributions.

Key features of this definition are:

- the instructor is explicitly using a contributing student pedagogical design;
- the pedagogy includes the requirement that students produce artifacts;
- the students are aware that the audience for these artifacts include other students; and
- these artifacts are shared with other students, for the purposes of their learning.

While this is a narrower definition than before, we think it an improvement, as it allows us to focus on those innovative (and brave) pedagogies that explicitly entail shifting power from instructor to students. Such pedagogies take a more significant leap forward than, for example, group work, problem based learning, minimal instruction methods [12], or the use of discussion forums.

Within this definition, the following activities are included: oral presentations by a student to the whole class; students providing comments on other students’ work (peer review); students creating example exam questions for students to practice on; and discussion forums where students are required to upload artifacts that other students will look at. Examples of activities that are not covered by this definition include: students working in teams to produce an item for assessment; students programming in pairs (pair programming); online discussion forums where students are not required to provide particular information or content; and blogs that are not accessible to other students.

4. AIM AND METHODOLOGY

The aim of this paper is to present a snapshot of tools currently used to support CSP activities in Computing Science education, to categorize them according to their use, and to identify gaps and limitations in the current use of CSP tools.

Prior to the working group convening at ITiCSE 2010, the four authors conducted a systematic search of the literature for papers in which technology was used to support CSP in the context of a computing course. In this search we used our original definition of CSP. We focused our search on papers published since 2006 in the key conferences and journals which report computing education research studies. The following conferences were searched:

- SIGCSE Technical Symposium on Computer Science Education (SIGCSE)
- Innovation and Technology in Computer Science Education (ITiCSE)
- International Computing Education Research Workshop (ICER)
- Australasian Computing Education Conference (ACE)
- Baltic Sea Conference on Computing Education Research (Koli Calling)
- Annual conference of New Zealand’s National Advisory Committee on Computing Qualifications (NACCQ).

The following journals were searched:

- Computer Science Education journal (CSE)
- Australasian Journal of Educational Technology (AJET)
- Journal of Informatics in Education

Several engineering journals were also searched, but they failed to produce any examples of CSP using technology. In addition, a collection of papers identified in a previous study [1] were also searched. The search produced a corpus of 85 papers.

During the working group meeting, the authors compiled their results and discussed and refined their definition of CSP. The papers were then re-evaluated according to this revised definition of CSP and a final set of 42 papers were decided upon. In addition, a further set of 10 papers were identified as reporting tools which were potentially useful for CSP, but had not been used in an intentional pedagogy; these are not discussed here. The distribution of papers by source is shown in Table 1.

<table>
<thead>
<tr>
<th>Source</th>
<th>Number of Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGCSE</td>
<td>8</td>
</tr>
<tr>
<td>ITiCSE</td>
<td>14</td>
</tr>
<tr>
<td>ICER</td>
<td>2</td>
</tr>
<tr>
<td>ACE</td>
<td>5</td>
</tr>
<tr>
<td>Koli Calling</td>
<td>5</td>
</tr>
<tr>
<td>Other</td>
<td>2</td>
</tr>
</tbody>
</table>

As a result of our group discussions and our findings in the literature, two independent dimensions emerged: characteristics of the tools (Section 5); and the nature of the CSP activity (Section 6).

Both these dimensions were used to categorise each of these 42 papers.
5. CHARACTERISTICS OF TOOLS THAT SUPPORT CSP

Our aim is to investigate tools and technology that support the creation and sharing of artifacts. This requires a repository for artifacts. The main characteristics of the repository relevant to this discussion are its access permissions and duration (who can access the artifacts, and for how long).

In addition, we identified the following characteristics of tools for supporting CSP:

1. The nature of interaction between students, as facilitated by the tool;
2. The number of students taking part in the sharing activity;
3. The nature of artifact created;
4. The extent to which the use of the tool has been evaluated;
5. Feedback mechanisms from students to the creator of the artifact;
6. Gathering data on student activity;
7. The extent to which the tool facilitates the entire CSP activity; and
8. Whether the tool is tied to the teaching of a single topic, or has wider use.

5.1 The nature of the interaction

Interaction between students can be mediated by technology in a variety of ways, and can be characterized as follows:

- **Identity**: The interaction can be anonymous, such as the case in a typical double-blind peer review, the identity of students can be disguised through the use of nicknames [pseudonyms], or the identity of the student is known, as occurs in social networking tools.
- **Synchronicity**: Interaction between students can occur synchronously, as it does in a chat session, or asynchronously as in the case of discussion forums.
- **Direction**: The interaction between students can occur in only one direction, as in a traditional presentation in which the audience receives knowledge but is unable to provide feedback, or the interaction may involve multiple contributors participating equally, as in the case of a discussion.

5.2 The number of students (“Constellation”)

Our CSP definition requires that students produce artifacts that are “consumed” by other students, with no restriction on the number of students producing and consuming. However, some CSP activities may require that the artifact be created by a group of students, and it may subsequently be used by all the other students, or only a subset. A particular constellation may be inherent in the design of the tool (for example, PeerWise), or it may be defined as part of the instructor’s pedagogical design.

5.3 The nature of the artifact

Importantly, in our definition of CSP, we require the existence of a concrete artifact; thus, verbal feedback or textual communications sent in synchronous chat rooms are not considered part of a CSP activity. Typically, the artifact will be stored electronically.

