

Where Making Construals meets Learning to Code

Meurig Beynon and Russell Boyatt
University of Warwick
Coventry
UK
+44 24 7652 3089
wmb@dcs.warwick.ac.uk

Ilkka Jormanainen and Andrés Moreno
University of Eastern Finland
Joensuu
Finland
+358 50 375 7289
ilkka.jormanainen@uef.fi

ABSTRACT

The role that computing has come to play in the contemporary world motivates two topical concerns: how to develop and educate the next generation of computing specialists, and how to help computing non-specialists to understand the broader significance of computing in human and social terms. In this short paper, we discuss how research into 'making construals' may contribute to achieving both of these goals.

BACKGROUND

Making construals is a new digital skill that aims to bridge the gap between computing specialists and non-specialists. Its focus is on using the computer as an instrument to make connections in experience – an activity that complements computational thinking. It involves creating interactive artefacts ('construals') to imitate the observables, dependencies and agency in a situation using a special-purpose environment for making construals (the 'MCE').

Developing principles and environments for making construals is a central theme that underpins the Empirical Modelling (EM) research project. The diagram below depicts the way in which making construals has been applied in this context. Since the personnel associated with the EM project have been staff and students of computer science, much attention has been devoted to making construals that relate directly to specialist computing subjects. Themes for making construals have included: heapsorting, statecharts (e.g. digital watch); educational programs (e.g. JUGS); relational database languages (e.g. SQL-EDDI); projection algorithms in graphics (e.g. graphics presentation). Such applications relate directly to the right-hand side of Figure 1. There have also been many applications of making construals to topics that are not technical in character.

The CONSTRUIT project

CONSTRUIT! is an ongoing EU Erasmus+ project that promotes making construals as a new digital skill for creating open educational resources. The project has developed a number of resources including construals exploring basic optics, shopping, graph colouring and playing the piano.

The project's current efforts are working to allow the wider audience to access these resources by reducing the dependence on specialist computing knowledge.

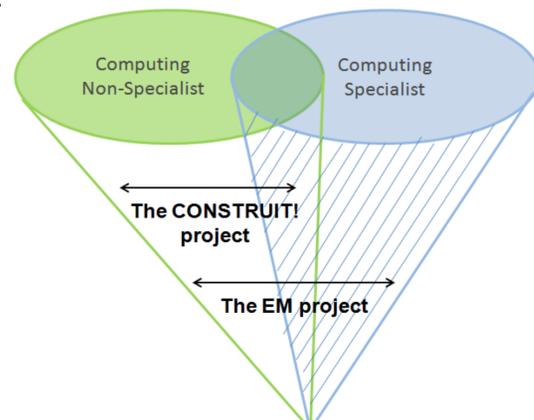


Figure 1: Specialist and non-specialist perspectives.

WHAT ARE CONSTRUALS?

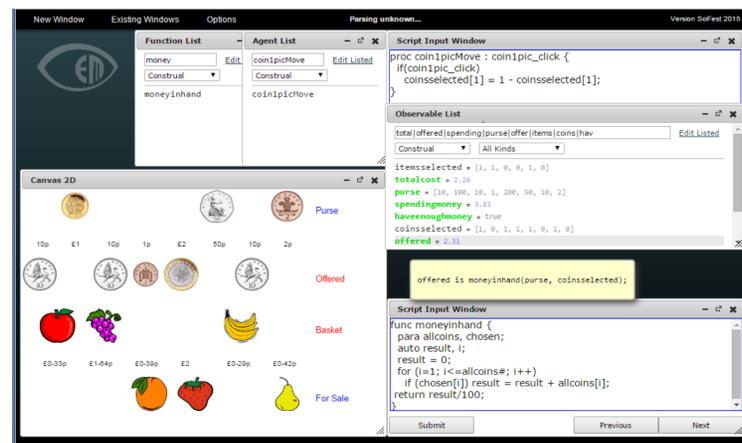
A construal is an interactive artefact that we develop to help explore and record our emerging understanding. The term construal was introduced by David Gooding to refer to the techniques that Faraday used to make sense of electromagnetic phenomena in the 19th century.

