Comics, Robots, Fashion and Programming: outlining the concept of actDresses

Ylva Fernaeus and Mattias Jacobsson
Swedish Institute of Computer Science
Box 1263, SE-164 29 Kista, Sweden
{ylva, majac}@sics.se

ABSTRACT
This paper concerns the design of physical languages for controlling and programming robotic consumer products. For this purpose we explore basic theories of semiotics represented in the two separate fields of comics and fashion, and how these could be used as resources in the development of new physical languages. Based on these theories, the design concept of actDresses is defined, and supplemented by three example scenarios of how the concept can be used for controlling, programming, and predicting the behaviour of robotic systems.

Author Keywords
Physical languages, semiotics, tangible interaction

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
One major stream of research within the areas of embedded and tangible systems design concerns investigations of new models and forms for programming. This has resulted in a large number of physical construction kits [e.g. 4, 24], physical tools for designing and controlling on-screen and acoustic media, and several more or less complete physical programming languages [19]. One application domain in this area concerns new ways of controlling the behaviour of electronic and robotic consumer products. As such products are inherently physical, often without a screen display surface, it is relevant to explore new and user-friendly ways for these to be controlled by physical means.

Interesting with respect to this is how people personalise their digital devices by different forms of physical means. Laptops are made personal by placing stickers on them, people buy or make their own customized cases, and they attach mascots and charms to their mobile phone handsets. Especially in the area of robotic consumer products, people tend to physically accessorize their technology, e.g., by sewing decorative covers for Roomba[1] and dressing up Pleo[2] for different occasions. Although the motivations for these practices may diverge drastically from programming (dressing up a vacuum cleaner may for instance be a decorative way of storing it when *not* in use), they do indicate an interesting starting point for exploration. This is also interesting as physical appearances of robots are increasingly explored in research, e.g. in studies of different head shapes [7] and costumes [25]. Given the fundamental role that clothing has in human culture, and the interest in surface appearance in product design, this may be used much more concretely also in physical interfaces.

After a short overview of current approaches to physical programming, the design concept of actDresses is described. The concept is theoretically inspired by two informal kinds of sign systems that are extensively used and understood in popular culture. The first area is the sign system used in comics, and how this has previously been used as inspiration in the design of visual programming languages. The second area that we explore concerns practices of clothing and accessorising, and how that could be used for controlling the behaviour of physical interactive devices. To make the discussion more concrete, we provide three example scenarios of how physical decoration and labelling can be used for controlling, programming, and predicting the behaviour of robotic systems.

APPROACHES TO PHYSICAL PROGRAMMING
A classic example of how robotic products are usually controlled and programmed can be seen in the robotic turtle used with the early developments of the Logo programming language [21]. That robot was intended as a physical representation and enactment of programming instructions, but the actual instructions were created and sent to the turtle from a separate workstation. Another example is the Lego Mindstorms robotic construction kit, where the intended functionality is partially manifested in the physical

[1] See http://www.myroombud.com/

[2] See e.g. user generated blog posts at www.pleoworld.com

design, but the actual behaviours can only be added, inspected and tuned in the context of using a personal computer. Similarly, changing the behaviour of most consumer robots of today, such as Pleo, Roomba or Aibo, requires a PC-based mode of interaction, which is essentially different from how users otherwise engage with these products. Moreover, when making programs in these settings, it is often hard to perceive what the expected behaviour will be like. For physical computing, this often results in extensive loops of uploading, debugging and fine tuning the interaction of the constructed system [13].

An intriguing question is what happens if the instructions in these cases would not be confined to other physical devices, but visualised and possible to manipulate and change in the ordinary use setting of the object controlled. This particular question has been explored in a range of projects on physical language design, including Patten et al's explorations of using physical string to program robotic systems [22], and McNerny's development of tangible Lego bricks to control and program electronic devices [19].