- **Form**: The artifact itself can take many concrete forms. It could be a document, a PowerPoint presentation, a multiple choice question or a concept map. It could be shown on the panel of a Tablet PC, on an overhead projector, or accessed via a webpage.
- **Duration**: Artifacts may exist over different time-scales: this may be the duration of a lecture or seminar session, or the whole semester. Exceptionally, the artifacts may be carried over to future courses, and used with a different cohort of students.
- **Access**: Once the artifacts have been stored, there are varying levels of possible access. The artifacts may only be accessible to the instructor, to the student who created the artifact, to a sub-set of students, or the whole class. While sharing the artifacts is a key part of the intentional CSP (and therefore it may seem that instructor-only access is against the CSP philosophy), artifacts may be shared before being stored.

5.4 Evidence of evaluation

We consider the extent to which the tools have been evaluated, and the nature of the evaluation performed. In some cases, the focus of the evaluation is the pedagogy; while in other cases, specific features of the tools have been evaluated. The evidence provided can be categorized as follows:

- None – There is no evidence of evaluation provided.
- Informal/ anecdotal – The observations of the instructor are reported, but no independent verification is possible.
- Student feedback – The opinions of the students are gathered and reported through surveys or interviews. These opinions may relate to enjoyment, attitudes, or the impact on learning, as perceived by students.
- Analysis of use – The logs of student use are collected and reported. These provide empirical evidence of the frequency (and potentially the nature) of use by students.
- Analysis of artifacts – The artifacts produced by students using the tools are analyzed. Such an analysis may provide evidence of higher thinking or measurable improvements in the quality of the product.
- Impact on subsequent performance – The impact of the tool is evaluated in terms of an external measure of performance, such as an exam at the end of the course.

5.5 Feedback to the contributor

Students will create artifacts for sharing with other students. In some CSP situations, the receiving students have the opportunity to provide feedback to the creator of the artifact. This feedback may itself be considered an artifact in the CSP learning process.

5.6 Student activity data

It is generally possible to record the identity of a contributor; however, tools do not always support the collection of data on the identity of students who learn from those contributions. For example, standard forums are able to record the identity of a student posting a message, but do not typically record and report the identity of students who have read the post.

5.7 How the tool is used

Some tools may only support the sharing of artifacts which are created by other means. To be used in a CSP activity, they would have to be used in tandem with another technology. They are typically tools that have a wider use outside of education (e.g. wikis,
blogs, and discussion forums). Other tools have been specifically designed and implemented for CSP use: these would usually support both the creation and the sharing of artifacts between students.

5.8 How widely is the tool used
Tools that have been designed to support specific material in a single subject (such as algorithm visualization) do not typically have potential for widespread use in other subject areas. However, tools that are designed to support a more general activity (such as discussion, or annotation) are normally able to support CSPs in many different subjects. In some more mature tools, this potential has been realized and the tool has already been used in more than one context.

6. THE NATURE OF THE TOOL-SUPPORTED CSP ACTIVITY
The existing tools support a variety of different CSP uses:

- **peer review**: students look at other students’ work, and provide comments.
- **dialogue and discussion**: the contribution comes about in the exchange of communications between students.
- **annotation**: students comment on existing materials and share their comments with other students. Note that these materials have not, in this case, been created by students.
- **content construction**: students create new learning materials embodying factual course content, for other students to learn from.
- **solution sharing**: students share their own solutions to problems with other students. Note that this is different from peer review in that the receiving students do not need to provide a review.
- **activity creation**: students create learning activities for other students to engage in.
- **making links**: students making connections between known concepts, or students searching for external resources that relate to the content.

In this section, we describe all the relevant papers we found in the literature, all of which fall into one or more of these categories.

6.1 Peer review
Peer review tools were (by far) the most prevalent systems described in the literature. These tools provide a means by which students can share solutions with their peers, and critically evaluate those solutions according to given criteria. This peer review process is purported to have numerous benefits to students, both for those producing the feedback and those who receive the feedback [13]. Numerous peer review tools have been developed to support online peer review in a variety of other subject areas [14], but here we consider only those tools that have been used in the context of Computer Science.

In most peer review systems, students engage in a double-blind review process. All the systems we describe have the capacity to accept student submissions and provide an interface for students to enter and submit a review. In some cases, the tool provides the ability for the recipient of a review to provide feedback to the reviewer.

In a typical peer review process, the allocation of submissions to reviewers occurs automatically, and other students are not permitted to access material that they have not been allocated, or the reviews of other students. However, both PeerWise [15] and StudySieve [16] use a model in which students can view as many of the learning artifacts as they wish, but a review is required for each artifact. In both systems, the review comments and ratings submitted by other students are publically visible.

We organize the tools into three broad categories following the approach taken by Luxton-Reilly [14]. Peer review tools are categorized as “general purpose” if they are flexible and could be used to review a wide range of artifacts of different types (e.g. code, essays, lecture material). A tool is categorized as “domain-specific” if the object of the review must be of a specific type (for example, only multiple choice questions, only code, only essays), but the tool has broad application within the limited domain. Tools that have been specifically built to support a particular activity in a particular classroom are categorized as “context specific”, and are unable to be used in other contexts without modification to the tool.

6.1.1 General purpose peer review tools
Aropá [17] is a flexible peer review tool that is designed to support a wide variety of peer review activities by providing customizable rubrics and workflow. Students submit their assignments to Aropá which allocates the assignments for review. The instructor defines a rubric that may consist of any combination of free-text responses and radio buttons. The instructor may require students to review their own work, and may also require them to review the reviews they receive, or the reviews submitted and received by other students. Aropá has been used in a number of Computer Science courses, and additionally used in a variety of other subjects. A comparison of peer ratings with the ratings assigned by tutors in Computer Science and Software Engineering courses revealed a significant positive correlation. The authors conclude that students are able to accurately assign grades for the work produced by their peers [17].