Making a construal differs from writing a program. Rather than specifying a recipe to achieve certain functional goals, it establishes an open-ended context for interaction within which the human and automated agency in a domain can be expressed.

RESEARCH QUESTIONS

Introducing the digital skill of making construals raises two broad research questions:

- 1) Making construals needs some technical computing knowledge. How much knowledge is needed? Can this knowledge be acquired by a computing non-specialist? – and if so, how?
- 2) To what extent can 'making construals' provide an introduction to coding for non-specialists? Can this be used to develop coding skills and learn to code generically and abstractly?



MAKING CONSTRUALS FOR NOVICES

The first obstacle for the computing novice is to appreciate the relationship between the intuitive notions of observable, dependency and agency and the computing concepts of definition, function and action. This requires the maker to be familiar with basic functions for manipulating values and house-keeping (such as operators for standard arithmetic, type conversion, list and string processing) and frames definitions in a formal program-style notation with strict syntactic conventions.

CONSTRUIT! is exploring several possible enhancements for the modelling environment:

- The environment can be enhanced to enable functions and actions to be derived semi-automatically, for instance, by allowing the maker to enact them using simple scripts and transforming these into procedural code automatically.
- The environment can include operators and the observables that are customized to the domain, so that makers of construals can frame observables, dependencies and agency in terms of familiar functions and actions.
- Techniques to simplify managing and specifying families of definitions are being developed.

MAKING CONSTRUALS AND CODING

Despite the strict and formal nature of the semantic model within which coding has been traditionally conceived, there is evidence that flexible and informal modes of interaction with code are vital in teaching programming effectively. Visualisation is a first step towards complementing formal understanding of code and engaging the programmer's intuition. In pursuing this notion, Moreno et al highlight the limited value of visualization and animation that is merely faithful to the formal code.

Interactive exploration of code is representative of how the perspective of the computing specialist can be supported from the orthodox rational computing perspective. This outlook is radically different from that promoted by making construals.

Firstly, the learning objective in teaching coding as an engagement with an unforgiving ontologically real computer is beyond what the non-specialist can typically achieve. Making construals is conceptually closely related to creating and using a spreadsheet. Developing a spreadsheet is an entirely different challenge from using a pre-built spreadsheet application.

Secondly, the nature of the interactive engagement involved in making construals is quite different, as it must be if the non-specialist is to have a point of entry to the domain of the computing specialist (cf. Chris Granger's contention that *Coding is not the new literacy*). The exploratory interactions most closely related to making construals are unlike those associated with purely abstract coding.

As Brian Cantwell-Smith observes in *Two Lessons of Logic* this interaction concerns 'the semantics of the semantics' of code. Where traditional semantics concerns how code is interpreted purely in terms of the changes of machine state it effects (without reference to how or even whether these state changes can be interpreted in an application domain), what concerns the programmer ultimately is how what the program does is related to meaningful observations and actions within the domain – what Smith calls the semantics of the semantics.

CONCLUSIONS

CONSTRUIT! aspires to promote making construals as an alternative approach to computing that is more accessible and intelligible to non-specialists. Making construals is no substitute for computer programming – there will always be a demand for technical specifically machine-centred expertise (cf. instrument makers, piano tuners, composers who enhance the technical range of instrumentalists etc). But making construals may help to build bridges between people with these skills and domain experts who have no specialist knowledge of computer science so as to achieve mutual comprehension.

It may also help those who *do* appreciate the technical aspects to approach computing from an experiential human and hermeneutic perspective. There are many reasons why this may be desirable: there is arguably a need to remedy and complement the incomplete story of abstract computation.

ACKNOWLEDGMENTS

We are indebted to Tim Monks, Nick Pope and Elizabeth Hudnott for developing the JS-EDEN environment. This project has been funded with support from the European Commission under the Erasmus+ programme (2014-1-UK01-KA200-001818). This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



For further information visit construit.org