Several robotic products are also programmable by means of physical training or demonstration, e.g. by bending or moving the artefact in different ways, which gets recorded and repeated by the program. Topobo [24] is a physical construction kit based on such a model of control, just as Curlybot [10], a wheeled robot that can be trained to move in patterns defined by the user. These and similar strategies for end-user specification of robotic movement have proven highly successful, especially for navigation and posture control tasks. However, making more complex programs sometimes requires some form of code that can be examined, edited, and revisited. This is available when programming robots on a PC, but is often lacking when controlling and training robots by physical means.

An increasingly common approach is also to use tangibles of different kinds. Examples include the use of Wii remote controls to control robot gesture [12], and a set of tangibles for controlling robotic teams [17]. Related explorations can be found in the domain of mechanical and interactive toys, where it has long existed dolls and other objects that can be explicitly switched into different modes by attaching different accessories to them (e.g. My real baby). The rich role-playing associated also with more conventional dolls (e.g. Barbie) is to a great extent due to clothing and accessorizing qualities as part of the interaction. A similar approach could be used for combining multiple programming constructs, i.e. collections of physical items that can be combined in many ways.

The general problem of making programming more concrete has also been explored extensively in the area of designing programming tools for children and end users, especially by making simplified higher level forms of representation [see e.g. 14]. However, additional work is needed to explore how to create visually effective and easily understandable ways to represent physical program actions (i.e. a behavioural correspondence to WYSIWYG). Here this design space is explored based on the fundamental aspect of programming languages concerned with signs and symbols as representing computational actions. Of relevance to this is the field of semiotics [5], which studies the use and interpretation of signs.

According to semiotic theory, signs may take more or less any form, including words, images, sounds, gestures, acts, and physical objects. However, they become signs only when they are loaded with meanings that make them stand for something other than themselves. Thus, a sign always consists of a combination of a *signifier* (the manifestation of the sign), and a *signified* (what the sign refers to). As an example, Chandler [5] explains how the same signifier, the word 'open', becomes a different sign if placed on a push-button inside a lift than if printed on a flag to indicate that a shop is open for business. Similarly, the concept 'open' could be represented by a range of different kinds of signifiers depending on context.

Here, focus will be on a scenario of using signs in the form of physical markings attached to a digital artefact, where the physical markings act as *signifiers*, and the actions or behaviours that they would make the artefact perform would be the *signified*. In the development of new physical languages, the challenges thereby include both the designs of the actual signs, as well as their coupling to meaningful computational action.

LEARNING FROM COMICS AND FASHION

Textiles and fashion has achieved increased attention lately in terms of crafting with new materials. Examples include Leah Buechley's work with the textile LilyPad toolkit [4], a range of reactive wearable costumes for playing and interaction [e.g. 23], and in combining conventional design practices, such as fashion, with digital features [20]. Fashion theory is also relevant to tangible computing as it essentially deals with physical features such as fabric, texture, transparency, shape, and so on.

Another motivation for exploring clothing and fashion in terms of physical language design concerns how people present themselves to others through their surface appearances. Clothes not only serve the purpose of being convenient for certain activities, they also serve a range of communicative functions, indicating e.g. appropriate behaviour, group belongings, and expected interactions [6, 11]. As a particular example, one could consider the richness in function of theatrical costumes. Similarly, markings on the physical outside of a digital artefact could be used to indicate to its users what mode it is currently in, and what behaviours and interactions that could be expected, at a certain point in time.

The visual sign language of comics is another form of cultural expression that is relevant to programming, because like conventional programming code, comics depict dynamic activities using a static representation [15].

Comics is also a form of language where it is acceptable to exaggerate and emphasise properties, e.g. powers and abilities, in a way that is playful, again using a form of language that is easy to read. For readers that have learned the principles and sign language of comics, the visual presentation can produce a very direct reading experience, creating an illusion of for example motion and sound, even though the medium itself is static and silent. [15, 18]

Here we explore how the semiotics of these two fields, fashion and comics, could work as inspirations to physical language design.