PRAISE [18] is a generic peer review tool designed to support reviews based on binary criteria. Students submit their assignments to the system and they are immediately allocated an assignment to review (if sufficient assignments have been submitted to the system). The review criteria are intended to be objective binary criteria (in other words, the assignment either meets the criteria or it fails to meet the criteria). If a student has major concerns over a given review, they can flag it as requiring instructor attention. When the reviews are sufficiently different from each other, the system flags the reviews for moderation. PRAISE has been used in a number of courses, including CS1 and a Masters level technology management course. Students were surveyed and usage logs were reported. The vast majority of students found PRAISE to be easy to use with respect to submission of assignments (100%) and reviewing assignments (96%), and they valued seeing the work produced by other students (92%) and felt that reviewing other students helped them to understand course material (77%). The authors report that students typically receive feedback within hours of submitting their work.

PRAZE [19] is designed to be flexible and support a wide variety of rubrics, including free-form text, Likert-scale, multiple choice options etc. Although it was used in the context of code review, the tool is not limited to one domain, but can accommodate the peer review of essays and other artifacts. Students using the system are expected to anonymously review the work of their peers, and to provide feedback to the reviewers on the quality of the
review. A student survey showed that students believe the seeing a variety of solutions helped them to learn (89%); feedback they received was useful (79%); the ability to respond to the reviewers was important (68%) and that overall, the peer review process was useful (95%).

Dooley [20] reports on the use of Moodle Workshop to perform peer reviews using an instructor defined rubric.

6.1.2 Domain-specific peer review tools

The OSBLE environment [21] is designed to support pedagogical code reviews (PCR). Students upload their solutions to an assignment, and the system guides students through a PCR primarily providing support for logging and viewing issues. The identity of the reviewers and those being reviewed are known. This is unusual in tradition peer review, but appropriate in this context as it is modeling the industry practice of code reviews which are not conducted anonymously. An evaluation of OSBLE showed that the use of an online system provided access to code prior to the PCR session, and access to the reviews of other students after the PCR session, both of which were not possible with paper. Compared with paper submission, the online system allowed students to log issues 37% faster. The authors report that they have recently extended the system to allow the instructor to specify the level of anonymity.

WebWork-JAG [22] is an extension to the WebWork system that runs student submitted Java code against a set of test cases. Students author both the Java code and the unit tests. The artifacts produced by students are peer reviewed using WebWork, and the peer reviews help students to reduce the number of errors present in test cases.

StudySieve [16] supports the peer review of student-generated free-response questions, and answers to those questions. Both the questions and answers submitted by students are peer reviewed by assigning a numeric rating and free-text formative feedback. The system reports the number of ratings given to a specific question or answer, allowing an author to identify the number of students that have viewed and evaluated the material. All submissions are anonymous.

PeerWise [15, 23-25] supports the review of multiple-choice questions (MCQs) and associated explanations. Students may choose to answer any number of MCQs. All content submitted to PeerWise is anonymous. Students can see the number of answers submitted to their questions, and can respond to comments on their questions made by other students.

Expertisa [26-27] is designed to peer review learning resources. Students are expected to perform reviews according to a specified rubric. The reviews themselves are rated according to another rubric. The resources reviewed by students can be learning activities, quizzes, or new content. Student surveys reveal that students believe they learn a lot from doing the peer review assignments, and that they enjoy participating in the peer review process [26]; and also that the feedback they received from their peers helped them to learn (1.9 on a Likert scale where 1 is strongly agree and 5 is strongly disagree to the question “The feedback I received helped me learn”) [27].

ClassCompass [28] supports the development of software designs by providing both automated assessment and facilities for peer review. Students critique each other’s designs by filling in a text area provided by the tool. Students can revise their work based on the critique, and may submit further critiques of the revised designs of their peers. The tool has been used in several undergraduate software engineering courses. Students in a fourth-year software engineering course were surveyed about the use of the tool. Of the students that responded to the questionnaire, 38% reported that they did not improve their understanding of design principles as a result of using the tool, but 18% said that their understanding did improve as a result of using the tool. The authors report that the tool might be improved by allowing students to respond to the critiques of others, but the value of the critiques in the current implementation was not clear.

Cutts [29] developed a small online system to help manage the peer review process. The instructor of the course allocated a number of essays for each student to anonymously review using the online system. Students responded to the reviews they received, and were required to grade the reviews written by a number of other students. No formal evaluation was conducted.

6.2 Dialogue and discussion

Systems that facilitate discussion between students are common: for example, email, online discussion forums, Facebook and bulletin boards all support communication between students. We focus on those tools where the explicit pedagogic intention of the instructor is that the tool be used for students to share artifacts with each other; thus, casual communication, when a student may provide some information that may be useful to another student, is excluded.

We found several examples of instructors using wikis or online discussion forums in their classes, with the intention that the students share artifacts, with varying success. Choy and Ng [30] used Wikka Wakka Wiki, and found that students were more willing to engage in question-and-answer discussion than to share artifacts.

Anderson and Lin [31] asked students to contribute to a blog and to communicate via a user group: they found that blog contribution was adopted more readily than the user group, and that blogs supported the equal contribution of under-represented groups. The students reported that blogs helped them with resource sharing and collaborative learning. Guerreiro and Georgouli [32] advocate the non-anonymous use of Moodle to facilitate direct communication between lecturers and students, and between students themselves – their preliminary use of the system was for the posting and answering of questions (these being the artifacts created), although there was no monitoring of which or how many students accessed these artifacts.

Bower [33] used Adobe Connect Meeting planner in a programming class both as a virtual classroom for scheduled sessions, and to facilitate collaborative group work, with students contributing a range of artifacts, from closed responses (true/false, classification) to discussions on existing program code and program code itself. Bower found that the more open tasks (e.g. program modification and debugging) elicited greater collaborative response and discussion than the more closed tasks. The use of Macromedia Breeze Presenter [34] allowed students to submit course content, code and audio discussions onto a shared whiteboard during synchronous sessions that were recorded and stored for students’ later use. Bower stresses the usefulness of the repository of recorded sessions to both the students and his own teaching peers.

However, Carrington et al [35] used Blackboard and Wimba Live Classroom as a means of facilitating communication with off-campus students that they had never met. While they had hoped that the students would use these tools for peer communication,
they found that most off-campus students considered them simply as a means of communication with the lecturer and accessing course materials – it seems that physical presence may actually affect online communication.

The MessageGrid system used by Pargas and Shah [36-37] allows students to post questions, summaries, explanations or ink diagrams during the teaching sessions itself – it facilitates direct communication between each student and the instructor. The instructor can then use these contributions as supporting material during the lecture, revealing the individual student contributions to the whole class as required. Used in conjunction with ‘clickers’ [38], they found that this approach enabled early detection of student misunderstandings.

6.3 Annotation

Annotation tools are those that support the ability for a student to make notes on material provided by an instructor, and to share those notes with their peers. These notes are intended to help other students to learn by providing supplementary information that clarifies concepts; providing variation in understanding; providing links to other related material; providing exemplars, or sample solutions; or illustrating flawed solutions that expose misunderstandings.

Classroom Presenter [39-41] is used to share digital ink annotations generated during a classroom lecture. Students attending a lecture are able to view a copy of the instructor’s overhead slides on their own Tablet PC. They may annotate the slides with digital ink and submit their annotations to the instructor, who selects a sample of the submissions to share with the class. The digital ink can take the form of text, diagrams or even programming code, and may be used as examples, or to motivate in-class discussions.

Ubiquitous Presenter [42] (an extension to Classroom Presenter) was combined with a tool called Noteblogger that records the annotations of a small number of students who are tasked with taking notes in the form of slide annotations. The annotated slides created by these students are archived and available both during and after the class.

Saeed and Yang [43] encouraged students to use an off-the-shelf blogging tool to critically reflect on the content of lecture material. Students used the blogs to comment on the things that they thought were most significant, and most confusing in the lecture material.

Rößling and Kothe [44] developed annotation support for any written document uploaded to a Moodle module. The annotations are in the form of text or highlighting, and can be private (i.e. only accessible to the author) or shared with other students by marking the annotations as public.

When an instructor mediates the student contributions, the identities of the contributors remain anonymous [39-40]. However, in the case of Noteblogger, the identities of the contributing students are known. Providing a small number of alternative annotations for the same lecture slides creates a competitive structure that encourages the bloggers to distinguish themselves. Such a structure allows for the possibility that students who are able to communicate their knowledge effectively could emerge as leaders of the learning community. The use of standard blogging tools revealed the identity of the students, which made them uncomfortable [43].

Although annotations typically occur in an asynchronous fashion, Noteblogger is unique in that it allows students to view the annotations of others in real time as the lecture progresses. All of the annotation tools are used to supplement the course content delivered by an instructor, and have uni-directional interaction. Only the Moodle module described by Rößling and Kothe [44] tracked which annotations had been read by a student. None of the other annotation tools provided any way to identify which of the students had read the annotations.

The blogging activity described by Saeed & Yang [43] was evaluated by a student survey. This found that only 59.3% of the class participated in the blogging process, but those who did participate liked the blogging activity (79.5%), felt that it improved their understanding (60.2%), and improved their collaboration and communications skills (67%). Overall, most of the participating students were satisfied (71.6%).

The artifacts created by students using the Noteblogging facility were examined with respect to Bloom’s taxonomy, and examples of student contributions were found at all levels of the taxonomy [42]. The majority (two-thirds) of students participating in a mid-semester survey reported that they watched blogs during class. These students had read the annotations. Students using Classroom Presenter [39] reported that the system had a positive effect on their learning (18 of 19), contributing solutions had a positive effect on their learning (18 of 19) and that seeing the solutions of other students had a positive effect on their learning (16 of 19). Student participation was high (69% for any given question), and 97% responded to at least one question in the lecture.

6.4 Content construction

One of our key requirements for CSP is the creation of an artifact. In many cases, and in line with our focus, there are several cases of instructors requiring students to create artifacts that embody the content of the course. In some cases the only way that this content is taught to students is via the artifacts that other students create; in other cases, the content may be mediated by the instructor.

There are several examples of students being asked to present information to other students in oral presentations. In some cases, the duration of the artifact produced (the presentations) is only the duration of the presentation session; in other cases, it is stored in a more concrete form to be accessed after the presentation.

Pears and Larzon [45] divided the class into groups and asked each group to present new material (a section of the textbook) to the class in an oral presentation, and to subsequently submit a report to a wiki. This was core material for the course, and this was the only way that it was taught to the students. All students and staff attended the presentation and had access to the report, which was available for the duration of the semester. The students in the class described by Crescenzi and Nocentini [46] asked students to create an animation of an algorithm and present it to the class using a visualization tool. This was supplementary material: the students knew the algorithm, but its visualization was a novel
creation of the contributing students. In a class survey, students said that the animations helped them with their examination revision. The authors do not state whether the student created visualizations were stored in a repository by the instructors.

Other content creation methods require that students upload material into an online system, the material then being made available to other students. Again, in some cases this was the only way the material was presented; in other cases, the material was supplementary to that provided by the instructor.

Students uploaded their own Scratch programs to an unnamed website in an high school introductory programming course described by Romeike [47]; all other students had access to these programs for the duration of the course, and a few of these programs were discussed at the beginning of the next lesson. A controlled study found that learning objectives were met, that students were highly motivated, and that their perception of computing was improved.