If signs and symbols known from fashion and comics should work as a form of code for controlling digital devices, one could assume that they would take a rather different and much less explicit character than the programming codes we are used to. Apart from historical shifts and variations in appropriate 'dress codes' for instance, Davis [6] lists three features that make fashion highly ambiguous as a code system: *"First it is heavily context dependent; second, there is variability in how its constituent symbols are understood and appreciated by different social strata and taste groupings; and third, it is – at least in Western society – much more given to "undercodings" than to precision and explicitness."[6, p.8]* However, given the multitude of forms in which different programming paradigms have been developed, finding and exploring possible parallels may still be relevant.

A basic property of clothing also concerns its modular qualities, where layers and collections of different items can be combined in a variety of ways. This property has been conceptually likened to the activity of higher-level programming by combining existing scripts or behaviours into new functioning programs [2]. Making programs by putting together existing pieces of code into new arrangements also relates to common popular practices of software development, e.g., the use of class libraries, interface widgets, and open source methodologies. Especially in educational settings and for novice programmers, such 'higher-level' modes of program construction have been found particularly useful.

Similarly, the visual language of comics is based on a combination of text, pictures, and visual markings such as labels and frames of different kinds. Comic book artists use a combination of such signs together to communicate dynamic features and actions that cannot be expressed using ordinary images. A well explored sign for expressing temporal aspects in comics is a sequence of panels, working to move the story forward [18]. Several research projects have shown that such sequences can be applied in visual programming for defining e.g. narrative structures and before-after scenarios [16]. Another kind of signs, and which may have further potential in the case of physical languages, is the family of visual signs used to depict dynamic actions that happen within a static visual frame. These signs include speech balloons, sound markings, speed lines, pain stars, etc. As such signs are always shown in the immediate visual context of the object or character that they refer to, semiotic theory refers to such signs as *contextual signs*. Systems that draw on this type of markings attempt to support a concrete high-level configuration that may be easier for users to read and interpret than more abstract modes of programming using lower level symbolic representations [9].

THE CONCEPT OF ACTDRESSES

In programming, as in interaction design at large, the physical form and how it shapes the behaviours that users predict and expect from a digital artefact is becoming increasingly relevant to explore. In HCI, this has been conceptualised in terms of what is *sensed, desired and expected* [1], and from a perspective of industrial design a similar concept has been framed as a balance between *appearance and action* [8]. However, knowing only the hardware is never sufficient for predicting how the system will actually work. The concept of clothing may then provide a valuable link between physical bodies, skills and capabilities, and the varying contexts of use.

For the purpose of physical programming, we define actDresses as a kind of physical markings that can be directly attached to a digital artefact, and that signifies some property, action, or behaviour of that artefact. Inspired by the method of visual programming with contextual signs [9], there are two main characteristics that distinguish actDresses from most other methods for programming and control of physical systems.

The first property is that the sign is shown in the *immediate physical context* of the object having the feature represented by the sign. This helps showing which object is controlled by a sign, and can give a direct visual impression of the resulting behaviour. This is not the case in most settings of programming robotic and tangible systems, where the 'code' is sometimes hidden, often at a completely separate hardware device. This also addresses the general desire for visible forms of program representations, which can be inspected and modified.

As a second property, actDresses are meant to represent and produce *perceivable actions* in the computerised system, in ways that end users may easily relate to and understand. As signs, these could take the forms of texts, pictures, or three-dimensional objects, drawing for instance on the material or cultural meanings of their manifestations.

The role of appearance is a central aspect that must be emphasised in this design concept, as the items simultaneously may work as controls as well as well as a form of decoration. Of importance to this is also awareness of existing cultural meanings of different signs, as well as concrete issues of readability. For instance, when attaching several signs or labels to the same object, a consequence may be that they obscure the object, so that it is difficult to clearly see the object underneath. The signs could themselves also overlap and obscure each other.

As with any higher level form of programming based on behaviours that should be possible to combine in a variety of ways, come aspects related to modularity. This concerns how different signs may be combined together, not only visually, but also in terms of their respective signifies, e.g. how different program actions can be practically and meaningfully combined. In terms of implementation, this also concerns aspects of program order and concurrency, and whether certain signs should be given higher priority than others when it comes to execution.

THREE EXAMPLE CASES

To illustrate how the concept could be realised concretely, we here provide three short interaction scenarios designed for different kinds of robotic artefacts. The first scenario is designed for the Pleo robot dinosaur, the second is designed for the experimental GlowBots platform, and the third case is designed for a prototype consumer vacuum cleaner robot. The examples are designed as physically embodied sketches, and an important aspect when selecting the scenarios was that they should be possible to realise with readily available technology.

Case 1: Role assemblage outfits for Pleo

Pleo is a robotic baby dinosaur, and one of the more sophisticated consumer products designed to simulate real life-like behaviour as a form of 'electronic pet'. The robot has approximately the same size as a cat or a small dog, and is equipped with a large number of sensors that make it responsive to touch, and has a sophisticated posture control system aimed to imitate how real animals respond to petting. The robot does not walk more than a few steps, and it makes sounds rather than talk or respond to talk. When users interact with this robot, this is done primarily through physical means, by carrying it around, by petting it, and by dressing it up in different costumes. Moreover, rather than treating Pleo like a mechanical device, users tend to play along with the pretention of a real animal, e.g. by naming it, tickling it, referring to actions such as being sleepy, happy, or wanting something (see user-generated blog posts at www.pleoworld.com).

Finding ways for end user to change the details of how this robot moves and interacts could map poorly with existing observed interaction patterns, and possibly even destroy the play on the illusion of interacting with a live pet. Therefore the mode of interaction must be carefully designed and grounded in existing practices of playing with Pleo. This includes both the form of physical manipulation, and the conceptual level of engagement.

In this particular example scenario we wanted to explore how the standard behaviour of the robot could be altered by letting it change mode or even acquire new abilities. At the same time we looked for a strategy that would seamlessly transfer to existing practices by not requiring much alteration of the original interaction style, nor affect the physical appearance of the robot's skin surface. The result was a set of scenarios such as putting a pyjama on to put the robot to sleep (see Figure 2), attaching a necklace to set it into guard mode, or shoes that make it walk.

Figure 1. Controlling Pleo using actDresses. From top-left: Playing with the robot in "watchdog" mode, as indicated by bracelet. When user changes the costume from bracelet to pyjama, the robot is slowly switching into sleep mode.

The implementation strategy of this interaction scenario is based on RFID technology, with a reader mounted on the robot and each of the available garments in the collection equipped with RFID tags. This combined with internal sensors, e.g. temperature and sound sensors could allow for a range of new and rich modes of interaction, e.g. putting a warm jacket on in a cold or warm environment may result in different robot actions. Items can also be designed as parts of full 'persona' outfits, turning the robot into modes that entail more complex behaviour patterns for special occasions or play scenarios.

Case 2: A behaviour pin collection for GlowBots

GlowBots are a collection of experimental robots with round LED-displays on their top, acceleration sensors for user interaction, protocols for robot-robot communication, and wheels underneath for autonomous movement (Figure 3). We use these robots here as a second illustration case to emphasise that the concept of actDresses not necessarily requires human-like or zoomorphic technology as its base.

We also wanted to specifically focus on actions and user interactions that can be sensed and performed by the GlowBots platform on a lower conceptual level than for that of Pleo. This was motivated by the more mechanical looking form of the device, and also on the envisioned actions that users would be likely to want to modify. These could be narrowed down to very basic groups of program actions: (1) Navigation in space, (2) Display patterns, (3) Generate sound, (4) Send and receive signals from other robots, and (5) Respond to user interactions, e.g. shaking

Figure 2. The GlowBots scenario. Selecting from a collection of physical behaviour "amulettes", attaching them to a GlowBot, and finally the GlowBots move, glow, and interact according to items on their bracelets.

Figure 3. The vacuum robot scenario. A set of signs that can put the robot into different modes. The "shy" sign has made the robot hide under the sofa, and is switched to another mode by the user, to make is spiral slowly and silently on the carpet.

and holding the robot in different ways. For each of these, a range of behaviours of different complexity could be developed. As with the Pleo case, the complexity of these behaviours may stretch from more elaborated behaviours that combine several actions, e.g. displaying a sequence of patterns on the screen, to more detailed coordination of movement and sound, or instruction for how the robots should move e.g. towards or away from one another.