Choy and Ng [30] used a voluntary scheme, whereby students were encouraged to submit materials covering course content to a wiki. These materials were also covered elsewhere during the course by the instructor, and the materials contributed by the students were intended to supplement the core materials. They found that the wiki was used extensively for students asking questions which were answered by other students, but that little actual content was uploaded. The authors indicated that they would like to extend their use of the wiki to encompass peer review and collaborative writing activities.

Conversely Lutteroth and Luxton-Reilly [23] made submission of course content to a wiki obligatory, and they did not teach any of the course content explicitly to the students – all the course material was created by the students and shared amongst the class. The instructors observed that this method required more self-motivation that traditional courses, and that weaker students required additional support. This method was similar to that used by Hamer [48] who asked the students to write an online hyper-text text-book covering all the material in the course. Groups of two or three students were allocated sections of the text book to write, and all students were encouraged to discuss the other students’ contributions using the wiki. Student survey responses indicated that the students valued the research element of the method and that they recognized that they had developed useful research skills in the process.

The Expertisa system allows for the creation of learning resources to ‘help future students learn the material’. Gehringer et al. [26]. Each student was allocated a different part of a not-yet-produced textbook, and asked to improve an explanation, make up an example of a concept, and to create exercises. These were peer reviewed, and the higher quality artifacts submitted to the author of a new textbook. It is not clear from the paper whether these finally selected artifacts were also made available to the current class in a widely accessible repository; although the authors suggest that they would be useful for subsequent classes. Their survey of students found that students typically found these assignments enjoyable and valuable to their learning.

As an example of artifacts being made accessible to students in another cohort, Shannugasundaram, Juell and Hill [49] asked pairs of students to produce visualizations to help other students understand object-oriented programming concepts (for example, event handling while loops). These visualizations were created on students’ own web pages, and were therefore accessible to all students in the class. In addition, they are carried forward to the following year – in fact, the authors say that they have collected sufficient visualizations of the concepts that the now only used student-created visualizations in their own classes. The authors reported that the students were highly motivated in creating the diagrams, but tended to not relate these diagrams to the code that they were meant to be illustrating.

It is clear that there are several factors to be considered when students are required to produce course content, in particular whether the content provided by students is the only way in which the students get this information, or whether it is also provided by the instructor. In addition, the use of a tool to support the collection of the student-produced artifacts can assist with allowing them to be accessed by other students. Even ephemeral artifacts like the slides for an oral presentation could be stored and made available to other students.

6.5 Solution sharing

Peer review activities include students sharing their solutions to exercises with other students, but there are also examples of tools that support students sharing their solutions, but not necessarily for the purposes of peer review. In these cases, typically, the solutions are shared during a lecture or class meeting, rather than being made accessible in a long-lasting durable form.

Classroom Presenter [40] allowed students to create in-class digital ink solutions to exam-style questions; these solutions were wirelessly (and anonymously) collected by the instructor, who chose some to display and discuss with the rest of the class. The solutions were not, however, stored and made available after the end of the class. The authors found that the students found these classes enjoyable, and that increased participation in the submission activity improved final exam performance.

Deming et al, Griswold [50] also used an in-class solution sharing activity. They asked students to create several different types of artifact using Tablet PCs during a lecture, including diagrammatic solutions to problems. Ubiquitous Presenter was used to then present a selection of these artifacts to the whole class. This exploratory research was, however, performed in a ‘mock lecture’ situation, and the authors focused their evaluation on the use of different modalities, rather than on the efficacy of the tool. These authors also note that this tool could be used in many different modes, with student-produced artifacts (whether produced in-class or out-of-class) can be automatically published on a web site to be accessed by all students and instructors [51].

It seems that there is a potential for tools to support solution sharing (aside from the peer review tools mentioned above). Tools that allow for several students to submit their solutions to an exercise, so that other students can see the range of possible answers would be useful. The only example we have in this category is the recently created StudySieve [16], where all students can submit solutions to (and comment on) student-proposed questions, but there is no specific peer-review requirement for each contribution.

6.6 Activity creation

We have classified ‘activity creation’ separately to ‘content creation’, as activity creation entails students creating resources that other students are expected to actively engage in (for example, potential examination questions, multiple choice questions). This is different from content creation (as described above) where the artifacts produced are the type of static material that students might otherwise get from a textbook or from an instructor.
Gehringer and Miller [27] also asked students to create physical active learning exercises (including props, diagrams, games and videos) and upload them to a wiki; for example, sorting exercises where students are allocated numbers and move physically move around the room to demonstrate the sorting method. Some of these activities were used in subsequent classes, led by the instructor. Feedback from students showed that they enjoyed creating (and acting out) the learning exercises, but were unsure that they actually learned anything by doing so.

Abad [52] required pairs of students to create interactive learning objects to support the learning of other students (both in their own class, and in future cohorts). Students created quizzes, the interactive learning object (for example, a tutorial or simulation) and theoretical background. These were submitted to a (unnamed and un-described) system, and anonymously peer-reviewed. The authors performed a survey evaluation of the class which found that the students enjoyed both creating and using the learning objects.

WeBWork is a system used by Gotel, Scharff and Wiedenberg [22] where students contribute programming problems to the system, and other students are required to create unit tests for these code fragments and to perform user acceptance testing. Students reported that this activity encouraged them to think about testing throughout the development of their code, rather than as an activity simply to be addressed at the end.

Two tools developed at the University of Auckland are good examples of activity creation: PeerWise allows a student to create multiple choice questions for other students to attempt, provide feedback on, and discuss. StudySieve extends this model to more generic question formulation using free text area for the specification of the solutions to the student-created exercises.