The set of behaviours designed for this platform consist of simple unit resistor pin-based decorations where a rich possibility for making different arrangements and combinations is a main quality. Combining several simple behaviours may lead to more complex patterns. For instance, Figure 3 shows a bracelet with behaviours for 'flash', 'play a tone', 'spin' and 'move forward' (will make it move in circle). Adding another 'move forward' will make it move faster and in larger circles.

Case 3: Comic Signs for a vacuum robot

In our third case we use commercial vacuum cleaning robots as a prototype base for exploring how the concept of actDresses could enhance a task-oriented robot. Earlier studies points out that even technomorphic looking robotic appliances can engage users 'socially'. In the case of Roomba, as with Pleo, specially designed cloth covers are available for purchase on the web. The main usage of such clothes may on the other hand not primarily be for functional purposes, but for personalisation and decoration.

Here we explore how comic book-style patches attached by magnetic tape can be used to extend the robot's basic behaviours with super positioned abilities. As inspiration, Braitenberg [3] introduced the concept of having very simple behaviours from hard-wired sensors-actuators. Such behaviours would for instance be light-seeking, light-avoidance, or turning in a direction as response to simple stimuli, resulting in behaviours that people tended to interpret as psychologically driven, e.g. being curious, aggressive, nervous, etc. Moreover having signs that represent e.g. light-avoidance, users would be able to use a flashlight on the vacuuming robot, making it more interactive. Other possibilities would be to use various super-patches to enhance a certain functionality e.g. speed.

In our example scenario, we have designed a set of comic book like magnetic patches that can be attached to the metallic shield on top of the vacuum robot (see Figure 3). Apart from the standard sensors of most commercially available robot vacuum cleaners, this would be equipped with a basic RFID-reader for reading of the signs, as well as a number of extra sensors to allow for the interaction patterns similar to those in the Braitenberg's Vehicles example, for responding to e.g. motion, sound, and light.

DISCUSSION

Computer programming has conventionally been concerned mostly with textual and symbolic modes of construction using screen displays as the general medium for representation. However, with increasingly physical systems, often not even equipped with screen displays, other systems and modes of program representations need to be explored. Theories of disparate fields such as comics and fashion could here be useful to explore as they are holding specific semiotic qualities, for instance that they are possible for ordinary people to read and relate to.

We have introduced the concept of actDresses, e.g. using physical clothing, labels, and accessories for controlling physically embodied systems. The work is motivated by existing practices of physical customisation of electronic devices, the current trend towards commercial products with increasingly advanced control mechanisms, and experiences from the domain of end-user programming.

As illustrated in our three interaction scenarios, the concept of actDresses could be visually manifested in a range of different ways. Many more scenarios could be envisioned i.e. bracelets attached to a robot with symbols representing program actions, a jacket with pockets containing behaviour descriptions in the form of comic strips, or tasks that the interactive artefact should perform. Thus, the shapes and forms that different representations could take include a near to infinite amount of options and variations.

Another aspect of the example cases concerns how the specific hardware platforms resulted in completely different sets of designs, based on what was considered likely for users to want to modify and control on the different devices. This illustrates the necessity to consider physical shape, behaviour capabilities, and interaction modalities together, grounded both in existing use patterns and capabilities of the respective platform. The example cases also point to how the level of abstraction could vary extensively while still being based on the same metaphor.

Of course, in comics as well as in fashion, all the visual signs and symbols closely interplay with characters, people, contexts and objects. Characters in comics usually display facial expressions and body postures that convey the action in the story, and every sign may be carefully crafted for a specific character drawing. Similarly, posture, figure, and physical environment play important roles in how clothes are interpreted and negotiated in social contexts. So even though actDresses may communicate effectively, they naturally take another form of expression when brought to the context of tangible computing.

In developing new ways of controlling, programming and predicting the behaviour of physical consumer products, a deeper understanding of the fundamental theories explored in these and other fields may hold further benefits. Projects like these suggests further potential in using loose physical items, such as garments, jewellery and visual signs as resources for controlling and programming physical computing systems.

REFERENCES

1. Benford, S., et al. (2005). Expected, sensed, and desired: A framework for designing sensing-based interaction. *ACM Trans. Comput.-Hum. Interact.*, *12* (1). 3-30.
2. Blackwell, A.F. (2006). Gender in Domestic Programming: from Bricolage to Séances d'Essayage *CHI'2006 workshop on End User Software Engineering*.
3. Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psycology*. MIT Press, Cambridge, Massachusetts.
4. Buechley, L., Elumeze, N. and Eisenberg, M. (2006). Electronic/Computational Textiles and Children's Crafts. Proc. of *IDC2006*, ACM.
5. Chandler, D. (2002). *Semiotics: The Basics*. Routledge, London.
6. Davis, F. (1994). *Fashion, Culture, and Identity*. University of Chicago Press.
7. DiSalvo, C.F., et al. (2002). All Robots Are Not Created Equal: The Design and Perception of Humanoid Robot Heads. Proc. of *DIS2002*, ACM Press.
8. Djajadiningrat, T., et al. (2004). Tangible products: redressing the balance between appearance and action. *Personal Ubiquitous Comput.*, *8* (5). 294-309.
9. Fernaeus, Y., Kindborg, M. and Sholz, R. (2006). Rethinking children's programming with contextual signs. Proc. of *IDC'06*, ACM Press, 121-128.
10. Frei, P., et al. (2000). Curlybot: designing a new class of computational toys. Proc. of *CHI'00*, ACM Press, 129-136.
11. Goffman, E. (1959). *The Presentation of Self in Everyday Life*. Anchor Books.
12. Guo, C. and Sharlin, E. (2008). Exploring the use of tangible user interfaces for human-robot interaction: a comparative study. Proc. of *CHI'08*, ACM.
13. Jonsson, M., Fernaeus, Y. and Tholander, J. (2008). Setting the stage - Embodied and spatial dimensions in emerging programming practices. *Interacting with computers* (Special issue on Physicality and Interaction Design).
14. Kelleher, C., Pausch, R. and Kiesler, S. (2007). Storytelling Alice Motivates Middle School Girls to Learn Computer Programming. Proc. of *CHI2007*, ACM.
15. Kindborg, M. (2003). Concurrent Comics - programming of social agents by children. Ph.D. Dissertation. Linköping University.
16. Kindborg, M. and McGee, K. (2005). Comic Strip Programs: Beyond Graphical Rewrite Rules. Proc. of *VCL 05*.
17. Lapides, P., Sharlin, E. and Sousa, M.C. (2008). Three dimensional tangible user interface for controlling a robotic team. Proc. of *HRI'08*, ACM.
18. McCloud, S. (1993). *Understanding Comics*. HarperCollins Publishers, New York.
19. McNerny, T.S. (2004). From turtles to Tangible Programming Bricks: explorations in physical language design. *Personal and Ubiquitous Computing*, *8*. 326-337.
20. Pakhchyan, S. (2008). *Fashioning Technology A DIY Intro to Smart Crafting*. Make Books.
21. Papert, S. (1980). *Mindstorms: Computers, Children, and Powerful Ideas*. Basic Books, New York.
22. Patten, J., Griffith, L. and Ishii, H. (2000). A Tangible Interface for Controlling Robotic Toys. Proc. of *CHI'00*, ACM, 277-278.
23. Post, E.R. and Orth, M. (1997). Smart Fabric, or "Wearable Clothing". Proc. of *International Symposium on Wearable Computers*, IEEE.
24. Raffle, H.S., Parkes, A.J. and Ishii, H. (2004). Topobo: a constructive assembly system with kinetic memory. Proc. of *CHI'04*, ACM Press, 647 - 654.
25. Robins, B., et al. (2004). Robots as Assistive Technology - Does Appearance Matter? Proc. of *RO-MAN 2004* IEEE Press, 277-282.