StudySieve highlights one of the problems with tools to support the creation of learning activities: the specification of the ‘correct’ solution of the exercise. While a student attempting a PeerWise student-created multiple choice question will get immediate feedback on whether their answer is correct or not, this cannot be the case with free-text solutions.

6.7 Making links
In our category of making links, we include examples where students are asked to make links between different concepts, the intention being that by sharing these links with other students, it will help those students understand the material better.

We found two (quite different) examples in this category: in one case, the links were “internal” (i.e. links between prior knowledge), in the other, they were “external” (i.e. links to newly discovered knowledge).

Marshall et al. [53] used GetSmart for students to create concept maps representing their own understanding of the course material. Individual students created their own concept map, which they shared with other students within the same group; these were used as a resource for a group presentation to the rest of the class. GetSmart supports the creation of visual concept maps: the whole class can access the system, and create their concept maps in the system. No information is given as to whether students can access concept maps created by students who are not in their own group.

Students were asked to collect “social bookmarks” [43] (shared links to web resources) and share them with the other students, so as to show the links that they had found between the course material and other online resources. An evaluation found that only a quarter of the students participated in the voluntary bookmarking activity, but that the bookmarks collected were all considered useful.

7. DISCUSSION
In this section, we discuss our findings in terms of the instructors’ motivations, the availability of the tools, tensions associated with “loss of control”, and gaps and limitations.

7.1 Motivation
The use and direction of CSP is primarily driven by instructors wishing to improve the quality of the education service they provide. Non-educational motivations may also be present, such as:

- reducing academic workloads (by having students perform some of the tasks normally undertaken by instructors);
- responding to institutional pressure for change.

However, these were not reflected to any significant degree in any of the reported literature. Rather, the reasons academics gave for introducing CSP teaching methods included:

- Acknowledging that traditional academic teaching takes place in a context markedly different from both academic research and industry practice, and that achieving desirable graduate outcomes requires restructuring the curriculum to more closely resemble industry and research practices;
- Concerns over students experiencing social isolation in large, introductory classes;
- Searching for ways to engage students more deeply with course material;
- Providing students with more meaningful coursework activities, by replacing routine, identical coursework with individual, purposeful activities;
- Stimulating high performing students;
- Providing additional support for struggling students;
- Exploring the potential of new electronic devices, such as tablet PCs, clickers, and cell phones.

We note that motivations were not always explicitly reported, but could often be inferred from the discussion. No dominant motivation themes emerged, and most instructors appear to have more than one outcome in mind when they propose a CSP intervention.

Overall, CSP is being instigated from the “grass roots”, rather than as a top-down, institutional-driven strategy. Many interventions appear to be opportunity-driven, being triggered by the availability of new software or hardware devices, and reports suggest a largely exploratory mode of use.

7.2 Availability
Almost all the tools reported are non-proprietary systems, and many authors actively encourage others make use of them. Despite this, only a relatively small number of reports involve specialist tools developed at another institution. In general, this is to be expected, as the effort required in writing software able to be deployed elsewhere is significant, and places an expectation on the developer to provide maintenance and support. The notable exceptions are the various wiki systems and course management software such as Moodle, which have established large user bases and are supported by numerous contributors and maintainers.

However, the great majority of the CSP tools reported are web-based systems, accessed through any standard browser. This cre-
ates the opportunity for systems to be offered as a hosted service, where new adopters do not need to install any software at all and developers have a single site to maintain. The barrier to entry for such tools is extremely low, and we predict that there will be growing trend toward making CSP tools available in this manner.

7.3 Tensions
There are evident tensions in the extent to which instructors are comfortable at relinquishing their traditional level of control, with most instructors hesitant at doing this. This statement is based on our own observations over several years of supporting instructors in a wide range of CSP activities. With any student-generated content used for instructional purposes, there are obvious concerns about the correctness of such information. Instructors face concerns about whether student-produced learning material will be correct. The overall impact on learning of some faulty information is unclear, and few studies have been conducted on the effects of erroneous material present in the corpus contributed by students. Further research about how students deal with inconsistent or incorrect information from their peers would provide valuable insight in this area.

7.4 Gaps and limitations
Extent: Given the substantial interest in active learning among the CS education community (see, for example, [54]), it is perhaps surprising that there are not more examples of tool-support for student-contributed learning activities. We expected to find more examples of such tools in our extensive search of the past four years of CS education literature. We did, however, find some examples of tools that had the potential to be used for CSP, but the authors of those papers had not recognized or realized this potential.

Wider use: Many of the tools we surveyed are reported in the educational research literature, but appear to be only used within the institution where they were developed (with few notable exceptions — Aropä, PeerWise and the various Classroom Presenter innovations). This is an obvious limitation of many of these tools — they are not promoted or supported for cross-institutional use, and there is much re-inventing of the wheel, especially with regards the peer review systems. Cross-institutional use would also encourage the sharing of best practice between instructors in a variety of educational contexts.

Access to artifacts: Peer review typically restricts access to student solutions to a small number of reviewers, and the reviews and feedback produced by others is limited to the author of the work reviewed (although in some cases, the feedback may be visible to other reviewers of the same material). The extension of this process to permit all students to view all submissions and feedback could increase the value of a peer review exercise. Only small modifications to many current peer review tools would be required to open up the reviews and solutions to the entire class and make the entire process open. Tools that support the public sharing of solutions are rare (StudySieve being the only example where a variety of solutions are presented for public critique), although studio-based learning and code reviews used in industry (and in some courses) are common.

Exploiting students’ interests: Students are continually communicating with their peers on social networking sites. Their obsession with such activities could be exploited by embedding CSP activities within them. For example, social bookmarking provides a way for people to share links to material that they value. Few tools currently support such activities, but we expect the number of tools actively taking advantage of these communication networks to increase.

Evaluation: A major limitation of several of these tools is that they have not been clearly evaluated. Many instructors report ‘success’ in using a CSP tool, without clearly indicating what ‘success’ means, or demonstrating how they have determined it. This is particularly the case for tools which have been used outside of the classroom in which they were developed.

7.5 Conclusion
The limited use of the tools we found in our investigation beyond a single classroom suggests a resistance on the part of other instructors to adopt student-contributing pedagogy in their own courses. This is despite many tools being web-based and easily adaptable to different situations. With minor changes, many existing tools could be accessible to a wider community of instructors, and the extent to which they support collaborative activities broadened. More imaginative use of existing tools (even if they have not explicitly been developed with CSP in mind) and, in particular, social networking technologies could enable collaborative learning activities to become easier to implement in the classroom than might be anticipated. Finally, the worth of CSP tools needs to be demonstrated in a clear and transparent manner: doing so may encourage further use by otherwise reluctant instructors.

Our survey has revealed a wide variety of tools for supporting Contributing Student Pedagogies – evidence of growing interest in collaborative learning and increasing investment in technological support. We expect this trend to continue, with more emphasis on cross-institutional activities, social networking metaphors, and educational evaluation.
<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Usage</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>*ALVIE</td>
<td>Algorithm visualization tool enabling the viewing and creation of visualizations (<a href="http://www.algorithm.org">http://www.algorithm.org</a>)</td>
<td>Content creation</td>
<td>[46]</td>
</tr>
<tr>
<td>Adobe Connect Meeting</td>
<td>Web-based platform including facilities to upload and share files, whiteboards and chat.</td>
<td>Dialogue/discussions</td>
<td>[33]</td>
</tr>
<tr>
<td>Aropä</td>
<td>Web-based system that supports submission and allocation of reviewers, weighted grade calculation and distribution of feedback. (<a href="https://aropa.ec.auckland.ac.nz">https://aropa.ec.auckland.ac.nz</a>)</td>
<td>Content creation &amp; Peer review</td>
<td>[23], [17]</td>
</tr>
<tr>
<td>Blog and social bookmarking (Google blogger &amp; Delicious)</td>
<td>Web-based tools repository allowing students to upload and share their resources. Google blogger (<a href="http://www.blogger.com">http://www.blogger.com</a>) and Delicious (<a href="http://del.icio.us">http://del.icio.us</a>)</td>
<td>Annotation &amp; Making links</td>
<td>[43]</td>
</tr>
<tr>
<td>Blogs and user groups</td>
<td>Web-based tools allowing users to post and reply to messages.</td>
<td>Dialogue/discussions</td>
<td>[31]</td>
</tr>
<tr>
<td>ClassCompass</td>
<td>Web-based collaborative software design tool providing facilities for automated feedback, manual peer-review, and instructor monitoring of solutions and feedback. (<a href="http://www.cs.ubc.ca/labs/spl/projects/classcompass">http://www.cs.ubc.ca/labs/spl/projects/classcompass</a>)</td>
<td>Peer review</td>
<td>[28]</td>
</tr>
<tr>
<td>Classroom Presenter</td>
<td>Tablet-based, wireless system that supports classroom submission of digital ink solutions to questions and exercises.</td>
<td>Annotation &amp; Solution sharing</td>
<td>[31, 39-41]</td>
</tr>
<tr>
<td>Expertiza</td>
<td>Web-based tool that supports uploading of created content, multiple stage peer review, and publishing of content as a learning object.</td>
<td>Content creation &amp; Peer review</td>
<td>[26-27]</td>
</tr>
<tr>
<td>GetSmart</td>
<td>Web-based system supporting the review of previously presented material through the authoring of concept maps and the annotation of course materials.</td>
<td>Making links</td>
<td>[53]</td>
</tr>
<tr>
<td>Java plus a website</td>
<td>Creation of visualizations using Java which are then uploaded to a website.</td>
<td>Content creation</td>
<td>[49]</td>
</tr>
<tr>
<td>Macromedia Breeze Meeting platform</td>
<td>Web-based platform offering a range of tools including, file upload, whiteboard, chat, and course administration.</td>
<td>Dialogue/discussions</td>
<td>[34]</td>
</tr>
<tr>
<td>Media Wiki</td>
<td>Standard wiki</td>
<td>Content creation</td>
<td>[48]</td>
</tr>
<tr>
<td>MessageGrid</td>
<td>Web-based tool allowing instructor and students to interact online both during and out of class using a structured discussion format.</td>
<td>Dialogue/discussions</td>
<td>[36-37]</td>
</tr>
<tr>
<td>Moodle Workshop (Moodle plugin)</td>
<td>Tool within Moodle enabling the creation and management of workshops, management of submission, grading and peer review.</td>
<td>Peer review</td>
<td>[20]</td>
</tr>
<tr>
<td>Noteblogger &amp; Ubiquitous Presenter (extension of Classroom Presenter)</td>
<td>Tablet PC application which allows students to create public noteblogs for a Ubiquitous Presenter classroom.</td>
<td>Annotation</td>
<td>[42]</td>
</tr>
<tr>
<td>OSBLE</td>
<td>Web-based tool that supports online submission, and synchronous and asynchronous reviewing of programming assignments.</td>
<td>Peer review</td>
<td>[21]</td>
</tr>
<tr>
<td>PeerWise</td>
<td>Web-based tool that supports the authoring of multiple choice questions that other students can use and provide feedback on. (<a href="http://peerwise.cs.auckland.ac.nz">http://peerwise.cs.auckland.ac.nz</a>)</td>
<td>Peer review &amp; Activity creation</td>
<td>[15, 23-25, 55-60]</td>
</tr>
<tr>
<td>Tool Name</td>
<td>Description</td>
<td>Features</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>PRAISE (Peer Review Assignments Increase Student Experience)</td>
<td>Web-based peer-review system that enables anonymous review and delivery of feedback.</td>
<td>Peer review</td>
<td>[18]</td>
</tr>
<tr>
<td>PRAZE</td>
<td>Web-based tool supporting submission, peer review and feedback to authors and reviewers. The tool has facilities for the monitoring of reviews.</td>
<td>Peer review</td>
<td>[19]</td>
</tr>
<tr>
<td>StudySieve</td>
<td>Web-based tool supporting the authoring of free response questions that other students can use and provide feedback on. (<a href="http://studysieve.cs.auckland.ac.nz">http://studysieve.cs.auckland.ac.nz</a>)</td>
<td>Peer review, Activity creation &amp; Solution sharing</td>
<td>[16]</td>
</tr>
<tr>
<td>Ubiquitous Presenter (extension of UW Classroom Presenter)</td>
<td>Web-based submission system enabling uploads from a range of devices including computers, laptops, Tablet PCs, cell phones, PDAs, etc.</td>
<td>Solution sharing</td>
<td>[50-51]</td>
</tr>
<tr>
<td>ViZCosh and eMargo (Moodle plugins)</td>
<td>Tools in Moodle to enable annotation of textual materials, visualization integration and uploaded.</td>
<td>Annotation</td>
<td>[44]</td>
</tr>
<tr>
<td>WeBWorK</td>
<td>Web-based formative assessment system used to generate, upload and grade exercises and distribute exercise solutions. (<a href="http://webwork.math.rochester.edu">http://webwork.math.rochester.edu</a>)</td>
<td>Peer review &amp; Activity creation</td>
<td>[22]</td>
</tr>
<tr>
<td>Wiki</td>
<td>Standard wiki</td>
<td>Activity creation &amp; Content creation</td>
<td>[23, 27] [45]</td>
</tr>
<tr>
<td>Wikka Wakka Wiki</td>
<td>Wiki tool enabling user authentication, access control, user pages, and tracking of user activities and user created contents.</td>
<td>Dialogue/discussions &amp; Content creation</td>
<td>[30]</td>
</tr>
<tr>
<td>Wimba Live Classroom</td>
<td>Web-based systems providing (amongst other things) text chat and whiteboard facilities. Wimba also provides audio chat and application sharing. Wimba (<a href="http://www.wimba.com">http://www.wimba.com</a>) and Blackboard (<a href="http://www.blackboard.com">http://www.blackboard.com</a>)</td>
<td>Dialogue/discussions</td>
<td>[35]</td>
</tr>
<tr>
<td>unnamed</td>
<td>Web-based repository enabling upload and sharing of resources.</td>
<td>Content creation</td>
<td>[47]</td>
</tr>
<tr>
<td>unnamed</td>
<td>Not enough information to describe tool.</td>
<td>Content creation &amp; Peer review</td>
<td>[52]</td>
</tr>
<tr>
<td>unnamed</td>
<td>Web-based tool to allow upload, peer review and distribution of feedback.</td>
<td>Peer review</td>
<td>[29]</td>
</tr>
</tbody>
</table>

* We have assumed that there exists a repository of algorithm visualization solutions although this is not clear from the paper.
REFERENCES


21. Hundhausen, C., A. Agrawal, and K. Ryan. The design of an online environment to support pedagogical code reviews. in 41st SIGCSE Technical Symposium on Computer Science Education. 2010. Milwaukee, WI, USA: ACM.

22. Gotel, O., C. Scharff, and Wildenberg. Teaching software quality assurance by encouraging student contributions to an open source web-based system for the assessment of programming assignments. in 13th Annual Conference on Innovation and Technology in Computer Science Education. 2008. Madrid, Spain: ACM.


33. Bower, M. Groupwork activities in synchronous online classroom spaces. in 38th SIGCSE Technical Symposium on
Computer Science Education. 2007. Covington, KY, USA: ACM.

34. Bower, M. Virtual classroom pedagogy. in 37th SIGCSE Technical Symposium on Computer Science Education. 2006. Houston, TX, USA: ACM.


36. Pargas, R.P. Reducing lecture and increasing student activity in large computer science courses. in 11th Annual Conference on Innovation and Technology in Computer Science Education. 2006. Bologna, Italy: ACM.

37. Pargas, R.P. and D.M. Shah. Things are clicking in computer science courses. in 37th SIGCSE Technical Symposium on Computer Science Education. 2006. Houston, TX, USA: ACM.


41. Razmov, V. and R. Anderson. Pedagogical techniques supported by the use of student devices in teaching software engineering. in 37th SIGCSE Technical Symposium on Computer Science Education. 2006. Houston, TX, USA: ACM.


44. Rößling, G. and A. Kothe. Extending Moodle to better support computing education. in 14th Annual Conference on Innovation and Technology in Computer Science Education. 2009. Paris, France: ACM.


46. Crescenzi, P. and C. Nocentini. The design of an online environment to support pedagogical code reviews. in 11th Annual Conference on Innovation and Technology in Computer Science Education. 2006. Bologna, Italy: ACM.


52. Abad, C.L. Learning through creating learning objects. in 13th Annual Conference on Innovation and Technology in Computer Science Education. 2008. Madrid, Spain: ACM.


60. Denny, P., et al. Coverage of course topics in a student generated MCQ repository. in 14th Annual Conference on Innovation and Technology in Computer Science Education. 2009. Paris, France: ACM.